

# hakin9

## Anti-Sniffing, Privacidad y VPN

Gosub

Artículo publicado en el número 5/2006 de la revista *hakin9*. Os invitamos a la lectura de toda la revista.  
Todos los derechos protegidos. Distribución gratuita admitida bajo la condición de guardar la forma y el contenido actuales del artículo. Revista *hakin9* Software-Wydawnictwo, ul. Bokserska 1, 02-682 Warszawa, [es@hakin9.org](mailto:es@hakin9.org)



Práctica

# Anti-Sniffing, Privacidad y VPN

Gosub



Grado de dificultad



Desde los orígenes de las redes, las comunicaciones masivas, internet y recientemente con las redes wifi, el tráfico de paquetes se ha vuelto de uso diario, prioritario y necesario. Si agregamos que las empresas y personas usan las redes para compras electrónicas, accesos al banco, intercambio de información confidencial y hay gobiernos/hackers mirando nuestro tráfico, no podemos dejar que la información fluya libremente sin garantizar su privacidad.

Los fundamentos de las redes informáticas fueron la de compartir, vincular, acercar, conectar y unir equipos y personas. *UNIX fue construido para compartir* (Dennis Ritchie).

Al aparecer los ordenadores personales y luego la PC se multiplicaron los ordenadores en los hogares. El ordenador personal permitió procesamiento local y aparece un ordenador en cada oficina y en cada casa. Desde los orígenes de TCP-IP y con la llegada de Internet a los hogares y empresas, nació una era de conectividad masiva mundial. Esos beneficios se multiplicaron aún más con la masificación de las redes wifis. En muchos hogares se instalan routers con conexión wifi, se multiplican los tele-trabajos, los portátiles vendidos, los hot-spots, los accesos a Internet públicos y en empresas. Aparecen conceptos como comercio via Internet, el Chat para usos comerciales, Email para intercambio de información entre empresas/personas y nuevas herramientas digitales para realizar negocios.

## Aparecen los riesgos

La comodidad de una red sin cables trae adherido un riesgo infinito de que nuestras comunicaciones sean interceptadas. El concepto

## En este artículo aprenderás...

- Como montar, instalar, configurar y poner en funcionamiento una red privada entre ordenadores, con varios sistemas operativos diferentes.
- Métodos para verificar la comunicación y como ver datos encriptados.
- Explicar los problemas actuales relacionados con las redes locales, wifi e Internet, explicar de una manera simple como armar una red privada virtual, y brevemente los beneficios de la solución propuesta.

## Lo que deberías saber...

- Instalar varios sistemas operativos.
- Instalar productos en diferentes sistemas operativos.
- Conocimientos básicos de TCP-IP.
- Conceptos básicos de diferentes algoritmos criptográficos.
- Conceptos básicos sobre VPN.
- Técnicas sobre hacking, sniffing, arp poison y otros métodos.
- Conceptos de vulnerabilidades existentes en redes y protocolo TCP-IP.

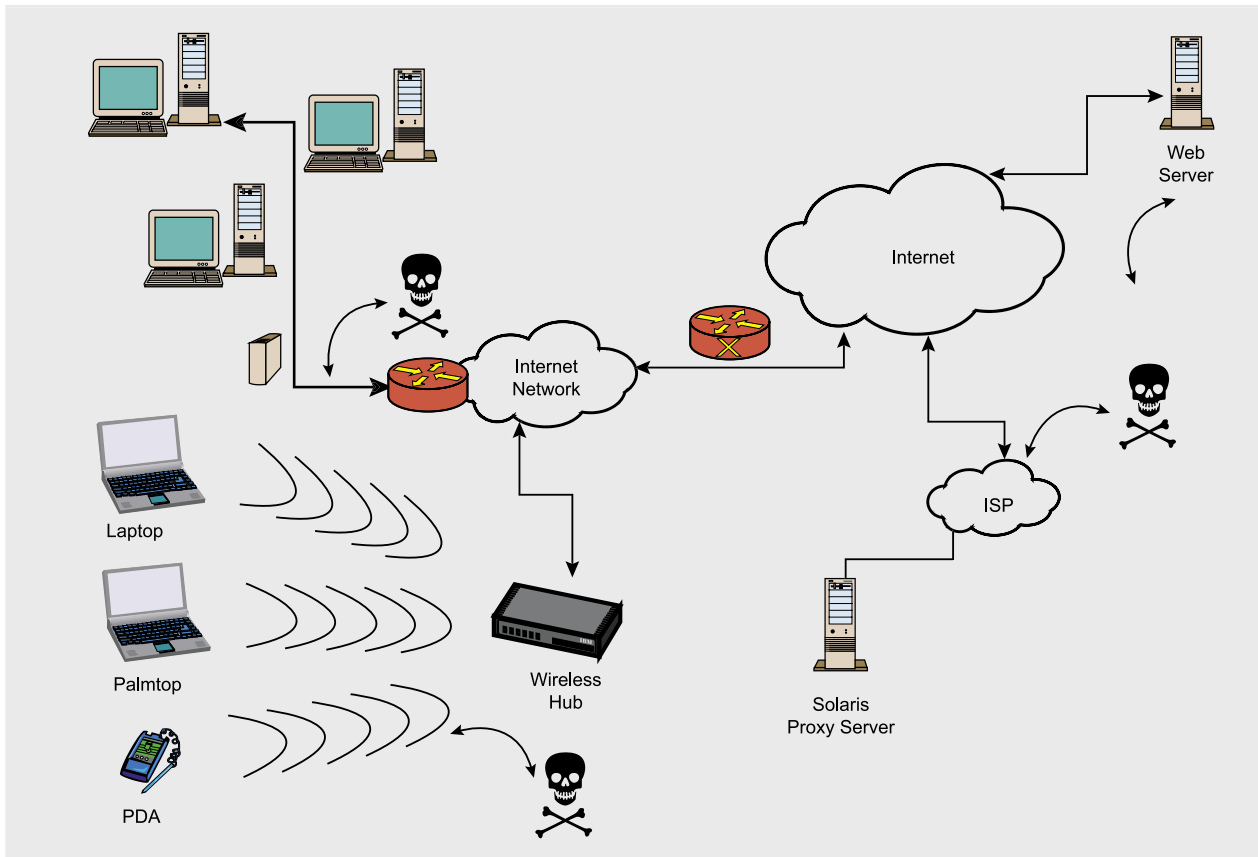


Figura 1. Gobierno y Hackers pueden ver nuestras comunicaciones

seguridad física no aplica en redes wifi, solo en redes en empresas y bajo ciertas condiciones.

Es demasiado fácil interceptar redes wifis, aún cuando la red utilice WEP o similares. Aún en el caso de utilizar cables existe el riesgo de que nuestro tráfico sea interceptado y visto. Toda empresa que mantenga comunicaciones usando Internet o redes wifi, debe tener un sistema que garantice su privacidad.

