IRAM-COMP-016

Revision: 0
2005-02-03

Contact Author

Virtual Private Network with OpenVPN

Owner Sebastien Blanchet

Approved by: A.Perrigouard	Date: Feb 2005	Signature:

Change Record

REVISION	DATE	AUTHOR	SECTION/PAGE AFFECTED	REMARKS

Content

1 Introduction

Fundamentally, a VPN is a set of tools which allow network at different locations to be securely connected, using a public network as the transport layer. VPNs use cryptography to provide protections against eavesdropping and active attacks. VPNs are most commonly used today for telecommuting and linking branch offices via secure WANs. The VPN concept is a cheap and secure alternative to a dedicated network to link branch offices.

1.1 IPSec

IPSec was the first major effort to develop a standard for secure networking. Unfortunately traditional IPSec implementations required a great deal of kernel code, complicating cross-plaform porting efforts. IPSec is complex for new users.

1.2 SSL and user-space VPNs

IPSec's slow progress and complexity caused many to turn to other solutions:
SSL (Secure Socket Layer) runs in user space, simplifying implementation and administration. Contrary to IPSEC, SSL matured quickly due to the heavy usage on the web.
The so-called SSL VPN is really just a web application that tries to give users the services they need without a full VPN implementation.

2 IPSec

Openswan is an implementation of IPSec for Linux. Unfortunately it is difficult to configure, and finally it never works. But I have learn many interesting things about digital certificates.

3 OpenVPN

By browsing the web to solve my IPSec problem, I have discovered the SSL VPN. I have chosen OpenVPN because it seemed to be easy to configure and because OpenVPN runs on many operating systems: Linux, Windows 2000/XP and higher, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris.

It uses a virtual network interface (tun or tap).

3.1 Tun and Tap interface

A tun interface is a virtual network adapter that looks like a point-to-point network hardware to the OS, but instead of pushings bits out a wire, the tun driver pushes them to the user space. A user space program can open the tun device just like a file and read and write IP packets from and to it.
A tap interface is also a virtual network adapter, but it only emulates the ethernet rather than point-to-point.

3.2 How it works

IP packets from tun or tap virtual network adapters are encrypted and encapsulated onto a UDP connection, and sent ot a remote host over the internet. The remote host decrypts, authenticates and de-encapsulates the IP packets, pumping then into a tun or tan virtual adapter at the other end.

The VPN is invisible to applications tunneling over it. One can apply routes or firewall rules to tun or tap interfaces in the same way that you can apply them to ethernet interfaces.

3.3 Installation

OpenVPN for windows
Download and install the windows client.

OpenVPN for Linux

Download the sources and build them

Install and build some libraries:

```
# tar xvfz lzo-1.08.tar.gz
# cd lzo-1.08
# ./configure
# make
# make check
# make test
# make install
```

Install missing rpms:

```
# rpm -i ./fc3/disc2/Fedora/RPMS/e2fsprogs-devel-1.35-11.2.i386.rpm
# rpm -i ./fc3/disc2/Fedora/RPMS/krb5-devel-1.3.4-7.i386.rpm
# rpm -i ./fc3/disc3/Fedora/RPMS/openssl-devel-0.9.7a-40.i386.rpm
./fc3/disc3/Fedora/RPMS/zlib-devel-1.2.1.2-1.i386.rpm
```

Build OpenVPN:

```
# tar xvfz openvpn-2.0_rc6.tar.gz
# cd openvpn-2.0_rc6
# ./configure
# make
# make install
```

Test the installation:

```
# openvpn --genkey --secret key
# openvpn --test-crypto --secret key
```

In two windows, in the same directory

```
./openvpn --config sample-config-files/loopback-client  (In one window)
./openvpn --config sample-config-files/loopback-server  (Simultaneously
in another window)
```

edit `/etc/modprobe.conf` to add `"alias char-major-10-200 tun"`

The command lines are valid for both Linux and Windows.

Example 1: A simple tunnel without security

On pctcp48:

```
# openvpn --remote pctcp72.iram.fr --dev tun1 --ifconfig 10.4.0.1
10.4.0.2 --verb 9
```

On pctcp72:

```
 # openvpn --remote pctcp48.iram.fr --dev tun1 --ifconfig 10.4.0.2
10.4.0.1 --verb 9
```

Use `ifconfig` to see the virtual ethenet adapter configuration.

on pctcp48: `ping 10.4.0.2`

Example 2: A tunnel with static-key security (i.e. using a pre-shared secret)

```
# openvpn --genkey --secret key
```

copy the key file on the both computer

On pctcp48:

```
# openvpn --remote pctcp72.iram.fr --dev tun1 --ifconfig 10.4.0.1
10.4.0.2 --verb 5 --secret key
```

