Global Information Assurance Certification Paper

Setting Up and Securing a Small Network with OpenBSD

GIAC Security Essentials Practical Assignment 1.4b Option 1

By Blair Heiserman

Introduction

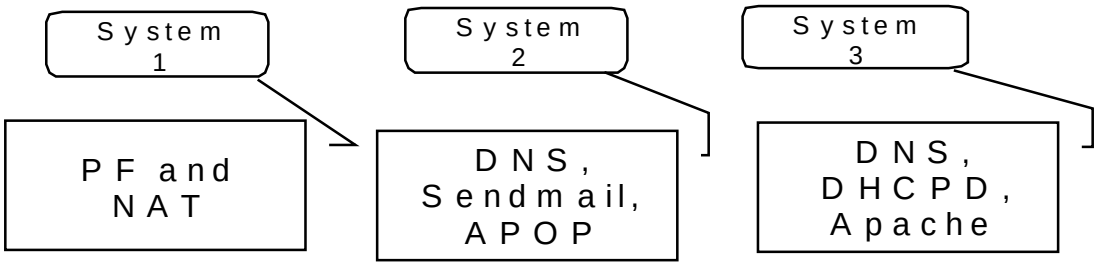
It is often difficult for a small business to figure out how to build a network with all of the services that they need. Often they have their network services managed by an ISP or a consulting company. By bringing their network services in house, they will gain the advantage of being able to make changes at their leisure and have the security of being able to easily review and audit their configurations for compliance with security policies. This paper intends to show how to setup a stable and secure small network, while keeping everything in house. I will cover setting up a small network and essential services for network connectivity and functionality. This will detail the most basic services that a small network is likely to need: DHCP, DNS, Email, Firewall, Network Address Translation, a POP server, and a Web Server. I will cover each in enough depth to provide a basic setup and understanding of the service.

It is important to be able to setup secure networks easily. I accomplished this task myself through countless hours of pouring through the Internet to find the right sources to be able to set up a small, cheap, reliable, and secure network. I desperately needed a good how-to guide to start me off in the right direction. Having now accomplished the hard part myself, I hope to make it easier for others to set up a secure network while they are educating themselves on network operation and security. At least a basic familiarity with the terminology associated with each of the subject areas is assumed, so that an exhaustive defining of terms and their use will be unnecessary. Thus, the following should be a good jumpstart to get a network setup and provide sufficient references so the readers can educate themselves further as they are able or as they are required.

OpenBSD Installation and Configuration

The following machine configurations are done on a machine with at least a Pentium II 300 MHz with 128MB of RAM, and four GB of disk space. I am using OpenBSD 3.1 on all of the machines. Best practices indicate that it is best to remove or isolate every service and, where possible, install a single service per machine.[1] Given the number of machines that would require, it is more realistic to ask the reader to use at least three machines: a firewall machine, primary and secondary DNS with all additional services distributed between the two DNS machines. This should create an acceptable level of separation of services, while not becoming prohibitive in cost. However, if additional machines are available they should be used. The machines required do not need to be particularly powerful. Using former desktop PCs should provide more than sufficient power for these servers. As your company grows, you can size the additional machines more appropriately for redundancy and power as your budget permits. However, to start you do not need much more then a few old PCs.

[1] Garfinkel, Simson. Page 485.



Obtaining the Source

OpenBSD is freely available via ftp on the web, to find a mirror view the Source code page.[2] You can also purchase the CDROM distribution. OpenBSD is being used due to its focus on security.[3] The OpenBSD project does an aggressive code auditing to remove programming errors in the source code. The process is a learning one and as they find a new type of programming error, they add it to their list of things to look for when auditing the code. This on-going code audit has allowed OpenBSD to drastically reduce the number of bugs in their code.[4]

Let us begin with the basic install process of OpenBSD. If you have purchased the CDROMs from the website you will have a bootable CDROM. Using this you can step through the basic install process. An example installation is also available with the CDROM. For the entire install process, unless otherwise noted the default options should be used.

Make certain that the BIOS boots to a CDROM drive first. After booting to the CD, you will be presented with a prompt to install, upgrade, or get a shell prompt for OpenBSD. Choose the install option. You will be asked to confirm the installation, type yes. Then specify a terminal type, unless you have an unusual monitor you should be able to accept the default. You can then choose a keyboard encoding if you wish.

Disk Setup

Next, we enter the disk configuration. First, you will have to select a disk to use. Once you have selected a disk, you are asked whether you wish to use the whole disk. You should choose to use the whole disk, as this will install a bootloader for OpenBSD on the disk. This puts you into a disk partitioning program. This allows you to allocate disk space to each of the different file systems that you may want. However, first you must delete all of the previous partitions on the disk before you can reconfigure the disk. Thus, you need to enter “p” which will print the disks current configuration. Then you can use “d” and the partition letter to delete the partition. You can remove all of the partitions except for c, which is a representation of the whole disk. Once you have removed all the partitions, you can begin adding new partitions.

---

[2] http://www.openbsd.org
[3] http://www.openbsd.org/security.html
[4] http://www.openbsd.org/security.html#process

Partition	Size (in MBs)
/	592
/usr	2048
/var	500
/tmp	500
/home	200
swap	256

At the most basic level you could allocate most of the disk to the root partition (/) and some to swap space. However, this is not a recommended setup. It is best to allocate space to each of the following partitions (/, /usr, /var, /tmp, /home). Adequate space, 100-200MB at a minimum, should be allocated to the (/, /var, /tmp, /home) directories. Swap space should be roughly double the amount of memory that you have in your system, if you have enough disk space available. Each of the different partitions holds it's own importance. Depending on how much you plan on using each partition should dictate the amount of space that you give it. The root directory (/) holds any of the file systems that you do not explicitly give another partition, thus it should be allocated more space if you do not create some of the other partitions. The /var directory will hold your log files which will allow you to monitor system activity after you have your system running. The /tmp directory is used as a scratchboard for the system and often program files, too small of a /tmp directory and the system may experience slowness. The /home directory is where all of the user file will be kept, unless you are planning on allowing extensive shell access to these machines a small /home directory will be more then adequate. The /usr directory holds all of the various programs, libraries, and the source code so should be allocated plenty of space. To add each partition use the 'a' command, for add partition. The first two partitions should be the root (/) partition and the swap space. After that assign partition labels as you like.

After you have assigned your partitions, use the "q" command to quit and commit your changes to the hard disk. You will be given a chance to confirm the file systems that you want each partition to have, once you are satisfied with the configuration type "done". You will again be given a chance to confirm your changes, if everything is correct say yes and continue. The next several minutes will be spent formatting the disk.