Alguien fuera de nuestra casa cómodamente sentado en su coche, nuestro vecino, un empleado del ISP, un administrador de servidores, o alguien bastante más malo podría ver también nuestros datos (ver Figura 1).

El protocolo TCP-IP tiene en su cabecera información del destino de cada paquete, de manera que los routers lo envíen donde corresponde y si llega a un ordenador que no es el destino, la placa de red de ese ordenador lo descarta. Una placa de red en modo promiscuo permite a la capa aplicación, ver información que no es para el (ver Figura 2).

Un paquete sale de un ordenador y puede ser visto por un ordenador en medio, aun sin ser el destino del paquete.

Se ha hablado mucho de los algoritmos fuertes para mejorar la seguridad y se han desarrollado sistemas de encriptación de comunicaciones. Hasta hace unos años esos

sistemas estaban diseñados para empresas, costaban demasiado dinero y no eran fáciles de implementar. Pero eso ha cambiado...

### Aparecen soluciones

Desde hace unos pocos años, es relativamente fácil montar un sistema de comunicaciones seguro, en

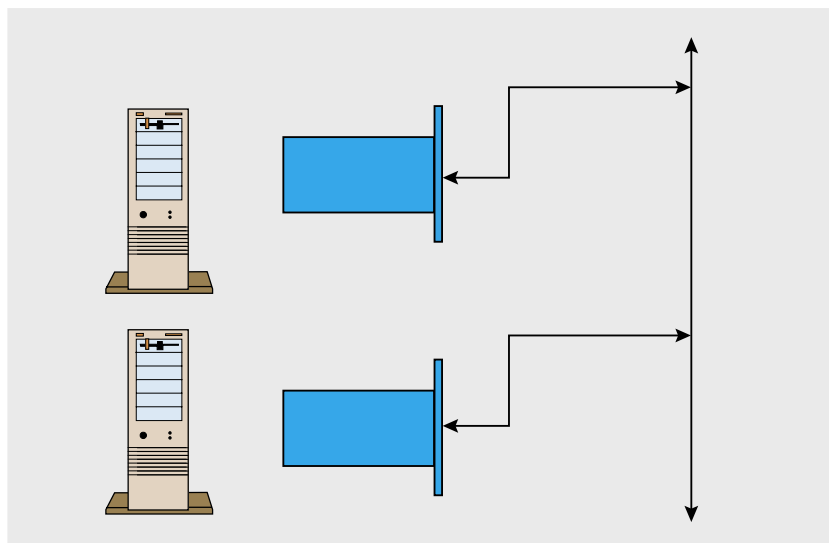


Figura 2. Un paquete puede ser leído en el camino

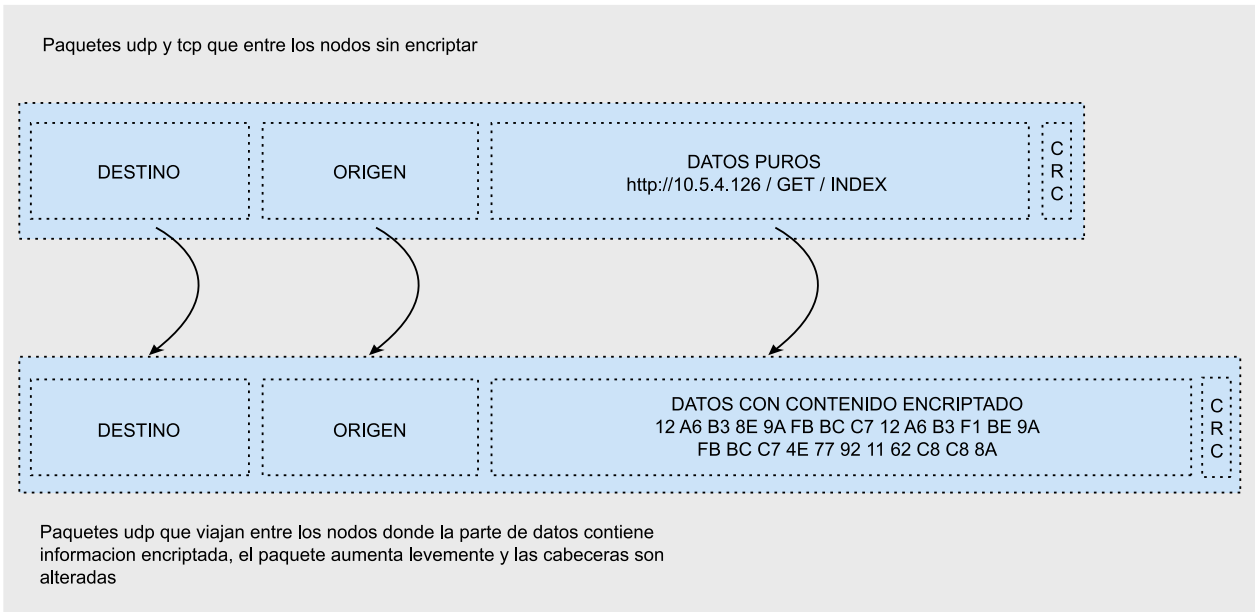


Figura 3. Un paquete sin encriptar y luego de ser encriptado

cualquier sistema operativo y que utilice algoritmos duros (no voy a decir inviolables). La encriptación, oculta y encapsula datos reales en paquetesTCP-IP (ver Figura 3).

Esas comunicaciones son transparentes y viajan por redes, (cables, wifi o Internet), garantizando que únicamente el destino podrá des-encriptar y ver el paquete original. Para probarlo, montaremos una VPN entre un server y dos clientes y analizaremos el tráfico generado. (con OpenVPN).

Usaremos un servidor Windows y dos clientes, uno con Linux y uno con un Unix. Generaremos una red privada de alta seguridad (ver Figura 4).

La conexión sera entre equipos Widnows, Linux y Unix.Usaremos Ethereal para ver los paquetes encriptados y sin encriptar. El tiempo total para probar el modelo es de 4 hs.