On pctcp72:

```
 # openvpn --remote pctcp48.iram.fr --dev tun1 --ifconfig 10.4.0.2
10.4.0.1 --verb 5 --secret key
```

Example 3: A tunnel with full TLS-based security

In this example we use the keys and certificates provide with openvpn. See the Appendix to know how to generate your own keys and certificates.

on pctcp48.iram.fr

```
 #  openvpn --remote pctcp72.iram.fr --dev tun1 --ifconfig 10.4.0.1
10.4.0.2 --tls-client --ca tmp-ca.crt --cert client.crt --key
client.key --reneg-sec 60 --verb 5
```

on pctcp72.iram.fr

```
 #  openvpn --remote pctcp48.iram.fr --dev tun1 --ifconfig 10.4.0.2
10.4.0.1 --tls-server --dh dh1024.pem --ca tmp-ca.crt --cert server.crt
--key server.key --reneg-sec 60 --verb 5
```

Note on windows, you can run OpenVPN as a service, by creating configuration files in the configuration directory. See the OpenVPN for additional details.

4 Generate your own keys and certificates

Create a directory to store the certificates:

```
[root@pctcp48 ~]# mkdir /var/sslca
[root@pctcp48 ~]# chmod 700 /var/sslca
```

edit `/usr/share/ssl/misc/CA`

and replace DAYS="-days 365" by DAYS="-days 7000"
Fill also the default value for country, state, organization name, etc.

4.1 Generate your Certificate Autorithy

A pass phrase, i.e. a password to protect your CA is required. This password will be required every time you want to sign a digital certificate.

```
# cd /var/sslca/
[root@pctcp48 sslca]# /usr/share/ssl/misc/CA -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
....................++++++
.++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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 [FR]:
State or Province Name (full name) [France]:
Locality Name [Saint Martin DHeres]:
Organization Name [IRAM]:
Organizational Unit Name [Computer Group]:
Common Name (eg, your name or your server's hostname) []:pctcp48
Email Address []:

[root@pctcp48 sslca]#
```

Let's also generate a crl file, which you'll need on your gateway boxes:

```
[root@pctcp48 sslca]# openssl ca -gencrl -out crl.pem
Using configuration from /usr/share/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
[root@pctcp48 sslca]#
```

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.

4.2 Generating a Certificate

You will need to generate a certificate for every machine that will be making a secure connection. **An other pass phrase is required**, to protect the certificate usage.

```
[root@pctcp48 sslca]# /usr/share/ssl/misc/CA -newreq
Generating a 1024 bit RSA private key
.........................++++++
.........++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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 [FR]:
State or Province Name (full name) [France]:
Locality Name [Saint Martin DHeres]:
Organization Name [IRAM]:
Organizational Unit Name [Computer Group]:
Common Name (eg, your name or your server's hostname) []:linuxclient
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
[root@pctcp48 sslca]#
```