Network Setup

Configuring the network is the next step. You can configure the network after you build the system, but the install process takes care of the many files that you would otherwise have to edit separately. Select "yes" to configure the network. You can now enter a hostname for the machine, note that this is not the whole name plus domain name, but just the host name (e.g. sauron). Next, enter your domain name. (e.g. example.com) After that, you choose which interfaces you wish to configure, this is particularly important if you have more then one network controller in the machine. You can enter either an IP address (e.g. 172.19.5.4) or dhcp for each card. Since the machines you are configuring are meant to be servers it is best to give them static address and because you have yet to setup your DHCP server. You are asked to confirm or change the host

name again. You are asked for the netmask next. You are then allowed to enter media directives, e.g. media 100baseTX[5], information for the network card, unless you have a strange network card the default setting should auto-configure the network card. You are then allowed to repeat the network configuration process for other cards or enter done to continue with the installation. After you have finished with the network cards you are asked for the default route or the network gateway, enter this.[6] Next, you need to enter addresses for the name servers. Use either your ISP's DNS servers or you can use the IP addresses for the servers you are planning to build as DNS servers. You can enter both primary and secondary name servers by separating them with a space. (e.g. 172.19.5.2 172.19.5.3) You are then asked if you wish to use the name server now. In this case, it is best not to since you are installing off the CDROM. You are then given a chance to edit the host table so that you could perform a network install if you desired, however, enter "no." Next you are given another opportunity to shell out and perform other network configuration, however this should be unnecessary.

You are asked for a password, which will be used by the root account. It is important to use a significantly strong password. It is recommended you use numbers and letters, with upper and lower case, and special characters if possible, and not a dictionary word or it's reverse.[7] You are asked to confirm the password before moving on to the next step of the configuration.

Package Installation

You are asked whether you plan to run the X Windows system. Since these machines are meant to be servers, it is best not to run X Windows as that adds another service, which could be exploited to break into the machine.

Now you are asked where you want to install from. Since we are installing from the CDROM, select that option. However as you can see there are a number of options available to do the installation: ftp, http, tape, CDROM, and a local hard disk. You are asked to confirm the CDROM drive that the installation CD is in. Then you are asked to confirm the path to the installation files.

You are now presented with the install packages available for the machine. All of the basic packages are selected, except for the X Windows packages, which we will not need. You may also wish to remove the games package with the -game31.tgz command. Once you have your final list of packages type done. You are asked for confirmation and then the packages are

[5] Man page: hostname.if

[6] Your router IP address should have been supplied by your ISP. Most ISPs will put the gateway or router address at the beginning of the address range that you have been assigned. Thus a class C network of 172.19.5.0/24 would have a gateway address of 172.19.5.1, however there is no requirement for it to be this way. You could just as easily have the gateway address in the middle or at the end of your address range.

[7] Frisch, Æleen. Pages 148-153.

installed. After the packages are installed, you are given the opportunity to install any additional sets of packages you may have available. Select “no.”

Now we set the time zone. The BIOS clock on the machine should be set to Greenwich Mean Time (GMT) if you are going to enter a time zone, otherwise the machine will not appropriately reflect the time. You can find your time zone by entering a question mark at the prompt for a list. In this case, we will use US/Eastern for Eastern Standard Time (EST).

The machine then finishes the install and unmounts the filesystems. You need to type reboot at the command prompt and your new system will come up. At this point, either remember to remove the CD from the CDROM drive or switch the BIOS back to booting from the hard drive first, or do both.

Post Install Configuration and Patching

Your machine will reboot and you now have a basic default configuration to work with. The majority of the files that you need to be concerned with to tighten OpenBSD’s security even further are in the /etc/ directory. The main ones that you need to review are /etc/rc.conf, /etc/sysctl.conf, and /etc/inetd.conf. These files contain the initialization routines for the services that will run on your OpenBSD system. While OpenBSD has only had a single remote hole in seven years[8], it still doesn’t hurt to remove services that you do not need on your machine.

Patching

With your basic install finished, it is now necessary to close the gaps and apply any patches that have been released for OpenBSD since the distribution was created. OpenBSD supports its distributions for a year after their release.[9] You can view all of the patches for the distributions via the Patches link on the main OpenBSD web page under OpenBSD Resources. There is a rollup of patches available each day and is posted at the top of each distribution’s patch page. You can download this and apply it to the source code. The source code can be downloaded either from the site or from the CD. The different methods are detailed on the OpenBSD page under Getting OpenBSD. For our purposes, it will be assumed that you ordered the CD and can copy it off the CD distribution. You want to untar the source file, src.tar.gz, into the /usr/src directory. Then you can untar the patch file rollup into the same directory. All of the patches take the form of 001_named.patch, where named is typically the name of section of code being patched. The directions on how to apply the patch are at the beginning the file. Typically, you type, “patch –p0 < 001_patchname.patch”, which applies the patch file to the source code. Then you need to follow the directions to rebuild and install the new piece of code. This almost invariably involves cleaning out any old code remnants (make cleandir), making the object files (make obj),

[8] http://www.openbsd.org
[9] http://www.openbsd.org/support.html

building it (make), and installing it (make install). This does vary from patch to patch so it is best to read the beginning of each patch file before you apply it.

Generally, at least one of the patches will require you to rebuild the kernel for the machine. This process is usually quite simple. Choose a configuration file. In this case, I would recommend using the GENERIC configuration file. This can be found in /usr/src/sys/arch/i386/conf/ directory. You type config GENERIC. Then you switch to the /usr/src/sys/arch/i386/compile/GENERIC directory. There you issue the commands: make cleandir, make obj and make. After the kernel has finished building, you move the old kernel (/bsd) to /bsd.old and then copy the newly built kernel to /bsd. This can be done with the command: mv /bsd /bsd.old; cp bsd /bsd. After this is accomplished, you can reboot the machine.

Services

After you have recompiled and installed all of the patches for your machine, you can begin to tweak the available services to match your desired configuration for each machine. Each machine should be locked down to an even more secure configuration initially and then turn the services turned back on as they are needed.