### Características del producto OpenVPN

OpenVpn funciona en Linux, Windows 2000/XP o superior, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris. El producto puede funcionar como bridge o router, en nuestro ejemplo utilizaremos el modo router (ver Figura 5). Sus características más importantes:

Tabla 1. Detalle de las claves generadas

Filename	Needed By	Purpose	Secret
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman parameters	NO
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES
client2.crt	client2 only	Client2 Certificate	NO
client2.key	client2 only	Client2 Key	YES
client3.crt	client3 only	Client3 Certificate	NO
client3.key	client3 only	Client3 Key	YES

#### Listado 1. Detalle del fichero server.ovpn

```

c:
cd \Archivos de programas
\openvpn\bin
openvpn server.ovpn
El fichero server.ovpn contiene:
#####04/2006 #####
local SERVER1
mode server
management localhost 7505
port 1194
proto udp
dev tap
dev-node vpn
ifconfig 192.192.192.
1 255.255.255.0
ca "key\ca.crt"
cert "key\server1.crt"
key "key\server1.key"
dh "key\dh2048.pem"
ifconfig-pool 192.192.192.
10 192.192.192.15
client-to-client
keepalive 10 120
tls-auth key1 0 #
This file is secret
tls-server
comp-lzo
max-clients 5
persist-key
persist-tun

status openvpn-status.log

log openvpn.log

verb 4 #puede ser 9 para ver mas
info al principio, luego
4 es suficiente
#####04/2006 #####

```

# ¡Ya a la venta!

También puedes comprarlo en nuestra tienda virtual:  
[www.buyitpress.com](http://www.buyitpress.com)

2 x DVD openSuSE 10.1 Instalación Configuración Paquetes adicionales

openSuSE 10.1

# openSuSE 10.1

Versión completa de una distribución segura de Linux

ISBN: 978-84-937-1763-3

openSuSE 10.1 Instalación Configuración Paquetes adicionales 2 x DVD

### Sólo aquí

Más de 3000 paquetes adicionales  
¡Paquetes para la reproducción de MP3 y las películas!

Última distribución de Linux - estable, eficaz, seguro, durable  
Fácil instalación para los principiantes  
Sistema operativo completo  
Suite de oficina completo  
Soporte del equipo más moderno  
Seguridad de uso de Internet

2x  
DVD

LINUX+  
Extra Pack



### Libros en PDF

Advanced Bash-Scripting Guide  
Bash Beginner's Guide  
Custom Porting Guide  
Introduction to Linux  
Linux Dictionary  
Linux Media Guide  
Securing and Optimizing Linux  
- The Ultimate Solution  
System Administrator's Guide

### Versiones completas

Software comercial para la empresa  
LeftHand CRM  
LeftHand Contabilidad simple  
LeftHand Contabilidad completa

### BONUS

openSUSE 10.1 LiveDVD  
¡Mira como funciona SUSE sin tener que instalarlo!  
10 tutoriales video  
Resuelve los problemas típicos mediante los tutoriales video

### SUPER 10.1

Una versión especial de openSUSE enfocada en la efectividad

[www.lpmagazine.org](http://www.lpmagazine.org)

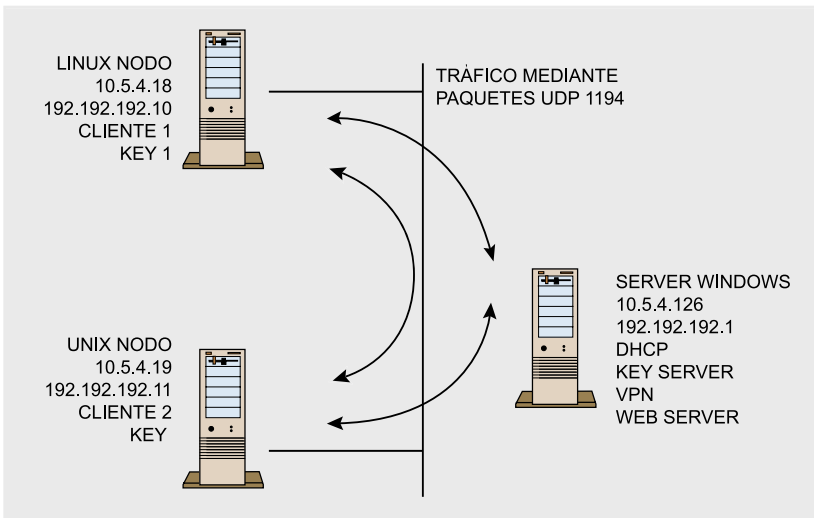


Figura 4. Esquema de la VPN que montaremos

- Libre
- Utiliza TLS
- Cross-platform
- Admite redes en estrella (1-N)
- Encapsula lógicamente
- Admite balance de carga
- Varios algoritmos de encriptación, Clave estática y/o certificados
- Tiene GUI p/Windows
- Soporta road warriors (DHCP)
- Una clave para cada cliente/nodo
- Puede actuar como router o bridge
- Genera un dispositivo virtual sobre una placa física
- Genera uno o mas dispositivos lógicos

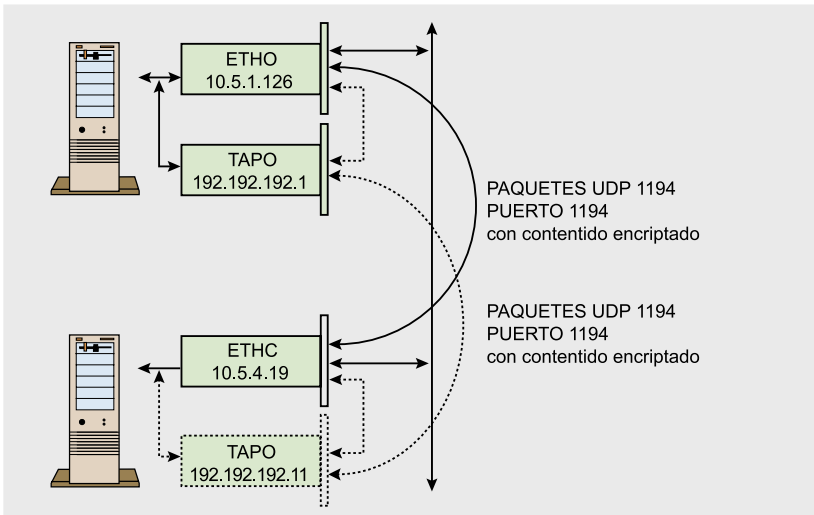


Figura 5. Las dos redes, la real y la red virtual VPN

La comunicación se realiza virtualmente entre las placas virtuales, pasando los paquetes a través de la placa física real (ver Figura 6).

### ¿Que dice OpenVPN del sistema de encriptación?

OpenVPN's security model can be summarized as such: Use the IPsec ESP protocol for tunnel packet security, but then drop IKE in favor of SSL/TLS for session authentication. This allows for a lightweight, portable VPN implementation that draws on IPsec's strengths, without introducing the complexity of IKE (openvpn oficial site).

