

[[about me](#)] [[geekery](#)] [[pix](#)] [[resume](#)] [[blog](#)]

This document describes how to get [Openswan](#) working with various other IPsec stacks, including Openswan and Windows 2000/XP. If you have any difficulties with this process, please e-mail the [Openswan mailing list](#), or if you can't get help from there, e-mail me at: ipsec@natecarlson.com. If you are using clients which benefit from receiving an IP address on the remote network (Windows, PocketPC, etc), you may want to consider L2TP over IPsec instead of the method described below. [Jacco de Leeuw's pages](#) cover this in a good amount of detail; I also have a basic walkthrough available at [my L2TP-over-IPsec page](#).

If you're not sure if IPsec is right for you, I have written a quick document about some of the various types of VPN available under Linux. It is available at: <http://www.natecarlson.com/linux/linux-vpn.php>. I hope this helps clear up some questions.

IMPORTANT NOTE: On March 1, 2004, the FreeS/WAN maintainers announced that the FreeS/WAN project is ending, for many reasons. The [Openswan project](#) is going to be taking over development. Openswan is based on Super FreeS/WAN, and already includes most of the patches that people wanted. I've updated these directions to use examples for Openswan 2.1.2; they should still run as-is on FreeS/WAN 2.0 with the X.509 patches, and will work with FreeS/WAN 1.99+X.509 and Openswan 1 with some minor modifications. They should also work as-is with Strongswan. I no longer cover patching FreeS/WAN with X.509; if you are going to start with a base FreeS/WAN installation, you will need to follow the directions at <http://www.strongsec.com/freeswan> on how to patch it.

IMPORTANT NOTE #2: As of June 17 2004, this document has been updated to reflect Openswan configuration instead of FreeS/WAN. I've also reorganized a few things; hopefully it will flow better now. Please let me know if you run into any problems with the new configuration. If you need it, the old page is available at: <http://www.natecarlson.com/linux/ipsec-x509-fs1.php>.

NOTE #3: Not nearly as important as above, but just wanted to note that I do occasionally post notes about new VPN options and such on my blog; see the VPN category at: <http://www.natecarlson.com/blog/category/geek-stuff/vpn>. Also, if you are interested in consulting services to help you set things up, I am available on a very limited basis - please see my [consulting page](#).

Contents:

[Changes made to this document](#)
[Setting up a Certificate Authority](#)
[Generating a Certificate](#)
[Installing Openswan](#)
[Installing the Certificate on your Gateway](#)
[Configuring Openswan on the Gateway Machine](#)
[Client Setup: Openswan](#)
[Client Setup: Windows 2000/XP with ipsec.exe](#)
[Some common errors, and resolutions for them](#)
[References used to write this document](#)

If you find this page helpful, and would like to help keep me motivated to update this site, feel free to donate with the button below. Any little bit helps!



Also, if you are looking for inexpensive webhosting with lots of features, check out 1&1 - they do a great job!



[Digital Dictate Software](#) - Professional dictation recorder for Windows and Mac. Free Download. www.nch.com.au/express

Referral Ads by Google

Changes made to this document

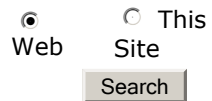
\$Id: ipsec-x509.php,v 1.46 2007-10-30 23:47:34 natecars Exp \$\br/>
 [03/18/05] Added config to ipsec.conf entries to disable OE
 [02/07/05] Update from Jacco regarding new ipseccmd.exe for sp2
 [06/17/04] Major updates to cover Openswan 2.1.2+ configuration.
 [04/06/04] In 'Common errors', add section on how to get rightca correct.
 [04/06/04] Some bugfixes to ipsec.conf examples from Paul of the Openswan team.
 [04/06/04] Added note under the cert copy section to make sure you set the pw, for non-newbies.
 [03/21/04] Added note about FreeS/WAN going away, and Openswan.
 [02/13/04] Added update note, saying it's based on 1.9x, and will be updated to 2.x.
 [11/13/03] Fix link to Strongsec site; thanks Jima!
 [05/06/03] Remove alternate way to get NAT working; add rightsubnetwithin note.
 [01/02/03] Link to Jacco de Leeuw's IPsec/L2TP page
 [11/21/02] Add a note for FreeS/WAN configuration files saying indentation is important. Thanks Stephen!
 [10/14/02] Updated RootCA.der entry in Client section to match with the Server.
 [10/10/02] Updated to new template for my main site. Cleaned things up a bit.
 [10/10/02] Put everything in CVS, finally.
 [10/09/02] Added paths for RH7.x/RH8.x (thanks for the reminder, Johan!)
 [08/08/02] Added note in the CA section to make sure that CA cert is longer than the client cert
 [06/19/02] Added color highlights to openssl commands; split into more sections
 [06/07/02] Switched from pre to classes to fix scrolling annoyances
 [05/30/02] Misc cleanup
 [04/30/02] Removed 'roadwarrior' conn from wireless section since it's not necessary; added example ipsec.exe output.
 [04/18/02] Added 'common errors' section; fixed some type-o's.
 [04/16/02] Added changelog.
 [04/15/02] Fixed type-o in Windows config section -- I had 'rightnet='; should be 'rightsubnet='.
 [04/01/02] Added section with example wireless setup.

