Use of IPSEC in Linux when configuring network-to-network and point-to-point VPN connections

Aleksandr Derevyanko (dio4@yandex.ru)
Engineer IT department
Consultant

Skill Level: Intermediate

Date: 15 May 2012
15 May 2012

This article takes a detailed look at the design principles, the basis for deploying VPN, and the IPSEC protocol concept, providing a description of the general features of IPSEC and of the mechanisms required for its implementation. This article was specially selected for translation by developerWorks Russia as an example of developerWorks world-wide offerings.

Introduction

Many organizations across the world use every available physical connection method to link up their individual offices. The option chosen can be both dedicated digital lines and Virtual Private Networks (VPN), which are significantly cheaper than their physical equivalents. VPNs, which generally deploy the same approaches as dedicated lines, can combine several LANs into one and encrypt the traffic to conceal the data being transmitted. When encryption is deployed in VPN technology, open standards are generally used. This involves the traffic being transmitted on top of IP and using datagrams as the transport level.

From a technical perspective, VPNs can be implemented using both software and hardware. In Linux®, FreeS/Wan technology has often been deployed, using the standard implementation of the security protocol IPSEC (Internet Protocol Security). These solutions, which are implemented using both software and hardware, operate like routers at the ends of the VPN connections. When a packet is transmitted by the client, it is sent to this dedicated router, which adds an Authentication Header (AH) to it. After this data is encrypted and instructions are added for decrypting and processing it, the packet is transferred to the dedicated router at the other end. Once the packet is received, the end router decrypts it, discarding the header, and transmits the clear packet to the user at the destination.

If encryption is used between the networks, the host in the LAN receives the packet already decrypted and starts processing it in the normal way. This means that when encryption is used between networks, the entire encryption/decryption process becomes transparent to the network's end host.

Several levels of authentication and encryption are deployed in VPNs, thus making them secure and effective enough to combine multiple remote nodes into a single intranet.

IPSEC is a popular implementation of the VPN standard which is reliable enough to meet the requirements of various customers in terms of connecting their branches or remote users to their networks.

General overview of IPSEC

IPSEC is generally used to support secure connections between nodes and networks throughout the Internet. It can operate in a node-to-node configuration (with one computer connected to another) or in a network-to-network configuration (with one LAN/WAN connected to another). IPSEC is implemented using the Internet Key Exchange (IKE) protocol developed by the Internet Engineering Task Force (IETF) for the mutual authentication and comparison of security parameters between systems or networks connecting to each other.

The IPSEC connection process is split into two logical phases. During the first phase, the IPSEC node establishes a connection with a remote node or network. The remote node/network verifies the credentials of the requesting node and both sides agree on the authentication method to be used in the connection. An algorithm with a pre-shared key is usually used for the IPSEC node's authentication. If the IPSEC connection uses a pre-shared key, both nodes must use the same key. It will then be possible to proceed to the second phase of establishing a connection.

The second phase of establishing an IPSEC connection between nodes is carried out using Security Association (SA). This involves configuration data, such as the encryption method, means of exchanging the session's secret keys and a few other parameters being imported into the SA database. This phase also manages the IPSEC connection between the nodes and networks distributed throughout the WAN.

Now consider the implementation of IPSEC based on the example of the CentOS Linux distribution. To deploy IPSEC on all the machines in the network (in the case of a node-to-node configuration) or on the routers (in the case of a network-to-network configuration), you must set up the relevant packets for managing the IPSEC configuration. These packets must include basic libraries, daemons, and configuration files that help establish the IPSEC connection, including the `/lib/libipsec.so` library containing the interface for managing the trusted key, PF_KEY, between the Linux kernel and the IPSEC implementation being used in CentOS Linux. In this case:

- `/sbin/setkey` provides key management and the IPSEC security attributes in the kernel. This program complies with the `racoon` daemon managing the keys. Further information about setkey is available in the `setkey`(8) man page (see Resources).
- `/sbin/racoon` is the daemon managing the IKE keys and supervises the key exchange and security association between the computers running IPSEC. This daemon can be set up after editing the file`/etc/racoon/racoon.conf`. Further information about racoon is available from the `racoon`(8) man page (see Resources).
- `/etc/racoon/racoon.conf` is configuration file where various IPSEC connection parameters are set, including the authentication methods and encryption algorithms. Detailed information on this subject is again available from looking at the conclusion of the command in the `racoon.conf`(5) man page (see Resources).