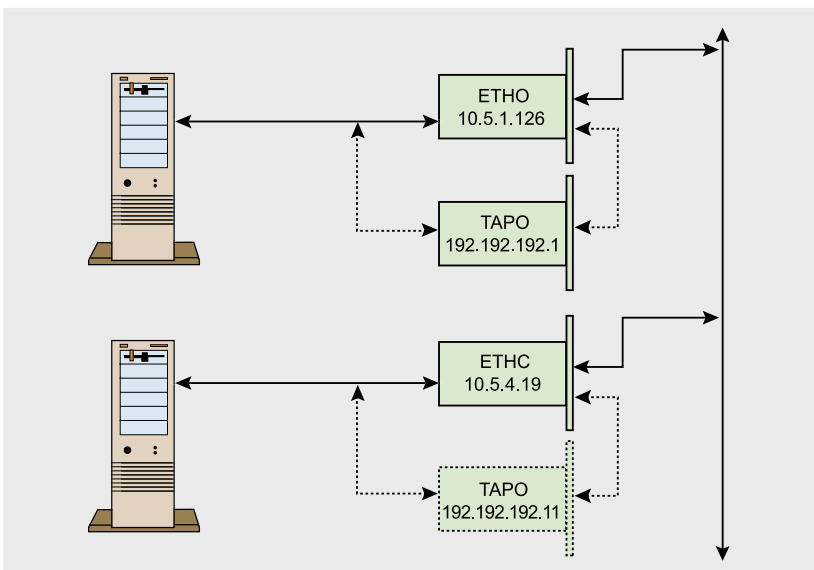


Figura 6. Un paquete pasa por la placa virtual, luego por la real y sale al medio

### Modo cliente servidor

El producto ofrece un modo eficiente y escalable del tipo Servidor con uno o mas clientes. Funciona en forma transparente y utiliza el puerto 1194 (se puede cambiar). Ver Figura 6.

Se generan placas logicas montadas sobre las placas físicas. En caso de tener firewalls solo hace falta habilitar ese puerto entre los equipos que formarán la VPN. OpenVpn genera una placa de red virtual y le define un rango de IP para las comunicaciones encriptadas entre los nodos. Ese proceso es transparente y automático. Para tener comunicaciones estables es conveniente (no excluyente) utilizar la misma versión del producto en todos los nodos.



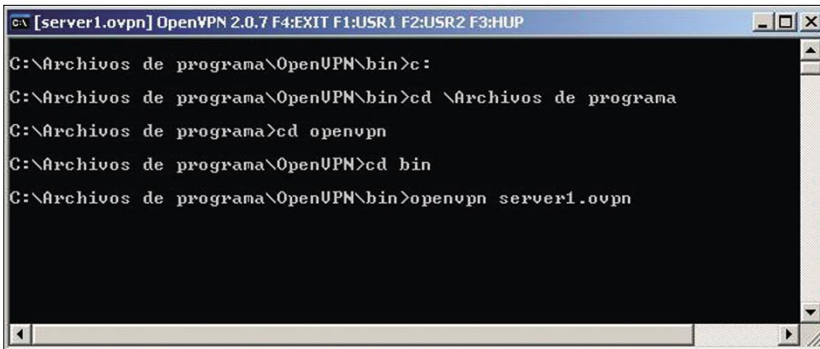


Figura 7. Ventana que aparece en Windows al arrancar el servidor OpenVPN

## Ahora la acción

En nuestra práctica utilizaremos tres equipos, un Windows XP, un Linux (Debian 3.1 2.6) y un Unix (FreeBSD 6.1)

### Paso 1. Instalar OpenVPN

LINUX: En entornos Linux se baja el paquete RPM, DEB o fuentes. En nuestro equipo haremos:

```
# apt-get install openvpn
```

UNIX: Se puede bajar el port o el paquete. Usaremos el port:

```
# cd /usr/ports/security/openvpn
# configure
# make
# make install
# make clean
```

WINDOWS: En Windows, bajamos el instalador y lo ejecutamos :

```
openvpn-2.0.7-install.exe
(http://openvpn.net/release/
openvpn-2.0.7-install.exe)
```

Información de los nodos:

- Servidor: Windows XP PRO SP2; IP 10.5.4.126; IPVNP 192.192.192.1
- Cliente 1: Debian 3.1r (unstable); IP 10.5.4.248; IPVPN 192.192.192.10
- Cliente 2: FreeBSD 6.1; IP 10.5.4.249; IPVPN 192.192.192.11
- Firewall:

Si tenemos un firewall deberá filtrar todas las comunicaciones entre el server y todos los nodos.

```

\OpenVPN\easy-rsa
set KEY_CONFIG=openssl.cnf
set KEY_DIR=keys
set KEY_SIZE=2048
set KEY_COUNTRY=ES
set KEY_PROVINCE=MA
set KEY_CITY=Madrid
set KEY_ORG=POINTGOV
set KEY_EMAIL=MINE@server.gov
    
```

Ejecutar otros batches para preparar entorno:

```

c:\> vars
c:\> clean-all
c:\> build-ca
    
```

el archivo generado (ca.crt) contiene algo como:

```

-----BEGIN CERTIFICATE-----
MIIDdTCCAt6gAwIBAgIJAND5/S7gDnIcMA
0GCSqGSIb3DQEBAUMAMIGEMQswCQYD
ZfTebSIOAtEj8ajHz+ZseLcAxv91gINXT4m
.....
5Cdx2RusYICEa0o7nWB3p80ubIxmKwKQ
vzn3odkXs1JXXQrk9r1Soo7DJimZ9F
RxtMvRN4h4w10c59Kkoh+zaFdg422UKLAQ==
-----END CERTIFICATE-----
    
```

La clave generada (ca.key) contiene algo como:

```

-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKQBDBpLcQubJwDvQvQJb
TKPQLJGeTolvV9xsFeYdxR0IBcvXScZ5DB
UTOEb7yK1BDCExlW+Uz7B1XwvTf3gM
    
```

Solo debemos activar el puerto 1194 (tcp y udp) en ambos sentidos.

### Paso 2. Crear claves

En el servidor Windows generaremos la master key (A), la server key (B) y una clave para los dos nodos (C). Recordar: cada nodo poseerá una clave única para el mismo.