Setting up your Certificate Authority

For the sake of this document, I'm assuming you want to use X.509 certificates for authentication. It is possible to use RSA



Google



Ads by Google

[Embedded IPsec and IKE](#)

Small Footprint.
High Performance.
Source code.
Royalty Free!
www.mocana.com/ipsec

[Free VPN Support Tools](#)

Tools to Support Remote VPN Users & Easy, Free Remote SSL VPN Access.
www.LogMeInITReach.com

[IPsec VPN Client Software](#)

Secure Your Remote Connections. For Large Enterprises. Free Trial!
www.TheGreenBow.com

[Remote Access Vpn](#)

Access Any Remote PC Online Sign Up For A Free Software Trial!
www.Bomgar.com

keys or pre-shared keys, but I find the X.509 method to be the most scalable and easiest to maintain for a decent-sized user base. I am also assuming that you will need your own Certificate Authority dedicated to VPN usage - if you already have access to a CA, you may just want to generate certificates from there (if that's the case, you can just skim this section.) If you need more details that I am going into here, please read the OpenSSL documentation -- it's fairly detailed. For CA certificate management, my examples use the utilities included with OpenSSL itself - there are third-party tools out there that make this a bit simpler, but I want to keep dependencies low. Note that you do not necessarily need to use your Openswan gateway as the Certificate Authority - it can be any box with OpenSSL installed. In fact, it may be better to use a different box, so if an attacker gains access to your Openswan gateway they don't have access to your CA, too. If you have any suggestions on how to make this process simpler, please let me know!

Now, on to the good stuff - let's start setting up our own CA.

1) Find your openssl.cnf file. This file has default values for OpenSSL certificate generation. Here's a few locations for various distributions:

```
Debian: /etc/ssl/openssl.cnf
RedHat 7.x+: /usr/share/ssl/openssl.cnf
```

Open this file in your favorite editor. We will need to change the following options:

'default_days': This is the length of time, in days, that your certificates will be valid for, and defaults to 365 days, or 1 year. I recommend setting this to '3650', as that will give you 10 years of validity on your certificates. Since this is for internal use, I am ok with the security ramifications of having a certificate valid for a long time - if you lose it or whatnot, you can revoke it without a problem.

'[req_distinguished_name]' section: You don't really *need* to change the options below req_distinguished_name; they just set the default options (such as location, company name, etc) for certificate generation. I find it's easier to set them here than re-type them for every certificate.

2) Create a directory to house your CA. I generally use something like /var/sslca; you can really use whatever you want. Change the permissions of the directory to 700, so that people will not be able to access the private keys who aren't supposed to.

3) Find the command 'CA.sh' (some distributions rename it to just 'CA'; don't ask me why.) Locations on various distributions:

```
Debian: /usr/lib/ssl/misc/CA.sh
RedHat 7.x+: /usr/share/ssl/misc/CA
```

Edit this file, and change the line that says 'DAYS="days 365"' to a very high number (this sets how long the certificate authority's certificate is valid.) Be sure that this number is higher than the number in Step 1; or else Windows may not accept your certificates. Note that if this number is too high, it can cause problems - I generally set it for 15-20 years.