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:

```
[root@pctcp48 sslca]# /usr/share/ssl/misc/CA -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:pemca
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Feb  2 08:54:20 2005 GMT
            Not After : Jan 31 08:54:20 2015 GMT
        Subject:
            countryName               = FR
            stateOrProvinceName       = France
            localityName              = Saint Martin DHeres
            organizationName          = IRAM
            organizationalUnitName    = Computer Group
            commonName                = linuxclient
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            Netscape Comment:
            OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            B7:49:14:97:C5:A7:B0:56:EA:7C:99:B1:A6:E5:76:DB:A4:D1:64:9E
            X509v3 Authority Key Identifier:

keyid:9C:80:2A:AE:C8:11:71:D8:A9:EF:EA:7F:2A:C4:EF:76:AF:EA:CC:90
            DirName:/C=FR/ST=France/L=Saint Martin
DHeres/O=IRAM/OU=Computer Group/CN=pctcp48
            serial:00

Certificate is to be certified until Jan 31 08:54:20 2015 GMT (3650
days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=FR, ST=France, L=Saint Martin DHeres, O=IRAM,
OU=Computer Group, CN=pctcp48
        Validity
            Not Before: Feb  2 08:54:20 2005 GMT
            Not After : Jan 31 08:54:20 2015 GMT
        Subject: C=FR, ST=France, L=Saint Martin DHeres, O=IRAM,
OU=Computer Group, CN=linuxclient
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d5:f5:91:8f:5c:cb:31:5d:f4:90:ed:45:08:bb:
                    7c:24:1a:25:ae:b8:d5:af:17:95:85:9c:96:42:d1:
                    8f:49:30:71:3a:44:9e:80:62:37:bc:96:7c:73:f3:
                    ad:2f:75:c3:9b:bc:15:bb:9d:03:82:d9:f2:40:0d:
                    e7:50:7c:da:c6:16:e6:a5:2c:35:25:b7:22:c6:9c:
                    00:9d:0d:b6:55:fb:31:4a:b7:9d:51:15:9e:29:e4:
                    12:d1:d9:44:96:03:5b:62:3c:09:fd:4d:8f:67:b1:
                    5f:de:4f:64:ce:bb:8e:d3:86:a4:12:d8:a0:83:18:
                    a8:ae:03:0d:a4:1a:c6:18:55
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            Netscape Comment:
            OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            B7:49:14:97:C5:A7:B0:56:EA:7C:99:B1:A6:E5:76:DB:A4:D1:64:9E
            X509v3 Authority Key Identifier:

keyid:9C:80:2A:AE:C8:11:71:D8:A9:EF:EA:7F:2A:C4:EF:76:AF:EA:CC:90
            DirName:/C=FR/ST=France/L=Saint Martin
DHeres/O=IRAM/OU=Computer Group/CN=pctcp48
            serial:00

    Signature Algorithm: md5WithRSAEncryption
        d8:89:1a:0a:fc:a9:04:ff:a3:27:ce:02:a2:b1:0c:7f:be:c4:
        62:b0:5f:82:86:c0:ea:aa:a5:a8:71:35:3f:35:c1:3c:68:0b:
        c6:e7:ab:7b:ec:4d:a0:9d:3a:86:9c:51:88:36:d3:b3:e5:45:
```

```
        71:55:49:4a:98:b5:30:db:7b:ae:12:ab:bf:af:80:2d:50:0d:
        9b:fc:dc:5d:96:74:65:de:c5:47:fd:c7:bd:1d:ba:4a:ab:d4:
        a6:57:21:1b:13:bb:4a:0f:cc:df:57:4c:ee:45:a0:07:88:4e:
        ad:fb:06:76:13:a3:9c:49:fe:5b:96:1e:f1:7f:8d:ee:13:4e:
        13:9a
-----BEGIN CERTIFICATE-----
MIIDZTCCAs6gAwIBAgIBATANBgkqhkiG9w0BAQQFADB2MQswCQYDVQQGEwJGUjEP
MA0GA1UECBMGRnJhbmNlMRwwGgYDVQQHExNTYWludCBNYXJ0aW4gREhlcmVzMQ0w
CwYDVQQKEwRJUkFNMRcwFQYDVQQLEw5Db21wdXRlciBHcm91cDEQMA4GA1UEAxMH
cGN0Y3A0ODAeFw0wNTAyMDIwODU0MjBaFw0xNTAxMzEwODU0MjBaMHoxCzAJBgNV
BAYTAkZSMQ8wDQYDVQQIEwZGcmFuY2UxHDAaBgNVBAcTE1NhaW50IE1hcnRpbiBE
SGVyZXMxDTALBgNVBAoTBElSQU0xFzAVBgNVBAsTDkNvbXB1dGVyIEdyb3VwMRQw
EgYDVQQDEwtsaW51eGNsaWVudDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
1fWRj1zLMV30kO1FCLt8JBolrrjVrxeVhZyWQtGPSTBxOkSegGI3vJZ8c/OtL3XD
m7wVu50DgtnyQA3nUHzaxhbmpSw1JbcixpwAnQ22VfsxSredURWeKeQS0dlElgNb
YjwJ/U2PZ7Ff3k9kzruO04akEtiggxiorgMNpBrGGFUCAwEAAaOB/jCB+zAJBgNV
HRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZp
Y2F0ZTAdBgNVHQ4EFgQUt0kUl8WnsFbqfJmxpuV226TRZJ4wgaAGA1UdIwSBmDCB
lYAUnIAqrsgRcdip7+p/KsTvdq/qzJCheqR4MHYxCzAJBgNVBAYTAkZSMQ8wDQYD
VQQIEwZGcmFuY2UxHDAaBgNVBAcTE1NhaW50IE1hcnRpbiBESGVyZXMxDTALBgNV
BAoTBElSQU0xFzAVBgNVBAsTDkNvbXB1dGVyIEdyb3VwMRAwDgYDVQQDEwdwY3Rj
cDQ4ggEAMA0GCSqGSIb3DQEBBAUAA4GBANiJGgr8qQT/oyfOAqKxDH++xGKwX4KG
wOqqpahxNT81wTxoC8bnq3vsTaCdOoacUYg207PlRXFVSUqYtTDbe64Sq7+vgC1Q
DZv83F2WdGXexUf9x70dukqr1KZXIRsTu0oPzN9XTO5FoAeITq37BnYTo5xJ/luW
HvF/je4TThOa
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
```

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

```
[root@pctcp48 sslca]# mv newcert.pem server.pem
[root@pctcp48 sslca]# mv newreq.pem server.key
```

Then move them to the appropriate location for your applications.

5 Biography

Openvpn website: http://www.openvpn.net/
Nate Karlson webpage about Openswan http://www.natecarlson.com/linux/ipsec-x509.php