(A) Generar clave maestra: INICIO – EJECUTAR – CMD; Microsoft Windows XP [Versión 5.1.2600]; (C) Copyright 1985-2001 Microsoft Corp.

```

C:\> cd \program files\openvpn\easy-rsa
C:\> init-config
C:\> notepad+ vars.bat
    
```

Modificamos el contenido del Fichero (mostramos el fichero completo):

```

@echo off
set HOME=%ProgramFiles%
    
```

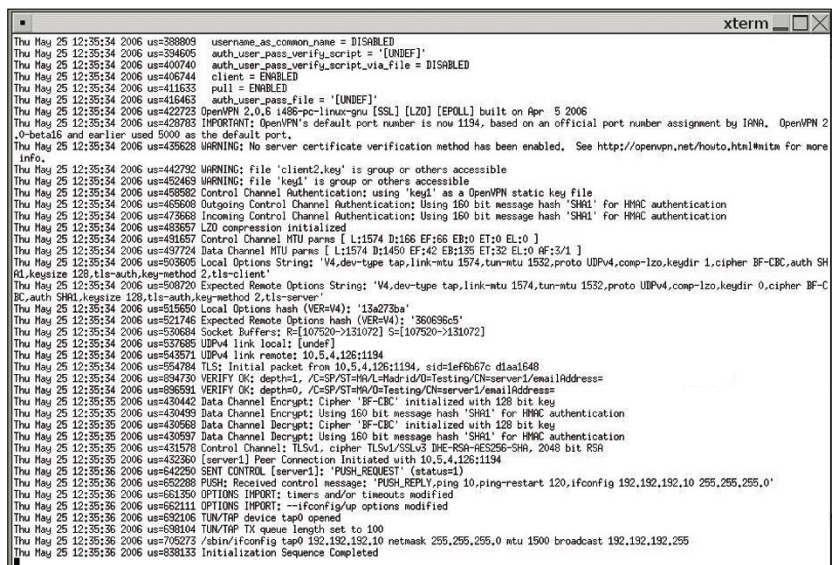


Figura 8. Pantalla LOG del arranque del cliente LINUX



```

MQFZ/YXV57G/2KvTVSQK3iJXSVx8kucb8E
-----
16rYgN+GGtv9wG+3PYKhIuRTruejBjAVp
Sp1CCXKsGgXu
-----END RSA PRIVATE KEY-----

```

### Generar clave para el servidor:

Ejecutamos:

```
C:\> build-key-server server
```

Poner opciones y un nombre para ese Server, ingresar una password y los datos del servidor1. El fichero generado (server.key) contiene algo como:

```

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCuzD
Kgf05ExGcu3ogHez9
Gh79mlf4RfP1S0d3TT3z1FczEfb4s
Hl0FZGz9rhvA8HxxTdg
OPPdDq+f+jywaS7
Y4SgByT1qVe0+ +Y
3XIVG9Is6KrkRd+
-----
W7o1r/Rh+tJimZv5Yt1FFk
GAJJo3Hc
RSaNatomhD+5H0g==
-----END RSA PRIVATE KEY-----

```

### Generar clave para cada nodo

Clave para el nodo LINUX. Ejecutamos:

```
C:\> BUILD-KEY CLIENTE1
```

La clave generada (client1.key) contiene algo como:

```

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDv2bv3
/diRvYn1ypqPL
nSgG3YrsE+6AGjXp
+ocUC1vG1YjVR70
aAOCJiJPKNoshcmKg
+ECQCFCflofa
AxqvnihsTYXipVmOwgRn4
ArUrVNj9UCmmp/F
-----
5ow/F8IVfTcQK+f/JZd/5
+2Gslj5BBgteNmW3/Zd4t8=
-----END RSA PRIVATE KEY-----

```

Clave para el nodo UNIX. Ejecutamos:

```
C:\> BUILD-KEY CLIENTE2
```

```

eth0    Link encap:Ethernet HWaddr 00:0C:29:42:CF:1E
        inet addr:10.5.4.131 Bcast:10.5.5.255 Mask:255.255.254.0
        inet6 addr: fe80::20c:29ff:fe42:c1e/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:712 errors:0 dropped:0 overruns:0 frame:0
        TX packets:741 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:740329 (722.9 KiB) TX bytes:112854 (110.2 KiB)
        Interrupt:169 Base address:0x1400

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:107 errors:0 dropped:0 overruns:0 frame:0
        TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:33137 (32.3 KiB) TX bytes:33137 (32.3 KiB)

tap0    Link encap:Ethernet HWaddr 4A:95:72:EA:70:1C
        inet addr:192.192.192.10 Bcast:192.192.192.255 Mask:255.255.255.0
        inet6 addr: fe80::4895:72ff:feea:701c/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:474 errors:0 dropped:0 overruns:0 frame:0
        TX packets:475 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:46054 (44.9 KiB) TX bytes:46038 (44.9 KiB)

```

Figura 9. Pantalla de las placas de red, donde se ve la placa física eth0, la placa virtual TAP0 y los IPs

La clave generada (client2.key) contiene algo como:

```

-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQC2g
M6BGB/iLKpZAac
Ye3lew/iux
A78bj6AHqcq
VXDD9MLFxaZzX
zHfVbn0pr08CpgrhIbyD
RQ/N4W7KJseXPI
Zu00x6cMTnfaLH5/5Zz
L2X0/pMtAsY
mFxNNBbrTH3DnY68p
VI0usmBqFtj
MdevdM85/5Ifv26XEgr
E32ASjBqHTQ1DAQAB
-----

```

```

BrTaC8IEcnfruWgLQc8CQ
QDRYmac8H
8aMe3ys070BW3JlBmD/fDp
3wfknbZWXC+
M2ZoDNiY6ZiiTiDNhE
QuQvRUonv
FNWeUill4hQtA/XH1
-----END RSA PRIVATE KEY-----

```

Clave DH para el server:

```

C:\> BUILD-DH
C:\Archivos de programa
\OpenVPN\easy-rsa>build-dh
Loading 'screen' into
random state - done
Generating DH parameters,
1024 bit long safe prime, generator 2

