Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

IPSec Interoperability between OpenBSD, Linux and Sonicwall

IPSec (IP Security protocol) has become an important tool for network engineers to provide enhanced security in company communications over the Internet. In the enterprise environment, hardware based Virtual Private Network (VPN) devices are available to handle hundreds of IPSec tunnels simultaneously. The OpenBSD project, the Linux FreeS/WAN project and Sonicwall Inc. have each produced cost effective IPSec implementations with excellent reliability. Collectively, they are helping to improve security for the small and...

Copyright SANS Institute

Author Retains Full Rights

AD

IPSec Interoperability between OpenBSD, Linux and Sonicwall

Practical: Track 1 – GIAC Security Essentials (GSEC)
GSEC Practical Requirements (v1.4b), Option 1
Author: Daniel de Young

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Summary

IPSec (IP Security protocol) has become an important tool for network engineers to provide enhanced security in company communications over the Internet. In the enterprise environment, hardware based Virtual Private Network (VPN) devices are available to handle hundreds of IPSec tunnels simultaneously. These expensive, high end solutions usually include advanced features to assist in the management of large numbers of connections and the authentication of many remote access users. In the smaller environment, however, businesses must choose cost effective solutions that are reliable and that can be quickly deployed to meet immediate needs. The OpenBSD project, the Linux FreeS/WAN project and Sonicwall Inc. have each produced cost effective IPSec implementations with excellent reliability. Collectively, they are helping to improve security for the small and medium sized organizations that comprise a large and growing portion of the Internet.

Due to the popularity of these solutions, it is not uncommon for interoperability to become an issue. The complexity of the IPSec protocol, the diversity between each solution's features and vast differences in device configuration can become a roadblock to setting up site to site VPN tunnels in a timely manner. Additionally, differing firewall configuration interfaces add to the challenge and therefore the viability of these otherwise useful technologies. This paper will briefly introduce these implementations and discuss some of the issues surrounding their interoperability. Additionally, a focused tutorial for quickly deploying these solutions is presented.

Assumptions

This discussion is aimed at network administrators and security analysts who are looking for information on how to implement VPN tunnels between the above named devices. A basic understanding of UNIX, networking and security fundamentals is assumed. On the other hand, because these devices are often deployed by novices, this paper will provide some high level discussion that the uninitiated reader may find informative. Also, links to supplementary or background information about a topic are periodically provided.

This paper will not cover the installation of operating systems, software, or hardware. These topics have been extensively discussed elsewhere any attempt to improve on them here would be redundant, exhausting and counter-productive to the purpose of this paper. Again, references to many of these documents will be included at logical points and should be examined if necessary.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Specifically, it is assumed that the reader has (at least) two working devices with (at least) two Ethernet adapters and with the operating systems and IPSec software installed and ready to configure.

Finally, note that it is not the goal of this paper to teach a specific operating system or general networking principles.

Before We Start

From this point on, the text will be mentioning specific technical terms and concepts that you will need to know in order to understand the discussion. For those readers with limited exposure to an applicable topic, I've included a short review of IPSec in the next section and I'm listing some pre-reading that may be useful below. See the Resources section for more information on these and other resources.

IP Security Architecture: http://www.ietf.org/rfc/rfc2401.txt

Networking Fundamentals: http://www.garykessler.net/library/tcpip.html

> **note: Once you have been introduced to TCP/IP, you are required by unwritten law to run (not walk) to the nearest bookstore and obtain a copy of W. Richard Stevens book, TCP/IP Illustrated Volume 1. If you work with networks and don't own this book, you are surely missing out.

UNIX Introduction: http://riceinfo.rice.edu/Computer/Documents/Unix/unix1.pdf

> **note: The single most useful and recommended UNIX book, in my experience, is the "UNIX System Administration Handbook" by Nemeth, Snyder, Seebass, Hein.

IPSec Review

While it is assumed that you have experience with IPSec, this is a brief review of basic concepts and terminology that are relevant to this discussion.

Internet Protocol Security (IPSec) is a collection of protocols designed to offer enhanced security features to IP (OSI Layer 3). These services can be broken into 4 main concepts:

Integrity: Data has not been modified during transit.

Authenticity: The source of the data can be trusted.

Replay protection: Data will only be processed once.

Confidentiality: Data can only be read by the intended recipient.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

IPSec uses the two traffic security protocols, authenticated header (AH) and encapsulating security protocol (ESP) to provide these services. AH provides integrity, authentication, and replay detection. ESP provides confidentiality, integrity, authentication, and replay detection. The IANA assigned protocol numbers 50 and 51 for AH and ESP, respectively.

IPSec traffic protocols can use either of two operation modes, transport or tunnel. In very simplistic terms, transport mode uses the original IP header of the packet while tunnel mode creates a new IP header. The new header (created in tunnel mode) contains the destination address of an IPSec gateway which, upon receipt, will strip away the new header, revealing the original IP header. The packet will then be forwarded to the originally intended destination. Interestingly, this is the basic premise for what has come to be known as a Virtual Private Network or VPN. The source and destination host do not have to be aware that they are communicating over secure channels, it happens transparently. But I digress.

For two hosts to successfully communicate using IPSec, they must agree upon a set of communication parameters. These parameters are defined in what is called a Security Association (SA). SAs are basically definitions of all parameters that apply to communicating with a particular host. Peer host address, encryption method, hashing algorithm, and related keys are among what is defined within an SA. Note that an SA is a description of a one way flow. Each IPSec communication will have two SAs, each describing the outgoing or incoming data flow.

IPSec SAs can be setup using manual keying or dynamically using Internet Key Exchange (IKE) protocol. IKE is considered to be the more secure of the two methods because keys are changed automatically and on a regular basis. IKE is a hybrid protocol that incorporates the previously published Oakley and SKEKE protocols under the framework of ISAKMP.

For the sake of simplicity, what you need to know is that IKE is used to provide secure authentication and negotiation services for SAs between IPSec peers. When prompted, IKE will contact the destination IPSec host using UDP port 500 and negotiate a key exchange. This is called a Phase 1 exchange. Its purpose is to authenticate and establish a secure connection so that subsequent traffic between the peers will be protected. Authentication occurs using one of various keying methods (e.g. pre-shared key, signed certificates, etc.). The Phase 1 exchange occurs using either main mode or aggressive mode. While Phase 1 is generally regarded as secure, there have been some concerns raised over its security in certain circumstances. Many persons familiar with IPSec on a technical level recommend avoiding the use of pre-shared keys with aggressive mode.