The first file to consider is the /etc/rc.conf file as it starts all of the varied services for the machine. In that file you want to page down and change the selection on the following choices to "NO": portmap, inetd, check_quotas, ntpd. By disabling these additional services, you have reduced the machines profile to just the SSH port. Portmap is unnecessary on all of the machines that we are configuring as the RPC services will not be needed and are not desired on these servers. By removing inetd, you have eliminated all of the avenues for using basic innocuous utilities to gather information on the machine. The check_quotas is unnecessary for the configurations that I am showing you. The ntp service is not installed by default on OpenBSD, but it is definitely good to have installed on the machine to make certain that all the logs on the different machines are synchronized to ensure easier identification of illegal activity if there is an intrusion into your network.[10] Since it is not installed, I will leave it out of the scope of this paper.

Communication with OpenBSD machines is done through Secure Shell (SSH). SSH is an encrypted replacement for telnet, ftp, and the r-services. It will allow you to communicate with your servers securely over the network. SSH configuration is done in the /etc/ssh directory. It is important once you have created an additional account with wheel access that you edit the sshd_config file to disable remote root access. Also, disable access with SSH version 1 as the protocol is not secure and has more problems then the updated SSH version 2.[11]

[10] Read about how to synchronize with ntp at O'Reilly. And http://www.linuxsecurity.com/feature_stories/feature_story-133.html

[11] CIAC. "M-017: Multiple SSH Version 1 Vulnerabilities"

You can make both of these changes by editing the sshd_config file and uncommenting the following lines

```
#Protocol 2,1
#PermitRootLogin yes
```

and change them to

```
Protocol 2
PermitRootLogin no
```

This should close the potential security holes of using the weaker and vulnerable version of SSH and restricting root access to either the console or someone with wheel privileges on the machine. By denying remote root logins, you eliminate a remote root brute force password hack attempt through repeated logons as well as creating better logging of who is using the root account and when.

OpenBSD has a built-in program that runs daily, which notifies you of the health and changes to the machine. Because of this, create a .forward file with an administrative email address in the file. You will receive all of the emails related to the systems' health and changes. Tracking changes should merit particular attention, as it will inform you when there have been changes or additions to libraries, programs, password files, and other important system components. It is a Tripwire-lite. [12] Monitoring these emails will keep you up to date on your system and informing you when there have been unauthorized or unintended changes to the system.

DNS

Businesses frequently leave their DNS operations to their Internet Service Provider sacrificing direct control and adding potential delay and frustration to DNS changes, as they must wait on the ISP to implement their changes. By running your own servers you can make changes internally and as frequently as needed. OpenBSD includes a heavily audited version of BIND 4.9.8.[13] Like the rest of the OS, the source code was audited to ensure that best programming practices had been followed. OpenBSD's release of BIND is frequently less vulnerable to the holes that appear in the older versions of BIND. From checking the errata pages on the OpenBSD website, it shows that from release 2.8 on that there have been either one or two vulnerabilities for BIND during each releases period of support.[14] Often the patches are for potential vulnerabilities rather then extant problems, in fitting with their code auditing procedures. The low number of patches and audited code makes OpenBSD's DNS implementation ideal for a simple DNS setup. OpenBSD's version of BIND can also handle IP v6 if your

[12] Frisch, Æleen. Page 255-257.
[13] OpenBSD FAQ 6.8.3.2.1
[14] By viewing the OpenBSD Errata pages, you can confirm the low number of patches for the parts of the named system. You can also note that frequently at the time of the patch, it was issued because of a potential, rather then actual vulnerability.

configuration requires that, however since IP v6 is not yet the deployed standard it is out of the scope of this paper.

You will be configuring a primary DNS server with a zone file for example.com on gandalf.example.com and a secondary DNS server on saruman.example.com. You need to edit /etc/rc.conf and change the named_flags=NO to "". Here is a sample configuration file (/etc/named.boot for example.com).[15] The comment symbol for the named files is a semi-colon and it comments out the remainder of that line.

```
;beginning of named.boot
directory      /namedb

xfernets 172.19.5.2&255.255.255.255 172.19.5.3&255.255.255.255

; type          domain                source host/file          backup file
cache           .                     root.cache
primary         0.0.127.in-addr.arpa  localhost.rev
primary         localhost             localhost.zone

; example primary server config:
primary         example.com           example
primary         5.19.172.in-addr.arpa example.rev
------------------------------------------------------------------------
; example secondary server config:
secondary       example.com           172.19.5.2      example.bak
secondary       5.19.172.in-addr.arpa 172.19.5.2      example.rev.bak
;end of named.boot
```

The first part of the configuration file defines the directory from which OpenBSD starts named. OpenBSD chroot's named so that even if BIND is somehow hacked it is limited in the areas that it has access to, thus reducing potential damage.

Then we have the xfernets directive.[16] This is useful for machines that will be hosting authoritative zone files. The xfernets directive limits the machines that are able to do a zone transfer with the name server, this is important since without this directive anyone can do a full zone transfer with your name servers. Thus you should have the xfernets directive on both the primary and secondary server configurations, even though the secondary server does not have tertiary servers with which it would do zone transfers. The xfernets directive takes IP address numbers and netmask in the form of IP address ampersand netmask. So, you can specify either a single machine or a range of machines.

[15] This example configuration is an amalgamation and simplification of the default OpenBSD named.boot file with the addition of another boot time directive.
[16] Cricket Liu explains the use of xfernets directive as well as other BIND 4.9.5 directives in this article.

Next, the machine defines zones and files for the localhost. This part of the configuration should be on all machines running DNS regardless of the role. A caching only server, one that does not host any primary zone files for any domain but simply queries and caches network information, would still need these files defined. The cache line defines all of the root servers. The next two lines define localhost information.

The examples for the primary and secondary server section are representative of two different types of machines. The primary section is for the main authoritative server for a domain. The secondary section is for any of the secondary servers for the domain. A primary server would contain the chroot, directory, xfernets, localhost, and primary server sections of the above configuration file. A secondary server would contain all of the same information, except that it would swap out the secondary server configuration section for the primary. You would need to alter the xfernets directive between both of the machines or you could simply put both of the nameservers in both of the files.

For a primary server entry, you enter "primary," the domain name and the name of the zone file that contains all of the domain information. For a secondary server entry, you enter "secondary," the domain name, primary server IP addresses and the zone filename where it will backup the information.