```

```

Thu May 25 12:37:13 2006 us=014653 cf_per = 0
Thu May 25 12:37:13 2006 us=014653 max_clients = 1024
Thu May 25 12:37:13 2006 us=014646 max_routes_per_client = 256
Thu May 25 12:37:13 2006 us=018268 client_cert_not_required = DISABLED
Thu May 25 12:37:13 2006 us=015479 user_name_as_common_name = DISABLED
Thu May 25 12:37:13 2006 us=015583 auth_user_pass_verify_script = (UNDEF)
Thu May 25 12:37:13 2006 us=015658 auth_user_pass_verify_script_via_file = DISABLED
Thu May 25 12:37:13 2006 us=015727 client = ENABLED
Thu May 25 12:37:13 2006 us=015809 pull = ENABLED
Thu May 25 12:37:13 2006 us=015887 auth_user_pass_file = (UNDEF)
Thu May 25 12:37:13 2006 us=016056 OpenVPN 2.0.6 [SSL] [LZO] built on May 24 2006
Thu May 25 12:37:13 2006 us=019340 IMPORTANT: OpenVPN's default port number is now 1194, based on an official port number assignment by IANA. OpenVP
N 2.0-beta5 and earlier used 5000 as the default port.
Thu May 25 12:37:13 2006 us=019840 WARNING: No server certificate verification method has been enabled. See http://openvpn.net/howto.html#mitm for a
core info.
Thu May 25 12:37:13 2006 us=019523 Control Channel Authentication: using 'keyd' as a OpenVPN static key file
Thu May 25 12:37:13 2006 us=012275 Outgoing Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
Thu May 25 12:37:13 2006 us=012908 Incoming Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
Thu May 25 12:37:13 2006 us=03164 LZO compression initialized
Thu May 25 12:37:13 2006 us=024086 Control Channel MTU parms [ L:1574 B:1450 EF:42 EB:136 EI:0 ET:0 E1:0 ]
Thu May 25 12:37:13 2006 us=026635 Data Channel MTU parms [ L:1574 B:1450 EF:42 EB:136 EI:0 ET:0 E1:0 ]
Thu May 25 12:37:13 2006 us=026448 Local Options String: 'V4,dev-type tap,link-mtu 1574,tun-mtu 1532,proto UDPv4,comp-lzo,keydir 1,cipher BF-CBC,auth
-SHA1,keysize 128,tls-auth,key-method 2,tls-client'
Thu May 25 12:37:13 2006 us=026852 Expected Remote Options String: 'V4,dev-type tap,link-mtu 1574,tun-mtu 1532,proto UDPv4,comp-lzo,keydir 0,cipher B
F-CBC,auth SHA1,keysize 128,tls-auth,key-method 2,tls-server'
Thu May 25 12:37:13 2006 us=027051 Local Options hash (VER=V4): '13a273ba'
Thu May 25 12:37:13 2006 us=027888 Expected Remote Options hash (VER=V4): '366886cd'
Thu May 25 12:37:13 2006 us=028443 Socket Buffers: R=[4096->65536] S=[8192->65536]
Thu May 25 12:37:13 2006 us=020911 UDPv4 link local: [undef]
Thu May 25 12:37:13 2006 us=023022 UDPv4 link remote: 10.5.4.126:1194
Thu May 25 12:37:13 2006 us=018820 TLS: Initial packet from 10.5.4.126:1194, sid=db2a412 72d1291
Thu May 25 12:37:14 2006 us=134101 VERIFY OK: depth=1, C=SP,R=CN=/sbin/0pTesting/CN=server1/emailAddress=
Thu May 25 12:37:14 2006 us=17146 VERIFY OK: depth=0, C=SP,S=CN=/sbin/0pTesting/CN=server1/emailAddress=
Thu May 25 12:37:14 2006 us=415286 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Thu May 25 12:37:14 2006 us=418038 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Thu May 25 12:37:14 2006 us=420140 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Thu May 25 12:37:14 2006 us=420215 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Thu May 25 12:37:14 2006 us=422628 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-WSS300-SHA, 2048 bit RSA
Thu May 25 12:37:14 2006 us=423167 [server:] Peer Connection Initiated with 10.5.4.126:1194
Thu May 25 12:37:15 2006 us=526355 SENT CONTROL [server1]: 'PUSH_REQUEST' (status=1)
Thu May 25 12:37:15 2006 us=619241 PUSH: Received control message: 'PUSH_REQUEST' (status=1)
Thu May 25 12:37:15 2006 us=578247 OPTIONS IMPORT: timers and/or timeouts modified
Thu May 25 12:37:15 2006 us=579049 OPTIONS IMPORT: --ifconfig-prio options modified
Thu May 25 12:37:15 2006 us=626622 TAP/TUN device /dev/tap0 opened
Thu May 25 12:37:15 2006 us=684800 /sbin/ifconfig tap0 192.192.192.11 netmask 255.255.255.0 mtu 1500 up
Thu May 25 12:37:15 2006 us=749472 Initialization Sequence Completed

```

Figura 10. Pantalla LOG del arranque del cliente LINUX



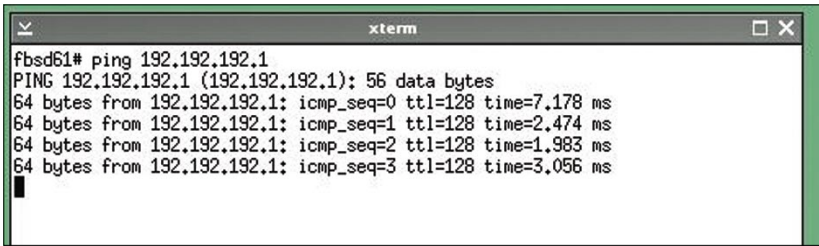


Figura 11. Ping desde Linux al servidor Windows usando encriptación

```
This is going to take a long time
.....
.....+.....+.....
.....+.....+.....
.....+.....+.....+.....+.....
```

Los archivos generados los copiamos a un pendrive (disco PGP) lo necesitaremos al configurar los clientes. Si los clientes son remotos,

se pueden enviar a cada uno de las claves que necesitarán por algún método seguro. Cada cliente utilizará un juego de claves diferente.

### Paso 3. Configurar servidor

Copiar la carpeta:

```
C:\Archivos de programa
\OpenVPN\easy-rsa\keys
```

A la carpeta:

```
C:\Archivos de programa
\OpenVPN\bin
```

La carpeta:

```
C:\Archivos de programa
\OpenVPN\easy-rsa\keys
```

Se debe pasar a un medio externo y asegurarla. Luego eliminarla del servidor. Se debe crear una cuenta para lanzar el servicio OVPN en el servidor, que tenga permisos en la carpeta:

```
C:\Archivos de programa
\OpenVPN\
```

Luego asegurar esa cuenta. En la carpeta:

```
C:\Archivos de programa
\OpenVPN\bin
```

Creo un fichero llamado *SERVER.BAT*. Agrego el contenido tal como vemos en el Listado 1.

Para nuestra prueba ejecutaremos cada nodo en modo interactivo, luego se pueden transformar en servicios o demonios. En nuestra prueba, hemos renombrado la placa en Windows XP a "VPN" para simplificar su nombre en los scripts.

### Paso 4. Configurar clientes

Cliente LINUX: en */etc/openvpn* creo un fichero llamado *cliente1.sh* que contiene:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
Openvpn Client1.ovpn
```

El fichero *cliente1.ovpn* contiene:

- Client
- Dev tap0
- Proto udp
- # ip server puerto server
- Remote 10.5.4.126 1194
- Resolv-retry infinit
- Nobind
- # user y group para asegurar el producto

#### Listado 2. Paquete sin encriptación

```
(se puede ver !"#%&'()*+,-./01234567, como texto plano)
0000 00 ff 92 2b 6c 52 00 bd 4e d2 0e 00 08 00 45 00 ...+lR.. N....E.
0010 00 54 06 61 00 00 40 01 72 ba c0 c0 c0 0b c0 c0 ..T.a..@. r.....
0020 c0 01 08 00 fc 5f 6f 0b 00 00 44 80 13 80 00 07 ....._o...D....
0030 49 8a 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 I.....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 67
```

#### Listado 3. Paquete ya encriptado