Once a secure communication channel is established by Phase 1, the Phase 2 (quick mode) negotiation will begin. Under Phase 2 the hosts will exchange one or more SA proposals to negotiate the transform under which the ESP and AH protocols will

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

operate. A transform is basically a cryptographic algorithm along with some extra information on how the algorithm will be used.

There are many configurable properties within the framework of IPSec. Ascertaining compatible values and configuring them correctly is the only way to successfully achieve interoperation. Here are some, often used, configurable settings:

Oakley Group - For both Phase 1 and Phase 2, an Oakley group is used for the Diffie Hillman key exchange. Most IPSec implementations offer a choice of 768 bit Group 1, 1024 bit Group 2, or 1536 bit Group 5. Note that Group 1 is considered by many to be insecure by today's standards.

Hashing Algorithm - For each phase a hashing algorithm is used. Typically these choices will include MD5 or SHA, but there are others.

Encryption Algorithm - Additionally, if confidentiality is desired (using ESP), an encryption algorithm must be used. These choices can vary greatly, but Data Encryption Standard (DES) and 3DES are usually available. DES is considered by many to be insecure because the time to break the encryption has dramatically shortened in the last decade. This leaves 3DES as the usable standard for IPSec interoperability. It should be noted that other algorithms such as Advanced Encryption Standard (AES) and Blowfish are considered to be very secure, yet provide faster performance than 3DES.

Perfect Forwarded Secrecy - PFS is a function of the Diffie Hillman key exchange that further protects encrypted data in the event that an encryption key is compromised. This is an optional layer of protection but usually must be set the same on both peers for Phase 2 to succeed.

In summary, what you need to know is this:

On each IPSec peer host you must:

1. Configure the IP addresses of the IPSec peers and/or protected networks.
2. Setup authentication between peers (pre-shared secret, signed certificates, etc.)
3. Define compatible settings for Phase 1.
4. Define compatible settings for Phase 2.

When using packet filtering:

1. If using AH, you must allow protocol 51 in and out.
2. If using ESP, you must allow protocol 50 in and out.
3. If using automatic keying (IKE), you must allow UDP port 500 in and out.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The Players

Before we try to get OpenBSD, Linux, and Sonicwall talking, let's take a brief look at each of these solutions. Each is unique and has differences we need to consider before we start. I will present a short introduction to each one and then include a feature table for more detail and as a quick reference.

OpenBSD

The OpenBSD operating system can be purchased from the OpenBSD project at http://www.openbsd.org for a reasonable cost (to support development) or even installed over the internet for free.

OpenBSD is an easy to install and exceptionally stable operating system. It runs on many platforms and has an exceptional record for security. The developers have spent thousands of hours auditing and rewriting code to the effect of releasing an OS that has had "only one remote hole in the default install, in more than 7 years" (OpenBSD Project). These characteristics make it an exceptional choice for an Internet gateway or LAN router and I've used it successfully many times in recent years.

IPSec has been tightly integrated into the OpenBSD operating system to meet the goals of performance, scalability, and portability. OpenBSD includes support for many encryption algorithms and other advanced cryptographic features that are not included in most other implementations. On the other hand, you will not find "bleeding edge" features such as NAT Traversal or Dead Peer Detection in the current OpenBSD software. When these features are finalized by the Internet Engineering Task Force (IETF) and/or found to be stable and secure technologies, you are then likely to find them fully implemented by the OpenBSD developers.

Sonicwall

Sonicwall firewall appliances are produced by Sonicwall Inc. and are available from many online retailers of computer and networking products. Depending on your specific needs, there are many models and add-on features to choose. See http://www.sonicwall.com for more information.

The Sonicwall device is easy to use, reliable, and inexpensive when compared to many "enterprise" level products. These devices make it possible for novice network administrators to quickly deploy DMZs, advanced NAT configurations, IPSec VPNs, content filtering, enforced anti-virus, VPN user radius authentication, and a lot more. Since the initial release, Sonicwall has regularly updated and extended their software features to the point where it is now competing with the more expensive enterprise offerings.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The Sonicwall IPSec implementation is focused on interoperation with other Sonicwall devices and is therefore limited to using pre-shared keys for authentication to most other IPSec gateways. Sonicwall recently added a "third party certificates" option for VPN authentication, but at this time it only works with other Sonicwall devices (according to Sonicwall support).

FreeS/WAN

FreeS/WAN is available at http://www.freeswan.org. Additionally, packages are available for many Linux distributions such as RedHat, Mandrake, Debian, and SuSE.

Commonly referred to as Freeswan, this has become the most common IPSec implementation used on the Linux operating system. One of the primary goals of the Freeswan project is a feature called Opportunistic Encryption (OE). This feature allows secure communication with unknown hosts over the Internet using IPSec by publishing public keys in your DNS record. These connections are setup transparently during normal TCP/IP communications and offer a way to easily protect against packet sniffing via intermediary hosts on the Internet. Currently you need Freeswan on both hosts for OE to work, but if the technology catches on, it could go a long way toward improving privacy in Internet communications.

The Freeswan project currently comes in two versions, the original and another project called "Super FreeS/WAN". Super Freeswan includes various patches that aren't considered secure or "ready" by the primary Freeswan developers. Because of this, support for original (single) DES and DH Group 1 is not included in Freeswan. However, by using various patches or Super Freeswan you can gain access to these and other features that you may need for specific applications or to interoperate with other devices. Overall Freeswan is considered to be a stable and mature IPSec offering and many companies and individuals use it successfully in production environments.

		OpenBSD	FreeS/WAN	Sonicwall
Encryption	DES	X	P	X
	3DES	X	X	X
	AES	X	P	
	CAST	X		
	Blowfish	X		
Hashing				
	MDS	X	X	X
	SHA1	X	X	X
Other				
	Aggressive Mode	X		X
	PFS	X	X	X
	NAT/T		P	X
	DHCP over IKE	X	P	X
	X509 Certificates	X	P	L
X = Supported P = Patch L = Limited				

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Preparing

The following configurations will use:

3DES encryption
SHA hashing
Perfect Forwarding Secrecy (PFS)
Pre-shared secret authentication

These were chosen specifically because all 3 gateways support them and because this simplifies the presentation. Feel free to experiment with different settings depending upon your environment and personal preferences.

Make sure that each gateway is properly installed and that there is connectivity established between each device.

If you have any problems while trying to get your boxes configured correctly, please see the Resources section for documentation, email lists and other resources to help you.