Now for the fun part, configuring the domain name files. The localhost files you can leave alone, because the default files that come with the system are already correctly configured. This just leaves us with setting up forward and reverse DNS entries for your enterprise. The zone files are located in /var/named/namedb. The following is a sample file /var/named/namedb/example.com for a very basic forward zone file for example.com.[17]

```
;start of example.com zone file
@       IN      SOA     gandalf.example.com.
dnsadmin.gandalf.example.com.  (
                        01082003      ; Serial
                        3600    ; Refresh
                        900     ; Retry
```

[17] You may have noted the Tolkein-esque naming conventions that I have been using. I blame the fact that the stereotype technical people and specifically systems and network administrators are portrayed as geeks and nerds, of which I am sure many are proud. I also blame the recent release of the Lord of the Rings movies. So, I feel the need to play into this stereotype. Naming schemes are a highly personal thing are can range from the obvious, comical, to the obscure. There is even an RFC available about naming systems. See RFC 2100. In these security conscious days, it may make more sense to add a little obfuscation to your naming conventions. Mind you, a dedicated attacker can still pull some of the information anyways by querying for NS and MX records. It is up to you to find the trade off, of an acceptable level of security to what you will be able to remember. Everyone can remember when the mail server is named mail, but will they be able to remember if it is called gandalf?

```
                        3600000 ; Expire
                        3600 )  ; Minimum
;Name Servers
                IN      NS          gandalf.example.com.
                IN      NS          saruman.example.com.
;Email Servers
                IN      MX          0       gandalf.example.com.
;Addresses
balrog          IN      A           172.19.5.1
gandalf         IN      A           172.19.5.2
saruman         IN      A           172.19.5.3
sauron          IN      A           172.19.5.4
frodo           IN      A           172.19.5.30
;end of example.com zone file.
```

Let us start with a quick breakdown of the file.[18] The first uncommented line is the beginning of the start of authority record or SOA. "gandalf.example.com" is the name of the machine that it is on. "dnsadmin.gandalf.example.com" is the email address that should be used to contact the dns administrator, though you wouldn't normally recognize it as such. You can determine the email address by substituting the first period with a @. Replace these items with your own information. Many people simply use root as the account for dns related emails, but given the ease with which you can setup either an alias or an account it may be better to give yourself a field to filter the message on.

The serial is effectively a version number. Everyone has their own preferred way of changing the serial number. What is important is that it gets incremented every time you change the zone file; otherwise, your secondary servers will not transfer the zone and get the updated information. In this example, I have simply used the date as the serial.

The remainder of the SOA record is self-explanatory. The "refresh" entry is how often the machine should check to see if it needs to refresh the information on the zone. The "retry" entry is how often it should attempt to reconnect with the primary zone computer if it cannot contact it on its refresh attempt. The "expire" entry is how long it should keep the data before it removes it. The "minimum" entry is the minimum amount of time the machine should keep the zone information.[19]

Next, we get to the different types of records that can be in the zone file. There are four that we will concern ourselves with and these are, A, MX, NS, and PTR. A is for an Address record and is used in the forward lookup files where names are translated to numbers. The MX record is used to indicate machines

[18] Man page: named.
[19] Man page: named.

that are mail servers. The number after the MX record is the priority number so that you can assign an order on how you prefer to have your email delivered. The lower the MX record number the higher the priority the machine will receive. The NS record indicates machines that are name servers. The PTR record translates IP address numbers to names in the reverse lookup files.[20]

Note that for the NS and MX records where a full machine name is given, there is a period at the end of the names. If you did not put a period at the end of the machine names, it would append the name of the domain whose file you are in, e.g. "gandalf.example.com" in that file without the period would resolve to "gandalf.example.com.example.com." It is critical that you have the period the name.[21] For the address records, note the form used is "machine_name IN A IP address" and by not using the period after the machine name, these machines will resolve to their appropriate full domain names.

```
;start of example.com reverse zone file
@       IN      SOA     gandalf.example.com.
dnsadmin.gandalf.example.com.  (
                        01082003      ; Serial
                        3600    ; Refresh
                        900     ; Retry
                        3600000 ; Expire
                        3600 )  ; Minimum
;Name Servers
                IN      NS      gandalf.example.com.
                IN      NS      saruman.example.com.
;Addresses
1               IN      PTR     balrog.example.com.
2               IN      PTR     gandalf.example.com.
3               IN      PTR     saruman.example.com.
4               IN      PTR     sauron.example.com.
30              IN      PTR     frodo.example.com.
;end of example.com reverse zone file
```

The name of the reverse domain zone is important. All of the reverse address records are part of the "in-addr.arpa" domain. As a part of the "in-addr.arpa" domain, the naming scheme is pre-defined to produce a reliable reverse mapping scheme. The order is the reverse of the IP address with the addition of "in-addr.arpa." (e.g. 172.19.5.1 is 1.5.19.172.in-addr.arpa) [22] To do a larger domain you can create a name server zone file for "5.19.172.in-addr.arpa," which will cover all of the hosts in 172.19.5.0/24. For the "in-addr.arpa" zone, you need to have PTR records so that address can be reverse mapped from IP

[20] Albitz, Paul and Liu, Cricket. Pages 413-422.
[21] http://www.ietf.org/rfc/rfc1537.txt Section 5.
[22] http://www.ietf.org/rfc/rfc1034.txt Page 9

address number to the hostname. The format for this is “address number IN PTR hostname.domainname.com. “

With the information outlined above, you should be able to enable the named service on a machine. Create your own named.boot configuration files for your own domain; and create forward and reverse zone files for your network.

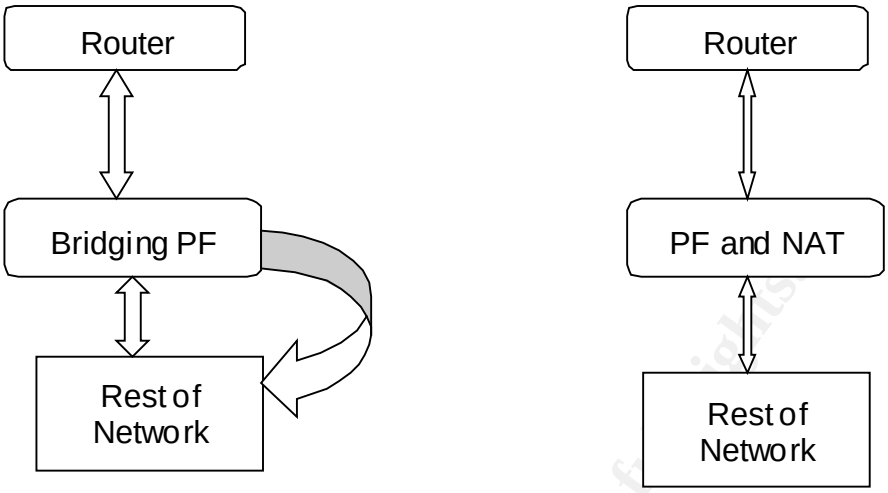
Firewall and NAT

A firewall is a critical piece of any security infrastructure. By limiting the number of ports that are able to enter and leave the network you vastly reduce the number of potential vulnerabilities. Network Address Translation (NAT) is another way to reduce potential vulnerabilities by using non-routable addresses inside your network while translating them to a pool of routable IP addresses when you need to contact the outside world. OpenBSD provides both NAT and a firewall. OpenBSD recently replaced its previous implementation of NAT and a firewall, Internet Packet Filter (IPF), with Packet Filter (PF) as its stateful firewall and NAT device. PF was written by Daniel Hartmeier and is extremely useful and configurable as both a stateful firewall and as a NAT device. [23]