4) Run the command 'CA.sh -newca'. Follow the prompts, as below. Example input is in red, and my comments are in blue. Be sure to not use any non-alphanumeric characters, such as dashes, commas, plus signs, etc. These characters may make things more difficult for you.

```
nate@example:~/sslca$ /usr/lib/ssl/misc/CA.sh -newca
CA certificate filename (or enter to create)
(enter)
Making CA certificate ...
Using configuration from /usr/lib/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++
.....+++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:(enter password) This is the password you will need to create any
other certificates.
Verifying password - Enter PEM pass phrase:(repeat password)
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US(enter) Enter your country code here
State or Province Name (full name) [Some-State]:State(enter) Enter your state/province
here
Locality Name (eg, city) []:City(enter) Enter your city here
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ExampleCo(enter) Enter your
company name here (or leave blank)
Organizational Unit Name (eg, section) []:(enter) OU, if you like. I usually leave it
blank.
Common Name (eg, YOUR name) []:CA(enter) The name of your Certificate Authority
Email Address []:ca@example.com(enter) E-Mail Address
nate@example:~/sslca$
```

Let's also generate a crl file, which you'll need on your gateway boxes:
nate@example:~/sslca\$ openssl ca -gencrl -out crl.pem
You'll need to update this CRL file any time you revoke a certificate.

That's it, you now have your own certificate authority that you can use to generate certificates.

Generating a Certificate

You will need to generate a certificate for every machine that will be making an IPSec connection. This includes the gateway host, and each of your client machines. This section details how to create the certificate, and convert it to formats needed for Windows and such.

Again, we'll be using the CA.sh script. Except this time, instead of telling it to create a new Certificate Authority, we're telling it to request, then sign a certificate:

```
nate@example:~/sslca$ /usr/lib/ssl/misc/CA.sh -newreq
Using configuration from /usr/lib/ssl/openssl.cnf
```

```

Generating a 1024 bit RSA private key
.....+++
.....+++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:(enter password) Password to encrypt the new cert's private key
with - you'll need this!
Verifying password - Enter PEM pass phrase:(repeat password)
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US(enter)
State or Province Name (full name) [Some-State]:State(enter)
Locality Name (eg, city) []:City(enter)
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ExampleCo(enter)
Organizational Unit Name (eg, section) []:(enter)
Common Name (eg, YOUR name) []:host.example.com(enter)This can be a hostname, a real
name, an e-mail address, or whatever
Email Address []:user@example.com(enter) (optional)

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:(enter)
An optional company name []:(enter)
Request (and private key) is in newreq.pem

```

What we just did is generate a Certificate Request - this is the same type of request that you would send to Thawte or Verisign to get a generally-accepted SSL certificate. For our uses, however, we'll sign it with our own CA:

```

nate@example:~/sslca$ /usr/lib/ssl/misc/CA.sh -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter PEM pass phrase:(password you entered when creating the ca)
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName      :PRINTABLE:'US'
stateOrProvinceName :PRINTABLE:'State'
localityName     :PRINTABLE:'City'
organizationName  :PRINTABLE:'ExampleCo'
commonName       :PRINTABLE:'host.example.com'
emailAddress      :IA5STRING:'user@example.com'
Certificate is to be certified until Feb 13 16:28:40 2012 GMT (3650 days)
Sign the certificate? [y/n]:y(enter)

1 out of 1 certificate requests certified, commit? [y/n]y(enter)
Write out database with 1 new entries
Data Base Updated
(certificates snipped)
Signed certificate is in newcert.pem

```

Next, move the output files to names that make a bit more sense for future reference.

```

nate@example:~/sslca$ mv newcert.pem host.example.com.pem
nate@example:~/sslca$ mv newreq.pem host.example.com.key

```

That's all that's required for Openswan boxes - you'll need these two files, along with the file 'cacert.pem' from the 'demoCA' directory, and the 'cr1.pem' file you generated earlier.

If this certificate is needed for a Windows box, you'll need to convert it to a p12 format:

```

$ openssl pkcs12 -export -in winhost.example.com.pem -inkey winhost.example.com.key
-certfile demoCA/cacert.pem -out winhost.example.com.p12

```

Installing Openswan

You'll need to install Openswan each Linux box you want to speak IPsec.

Openswan now integrates all of the important patches, including X.509 and NAT Traversal. If you want to build it from scratch, you can download it from <http://www.openswan.org/code>, and follow the installation directions included with the package.

You now have two options for which IPsec stack you want to install in the kernel - you can either use Openswan's IPsec stack (Klips), or use the built-in IPsec stack in the 2.6 kernel (26sec). If you are running on a stock 2.4 kernel, the only option is Klips. You'll need to patch NAT Traversal support into your kernel (if you intend to use it), and build the ipsec.o kernel module. Otherwise, if you are using a 2.6 kernel or a 2.4 kernel with backported 26sec support (such as the kernel Debian provides), you don't need to touch the kernel-land at all - you can just install the Openswan user-land utilities and go. Note that there isn't as of yet an option to use Klips on the 2.6 kernel; it is on the Openswan developer's to-do list, but isn't a real high priority.