To demonstrate interoperability between these IPSec solutions, we will use a simple network architecture consisting of three Internet gateway devices connected to a common network segment which will represent the "hostile" network. Each of these devices will have a second Ethernet interface attached to its own "private" network. All devices will be configured to forward packets between its attached segments. Network Address Translation (NAT) and packet filtering will also be configured on all devices. *see figure 1*

For this exercise, the following hardware and software was used:

OpenBSD: OpenBSD 3.3 stable on a Pentium II, 550MHz w/128MB RAM. A base install was performed without X or games packages. The hard drive was partitioned with 1GB /, 400MB /var, and 128MB Swap.

> **note: I have successfully used OpenBSD on a 486 DX2 w/32 MB RAM as an Internet gateway and VPN box. Check the install documentation to make sure your network cards (and other equipment) are supported.

Linux / FreeS/WAN: Mandrake 9.1 on an Athlon XP 2.0, w/512MB RAM. A "minimal" install with system documentation was performed without X. The hard drive was partitioned with 3GB /, 400MB /var, and 512MB Swap. Mandrake was chosen because it has a decent "minimal" install option, the kernel is pre-patched with the Freeswan extensions, and Freeswan packages are included on the installation CDs. All commands demonstrated in this tutorial will also work correctly on Red Hat Linux.

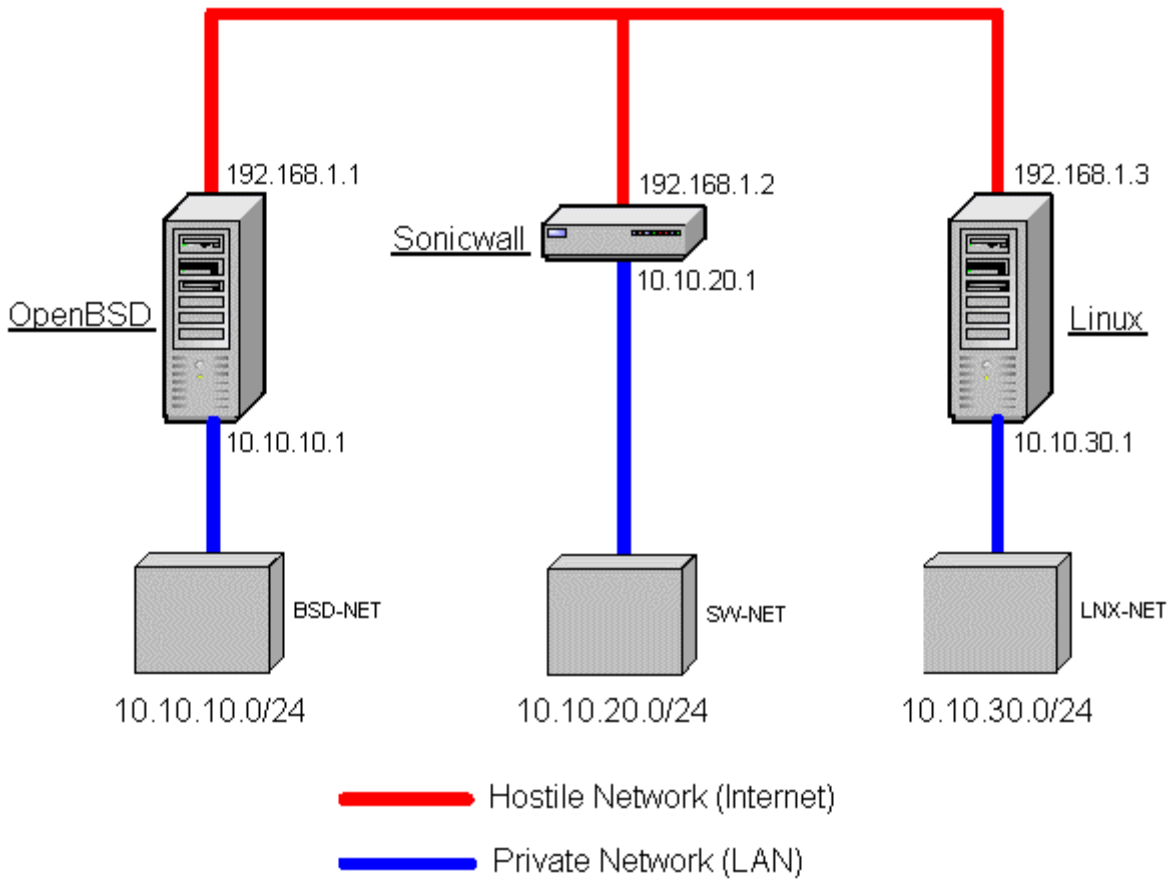
> **note: A faster Pentium with 64MB RAM would likely do well in this application.

Sonicwall: Sonicwall TELE3 166Mhz rev. 6.5.0.2 w/16MB RAM. This device supports up to 6 IPSec SAs (including 1 built-in for remote admin).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

**note: Some less expensive models require an upgrade to enable IPSec support. It's preferable to have a firmware revision of 6.4.0.0 or greater due many to IPSec bugs that were fixed in previous versions.

[figure.1] Diagram of the Test Network



OpenBSD and Sonicwall

OpenBSD

IPSec is enabled by default in the standard kernel that ships with OpenBSD. Verify that ESP protocol is enabled with the "sysctl" command.

```
openbsd# sysctl net.inet.esp.enable
net.inet.esp.enable = 1
```

If you need to, use the "grep" command to verify that ESP is not disabled in your /etc/sysctl.conf file. It should appear as below.

```
openbsd# grep esp /etc/sysctl.conf
#net.inet.esp.enable=0
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Since we are setting up a gateway, IP forwarding must be enabled.

Uncomment the “net.inet.ip.forwarding=1” line in /etc/sysctl.conf. You may either reboot or issue the following command to turn on IP forwarding.

```
openbsd# sysctl -w net.inet.ip.forwarding=1
net.inet.ip.forwarding: 0 -> 1
```

Make sure that isakmpd is configured to start automatically upon boot.

```
openbsd# grep isakmpd /etc/rc.conf
isakmpd_flags=""        # for normal use: ""
```

Now we will create the files necessary to configure IKE. The isakmpd.conf file is used to define your connections (SAs). The isakmpd.policy file is used to define authentication policies for IKE. Create these two files now.

```
openbsd# umask 177; touch /etc/isakmpd/isakmpd.conf
openbsd# touch /etc/isakmpd/isakmpd.policy
```

Verify that each file can only be read or written to by the superuser account.

