Introduction to IPSec

Brad Marshall <bmarshal@pisoftware.com>

June 10, 2002

Contents

1 Introduction to IPSec

Many of you would have heard of IPSec, but may know nothing about it. It was first proposed in RFC1825 way back in August of 1995 (later updated in RFC2401 in November of 1998) as a means of embedding security into both IPv4 and IPv6. There were several properties to consider for data - authentication, integrity, confidentiality and non-repudiation. There are occasions when sometimes only one of these are important, and times when you require all.

The IPSec architecuture doesn't force any particular encryption or authentication algorithms on the user - it is all completely replaceable with whatever algorithm you want. However, the RFCs specify certain algorithms that should be implemented for compatibility reasons.

IPsec can be deployed to protect data going between two hosts, between a host and a router / firewall, or between two routers / firewalls. It also gives

fine grained control as to how the security is implemented - what services to use where, what combination, and what algorithms to use.

However, IPSec is not an end to end solution - it can only encrypt traffic as it travels over the network. You will still need to protect the data on the destination machines, if required. IPSec can also only authenticate machines, not users - you will still need to have user based authenticaton on your applications etc.

1.1 IPSec Protocols

There are several protocols used in IPSec that are important to understand. The Authentication Header, or AH, is used as the name suggests to authenticate the packets, and optionally anti-replay protection. This is used soley to ensure knowledge of who the sender is, and by itself gives no confidentiality of data. However, due to various crypto laws around the world it is perhaps easier to implement politically due to the lack of encryption.

IP Encapsulating Security Payload header, or ESP, is used to ensure that the data transmitted between the two hosts securely, optionally with authentication and integrity checking. Regardless, using either (or both) of these protocols gives you access control over the data.

The Internet Key Exchange, or IKE, negotiates the connection parameters, which includes what type of connection, what encryption algorithms to use, and what keys are used.

2 FreeS/WAN

The Linux implementation of IPSec is FreeS/WAN. It is available from http://www.freeswan.org/.

FreeS/WAN consists of 3 main parts:

KLIPS - Kernel IPSec, implements AH and ESP

Pluto - IKE daemon

2.1 Installing FreeS/WAN

Installing FreeS/WAN consists of two main parts - patches to the kernel, and some userspace tools and daemons. The example that follows assumes you are using a recent version of Debian GNU/Linux.

2.1.1 Kernel Patches

The following is a brief description of how to install, patch and compile a Debian kernel package with support for FreeS/WAN. It assumes you have a recent 2.4 kernel (2.4.18 as of time of writing) untarred in /usr/src/linux.

```
$ sudo apt-get install kernel-package kernel-patch-freeswan
$ cd /usr/src/linux
$ export PATCH_THE_KERNEL=YES
$ make-kpkg --config=menuconfig --revision=host1.0 configure
```

At a minimum, you need the following kernel config options selected:

```
CONFIG_IPSEC
CONFIG_IPSEC_IPIP
CONFIG_IPSEC_AH
CONFIG_IPSEC_ESP
```

Once you've configured the kernel appropriately, and optionally saved your kernel configuration somewhere, run the following:

```
$ fakeroot make-kpkg binary-arch
```

Now, install the kernel package (it will be in /usr/src, and named something like kernel-image-2.4.18_host1.0_i386.deb) and reboot.

2.1.2 Userspace daemons and tools

Under Debian installing the userspace daemons and tools is fairly simple - you do something like:

```
$ sudo apt-get install freeswan
```

This will install all the necessary init scripts, and optionally create an x509 or RSA key for you.

2.2 Configuration

FreeS/WAN configuration is done by `/etc/ipsec.conf`, and RSA keys and preshared secrets are kept in `/etc/ipsec.secrets`.

There are 3 main types of networks that IPSec can provide a solution for - road warriors, subnet to subnet, and client behind NAT.

2.2.1 Road Warrior

A road warrior is simply a machine that's out on the internet somewhere, which may or may not have a static ip address, and wishes to communicate back to the office. The most common example of this situation is that of a laptop dialed up to some ISP. This is perhaps the simplest case, and is not very hard to setup.