You'll also need the user-land utilities. If you are installing from source, 'make programs ; make install' should get you what you need. Otherwise, if you are running Debian testing or unstable, you can just run 'apt-get install openswan' to get the user level utilities. ATrpms provides a Openswan package for recent versions of RedHat and Fedora Core; for more information on that, see <http://atrpms.net>.

Once you've selected and set up your IPsec stack and installed the user-land programs, you're ready to move on to configuring Openswan.

Installing the Certificate on your Gateway

This discusses how to install the certificate on your gateway machine. These same steps apply for installing the cert on

Openswan clients, too. I'm assuming you've already created a certificate for each machine (see the "Generating a Certificate" section) - if that's not the case, please go back and do that now.

1) Install the files in their proper locations (if installing to a remote machine, please be sure to copy the files in a secure manner):

```
$ cp /var/sslca/host.example.com.key /etc/ipsec.d/private
$ cp /var/sslca/host.example.com.pem /etc/ipsec.d/certs
$ cp /var/sslca/demoCA/cacert.pem /etc/ipsec.d/cacerts
$ cp /var/sslca/crl.pem /etc/ipsec.d/crls/crl.pem
```

Configuring Openswan on the Gateway Machine

1) Configure ipsec.secrets:

/etc/ipsec.secrets should contain the following:

```
: RSA host.example.com.key "password"
```

The password above should be the PEM passphrase that you entered while generating the SSL certificate.

2) Configuring ipsec.conf

/etc/ipsec.conf should look something like the configuration below (note that the indentation is important; without it, openswan will fail):

```
version 2.0

config setup
    interfaces=%defaultroute
    nat_traversal=yes
    virtual_private=%v4:10.0.0.0/8,%v4:172.16.0.0/12,%v4:192.168.0.0/16

conn %default
    keyingtries=1
    compress=yes
    disablearrivalcheck=no
    authby=rsasig
    leftrsasigkey=%cert
    rightrsasigkey=%cert

conn roadwarrior-net
    leftsubnet=(your_subnet)/(your_netmask)
    also=roadwarrior

conn roadwarrior
    left=%defaultroute
    leftcert=host.example.com.pem
    right=%any
    rightsubnet=vhost:%no,%priv
    auto=add
    pfs=yes

conn block
    auto=ignore

conn private
    auto=ignore

conn private-or-clear
    auto=ignore

conn clear-or-private
    auto=ignore

conn clear
    auto=ignore

conn packetdefault
    auto=ignore
```

This configuration will set things up so anyone with a valid certificate signed by your CA will be able to connect to your host. There are two connection profiles: one for a connection directly to the gateway, and one for the client to connect to the network behind the gateway. This configuration also includes NAT Traversal configuration that will allow anyone a host behind a NAT gateway using RFC1918 private addresses (defined in the 'virtual_private' line) to connect. All of the 'auto=ignore' entries are used to disable Opportunistic Encryption (OE), as it can cause problems if not configured properly.

If you are planning on having Windows boxes connect to your host using L2TP over IPSec, you'll also need the following connections, somewhere above the 'roadwarrior' definition:

```
conn roadwarrior-l2tp
    pfs=no
    leftprotoport=17/0
    rightprotoport=17/1701
    also=roadwarrior

conn roadwarrior-l2tp-updatedwin
    pfs=no
    leftprotoport=17/1701
    rightprotoport=17/1701
```

```
also=roadwarrior
```

In addition, if you want to have clients tunnel all traffic via IPSec, you'll need a connection that allows that. The following is what I recommend (again, add above roadwarrior):

```
conn roadwarrior-all
    leftsubnet=0.0.0.0/0
    also=roadwarrior
```

Client Setup: Openswan

1) Follow the steps under '[Generating a Certificate](#)' to create a new certificate for the client machine, modifying file names and such as needed. (We will refer to the files for this client as 'clienthost.example.com'.)