```
openbsd# cd /etc/isakmpd; ls -l isakmpd.*
-rw-------  1 root  wheel  1429 Jul 21 15:39 isakmpd.conf
-rw-------  1 root  wheel   207 Jun 21 14:07 isakmpd.policy
```

isakmpd.conf uses the .INI style format which may be familiar to you. The following is the isakmpd.conf file used for the lab setup outlined earlier. Comments have been added to clarify and explain certain sections. Choose your own secret and read “man isakmpd.conf” before attempting to adopt this file to your application.

```
# /etc/isakmpd/isakmpd.conf

# This section outlines global settings for all connections.
# Set the Listen-on= value to the IP address on the external
# interface.
[Global]
Retransmits=            5
Exchange-max-time=      120
Listen-on=              192.168.1.1

# Begin the SA definitions.

# Define the IP address of your Sonicwall gateway and give the
# gateway a nametag for the Phase 1 definition.
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
[Phase 1]
192.168.1.2=            sonicwall

# Create a nametag for the Phase 2 definition(s).
[Phase 2]
Connections=            obsd-sonicwall

# Define Phase 1 parameters including our pre-shared key.
[sonicwall]
Phase=                  1
Transport=              udp
Local-address=          192.168.1.1
Address=                192.168.1.2
Configuration=          Default-main-mode
Authentication=         !Sst@^CtO8$3@46Rh)%

# Define Phase 2 parameters including where to find the
# definitions of our protected networks.
[obsd-sonicwall]
Phase=                  2
ISAKMP-peer=            sonicwall
Configuration=          Default-quick-mode
Local-ID=               BSD-NET
Remote-ID=              SW-NET

# Define the network connected to OpenBSD
[BSD-NET]
ID-type=                IPV4_ADDR_SUBNET
Network=                10.10.10.0
Netmask=                255.255.255.0

# Define the network connected to Sonicwall
[SW-NET]
ID-type=                IPV4_ADDR_SUBNET
Network=                10.10.20.0
Netmask=                255.255.255.0

# Define main mode (ID_PROT) for Phase 1 and transform(s).
[Default-main-mode]
DOI=                    IPSEC
EXCHANGE_TYPE=          ID_PROT
Transforms=             3DES-SHA-GRP2

# Define Phase 2 transform(s).
[Default-quick-mode]
DOI=                    IPSEC
```

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-SHA-PFS-GRP2-SUITE

# END
```

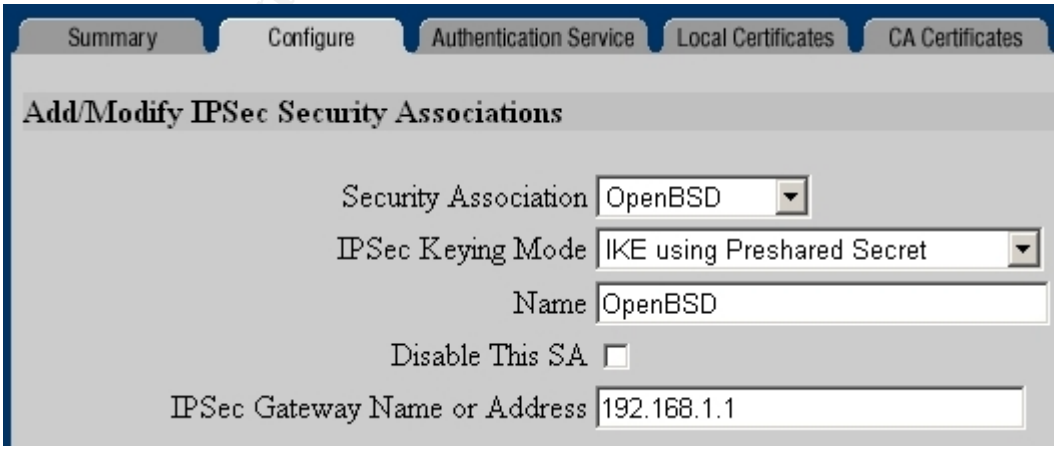
Next we need to define a simple policy in our isakmpd.policy file. This policy allows any connection with a valid pre-shared secret. Read “man isakmpd.policy” for more information.

```
# /etc/isakmpd/isakmpd.policy
Comment: Policy == any correct password is OK.
Authorizer: "POLICY"
Conditions: app_domain == "IPsec policy" &&
                esp_present == "yes" &&
                esp_enc_alg != "null" -> "true";
# END
```

This concludes the IPSec setup for the OpenBSD side. If you’re using pf to filter packets (and you should be), see OpenBSD in the Firewall Considerations section.

Sonicwall

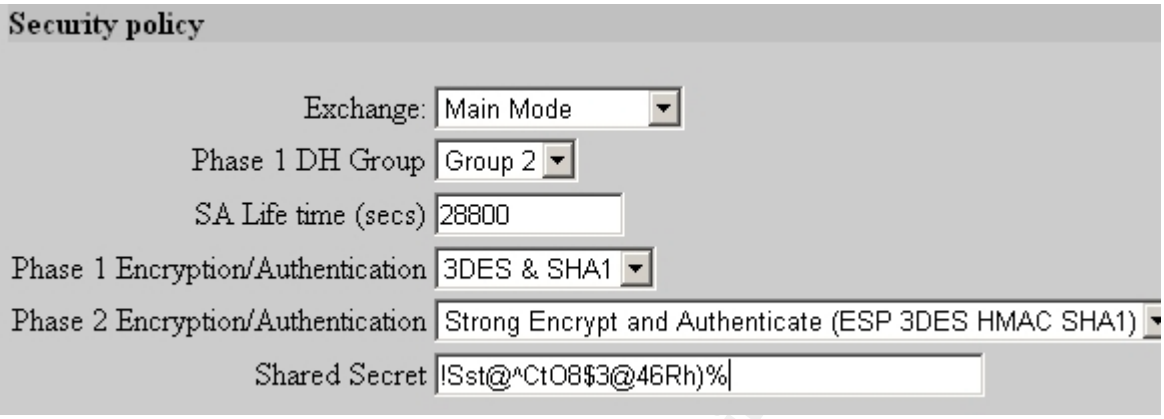
1. Login to the Sonicwall.
2. Go to the VPN page.
3. Make sure that “Enable VPN” is checked.
4. Go to the “Configure” tab.
5. Select “Add new SA” from the Security Association selector.
6. Choose “IKE using Preshared Secret” for “IPSec Keying Mode”.
7. Choose a descriptive name for this SA.
8. Make sure the “Disable this SA” box is unchecked.
9. Enter the IP Address of the OpenBSD gateway (e.g. 192.168.1.1).