Setting up IPSEC for node-to-node configuration

IPSEC can be used to connect one workstation to another, based on a node-to-node connection. With this kind of connection, the network to which both nodes are connected is used to create a secure tunnel. The nodes only need a permanent connection to the Internet or another constantly operating network to establish the IPSEC connection.

The following data is required to create a node-to-node connection:

- IP addresses for both nodes
- A unique name for the IPSEC connection, differentiating it from the other devices or connections (`ipsec0`)
- An encryption key which is permanent or created automatically using `racoon`
- An authentication pre-shared key used to establish the connection and exchange encryption keys during the connection session

Now look at the scenario where two hosts establish a connection with each other. They will do this using the shared key `my_key`and the daemon `racoon`to automatically create and exchange an authentication key. The connection name will be `ipsec0`.

The workstation uses the `ifcfg` file, in Listing 1, to establish the IPSEC node-to-node connection with the other workstation. Listing 1 shows the format of the file `/etc/sysconfig/network-scripts/ifcfg-ipsec0`.

Listing 1. Format of ifcfg-ipsec0

```
DST=X.X.X.X'
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
```

On the first computer the letters *X.X.X.X* must be replaced by the second computer's IP address, and vice versa on the second computer. This connection is established

during boot-up (`ONBOOT=yes`) and uses the pre-shared key authentication method (`IKE_METHOD=PSK`).

The file with the shared key (`/etc/sysconfig/network-scripts/keys-ipsec0`), shown in Listing 2, is needed by both computers for mutual authentication. The content of this file must be the same on both computers, and only the root user must be able to access it: `IKE_PSK=my_key`.

Issue the following command to restrict access to the file `keys-ipsec0` :

Listing 2. File with the shared key

```
chmod 600 /etc/sysconfig/network-scripts/keys-ipsec0.
```

To change authentication key at any time, you must edit the keys-ipsec0 file on both computers. Both keys must be identical to establish the connection.

Listing 3 examines the configuration process for the first phase in connecting to a remote node. This file is called *X.X.X.X.conf* (*X.X.X.X* is replaced by the IP address of the remote IPSEC router). Remember that this file is generated automatically when the IPSEC tunnel is activated and is not edited manually.

Listing 3. Format of X.X.X.X.conf file

```
;
remote X.X.X.X
{
        exchange_mode aggressive, main;
        my_identifier address;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm sha1;
                authentication_method pre_shared_key;
                dh_group 2 ;
        }
}
```

- The command `remote X.X.X.X` indicates that the following strings in the configuration file are only applied to the remote node being assigned to the address *X.X.X.X*.
- `exchange_mode aggressive` in the default IPSEC configuration in CentOS Linux is an authentication method that allows different IPSEC connections with multiple nodes.
- `my_identifier address` defines the identification method that will be used for node authentication. CentOS Linux identifies the nodes at the IP addresses.
- `encryption_algorithm 3des` defines the encryption algorithm used for authentication. The default method is the Triple Data Encryption Standard (3DES).
- `hash_algorithm sha1` indicates the hash calculation algorithm used during the first phase of the connection.

- `authentication_method pre_shared_key` defines the authentication method used when synchronizing nodes.
- `Bdh_group 2` indicates the number of the Diffie-Hellman group for selecting the dynamically generated session keys. The 1024-bit group is used by default.