2) Copy the following files (in a secure fashion) over to your client:

```
host.example.com.pem (your gateway's certificate file)
clienthost.example.com.key
clienthost.example.com.pem
cacert.pem
crl.pem
```

3) Copy the files into their proper locations:

```
$ cp clienthost.example.com.key /etc/ipsec.d/private
$ cp clienthost.example.com.pem /etc/ipsec.d/certs
$ cp host.example.com.pem /etc/ipsec.d/certs
$ cp crl.pem /etc/ipsec.d/crls
$ cp cacert.pem /etc/ipsec.d/cacerts/cacert.pem
```

4) Configure ipsec:

ipsec.secrets:

```
: RSA clienthost.example.com.key "password"
```

ipsec.conf:

```
version 2
```

```
config setup
```

```
    interfaces=%defaultroute
    nat_traversal=yes
```

```
conn %default
```

```
    keyingtries=1
    compress=yes
    authby=rsasig
    leftrsasigkey=%cert
    rightrsasigkey=%cert
```

```
conn roadwarrior-net
```

```
    leftsubnet=(your_subnet)/(your_netmask)
    also=roadwarrior
```

```
conn roadwarrior
```

```
    left=(ip.of.host)
    leftcert=host.example.com.pem
    right=%defaultroute
    rightcert=clienthost.example.com.pem
    auto=add
    pfs=yes
```

```
conn block
```

```
    auto=ignore
```

```
conn private
```

```
    auto=ignore
```

```
conn private-or-clear
```

```
    auto=ignore
```

```
conn clear-or-private
```

```
    auto=ignore
```

```
conn clear
```

```
    auto=ignore
```

```
conn packetdefault
```

```
    auto=ignore
```

5) Start the VPN link, and make sure everything works:

```
# /etc/init.d/ipsec restart
$ ipsec auto --up roadwarrior
$ ipsec auto --up roadwarrior-net
```

6) If you would like to have the links start automatically, change 'auto=add' to 'auto=start'.

Client Setup: Windows 2000/XP

NOTE: If you have previously installed SSH Sentinel, and want to use the built-in Windows IPSec stack, you will need to

uninstall (or disable) SSH Sentinel, and enable the 'ipsec' service. I know this has tripped a few people up. This also applies for any other IPsec client you may have installed - you *need* to make sure it's disabled before trying to use the built in IPsec service.

NOTE #2: The HTML guy at my previous employer went through and made screenshots of the process of importing a certificate. These screenshots are available at <http://support.real-time.com/open-source/ipsec/index.html>. Please do NOT e-mail Real Time with any questions related to this; I no longer work there, and don't want them to get a flood of questions about this.

1) Create the certificate, again following the steps under '[Generating a Certificate](#)'. We'll assume that you call the Windows 2000 certificate 'winhost.example.com'. You'll need to follow the directions to output a .p12 file.

Also run the following, and make a note of it's output:

```
$ openssl x509 -in demoCA/cacert.pem -noout -subject
```

You will need this for your VPN configuration.

2) Copy this file over to the Windows machine in a secure fashion, such as 'scp' or with a floppy disk. Don't use FTP!

3) Download Marcus Müller's ipsec.exe utility from <http://vpn.ebootis.de> and unzip it to some directory on your Windows machine (I generally use c:\ipsec)

4) Create a IPSEC + Certificates MMC

Start/Run/MMC

File (or Console) - Add/Remove Snap-in

Click on 'Add'

Click on 'Certificates', then 'Add'

Select 'Computer Account', and 'Next'.

Select 'Local computer', and 'Finish'.

Click on 'IP Security Policy Management', and 'Add'.

Select 'Local Computer', and 'Finish'

Click 'Close' then 'OK'

5) Add the certificate

Click the plus arrow by 'Certificates (Local Computer)'

Right-click 'Personal', and click 'All Tasks' then 'Import'

Click Next

Type in the path to the .p12 file (or browse and select the file), and click 'Next'

Type the export password, and click Next

Select 'Automatically select the certificate store based on the type of certificate', and click Next

Click Finish, and say yes to any prompts that pop up

Exit the MMC, and save it as a file so you don't have to re-add the Snap Ins each time

6) Set up the IPsec utility

Install ipsecpol.exe (Windows 2000) or ipseccmd.exe (Windows XP) as described in the documentation for the ipsec utility.

Note that for Windows XP SP2, you'll need a new version of ipseccmd.exe - it can be downloaded from

<http://support.microsoft.com/default.aspx?scid=kb;en-us;838079>.

Edit your ipsec.conf (on the windows machine), replacing the "RightCA" with the output of the 'openssl x509 -in demoCA/cacert.pem -noout -subject'; reformatted as below (you need to change the /'s to commas, and change the name of some of the fields -- just follow the example below):

```
conn roadwarrior
    left=%any
    right=(ip_of_remote_system)
    rightca="C=US,S=State,L=City,O=ExampleCo,CN=CA,Email=host@example.com"
    network=auto
    auto=start
    pfs=yes
```

```
conn roadwarrior-net
    left=%any
    right=(ip_of_remote_system)
    rightsubnet=(your_subnet)/(your_netmask)
    rightca="C=US,S=State,L=City,O=ExampleCo,CN=CA,Email=host@example.com"
    network=auto
    auto=start
    pfs=yes
```

If you would like to encrypt all data over the tunnel, the following should work (if you have set up the Linux side properly):

```
conn roadwarrior-all
    left=%any
    right=(ip_of_remote_system)
    rightsubnet=*
    rightca="C=US,S=State,L=City,O=ExampleCo,CN=CA,Email=host@example.com"
    network=auto
    auto=start
    pfs=yes
```

7) Start the link

Run the command 'ipsec.exe'. Here's example output:

```
C:\ipsec>ipsec
IPSec Version 2.1.4 (c) 2001,2002 Marcus Mueller
Getting running Config ...
Microsoft's Windows XP identified
Host name is: (local_hostname)
No RAS connections found.
LAN IP address: (local_ip_address)
Setting up IPSec ...

Deactivating old policy...
Removing old policy...
```

```
Connection roadwarrior:
MyTunnel : (local_ip_address)
MyNet : (local_ip_address)/255.255.255.255
PartnerTunnel: (ip_of_remote_system)
PartnerNet : (ip_of_remote_system)/255.255.255.255
CA (ID) : C=US,S=State,L=City,O=ExampleCo,...
PFS : y
Auto : start
Auth.Mode : MD5
Rekeying : 3600S/50000K
Activating policy...
```

```
Connection roadwarrior-net:
MyTunnel : (local_ip_address)
MyNet : (local_ip_address)/255.255.255.255
PartnerTunnel: (ip_of_remote_system)
PartnerNet : (remote_subnet)/(remote_netmask)
CA (ID) : C=US,S=State,L=City,O=ExampleCo,...
PFS : y
Auto : start
Auth.Mode : MD5
Rekeying : 3600S/50000K
Activating policy...
```

```
C:\ipsec>
```

Now, ping your gateway host. It should say 'Negotiating IP Security' a few times, and then give you ping responses. Note that this may take a few tries; from a T1 hitting a VPN server on a cable modem, it usually takes 3-4 pings. Do the same for the internal network on the remote end, and you should be up!

Some common errors, and resolutions for them

I've tried to make it as simple as possible to follow the above instructions, but sometimes it just doesn't quite work right. :) If you have trouble, feel free to [e-mail me](#), or join the FreeS/WAN mailing list and ask your questions there (many times, you will get a quicker response there, as there are more people listening at any given time, and most of them are smarter than me!). But, just in case you've got one of the really common problems, here's a few problems and solutions:

1) Logging on the Windows side (helps troubleshoot certificate errors, etc)

Yes, it is actually possible to enable logging on the Windows box! To do this, follow the directions at Microsoft's [Basic IPsec Troubleshooting in Windows 2000](#) page -- look for the section entitled 'Obtaining an Oakley Log'.

2) Pinging from the Windows side shows 'Negotiating IP Security', but the tunnel never comes up!

This is one of the most common problems people have, and is usually caused by problems with rightca= on the Windows side. To verify that you have that set properly, follow these instructions:

- Load the IPSec MMC you created earlier
- Click IP Security Policies; double-click on the FreeSwan tunnel
- Double-click roadwarrior-Host filter
- Click on the 'Authentication Methods' tab
- Click 'Add', then 'Use a certificate from this CA'
- Click Browse, find your CA
- Copy/paste the text in the grayed-out box into your ipsec.conf

In many cases, that'll clear up the issues - if it doesn't, check your log for errors.

More troubleshooting tips to come soon, assuming I get time to write them. :)

Let me know if the above doesn't make sense, and I'll try to help you out. :)

References

FreeS/WAN Documentation: <http://www.freeswan.org>

X.509 Patch Documentation: <http://www.strongsec.com/freeswan>

The Windows 2000 VPN Tool Documentation: <http://vpn.ebootis.de>

Microsoft's Basic IPsec Troubleshooting page: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q257225>