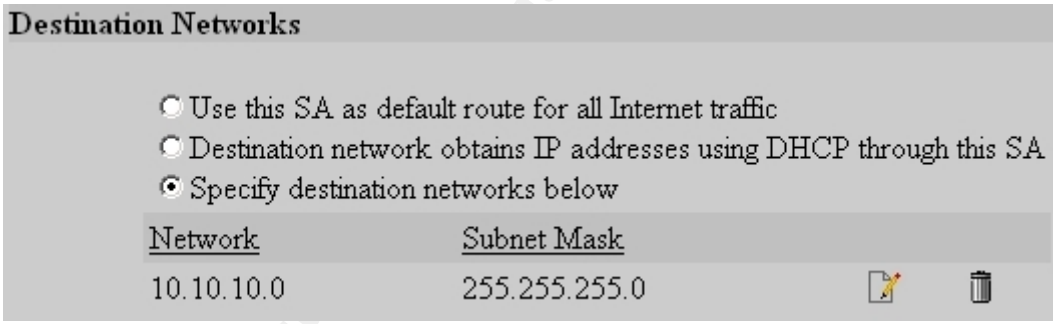
10. In the Security policy section, choose Main Mode for “Exchange”.
11. For Phase 1 DH Group choose “Group 2”.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

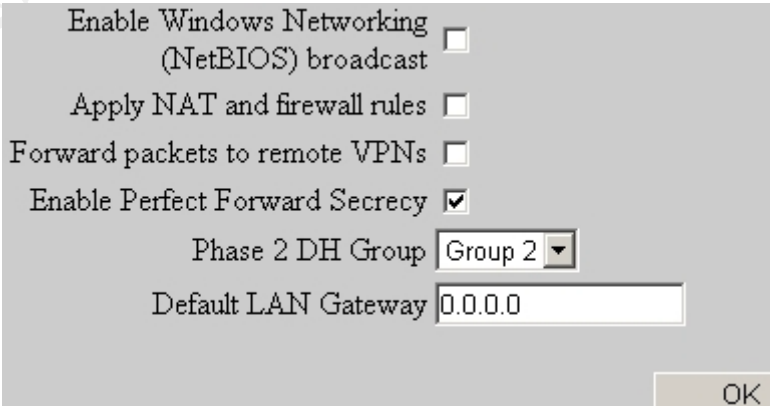
12. Set SA Life time to desired value (e.g. 28800).
13. Choose “3DES & SHA1” for Phase 1 Enc/Auth.
14. Choose “ESP 3DES HMAC SHA1” for Phase 2 Enc/Auth.
15. Enter your Shared Secret (e.g. !Sst@^CtO8$3@46Rh)%).



16. For “Destination Networks” select “Specify destination networks below”.
17. Click on “Add New Network”.
18. When the “Edit VPN Destination Network” box appears, enter the private network and its subnet mask behind the OpenBSD gateway. Click OK.



19. Click the “Advanced Settings” button.
20. When the “Edit Advanced Settings” box appears, check the “Enable Perfect Forward Secrecy” box. Choose “Group 2” for Phase 2 DH Group. Click OK.



© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

21. Click the “Update” button on the bottom of the Configuration page.

The Sonicwall configuration is complete. The Sonicwall will automatically add the necessary rules to pass the ESP and IKE traffic properly.

Now, bring up the connection. First check to make sure that isakmpd is running.

```
openbsd# ps ax |grep isakmpd
27358 ??  Is      1:37.81 isakmpd
```

If not, just type “isakmpd” at the command prompt (as root). When isakmpd starts, it creates a fifo interface at /var/run/isakmpd.fifo. Commands that we direct there, will control the daemon. See “man isakmpd” for more information.

```
openbsd# echo "c obsd-sonicwall" > /var/run/isakmpd.fifo
```

Let’s ping to see if our connection is active. We will need to specify our internal interface on the command line because our SA only applies to packets originating from the private network. You will need to create a second SA on each peer if you need direct connectivity from one to the other over the tunnel. I usually don’t because both ping and ssh allow you to specify the internal address using the –I and –b switches respectively.

```
openbsd# ping -I 10.10.10.1 10.10.20.1
PING 10.10.20.1 (10.10.20.1): 56 data bytes
64 bytes from 10.10.20.1: icmp_seq=0 ttl=64 time=16.623 ms
64 bytes from 10.10.20.1: icmp_seq=1 ttl=64 time=11.380 ms
64 bytes from 10.10.20.1: icmp_seq=2 ttl=64 time=13.074 ms
...
```

Linux and Sonicwall

Linux

Make sure that IP forwarding is enabled in your startup scripts. The “`net.ipv4.ip_forward`“ value must be set to “1”.

```
linux# grep ip_forward /etc/sysctl.conf
net.ipv4.ip_forward=1
```

Reboot or use “sysctl” to enable forwarding if the above setting was changed:

```
# sysctl -w net.ipv4.ip_forward=1
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Start the KLIPS (kernel IPSec) and Pluto (IKE) processes with the ipsec init script.

```
linux# service ipsec start
Starting IPsec
```

Make sure that Freeswan is installed correctly using the “ipsec verify” command. You can safely ignore failed DNS checks.

```
linux# ipsec verify
Checking your system to see if IPsec got installed and started
correctly
Version check and ipsec on-path                                 [OK]
Checking for KLIPS support in kernel                            [OK]
Checking for RSA private key (/etc/freeswan/ipsec.secrets)      [OK]
Checking that pluto is running                                  [OK]
```

If you have problems, read this URL:
http://www.freeswan.org/freeswan_trees/freeswan-1.99/doc/trouble.html#install.check

Configure Freeswan to run at all “active” run levels (whether you use them or not). Use the “chkconfig” command to set and verify when the init script will start Freeswan.

```
linux# chkconfig --level 2345 ipsec on
linux# chkconfig --list ipsec
ipsec      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Locate and edit the ipsec.conf file with the following:

```
# /etc/freeswan/ipsec.conf

# basic configuration
config setup
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes

conn sonicwall
        authby=secret
        left=192.168.1.2
        leftsubnet=10.10.20.0/24
        leftnexthop=%defaultroute
        right=192.168.1.3
        rightsubnet=10.10.30.0/24
        auto=start
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
# END
```

Next, authentication must be setup in the ipsec.secrets file. This file holds our pre-shared secrets and tells Freeswan when to use them. You must include the local and remote external IP addresses, the PSK designator, and the secret (in quotes). Choose your own secret.