The files that are critical to configuring NAT and PF are /etc/nat.conf, /etc/pf.conf, and /etc/sysctl.conf. These three files contain all of the configuration information for the system. You need to tweak the sysctl.conf if you are using NAT, but not if you are using PF.

You will setup PF, or NAT, or both on “sauron.example.com” as is appropriate for your desired network setup. It is best to setup PF on a machine with three network cards so that you can direct traffic through a two card bridge from your router. Reserving one card to use to reach the machine internal so that your firewall is not exposed to external access. You can also assign the bridge an IP number. This setup is ideal where your network includes routable addresses. In a network where the internal addresses are not routable, only two network cards are needed, as you will be using NAT to translate instead of bridging to direct network traffic.

[23] Andrew, Jeremy. “Interview with Daniel Hartmeier.”



To begin you must first edit /etc/rc.conf. Find the section with pf=NO and change the "NO" to "YES." Save the file. If you are also running NAT you will need to edit the /etc/sysctl.conf and uncomment the net.inet.ip.forwarding=1 option by removing the # and saving the file. You can now edit the /etc/nat.conf and /etc/pf.conf to put in a basic setup that will run when you restart the machine. Once you have established a baseline, you can restart the system to implement the changes. Alternatively, you can simply restart and then enable or disable pf and nat with the pfctl command.

NAT (Network Address Translation)

I will cover NAT first. I always find it simpler to work with a sample configuration file in hand. Here is a sample /etc/nat.conf

```
#start /etc/nat.conf example
#nat on $interfacename from $internaladdresses to any -> $externaladdresses
nat on fxp0 from 172.19.5.0/24 to any -> 1.2.3.4./32
#End /etc/nat.conf example
```

Surprisingly simple. This setup allows you to translate packets from internal addresses to an external address. Between this and pf, you can setup a reasonably strong network perimeter. One thing to be aware of is that all of the NAT rules are processed before they are run through PF, and they only apply to that specific interface in the direction specified by the rule. Thus a NAT rule for outgoing traffic will not translate traffic coming into the interface. The interface name is the one used for the system with ifconfig. The $internaladdresses should be replaced with IP address and netmask (e.g. 172.19.5.0/24) and the $externaladdresses should be replaced with the external IP address numbers and netmask (e.g. 1.2.3.4/32) that your ISP has supplied you with.

PF (Packet Filter)
Here is an example file for pf.conf for the firewall configuration.

```
#Start example /etc/pf.conf
# Interface Setup for ease of use
bridgeext = fxp0
bridgeint = fxp1
localif = sis0
mail1 = "172.19.5.2"
dns1 = "172.19.5.2"
dns2 = "172.19.5.3"
www = "172.19.5.3"
NoRouteIPs="{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8,
255.255.255.255/32 }"
#Normalize all incoming traffic
scrub in all
# default policy
block in log inet6 all
block in log on $bridgeext all
#Block outgoing addresses that aren't from our address range
block out log quick on $bridgeext from ! 172.19.5.0/24 to any
# don't allow anyone to spoof non-routable addresses
block in log quick on $bridgeext from $NoRouteIPs to any
block out log quick on $bridgeext from any to $NoRouteIPs
#allow ICMP types
pass in  on $bridgeext inet proto icmp from any to {$dns1, $dns2} icmp-type 8
code 0 keep state
# Block NMAP attempts
block in log quick on $bridgeext proto tcp all flags FUP
#block Netbios
block in log quick on $bridgeext proto tcp from any to any port {137,138,139,445}
#DNS, SSH, Mail, POP, VPN, HTTP & HTTPS, FTP
pass in quick on $bridgeext proto udp from any to {$dns1,$dns2} port = 53 keep
state
pass in quick on $bridgeext proto tcp from any to {$dns1,$dns2} port = 53 flags S
modulate state
pass in quick on $bridgeext proto tcp from any to {$dns1,$dns2} port = 22 flags S
modulate state
pass in quick on $bridgeext proto tcp from any to $mail1 port = 25 flags S
modulate state
pass in quick on $bridgeext proto tcp from any to $www port {80,443} flags S
modulate state
#Unrestricted access out.
pass out quick on $bridgeext proto tcp all flags S modulate state
pass out quick on $bridgeext proto udp all keep state
pass out quick on $bridgeext proto icmp all flags S keep state
```

#End example /etc/pf.conf

The above example file may seem a bit intimidating, however thankfully the pf.conf file is extremely easy to understand. Each rule defines whether something is passing in or out of the network, on which interface, whether or not to log the traffic, which protocol, and potentially from which host and port as well as which host and port to send the traffic to. PF is also very useful as it supports variable interpolation. As you can see from the beginning of the file, you are able to define variables once so that you can then use them in the rest of the file and not worry about forgetting to change an instance of an interface or a host address. It is important to note that PF evaluates the set of rules is evaluated from top to bottom, and the last matching rule decides what action is performed.[24]

The first thing you want PF to do when it gets traffic is to scrub it. Scrubbing the traffic normalizes it, by recombining packets so they can be analyzed fully. Scrubbing the packet stream normalizes and verifies the entire data stream so that malformed packets will not pass through into your network.[25]

The next set of rules helps to establish a baseline for the rest of the file. By blocking all traffic by default, you then can be certain that only specifically defined exceptions will make it through. Consider the log statement, it may not be appropriate for your network or a machine with low disk space as it will log all traffic that is not allowed which can be quite voluminous. The block rule for inet6 will block all IPv6 traffic; this includes virtual private networks that utilize the Encapsulating Security Payload. It is better to block this entire section of traffic until you are certain you need it.