The `/etc/racoon/racoon.conf` must also be identical at all the IPSEC nodes, *apart from* the operator `include "/etc/racoon/X.X.X.X.conf"`, which will have a different IP address. Both the operation and the file are generated when the IPSEC tunnel is activated. Listing 4 shows the typical `racoon.conf` produced when the IPSEC connection is established:

Listing 4. Format of racoon.conf file

```
# Racoon IKE daemon configuration file.

# See 'man racoon.conf' for a description of the format and entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
 pfs_group 2;
 lifetime time 1 hour ;
 encryption_algorithm 3des, blowfish 448, rijndael ;
 authentication_algorithm hmac_sha1, hmac_md5 ;
 compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf"
```

The description of all the parameters for the configuration files falls outside the scope of this article. You can obtain them from the appropriate manuals.

To establish the connection, either reboot the computer or execute the following command at each node on behalf of root: `/sbin/ifup ipsec0`.

Launch the utility `tcpdump` to check the IPSEC connection. When doing so, the packet must contain the AH and ESP data. ESP means that encryption is working. Listing 5 shows an example tcpdump:

Listing 5. Sample tcpdump

```
17:13:20.617872 pinky.example.com > ijin.example.com: \
AH(spi=0x0aaa749f,seq=0x335): ESP(spi=0x0ec0441e,seq=0x335) (DF)
```

Setting up IPSEC for network-to-network configuration

With IPSEC, you can connect whole networks to other network segments by organizing an internetwork. IPSEC routers need to be installed in each network so that traffic from the node of one network can be processed transparently and reach the node of the other network. IPSEC routers, which can authenticate and combine these networks through a secure tunnel, must be operating in these networks, with traffic flowing through the Internet or any other network. If the packets are

intercepted during transmission, you'll need to use the search method to crack the code protecting them, which can be a fairly long process and might not even be relevant by the end of the procedure.

The following information is required to establish an internetwork connection:

- The public IP addresses of the dedicated IPSEC routers
- The IP addresses of the interfaces of the gateways routing the network node traffic to the Internet
- The unique name for the IPSEC connection (such as, `ipsec0`)
- The encryption key created using `racoon`
- The authentication pre-shared key

We will look at the example of an IPSEC tunnel between the network my_net1.com and the network my_net2.com. The address of the first network is 192.168.1.0/24, while the address of the second one is 192.168.2.0/24. The IP address of the gateway in the first network is 192.168.1.254, while that in the second is 192.168.2.254. The routers are implemented separately from the gateways and use two network interfaces: eth0 has the static public IP address dedicated to the Internet, whereas eth1 receives and processes packets from the LAN.

The IPSEC connection between the networks uses the pre-shared key `r3dh4tl1nux`. The content of the file `ifcfg`, which was created for the IPSEC internetwork connection in the first network, is shown in Listing 6. In the example this connection has the unique name ipsec1. Listing 6 shows your example /etc/sysconfig/network-scripts/ifcfg-ipsec1.

Listing 6. Example ifcfg-ipsec1

```
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
SRCGW=192.168.1.254
DSTGW=192.168.2.254
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

This connection is established during boot-up (`ONBOOT=yes`) and uses the pre-shared key authentication method (`IKE_METHOD=PSK`).

The content of the file with the pre-shared key (called `/etc/sysconfig/network-scripts/keys-ipsecX`, where *X* is equal to 0 for the first network and to 1 for the second network) is shown in Listing 7.

You can change the authentication key, which requires the file `keys-ipsecX` to be edited on the IPSEC routers, at any time. The keys must be identical. Listing 7 shows the example /etc/racoon/racoon.conf.

Listing 7. Example racoon.conf file

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
 pfs_group 2;
 lifetime time 1 hour ;
 encryption_algorithm 3des, blowfish 448, rijndael ;
 authentication_algorithm hmac_sha1, hmac_md5 ;
 compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf"
```

Listing 8 shows the configuration of the specific connection with the remote network. The name of this file is `X.X.X.X.conf` (where *X.X.X.X* is the IP address of the remote IPSEC router).