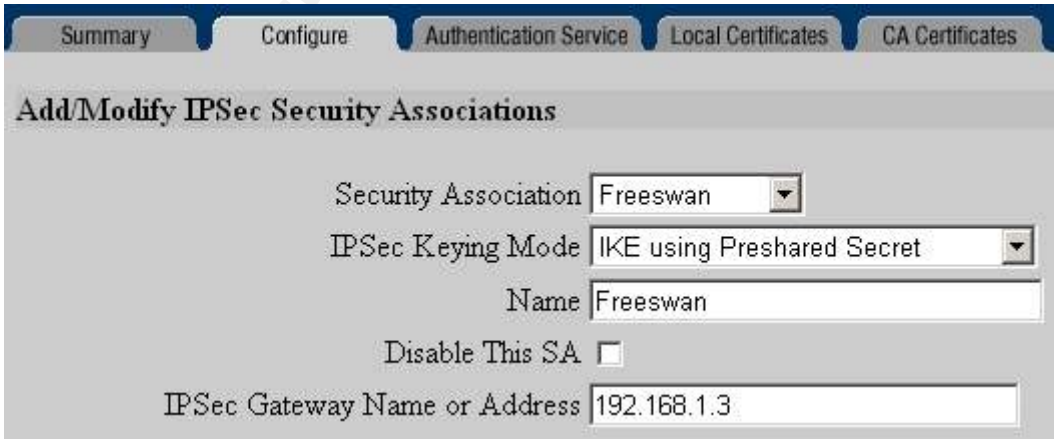
```
# /etc/Freeswan/ipsec.secrets

192.168.1.3 192.168.1.2 : PSK "%&Hs0#2k81S_qB!^v*7"
```

The Linux side is now setup, please refer to the Firewall Considerations section for information on setting up Linux “iptables” correctly.

Sonicwall

The Sonicwall configuration is identical to the previous example with OpenBSD except for the IP addressing settings and the pre-shared secret.

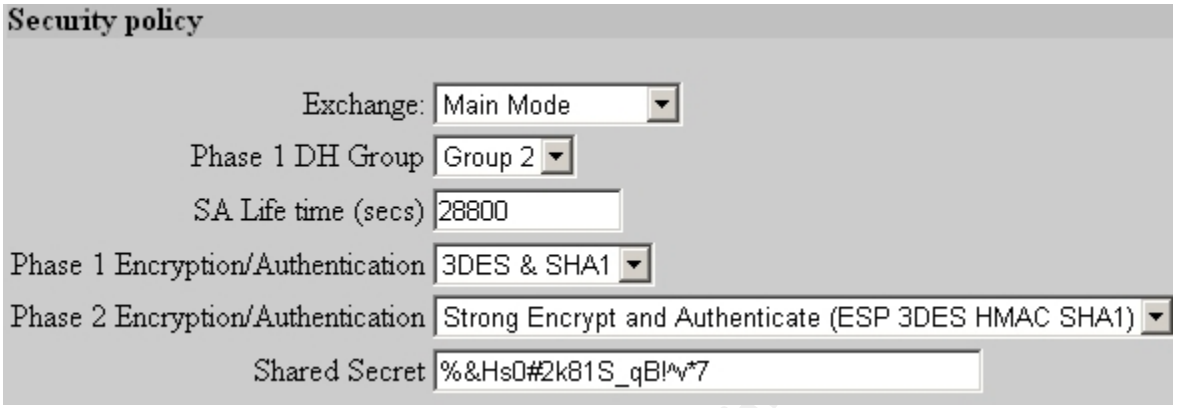
1. Login to the Sonicwall.
2. Go to the VPN page.
3. Make sure that “Enable VPN” is checked.
4. Go to the “Configure” tab.
5. Select “Add new SA” from the Security Association selector.
6. Choose “IKE using Preshared Secret” for “IPSec Keying Mode”.
7. Choose a descriptive name for this SA.
8. Make sure the “Disable this SA” box is unchecked.
9. Enter the IP Address of the Linux gateway (e.g. 192.168.1.3).



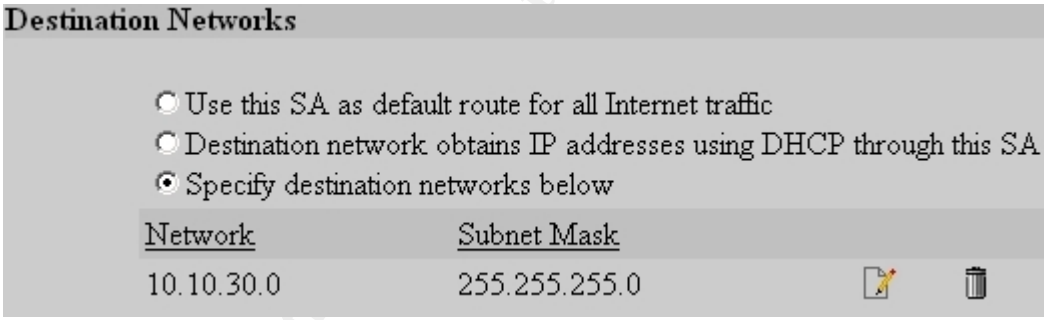
10. In the Security policy section, choose Main Mode for “Exchange”.
11. For Phase 1 DH Group choose “Group 2”.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

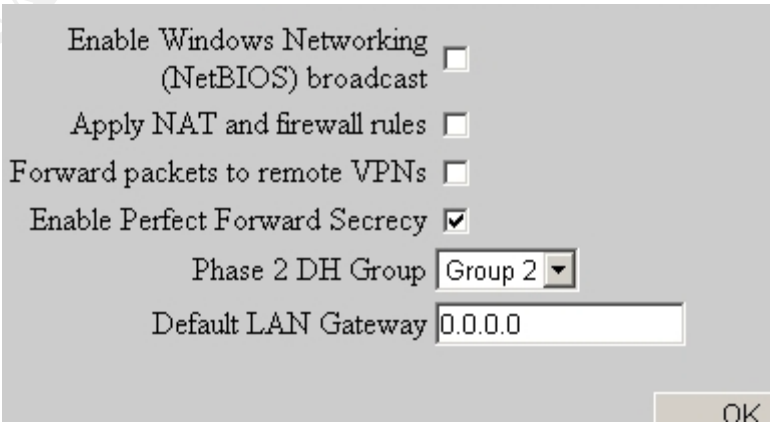
12. Set SA Life time to desired value (e.g. 28800).
13. Choose “3DES & SHA1” for Phase 1 Enc/Auth.
14. Choose “ESP 3DES HMAC SHA1” for Phase 2 Enc/Auth.
15. Enter your Shared Secret (e.g. %&Hs0#2k81S_qB!^v*7).