For the road warrior, `/etc/ipsec.conf` looks like:

```
conn road-warrior
    left=%defaultroute
    leftid=@hostname.identifier
```

```
    leftrsasigkey=0s ... rsa sig ...
    right=ip.add.ress
    rightid=@gw.hostname.identifier
    rightrsasigkey=0s ... rsa sig ...
    rightsubnet=local.ip.range.0/24
    rightnexthop=gw.default.route
    authby=rsasig
    auto=add
```

On the server, `/etc/ipsec.conf` looks like:

```
conn vpn
    left=%any
    leftid=@hostname.identifier
    leftrsasigkey=0s ... rsa sig ...
    right=ip.address
    rightid=@stallman.pisoftware.com
    rightrsasigkey=0s ... rsa sig ...
    rightsubnet=local.ip.range.0/24
    rightnexthop=gw.default.route
    authby=rsasig
    auto=add
```

2.2.2 Subnet to Subnet

This is the situation where you wish to have a subnet behind a router of some kind talk to the office subnet. This is most common in the situation where you have a machine at home providing network access for the rest of the house, and you wish the entire network to see the "office".

The "client" `/etc/ipsec.conf` extract looks like:

```
conn client-subnet-vpn
    left=%defaultroute
    leftid=@host.identifier
    leftrsasigkey=0s...rsa sig...
    leftsubnet=local.ip.range.0/24
    right=server.ip.addr
    rightid=@gw.host.identifier
    rightrsasigkey=0s...rsa sig...
    rightsubnet=local.ip.range.0/24
    rightnexthop=gw.default.route
    authby=rsasig
    auto=start
```

The "server" `/etc/ipsec.conf` extract looks like:

```
conn server-subnet-vpn
```

```
    left=%any
    leftid=@host.identifier
    leftrsasigkey=0s...rsa sig...
    leftsubnet=local.ip.range.0/24
    right=server.ip.addr
    rightid=@gw.host.identifier
    rightrsasigkey=0s...rsa sig...
    rightsubnet=local.ip.range.0/24
    rightnexthop=gw.default.route
    authby=rsasig
    auto=add
    # we don't want to retry if IP connectivity is gone
    keyingtries=1
    keylife=30m
    ikelifetime=30m
```

2.2.3 Client behind NAT

In some cases it may be necessary for the client that initiates the ipsec connection to be behind NAT. In certain situations, this is indeed possible, regardless of what you may have read elsewhere.

In this network setup, the machine doing the NAT was running a 2.4 linux kernel, with ip masquerading for the entire subnet behind it. The client that is initiating the connection needs to have a static private IP address.

```
conn nat-vpn
    authby=rsasig
    left=%defaultroute
    leftsubnet=client.ip.add.ress/32
    leftnexthop=
    leftid=@client.identifier
    leftrsasigkey=0s...rsa sig...
    right=server.ip.add.ress
    rightid=@gw.identifier
    rightrsasigkey=0s...rsa sig...
    rightsubnet=local.ip.range.0/24
    rightnexthop=gw.default.route
    auto=add

conn nat-vpn
    authby=rsasig
    left=0.0.0.0
    leftsubnet=client.ip.add.ress/32
    leftnexthop=
    leftid=@client.identifier
    leftrsasigkey=0s...rsa sig...
    right=server.ip.add.ress
```

```
rightid=@gw.identifier
rightrsasigkey=0s...rsa sig...
rightsubnet=local.ip.range.0/24
rightnexthop=gw.default.route
auto=add
```

2.2.4 Complex combinations

There are many possible network topologies that can be used with IPSec, and the few that are covered above will not necessarily be suffient for all needs. However, in the general case, it is possible to build up the network configurations required with "blocks" of the previously described setups.

3 Firewall Issues

There are 3 protocols you have to allow though a firewall for complete operation of IPSec. These are AH, ESP and IKE. These firewall rules are examples from the Linux kernel 2.4 firewalling tool, iptables. Converting to your preferred firewalling tool shouldn't be hard.

```
# Allow IKE
/sbin/iptables -A INPUT -p udp --sport 500  -j ACCEPT
/sbin/iptables -A INPUT -p udp --dport 500  -j ACCEPT
# Allow ESP
/sbin/iptables -A INPUT -p 50 -j ACCEPT
# Allow AH
/sbin/iptables -A INPUT -p 51 -j ACCEPT
```

4 Conclusion

As you've seen, there are many useful things you can do with IPSec, but it is not a silver bullet for security. You need to have proper management of setting up the VPNs, and to be careful that you are not opening yourself up to problems. By allowing another network access to yours over a VPN it is important to recognize that the security policy of the remote end will affect yours.