```
0000 00 13 21 00 0d 2c 00 0c 29 ea 33 9a 08 00 45 00 ..!.,,.. ).3...E.
0010 00 a1 06 76 00 00 40 11 56 cf 0a 05 04 80 0a 05 ...v..@. V.....
0020 04 7e f9 4a 04 aa 00 8d 6f 36 34 65 8e 49 5e 3a ..~.J... o64e.I^
0030 31 0e 8b b7 fc 75 87 b2 a9 86 d5 9a c4 9d 26 e5 1.....&.
0040 a8 c6 47 4b 5b 5a 98 ac 47 0c ce fe 91 c5 fa 57 ..GK[Z.. G.....W
0050 ef 01 e6 f2 d6 e8 7f c8 1c f5 18 3a d7 94 f2 cf .....
0060 78 2e 8e 55 7e f5 88 ac 3d 58 55 8b ff a9 c8 b6 x..U~... =XU....
0070 e7 0d ed 85 cd 63 fd e0 24 aa 2e ee 8d 4e 48 cd .....c. $....NH.
0080 16 b5 2b 14 42 44 7b b5 57 a8 c6 da 22 c7 dd 0b ..+.BD{. W..."...
0090 ce 7b 85 6f 72 33 e8 07 e8 47 f4 50 18 b8 3f 8c .{.or3.. .G.P...?
00a0 5f ac 5e d4 f0 f6 90 04 73 09 24 7d dd 92 0d ...^..... s.$)...
```

#### Listado 4. Paquete Replay encriptado

```
0000 00 0c 29 ea 33 9a 00 13 21 00 0d 2c 08 00 45 00 ..).3... !.....E.
0010 00 a1 6e 9f 00 00 80 11 ae a5 0a 05 04 7e 0a 05 ...n.....~...
0020 04 80 04 aa f9 4a 00 8d 39 72 34 d1 c8 48 b9 47 .....J... 9r4..H.G
0030 1c d7 74 d1 67 ee 86 82 35 5e d5 6b 1e ba 02 e3 ..t.g... 5^..k....
0040 59 e1 b8 7b 01 dc 6b 79 de 49 68 37 a8 3d 98 46 Y..{..ky .Ih7.=.F
0050 3c 34 fe 32 d0 f7 c2 e6 5e f6 14 6b 6a ff 8e 8f <4.2..... ^..kj...
0060 e3 87 13 99 3a 37 3d 5f dc bb c5 07 b5 58 ff d7 .....:7=_ .....X..
0070 1e df d5 d4 8c 95 96 83 ae 83 8d b4 9f ab 09 01 .....
0080 94 b7 48 2f ee 5a 1d 79 20 71 c8 bf 87 8f 5f 1a ..H/.Z.y q....._
0090 d6 9f d2 c4 5a 3f 4d 59 18 c7 14 d0 d1 18 47 78 ....Z?MY .....Gx
00a0 38 b8 d0 0b 45 7d 7d 08 b9 9e 68 4c 4f 46 b7 8...E)}. ..hLOF.
```



```

xterm
64 bytes from 192.192.192.1: icmp_seq=395 ttl=128 time=2.19 ms
64 bytes from 192.192.192.1: icmp_seq=396 ttl=128 time=2.52 ms
64 bytes from 192.192.192.1: icmp_seq=397 ttl=128 time=25.8 ms
64 bytes from 192.192.192.1: icmp_seq=398 ttl=128 time=2.13 ms
64 bytes from 192.192.192.1: icmp_seq=399 ttl=128 time=2.63 ms
64 bytes from 192.192.192.1: icmp_seq=400 ttl=128 time=2.22 ms
64 bytes from 192.192.192.1: icmp_seq=401 ttl=128 time=2.16 ms
64 bytes from 192.192.192.1: icmp_seq=402 ttl=128 time=1.98 ms
64 bytes from 192.192.192.1: icmp_seq=403 ttl=128 time=2.01 ms
64 bytes from 192.192.192.1: icmp_seq=404 ttl=128 time=2.05 ms
64 bytes from 192.192.192.1: icmp_seq=405 ttl=128 time=2.21 ms
64 bytes from 192.192.192.1: icmp_seq=406 ttl=128 time=1.58 ms
64 bytes from 192.192.192.1: icmp_seq=407 ttl=128 time=1.65 ms
64 bytes from 192.192.192.1: icmp_seq=408 ttl=128 time=2.20 ms
64 bytes from 192.192.192.1: icmp_seq=409 ttl=128 time=2.55 ms
64 bytes from 192.192.192.1: icmp_seq=410 ttl=128 time=2.13 ms
64 bytes from 192.192.192.1: icmp_seq=411 ttl=128 time=2.11 ms
64 bytes from 192.192.192.1: icmp_seq=412 ttl=128 time=2.54 ms
64 bytes from 192.192.192.1: icmp_seq=413 ttl=128 time=5.04 ms
64 bytes from 192.192.192.1: icmp_seq=414 ttl=128 time=5.99 ms
64 bytes from 192.192.192.1: icmp_seq=415 ttl=128 time=2.11 ms
64 bytes from 192.192.192.1: icmp_seq=416 ttl=128 time=2.02 ms
64 bytes from 192.192.192.1: icmp_seq=417 ttl=128 time=1.61 ms
64 bytes from 192.192.192.1: icmp_seq=418 ttl=128 time=2.02 ms
64 bytes from 192.192.192.1: icmp_seq=419 ttl=128 time=1.59 ms
64 bytes from 192.192.192.1: icmp_seq=420 ttl=128 time=2.34 ms
64 bytes from 192.192.192.1: icmp_seq=421 ttl=128 time=1.98 ms
64 bytes from 192.192.192.1: icmp_seq=422 ttl=128 time=2.00 ms
64 bytes from 192.192.192.1: icmp_seq=423 ttl=128 time=2.10 ms
64 bytes from 192.192.192.1: icmp_seq=424 ttl=128 time=2.09 ms
64 bytes from 192.192.192.1: icmp_seq=425 ttl=128 time=2.23 ms
64 bytes from 192.192.192.1: icmp_seq=426 ttl=128 time=3.06 ms
64 bytes from 192.192.192.1: icmp_seq=427 ttl=128 time=2.00 ms
64 bytes from 192.192.192.1: icmp_seq=428 ttl=128 time=1.60 ms
64 bytes from 192.192.192.1: icmp_seq=429 ttl=128 time=2.34 ms
64 bytes from 192.192.192.1: icmp_seq=430 ttl=128 time=2.44 ms
64 bytes from 192.192.192.1: icmp_seq=431 ttl=128 time=2.04 ms
64 bytes from 192.192.192.1: icmp_seq=432 ttl=128 time=1.99 ms

```

Figura 12. Ping entre placas VPN, TTLs, y tiempos

- User poco
- Group poco
- Persist-key
- Persist-tun
- Ca ca.crt
- # ficheros con claves
- Cert client1.crt
- Key client1.key
- Tls-auth key1 1
- Comp-lzo
- # nivel de info que enviará a pantalla
- Verb 4

Asigno permisos a ese fichero:

```
Chmod 700 cliente1.sh
```

Copiar del PenDrive a `/etc/openvpn`: CA.CRT, CLIENT1.CRT, CLIENT1.KEY, KEY1

ClienteUNIX: en la carpeta `Cd/user/local/etc/openvpn` creo un fichero llamado `vi cliente2.sh` que contiene:

```
# kldload bridge
# kldload if_tap
# openvpn -config cliente2.ovpn
```

El fichero `cliente2.ovpn` contiene:

- Client
- Dev tap0
- Proto udp
- # ip server puerto server
- Remote 10.5.4.126 1194
- Resolv-retry infinit
- Nobind
- # user y group para asegurar el producto
- User poco
- Group poco

- Persist-key
- Persist-tun
- # ficheros con claves
- Ca ca.crt
- Cert client2.crt
- Key client2.key
- Tls-auth key1 1
- Comp-lzo
- # nivel de info que enviará a pantalla
- Verb 4

Asigno permisos a ese fichero:

```
Chmod 700 cliente2.sh
```

Copiar del PenDrive a `/usr/local/etc/openvpn`: CA.CRT, CLIENT2.CRT, CLIENT2.KEY, KEY1

## Paso 5

UP & Running

## Servidor Windows

Doble Clic al batch `Server.bat` (abre una ventana de monitorización, en caso de ser servicio solo guardará info en un fichero de logs). Ver Figura 7.

## Cliente Linux

Ver Figura 8 y 9.

```
# Cliente1.sh
```

## Cliente Unix

Ver Figuras 10.

```
# cliente2.sh
```

Verificación del funcionamiento: activar un sniffer (por ejemplo `ethereal`) y capturar tráfico entre los nodos. En el `server Windows`:

- ping 192.192.192.1 -t
- ping 192.192.192.10 -t
- ping 192.192.192.11 -t

Desde los nodos:

- ping 192.192.192.1
- ping 192.192.192.10
- ping 192.192.192.11

Ver Figuras 11 y 12. Ver el fichero `OpenVpn.log` (win) donde mostrará

información del arranque de la parte servidor. Se puede leer:

```
.....
Thu Jun 01 15:16:28 2006 us=
105511
Diffie-Hellman initialized
with 2048 bit key
.....
Thu Jun 01 15:16:38 2006 us=
119418
Initialization Sequence Completed
```

Desde un cliente, entrar a un navegador :

```
http://192.192.192.1
```

(suponiendo que hay un web server en el servidor central)

Ver los paquetes capturados y el contenido. Paquete ping sin encriptar se encuentra en el Listado 2. El mismo paquete ya encriptado podemos ver en el Listado 3. Un paquete Reply también encriptado se encuentra en el Listado 4.

Se puede cambiar la longitud de las claves. En el caso de road warriors, accediendo a un servidor con IP Fija, se debe cambiar la configuración de clientes.ovpn, Indicando el nombre del servidor. Donde dice:

```
Remote 10.5.4.126 1194
```

Poner:

```
Remote server.empresa.gov 1194
```

(verificar que ese nombre sea resuelto). Paquete ARP Broadcast, para buscar que MAC es 192.192.192.11:

```
0000 ff ff ff ff ff ff 00 ff 92 2b
6c 52 08 06 00 01 ..... +1R....
0010 08 00 06 04 00 01 00 ff 92
2b 6c 52 c0 c0 c0 01 ..... +1R....
0020 00 00 00 00 00 00 c0 c0
c0 0b ..... ..
```

Paquete ARP Informando MAC del 192.192.192.11:

```
0000 ff ff ff ff ff ff 00 ff 92 2b
6c 52 08 06 00 01 ..... +1R....
0010 08 00 06 04 00 01 00 ff
```

### En la Red

- [www.openvpn.net](http://www.openvpn.net) - Sitio oficial
- <http://openvpn.net/security.html> - Security Tips
- <http://www.ethereal.com/> - Ethereal Pagina oficial
- [www.debian.org](http://www.debian.org) - Linux
- [www.freebsd.org](http://www.freebsd.org) - Unix
- [www.microsoft.com](http://www.microsoft.com) - Windows

### Sobre el Autor

Gosub (anonymous) Argentino, Italiano y Español. Informático de profesión, en 1983 y con 16 años comenzó su vida digital con una TI99/4 a. Estudió Técnico Informático, Ingeniería en Sistemas y un Master en Sistemas Informáticos (at&t – USA). Trabaja para Gobiernos, Multinacionales y Grandes empresas. Se desempeña en Tecnología desde 1993 hasta la fecha, oficialmente fue programador, analista, Leader de Proyectos y DBA (Db2, SQL y Oracle simultáneamente) los últimos 12 años. Extra-oficialmente, investigador, usuario de Linux y FreeBSD, trabaja para prensa y Consultoría en Servidores y Seguridad. Desde el 2003 vive en España pero es más fácil encontrarlo en internet. Contacto con el autor: [hakin9@hakin9.org](mailto:hakin9@hakin9.org)

```
92 2b 6c 52 c0 c0 c0 01 ..... +1R....
0020 00 00 00 00 00 00 c0 c0
c0 0b ..... ..
```

### Opciones

Se puede cambiar el puerto 1194 por otro diferente. (ej: 11194) para intentar ocultar un poco el servicio. (Personalmente, prefiero que el servidor OpenVPN sea un FreeBSD).

Hay que recordar el ajustar los filtros de nuestros firewalls. Si la conexión de corta por algun motivo, OpenVpn reintentará y vuelve a establecerla. Hacer copias de las claves, res-

guardarlas y de ser posible la carpeta de KEYS que sea READ/ONLY. En los tres sistemas operativos, no tuve que instalar ningun driver de placa, ninguna placa virtual.

Como se usa una clave para cada nodo, el tráfico para uno solo puede ser visto por ese, cualquier otro nodo, aunque sea válido y cliente del mismo server, no puede decodificar el paquete. Aunque la instalación del producto instala una placa virtual (windows), se pueden generar otras. A partir de aquí, a jugar un poco con esas comunicaciones. ●

### Tips

- UNIX TIP:  
Si queremos que funcione como servicio, podemos copiar el cliente2.ovpn como `openvpn.conf (/usr/local/etc/openvpn)`
- LINUX TIP  
Si queremos que funcione como servicio, podemos copiar el cliente1.ovpn como `/etc/openvpn/openvpn.conf`
- WINDOWS TIP  
Al instalar podemos indicar que queremos que sea un servicio y luego ponerlo el AUTOMATIC.
- TIP EN PRUEBAS

Mientras realizamos las pruebas, se pueden lanzar a mano los batches, con el modo VERB 4 hay abundante información, luego con VERB 1 es suficiente.