16. For “Destination Networks” select “Specify destination networks below”.
17. Click on “Add New Network”.
18. When the “Edit VPN Destination Network” box appears, enter the private network and its subnet mask behind the Linux gateway. Click OK.



19. Click the “Advanced Settings” button.
20. When the “Edit Advanced Settings” box appears, check the “Enable Perfect Forward Secrecy” box. Choose “Group 2” for Phase 2 DH Group. Click OK.



© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

21. Click the “Update” button on the bottom of the Configuration page.

The Sonicwall configuration is complete. The Sonicwall will automatically add the necessary rules to pass the ESP and IKE traffic properly.

To bring up the connection, use the “ipsec auto” command from the Linux root prompt:

```
linux# ipsec auto --up sonicwall
112 "sonicwall" #11: STATE_QUICK_I1: initiate
004 "sonicwall" #11: STATE_QUICK_I2: sent QI2, IPsec SA
established
```

Test the connection:

```
linux# ping -I eth1 10.10.20.1
PING 10.10.20.1 (10.10.20.1) from 10.10.30.1 eth1: 56(84) bytes
of data.
64 bytes from 10.10.20.1: icmp_seq=1 ttl=255 time=0.399 ms
64 bytes from 10.10.20.1: icmp_seq=2 ttl=255 time=0.196 ms
64 bytes from 10.10.20.1: icmp_seq=3 ttl=255 time=0.177 ms
...
```

OpenBSD and Linux

This section contains information on how to add the third tunnel. Please refer to the previous sections for more detailed information on how to set up each IPSec configuration. Information that is to be added will appear in black and existing lines will appear grayed out.

OpenBSD

Add the following lines to your isakmpd.conf file.

```
# /etc/isakmpd/isakmpd.conf

[Global]
Retransmits=            5
Exchange-max-time=      120
Listen-on=              192.168.1.1

[Phase 1]
192.168.1.2=        sonicwall
192.168.1.3=        linux
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
[Phase 2]
Connections=            obsd-sonicwall,obsd-linux

[obsd-sonicwall]
Phase=                  2
ISAKMP-peer=            sonicwall
Configuration=          Default-quick-mode
Local-ID=               BSD-NET
Remote-ID=              SW-NET

[obsd-linux]
Phase=                  2
ISAKMP-peer=            linux
Configuration=          Default-quick-mode
Local-ID=               BSD-NET
Remote-ID=              LINUX-NET

[sonicwall]
Phase=                  1
Transport=              udp
Local-address=          192.168.1.1
Address=                192.168.1.2
Configuration=          Default-main-mode
Authentication=         openbsd_vpn

[linux]
Phase=                  1
Transport=              udp
Local-address=          192.168.1.1
Address=                192.168.1.3
Configuration=          Default-main-mode
Authentication=         HUu&^$38D3=ksI(*4Y9

[BSD-NET]
ID-type=                IPV4_ADDR_SUBNET
Network=                10.10.10.0
Netmask=                255.255.255.0

[SW-NET]
ID-type=                IPV4_ADDR_SUBNET
Network=                10.10.20.0
Netmask=                255.255.255.0

[LINUX-NET]
ID-type=                IPV4_ADDR_SUBNET
Network=                10.10.30.0
Netmask=                255.255.255.0
```

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
[Default-main-mode]
DOI=                    IPSEC
EXCHANGE_TYPE=          ID_PROT
Transforms=             3DES-SHA-GRP2

[Default-quick-mode]
DOI=                    IPSEC
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-SHA-PFS-GRP2-SUITE

# END
```

Add the new connection to your ipsec.conf file.

```
# /etc/freeswan/ipsec.conf

# basic configuration
config setup

        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes

conn sonicwall
        authby=secret
        left=192.168.1.2
        leftsubnet=10.10.20.0/24
        leftnexthop=%defaultroute
        right=192.168.1.3
        rightsubnet=10.10.30.0/24
        auto=start

conn openbsd
        authby=secret
        left=192.168.1.1
        leftsubnet=10.10.10.0/24
        leftnexthop=%defaultroute
        right=192.168.1.3
        rightsubnet=10.10.30.0/24
        auto=start

#END
```

© SANS Institute 2003, Author retains full rights

Choose a secret and add a new entry to ipsec.secrets:

```
# /etc/Freeswan/ipsec.secrets

192.168.1.3 192.168.1.2 : PSK "%&Hs0#2k81S_qB!^v*7"
192.168.1.3 192.168.1.1 : PSK "HUu&^$38D3=ksI(*4Y9"

# END
```

Bring up the connection from Linux:

```
# ipsec auto --up openbsd
112 "openbsd" #15: STATE_QUICK_I1: initiate
004 "openbsd" #15: STATE_QUICK_I2: sent QI2, IPsec SA
established
```

Test the connection:

```
linux# ping -I eth1 10.10.10.1
PING 10.10.10.1 (10.10.10.1) from 10.10.30.1 eth1: 56(84) bytes
of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=255 time=0.399 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=255 time=0.196 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=255 time=0.177 ms
...
```

Firewall Considerations

Since each of these boxes is acting as the Internet gateway for a private network, they should be packet filtering all network traffic. Each of them provides there own interface to create and manage its filter rules. This section will list exactly what traffic needs to be passed by the firewalls and then how to configure each gateway.

Each example accomplishes the following:

1. Block all inbound traffic on external interface by default.
2. Allow all traffic from our internal (private) network.
3. Allow Encrypted packets (ESP).
4. Allow IKE packets (UDP 500).
5. Allow inbound SSH connections.
6. Allow ping packets.
7. Allow traffic to and from IPSec tunnels.
8. NAT traffic from internal network when exiting external interface.

These simple examples are stripped down in order to demonstrate basic rules and their syntax. These rules do not provide any logging or spoofing protection and very little protection against common firewall attacks. Additionally, they are much too relaxed for most environments.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Do not use these rules in a production environment. You have been warned.

Since the Sonicwall automatically creates corresponding filtering rules during its initial setup and as you configure your IPSec connections, I will not cover it here.

OpenBSD pf

OpenBSD uses "pf" (packet filter) to control network connections. Your /etc/rc.conf file must include the following minimum configuration.

```
openbsd# grep pf /etc/rc.conf
pf=YES                          # Packet filter / NAT
pf_rules=/etc/pf.conf           # Packet filter rules file
pflogd_flags=                   # add more flags, ie. "-s 256"
```

Copy the following example into your /etc/pf.conf file. Make a backup of the original file first as it contains good information that you may need to create a more suitable rule set later.