Listing 8. Sample X.X.X.X.conf file

```
;
remote X.X.X.X
{
        exchange_mode aggressive, main;
        my_identifier address;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm sha1;
                authentication_method pre_shared_key;
                dh_group 2 ;
        }
}
```

Prior to launching the IPSEC connection, activate IP routing in the kernel by editing the file /etc/sysctl.conf and setting `net.ipv4.ip_` forward equal to 1.

To ensure that the amendment is implemented, issue this command: `sysctl -p /etc/sysctl.conf`.

To establish the connection, either reboot the routers or run the following command on the routers on behalf of root: `/sbin/ifup ipsec0`.

The routers automatically generate the initialization scenario activated by the command `ifup` when the IPSEC connection is established.

With the following command, check the list of network routers: `/sbin/ip route list`.

Run the utility tcpdump to check the IPSEC connection, for example: `tcpdump -n -i eth0 host my_net1.com`

The packet must contain the AH and ESP data. In this case, the presence of the ESP will mean that encryption is working. Listing 9 shows an example of the check carried out on such a packet from the established connection:

Listing 9. Sample packet with encryption

```
12:24:26.155529 my_net2.com > my_net1.com: AH(spi=0x021c9834,seq=0x358): \
my_net2.com > my_net1.com: ESP(spi=0x00c887ad,seq=0x358) (DF) \
(ipip-proto-4)
```

Conclusions

In this article, you took a detailed look at the design principles, the basis for deploying VPN, and the IPSEC protocol concept, with a description of the general features of IPSEC and of the mechanisms required for its implementation.

You examined the arrangement of the methods for designing the protected connections using the IPSEC protocol. A detailed description of these implementations in relation to the two connection schemes, node-to-node and network-to-network, was provided alongside real examples.

Resources

Learn

- See technical documents on IPSEC on the IP Security Maintenance and Extensions, which shows the related RFCs and other documents.
- See details on the Internet Key Exchange IKE in rfc2409.
- Learn more about Linux FreeS/WAN on the project home page.
- Read the `setkey`(8) man page online.
- Read the `racoon`(8) man page online.
- Read the `racoon.conf`(5) man page online.
- Read Heterogeneous IPSec solution between AIX and Windows (Anto A. John and Akshay Kaushik, developerWorks, August 2010) to explore IPSEC in other environments.
- In the developerWorks Linux zone, find hundreds of how-to articles and tutorials, as well as downloads, discussion forums, and a wealth of other resources for Linux developers and administrators.
- The Open Source developerWorks zone provides a wealth of information on open source tools and using open source technologies.
- Stay current with developerWorks technical events and webcasts focused on a variety of IBM products and IT industry topics.
- Attend a free developerWorks Live! briefing to get up-to-speed quickly on IBM products and tools, as well as IT industry trends.
- Watch developerWorks on-demand demos ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Follow developerWorks on Twitter, or subscribe to a feed of Linux tweets on developerWorks.

Get products and technologies

- Explore CentOS Linux on the distribution home page.
- Evaluate IBM products in the way that suits you best: Download a product trial, try a product online, use a product in a cloud environment, or spend a few hours in the SOA Sandbox learning how to implement Service Oriented Architecture efficiently.

Discuss

- Check out developerWorks blogs and get involved in the developerWorks community.
- Get involved in the developerWorks community. Connect with other developerWorks users while exploring the developer-driven blogs, forums, groups, and wikis.

About the author

Aleksandr Derevyanko

Aleksandr Derevyanko has been working in the IT sector since 1990. He has worked in positions ranging from a “Category 1” engineer to head of an IT department. He is currently working as senior expert consultant in the department for engineering and technical support in the Administrative Office of Krasnodar Territory, Russian Federation.