You may have noticed that all of the filtering is going on the external interface on the machine. This is so that PF does not evaluate the rules for the packets more then once.[26] Since PF can analyze the traffic every time it hits a network interface. It is important to pick a single interface to do all of the filtering from, otherwise you may end up filtering the traffic more than once, or confusing yourself about where the traffic enters and leaves. By filtering the incoming and outgoing traffic on a single interface, you reduce the number of rule evaluations and keep the configuration simple.

The following rule is an egress filter. It stops your network from allowing illegitimate traffic out onto the Internet. This statement, along with the next two rules, block non-routable or private IP addresses from entering or leaving your network is a part of being a good Internet citizen. These simple safeguards make it harder for hackers to utilize your machines were they to break into your network.

[24] man pf.conf
[25] Coene, Walter. "Packet Filter How-To."
[26] Hartmeier, Daniel. "Re: pf and stateful filtering on a bridge."

The ICMP rule is to allow the Internet to ping your DNS server. It is good form to allow others to verify that your machine is up and running, without allowing tools like traceroute to map your network. This is done by only allowing one specific type of icmp, which ping uses.

The NetBIOS and NMAP rules are examples of rules you may wish to use. PF allows you to have additional redundant rules like these to log traffic. If your original block statements did not include the log directive, you could create rules like these so that you could monitor your firewall for network probes on common ports or tools, such as the Windows NetBIOS ports and nmap's port scanning.

Then we have the rules to allow machines from the Internet to connect to your publicly available services. Note how each different protocol type is listed separately, this is because UDP is stateless and we want to use stateful filtering for our TCP connections. Stateful filtering allows us to reduce the number of rules that need to be evaluated to allow traffic on your network. It also allows us to apply the modulate state directive which strengthens the TCP sequence number randomization, which is weak in some TCP stack implementations.[27] The flags statement lets us define when to start using stateful filtering, by using the S flag it begins with the initial connection and then allows that specific connection to bypass the re-evaluation by the PF rules, which saves time.

The last batch of rules is to allow unrestricted access from your network to the outside world. Because of the state rules, your connections leave your network and are able to return securely. This is important, as you do not have to leave outside ports open to get traffic from the Internet.

You should be able to configure NAT as well as PF at this point. PF will be basic, but by using the different reference sources, you should be able to customize PF into a strong firewall for your enterprise.

DHCP

The Dynamic Host Configuration Protocol (DHCP) is used to dynamically assign IP addresses to computers on your network and can be very useful if you have laptop users or even frequent network changes that would usually require going to each machine to make network changes. DHCP allows you to make changes on a central server and have those changes to the network propagate automatically to all of the machines. This also allows you to sit down at any active network jack and be assured that you will be able to get an IP address for the machine.

You will configure DHCPD on "saruman.example.com." DHCP is controlled by the DHCP daemon which is launched by the /etc/rc.conf file at boot time. You need to edit the /etc/rc.conf file to get the DHCP server to start by

[27] Zalwaski, Michal. "Strange Attractors and TCP/IP Sequence Number Analysis"

changing the entry from NO to “-q”. You also need to edit the dhcpd.conf and dhcpd.interfaces files. The interfaces files determine which network card to run the server on. It can run it on a single or multiple network cards. You simply need to put the name of the network card(s) in the file. The /etc/dhcpd.conf file is a little more complex.

Here is a sample /etc/dhcpd.conf configuration.

```
shared-network example {
        option  domain-name "example.com";
        option  domain-name-servers 172.19.5.2, 172.19.5.3;
        option  subnet-mask 255.255.255.0;

        subnet 172.19.5.0 netmask 255.255.255.0 {
                option routers 172.19.5.1;
                option  subnet-mask 255.255.255.0;
                option  domain-name-servers 172.19.5.2, 172.19.5.3;
                range 172.19.5.50 172.19.5.254;
        }
        host frodo {
                hardware ethernet 00:10:83:a1:d5:b3;
                fixed-address 172.19.5.30;
        }
}
```

You can use the above to create a simple network that assigns addresses to all your client PCs. The servers should have static addresses and so are not included in this list. You can create additional host entries if you wish to have the same dynamic addresses assigned to your machines. You can get their hardware MAC address from either the machine itself with ipconfig or ifconfig or the appropriate command for the OS, or you can simply look in the /var/db/dhcpd.leases file. As long as you have given all of the machines identifiable host names, those should show up in the leases file. By taking that entry and moving it into the dhcpd.conf file and restarting the DHCPD daemon you can get all of your machines to have a static dynamic address. You will need to edit the entry to get it to conform to the configuration rules for DHCPD.

As you can see, the dhcpd.conf file has three distinct parts. There is the overall network configuration with options for the domain name, DNS servers, and the subnet mask. Then you have sections for subnets inside of the overall network, which allow you to input much the same information as well as the default gateway. The range option is used to define the IP address pool for machines receiving dynamic IP addresses. The final section is for those that wish to define static mappings for their dynamic addresses. It contains the hosts name, hardware mac address and the fixed address to which it is assigned.

DHCPD will provide you with an easily configured network with plug and play connectivity. The use of the hosts options will allow you to keep users on static addresses even if they roam about the office with their laptops.

Sendmail and APOP

Sendmail is the default mailer included with OpenBSD. The coders for OpenBSD have made many changes to Sendmail's operation on OpenBSD to further ensure its security. Sendmail is now chrooted like named and much of the root privileges have been removed from the program. Everyone wants their own email server, since it definitely one of the main reasons to have any kind of Internet presence. Sendmail and APOP provide that for you. Sendmail delivers your email messages and APOP lets you receive them.

You will configure Sendmail and APOP on "gandalf.example.com." All of the configuration files for Sendmail are in the /etc/mail directory, except for the initial invocation of Sendmail for startup which is in the rc.conf file in the /etc directory. If you want to run Sendmail as your main email server you will need to edit the /etc/rc.conf to change the directive for Sendmail from sendmail_flags="-L sm-mta -C/etc/mail/localhost.cf -bd -q30m" to sendmail_flags="-L sm-mta -bd -q30m."

```
#start sendmail.cf subsection example
####################
#   local info   #
####################

# my LDAP cluster
# need to set this before any LDAP lookups are done (including classes)
#D{sendmailMTACluster}$m

Cwlocalhost gandalf.example.com example.com
#end sendmail.cf subsection example
```

The above example shows you how to get Sendmail to recognize itself as something that is not its exact host name. The machine will recognize itself as "gandalf.example.com" without putting that in the Cw directive, but it won't recognize itself as the mail server for example.com. You can edit the /etc/mail/sendmail.cf file and change the Cw directive or you can add all of the host names that the machine should recognize itself as in the /etc/mail/local-host-names file. Either will work.[28]