```
# SAMPLE RULES - NOT FOR PRODUCTION USE
# Required order:
# options, normalization, queueing, translation, filtering.
# Translation rules are first match while filter rules are last.

ext_if="sis0"
int_if="dc0"
int_net="10.10.10.1/24"

# Normalize packets
scrub in all

# NAT
nat on $ext_if from $int_net to any -> ($ext_if)

# Default rules.
block in all
pass out all keep state

# Allow inbound SSH
pass in proto tcp from any to any port 22 keep state

# Allow all traffic from the internal network.
pass in on $int_if all keep state

# Allow ping
pass in inet proto icmp all icmp-type 8 code 0 keep state
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
# Allow ESP and IKE
pass in on $ext_if proto esp all
pass in on $ext_if proto udp from any to any port 500

# Allow traffic inbound from tunnels
pass in on enc0 all

# SAMPLE RULES - NOT FOR PRODUCTION USE
# END
```

After you have created the file, make sure that "pf" is enabled and then apply the new rules.

```
openbsd# pfctl -e
openbsd# pfctl -F all -f /etc/pf.conf
rules cleared
nat cleared
altq cleared
states cleared
pf: statistics cleared
0 tables deleted.
```

If you need help or more information, please use the following resources:

The PF man pages: `pfctl(8)` and `pf.conf(5)`
The PF FAQ and user guide: http://www.openbsd.org/faq/pf/index.html
The PF email list and archive: http://www.benzedrine.cx/pf/

Linux iptables

Linux uses "iptables" (Netfilter) to perform its packet filtering. First make sure that iptables is setup to start upon boot:

```
linux# chkconfig --level 2345 iptables on
linux# chkconfig --list iptables
iptables   0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Next, save the following script as /root/firewall.rc

```
#!/bin/sh
# SAMPLE RULES - NOT FOR PRODUCTION USE

LAN_NET='10.10.30.0/24'
LAN_INT='eth1'
WAN_INT='eth0'
VPN_INT='ipsec0'
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
# Flush all tables
iptables -t nat -F POSTROUTING
iptables -t nat -F PREROUTING
iptables -t nat -F OUTPUT
iptables -F

# Default rules
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Allow SSH from anywhere
iptables -A INPUT -j ACCEPT -p tcp --dport 22

# MASQ and Forwarding (w/state)
# We must specify –o $WAN_INT in the nat rule so that
# packets bound for ipsec tunnels are not modified
iptables -t nat -A POSTROUTING -s $LAN_NET -o $WAN_INT –j \
MASQUERADE
iptables -A FORWARD -j ACCEPT -i $LAN_INT -s $LAN_NET
iptables -A FORWARD -m state --state ESTABLISHED,RELATED –j \
ACCEPT

# Allow ping traffic for troubleshooting
iptables -A INPUT -j ACCEPT -p icmp --icmp-type echo-reply
iptables -A INPUT -j ACCEPT -p icmp --icmp-type echo-request

# Allow ESP and IKE
iptables -A INPUT -j ACCEPT -p 50
iptables -A INPUT -j ACCEPT -p udp --dport 500

# Allow vpn tunnel traffic
iptables -A FORWARD -j ACCEPT -i $VPN_INT

# Allow active connections back in (router state)
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# SAMPLE RULES – NOT FOR PRODUCTION USE
# END
```

Once the file is created, simply run it from the command line:

```
linux# sh /root/firewall.rc
```

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Make sure that everything is working properly and if so, save your new rules for use during startup.

```
linux # service iptables save
Saving current rules to /etc/sysconfig/iptables:        [  OK  ]
```

If you need help or more information, please use the following resources:

The iptables man pages:
`iptables(8), iptables-save(8), iptables-restore(8)`
The Netfilter Documentation: http://www.netfilter.org/documentation/
The Netfilter email list and archives: http://www.netfilter.org/contact.html#list

Conclusion

From the previous discussion and tutorial, it is evident that interoperability between OpenBSD, Linux and Freeswan is possible. Additionally, these VPN solutions can be implemented in a timely manner by persons familiar with UNIX and Networking basics. Please review the resources section for more information about these topics and how to utilize them in your environment.

© SANS Institute 2003, Author retains full rights

Resources

Brad. "OpenBSD Platforms." OpenBSD Project. v1.49 15 June 2003.
URL: http://www.openbsd.org/plat.html (21 June, 2003).

Miod. "OpenBSD Security." OpenBSD Project. v1.248 1 June 2003.
URL: http://www.openbsd.org/security.html (21 June 2003).

Jason. "OpenBSD Cryptography." OpenBSD Project. v1.121 16 June 2003.
URL: http://www.openbsd.org/crypto.html (21 June 2003).

David. "OpenBSD." OpenBSD Project. v1.425 17 June 2003
URL: http://openbsd.org (21 June 2003).

Keromytis, Angelos D. "Implementing Internet Key Exchange (IKE)" 4 May 2000 URL: http://www.usenix.org/publications/library/proceedings/usenix2000/freenix/full_papers/hallqvist/hallqvist_html/ikepaper.html (20 June 2003)

"Netfilter Documentation." The Netfilter Project.
URL: http://www.netfilter.org/documentation/ (22 June 2003).

"Netfilter Mailing List." The Netfilter Project.
URL: http://www.netfilter.org/contact.html#list (22 June 2003).

Nick. "PF FAQ." OpenBSD Project. v1.7 29 June 2003.
URL: http://www.openbsd.org/faq/pf/index.html (22 June 2003)

"PF Mailing List" OpenBSD Project.
URL: http://www.benzedrine.cx/pf/ (22 June 2003)

"Linux FreeS/WAN Troubleshooting Guide." Linux FreeS/WAN Project. URL: http://www.freeswan.org/freeswan_trees/freeswan-1.99/doc/trouble.html#install.check (20 June 2003)

"Linux FreeS/WAN Project"
URL: http://www.freeswan.org (20 June 2003)

Kent, S. "Security Architecture for the Internet Protocol." November 1998.
URL: http://www.ietf.org/rfc/rfc2401.txt (19 June 2003)

Kessler, Gary. "An Overview of TCP/IP Protocols." 25 March 2003
URL: http://www.garykessler.net/library/tcpip.html (22 June 2003)

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

"Introduction to the UNIX Operating System on IT Systems." 11 July 2002
URL: http://riceinfo.rice.edu/Computer/Documents/Unix/unix1.pdf (22 June 2003)

"The Linux Documentation Project." 10 June 2003
URL: http://www.tldp.org (22 June 2003)

© SANS Institute 2003, Author retains full rights

Last Updated: May 23rd, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced