



The Application Layer is where the higher protocols are (TCP, UDP, etc.) and has an interaction with the Network Layer that is transparent to software and users. If the Network Layer is secure, everything that goes on the Application Layer is secure as well. With this, attacks such as "Man in the Middle" and Eavesdropping are no longer valid.

There are several ways to use IPSec, and it is important to decide what is best for your network and your data. If you want to authenticate a client accessing the server, or you want to make sure that the data being sent has not been changed, but you do not feel the need to encrypt that data, the best thing to use is just authentication on the packets. If you want to protect the data from being monitored while it travels over the network, the best thing would be encryption and authentication. The encryption protocol is called Encapsulated Security Payload (ESP) and can also be used for authentication. The authentication portion is called the Authentication Header (AH) and can only be used for authentication. In the example below we are going to use both ESP and AH. It has been demonstrated that Encryption without Authentication can open possibilities of active attacks that may allow the breaking of the encryption.

Another important thing to know about IPSec is the Key Exchange. The two main methods of key exchange are Manual Key Exchange (MKE) and Automatic Key Exchange (AKE). When the choice is MKE, the administrator must set the keys for the communication; this method is mostly used when debugging the software that implements IPSec, but not used often on general IPSec connections. The most used method is AKE. Besides being easier for the administrator, AKE is much more secure than MKE as the key can be changed automatically every few minutes, without shutting down the connection. Manually generated keys must be exchanged between the machines using IPSEC, meaning the administrator must find a way for taking the key from one machine to another without leaving it being exposed on the network. This does not happen all the time. Some Automatic Key Exchange Protocols: IKE - the Internet Key Exchange protocol - not the best protocol, but the most used; ISAKMP - the Internet Security Association and Key Management Protocol - also used quite often, and also has some known flaws; Photuris, although still experimental, is one of the best protocols for key exchange, and is currently implemented in OpenBSD's IPSec.

IPSec can work in Tunneling Mode or Transport Mode. Both behave the same with regards to key exchange, encryption and authentication. In Tunneling Mode the end hosts do not need to have IPSec implemented, they just have to route the packets to gateways that talk to each other through IPSec interfaces. Transport Mode differs from Tunneling Mode in a way that

every host has IPsec interfaces.

There are many different implementations of IPsec, with different resources. Below is a brief list with some of the most popular implementations.

Product	Platform	Available at
FreeS/WAN	Linux	http://www.freeswan.org
CheckPoint VPN-1	Microsoft Windows	http://www.checkpoint.com
	NT 4.0 (SP3 & SP4)	
	Sun Solaris 2.6	
	Solaris 7	
	HP-UX 10.20, 11.0	
SunScreen SKIP	IBM AIX 4.2.1, 4.3.2	
	Sun Solaris 2.6	http://www.sun.com
	Solaris 7	
	Microsoft Windows	
	NT 4.0, 95, 98	

OpenBSD comes with IPsec. For a list with 41 different IPsec implementations, go to: <http://www.mit.edu/~tytso/ipsec/results9710.html>

Now that the basic issues about IPsec have been covered, next comes the practical part: setting up an example. For this you are going to need at least three machines connected on the same network. If you are going to use a switch instead of a hub, you will have to use one of the machines as a gateway between the other two, like two machines, on two networks, talking to each other. These machines should be using Linux (kernel 2.0.38).

So, let's do it step by step, follow the steps below on two machines:

1. Download and unpack FreeS/WAN. To quote their readme:

1. Download:

```
ncftp ftp://ftp.xs4all.nl/pub/crypto/freeswan/freeswan-*
```

2. You should now have:

```
freeswan-1.3.tar.gz
freeswan-1.3.tar.gz.sig
freeswan-sigkey.asc
```

3. Check the files' integrity:

```
pgp freeswan-1.3.tar.gz.sig freeswan-1.3.tar.gz
```

4. Unpack the FreeS/WAN sources where you would normally do so.

5. Before "making" anything, cd into the top-level FreeS/WAN source

directory and apply any patches (note: at this time, there are no patches for version 1.3):

```
cd ../freeswan-1.3/
zcat ../freeswan-1.3.patch1.gz | patch -p0 > PATCH.LOG 2>&1 || cat PATCH.LOG
```

6. The result should be console output of the form 'patching file <file>'. It will also be in PATCH.LOG. If you see any 'FAILED' messages or mention of files with a suffix of '.rej', the patch has failed. Report the problem and do not proceed.

7. Now continue with the normal freeswan-1.3 installation by completely reading README and INSTALL in the top-level FreeS/WAN directory. Check the FreeS/WAN errata page at <http://www.freeswan.org/errata.html> for further updates.

You should unpack the FreeS/WAN distribution at /usr/src or where your kernel is.

2. Go to the top level directory of FreeS/WAN and prepare to re-compile your kernel, but first make sure your network is working O.K. without IPsec and that this kernel source has no errors (compiling without FreeS/WAN first is a good way to test this). Choose whatever configuration method you want (make xconfig, make menuconfig, etc.). When compiling the kernel, DO NOT DISABLE this:

```
IP: forwarding/gatewaying;
Kernel/User network link driver;
Kernel/User netlink socket.
```

You probably will notice the IPsec options, they are O.K. for now. Just finish setting up the kernel

and compile it.

3. Install the new kernel by doing: make kinstall (this installs the modules as well), and finish your kernel installation (lilo, etc.).
4. Now you are going to edit /etc/ipsec.conf to fit your network:

```
config setup
    interfaces="ipsec0=eth0"
    klipsdebug=none
    plutodebug=none
    plutoload=%search
    plutostart=%search
```

Note that I used eth0, but you should use what interface you want to setup as IPsec.

```
conn %default
    keyingtries=0
    keylife=2h
conn left-right
    left=172.16.4.12
    leftnexthop=
    right=172.16.4.23
    rightnexthop=
    auto=start
    auth=ah
```

A host being left or right is a choice that you have to do. In fact, it does not matter which is which, as long as you use always the same side for the same host. Just fix "left=" and "right=" for whatever suits your network. The "auto=start" options means that when Pluto (the automatic key exchanger) is started, this connection is also started. For other options, go to:

http://www.freeswan.org/freeswan_trees/freeswan-1.3/doc/setup.html

5. Time to setup /etc/ipsec.secrets, first you need to create a random key. FreeS/WAN comes with a utility for that, just run:

```
umask 177
ipsec ranbits 256 > delete.me
```

Now open your /etc/ipsec.secrets. It should look something like this:

```
172.16.4.12 172.16.4.23 "0xf568175c_97462413_6db3d6ae_f2b46f40_d4e891fc_99d422f4_d6160755_0410164c"
```

Where the huge string of hexadecimal numbers should be what you got on that file created by ranbits (delete.me). Do not forget to delete delete.me after using it!

6. Copy /etc/ipsec.conf and /etc/ipsec.secrets to the other machine SECURELY. Use PGP or any other cryptography software for it. Remember that these files MUST be kept securely (rw-----).
7. Reboot the machines.
8. Check to see if everything is right:
 - o go to /proc/net and look for IPsec files: ls ipsec*
 - o is the relationship between IPsec and your interfaces right? Find out by doing:


```
cat /proc/net/ipsec_tncfg
```
9. Start IPsec in both machines by doing:
 - a. ipsec auto --add left-right on BOTH machines
 - b. ipsec auto --up left-right on ONE machine
10. Test it with 'ipsec look'. If you see a table with the connections you probably did everything right. If you ever want to take the IPsec interface down, just run


```
ipsec auto --down left-right
```

Let's work a little with the third machine. First of all, download and install Sniffit from: <http://reptile.rug.ac.be/~coder/sniffit/sniffit.html>. Learn how to use it, it is a very good tool.

Now you have three machines, two with IPsec and one with a sniffer. The final example on this article is an analysis of the traffic with and without IPsec.

With the IPsec interfaces down (ipsec auto --down left-right) and with your sniffit running on machine three, telnet from machine one to machine two. See how open is the traffic? Well, you probably could see your password going out clearly.

Now start the IPsec interfaces like you did on step #9 above. With you sniffit running on machine three, telnet from machine one to machine two. Can you see the difference? If you generated the key with ranbits 256 you are seeing your traffic encrypted with a 256 bits key.

For interoperation of FreeS/WAN with other implementations of IPSec, check out: http://www.freeswan.org/freeswan_trees/freeswan-1.3/doc/compatibility.html#intero

If you want to learn more about IPSec and other implementations, below are some resources that I found useful:

Requests For Comment:

RFC-2401: Security Architecture for the Internet Protocol

RFC-2411: IP Security Document Roadmap

Internet Engineering Task Force Drafts:

draft-ietf-ipsec-policy-schema-00: IPsec Policy Schema

draft-ietf-ipsec-dhcp-04: DHCP Configuration of IPSEC Tunnel Mode

IPSec is going to be embedded in IPv6, so it is probably going to be a standard soon. An Internet that reduces the possibilities of eavesdropping and protects the privacy of the users is a promise that IPSec can partly fulfill. With this article as starting point, you can learn how to setup IPSec gateways and tunnels. I strongly recommend reading the IPSec man pages on OpenBSD, if you plan to build a secure gateway.

[Privacy Statement](#)

Copyright 2006, SecurityFocus