To get Sendmail to accept email messages from other computers on your network you will likely have to create the file /etc/mail/relay-domains. In this file, you can place a number of different statements to get Sendmail to allow your

[28] http://www.sendmail.org/faq/section4.html Subject Q4.5

workstations to send mail through your mail server. This includes listing domainname.com, hostname.domainname.com, or IP address. By listing your computers in the /etc/mail/relay-domains file Sendmail will accept email message from those hosts, instead of denying relaying.[29]

Another file you should consider for Sendmail is the /etc/mail/aliases file. It allows you to create forwarding or redirection for your email accounts.[30] Thus if you wish to create additional email names or aliases, (e.g. John Smith's account is jsmith, but wishes to have John.Smith@example.com as his email address) you would enter those into the /etc/mail/aliases file. You enter it in the form alias:email_account. (e.g. John.Smith:jsmith) You are not limited to local accounts but can also redirect the email to accounts on other servers. (e.g. John.Smith:jsmith@hotmail.com) After you have made your changes to the file you need to issue the newaliases command, which rebuilds the email aliases database.

To enable APOP to receive email from your mail server you must first enable inetd.conf in the /etc/rc.conf file. Do this by changing the inetd directive from "NO" to "YES." Now you can enable the built in popserver for OpenBSD by uncommenting the pop3 statement in the /etc/inetd.conf file. You should also comment out all of the other services in the inetd.conf file, as you will not be using them.

Sendmail will let you send your messages from your own domain name. With APOP enabled, you will enable your users to receive their email as long as they have an account on the system or you setup an email forwarding address.

Apache

Apache is installed by default on OpenBSD, but is not enabled. You can setup a simple website with Apache, though with greater experience you can run a large website on a machine with Apache. According to Netcraft in December of 2002 Apache was running 65.2% of all websites.[31]

You will configure Apache on saruman.example.com. All of the files for Apache on OpenBSD are found in the /var/www directory. This includes the configuration directory /var/www/conf, log directory /var/www/logs, and document directory /var/www/htdocs. You also need to edit /etc/rc.conf to change the httpd directive to "". The following tells you how to setup a simple site and websites for each user on the machine.

Here are the directives you need to change in the /var/www/conf/httpd.conf file.

[29] http://www.sendmail.org/tips/relaying.html
[30] Costales, Bryan and Allman, Eric. Pages 24-26.
[31] http://www.netcraft.com/survey/

```
ServerAdmin root@www.example.com

DirectoryIndex index.html default.html Index.html Default.html index.htm

UserDir www
<Directory /home/*/www>
    AllowOverride None
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
       Order allow,deny
       Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK
UNLOCK>
       Order deny,allow
       Deny from all
    </Limit>
</Directory>
```

Note the three separate directives. The first you need to set so that you have provided an administrative contact for you web server, this can be any email address you wish. The next directive adds potential default file names, so that users are appropriately directed to the base file in the directory. The last set of rules allows users to have their own folders on the web server. This allows them to setup a www directory in their home directory, which will allow them to have webpages accessible at http://www.example.com/~username. The directory needs to be world readable, as do the files in the directory to display properly.[32]

Remove or relocate the default files in the /var/www/cgi-bin and /var/www/htdocs. This will leave you with a bare and functional website. You need only to place your content inside the /var/www/htdocs directory and you have your website.

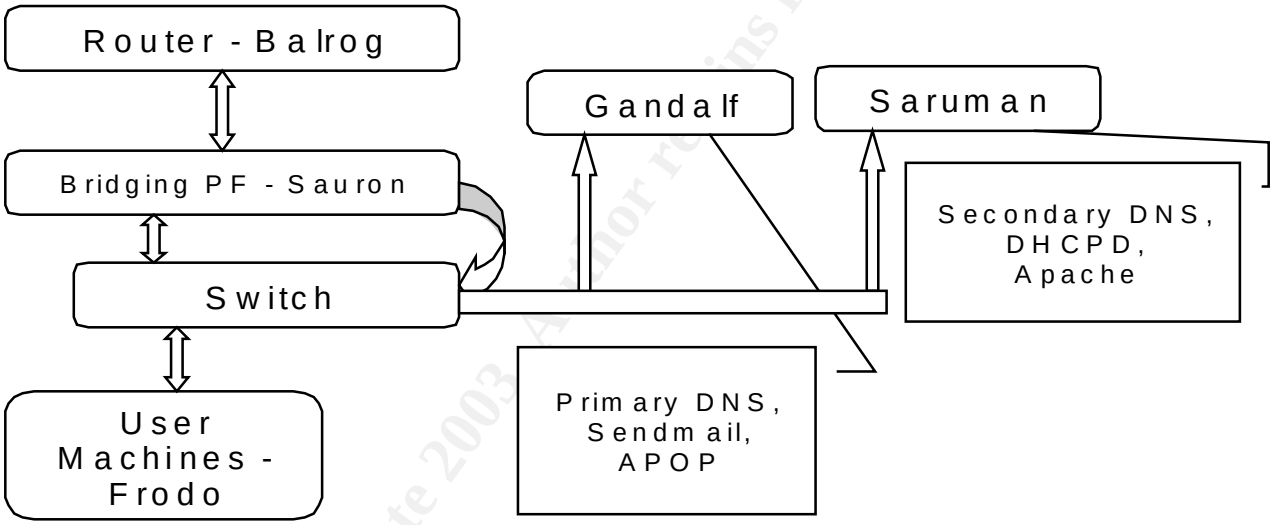
With Apache enabled you can have a relatively high performance static website, and once you gain the expertise you can switch it over to a more dynamic web site.

Summary

You should now have three machines. Gandalf should be your primary DNS, APOP and email server. Saruman should be your secondary DNS, DHCPD and web server. Sauron should be either your bridging PF firewall or your NAT and PF firewall for your network. You could install the services on different machines as you wish. The instructions above are modular enough that you can simply go through the process of enabling them on another machine.

[32] Laurie, Ben and Laurie, Peter. Pages 160-161.

If you read through to here, you should have a better understanding of how to setup your own small network. This paper provided you with enough information to get a moderate understanding of each of the services involved in a small network, while leaving the more in depth coverage of each subject to other references. It provided all of the information in one place, which was the main goal. It also covered how to keep your basic setup secure through the elimination of services and ensuring that your configuration was minimal enough to keep potential vulnerabilities low. Another thing that should be clear from the footnotes is that often all of the information you need to configure network services securely is at your fingertips, either in the man pages for that topic or through a quick Internet search. If the man pages are not clear enough you can always find further sources for clarification on the Internet and at your local bookstore.



References

Albitz, Paul and Liu, Cricket. "DNS and BIND: Third Edition." O'Reilly & Associates, 1998.

Andrew, Jeremy. "Interview with Daniel Hartmeier." October 29, 2002. URL: http://www.kerneltrap.org/node.php?id=477 (January 14, 2003)

Beertema, Piet. "Common DNS Data File Configuration Errors." October 1993. URL: http://www.ietf.org/rfc/rfc1537.txt (January 14, 2003).

CIAC. "M-017: Multiple SSH Version 1 Vulnerabilities." November 16th, 2001. URL: http://www.ciac.org/ciac/bulletins/m-017.shtml (January 14, 2003)

Coene , Wouter. "The OpenBSD Packet Filter How-To." April 5, 2002. URL: http://www.inebriated.demon.nl/pf-howto/html/ (January 14, 2003)

Costales, Bryan and Allman, Eric. "Sendmail: Second Edition." O'Reilly & Associates, 1997.

Frisch, Æleen. "Essential System Administration:Second Edition." O'Reilly & Associates, 1995.

Garfinkel, Simson and Gene Sprafford. "Practical UNIX & Internet Security: Second Edition." O'Reilly & Associates, 1996.

Hartmeier, Daniel. "Re: pf and stateful filtering on a bridge." October 4, 2001. URL: http://marc.theaimsgroup.com/?l=openbsd-tech&m=100220976320265&w=2 (January 14, 2003)

Laurie, Ben and Laurie, Peter. "Apache: The Definitive Guide." O'Reilly & Associates, 1999.

Liu, Cricket. "Name Server Security Features in BIND 4.9.5." October 31, 1997. URL: http://www.w3j.com/7/s3.liu.wrap.html (January 14, 2003).

Mockapetris, P. "DOMAIN NAMES - CONCEPTS AND FACILITIES" November 1987. URL: http://www.ietf.org/rfc/rfc1034.txt (January 14, 2003)

Netcraft. "Netcraft Webserver Survey." December 2002. URL: http://www.netcraft.com/survey/ (January 14, 2003).

OpenBSD. "OpenBSD Homepage." URL: http://www.openbsd.org (January 13, 2003)

OpenBSD. "OpenBSD Security." URL: http://www.openbsd.org/security.html (January 14, 2003)

OpenBSD. “Manual Pages: Pf.conf.” July 8, 2001. URL: http://www.openbsd.org/cgi-bin/man.cgi?query=pf.conf&apropos=0&sektion=0&manpath=OpenBSD+3.1&arch=i386&format=html (January 14, 2003)

OpenBSD. “Manual Pages: hostname.if.” September 2, 1999. URL: http://www.openbsd.org/cgi-bin/man.cgi?query=hostname.if&apropos=0&sektion=0&manpath=OpenBSD+3.1&arch=i386&format=html (January 14, 2003)

OpenBSD. “OpenBSD FAQ 6.8.3.2.1 - Security Note” URL: http://www.openbsd.org/faq/faq6.html#DNS (January 14, 2003)

OpenBSD. “OpenBSD Errata and Patch list.” URL: http://www.openbsd.org/errata.html (January 14, 2003)

Sendmail. “Sendmail FAQ, Section 4 General Sendmail Issues Q4.5” January 24, 2002. URL: http://www.sendmail.org/faq/section4.html (January 14, 2003)

Sendmail. “Allowing controlled SMTP relaying in Sendmail 8.9 and later” September 24, 2002. URL: http://www.sendmail.org/tips/relaying.html (January 14, 2003)

Zalewski, Michal. “Strange Attractors and TCP/IP Sequence Number Analysis” April 21, 2001. URL: http://razor.bindview.com/publish/papers/tcpseq.html (January 14, 2003).

Further Reading

OpenBSD

OpenBSD Website. URL: http://www.openbsd.org

OpenBSD Man Pages. URL: http://www.openbsd.org/cgi-bin/man.cgi

OpenBSD Mailing List Archives. URL: http://www.monkey.org/openbsd/

Geodsoft. “Hardening OpenBSD Internet Servers.“ December 20, 2001. URL: http://geodsoft.com/howto/harden/ (January 14, 2003)

Horn, John. “I've been hacked! How OpenBSD saved our project.” October 1999. URL: http://www.daemonnews.org/199910/openbsd.html (January 14, 2003)

OpenBSD Journal. URL: http://www.deadly.org

DNS and BIND
Man pages: named.
Albitz, Paul and Liu, Cricket. "DNS and BIND: Third Edition."

Internet Software Consortium. "BIND" URL: http://www.isc.org/products/BIND/ (January 14, 2003)

PF and NAT
Man pages: pf.conf, pfctl, nat.conf.
Coene , Wouter. "The OpenBSD Packet Filter How-To." See Above for URL.

Hartmeier, Daniel. "OpenBSD Packet Filter." December 11, 2002. URL: http://www.benzedrine.cx/pf.html (January 14, 2003)

Wes. "How To: Invisible Firewall." August 24, 2000. URL: http://www.openlysecure.org/openbsd/how-to/invisible_firewall.html (January 14, 2003)

DHCPD
Man pages: dhcpd, dhcpd.conf.

Internet Software Consortium. "DHCP" URL: http://www.isc.org/products/DHCP/ (January 14, 2003)

Sendmail and APOP
Man pages: sendmail, aliases, newaliases.
Sendmail.org. URL: http://www.sendmail.org/

Sendmail. "Allowing controlled SMTP relaying in Sendmail 8.9 and later"

Costales, Bryan and Allman, Eric. "Sendmail: Second Edition."

Apache
Man pages: httpd.

Apache Software Foundation. "Apache Website." URL: http://www.apache.org (January 14, 2003)

Laurie, Ben and Laurie, Peter. "Apache: The Definitive Guide."

Other Services

NTP
Mills, Dave. "TimeSynchronication Server Website." October 9, 2000. URL: http://www.eecis.udel.edu/~ntp/ (January 14, 2003)

Dunston, Duane. “No 'A' Word In Time” December 23, 2002. URL: http://www.linuxsecurity.com/feature_stories/feature_story-133.html (January 14, 2003).

SSH

OpenSSH. “OpenSSH Website.” December 26, 2002. URL: http://www.openssh.org/ (January 14, 2003)