CREAZIONE DI UNA VPN UTILIZZANDO OPENVPN, TRA SISTEMI LINUX E WINDOWS.

Il presente scritto NON e' esaustivo. In rete esistono ottime guide su openvpn. Saro' qui molto breve, vedremo nel dettaglio come installare openvpn su linux slackware e su windows e di conseguenza come creare un tunnel sicuro che unisce due punti, attraversando una rete insicura, tipicamente internet. Inoltre vedremo come configurare un server vpn affinche' accetti piu' di una connessione vpn. Il software ce lo procuriamo dal sito di openvpn:

http://openvpn.net/download.html

ci occorre:

1) openvpn
2) le librerie lzo (per la compressione dei dati)
3) le bridge utility

I download si eseguono dai seguenti links:

1)openvpn = http://openvpn.net/download.html
2)lzo = http://www.oberhumer.com/opensource/lzo/download/LZO-v1/
lzo-1.08.tar.gz
3)bridge utility = http://sourceforge.net/project/
showfiles.php?group_id=26089

Per windows e' molto comoda anche la semplice gui reperibili all'url

http://openvpn.se/download.html

1) INSTALLAZIONE SU LINUX SLACKWARE

Formattiamo un pc ed installiamo l'ottima ed insuperabile slackware.
Nel presente testo faro' riferimento a " slackware " intendendo indistintamente slackware 10.2 e 10.1 . Posizioniamo il software in /root/ [i puristi della sicurezza non storcano il naso.] e ci accingiamo alla compilazione. Ovviamente dobbiamo assumere l'identita' di root.
cominciamo con lzo:

```
cd /root/
tar zxvf lzo-1.08
cd  lzo-1.08
./configure
./mate test
./make test
./make install
```

Ci dedichiamo ora ad openvpn

```
cd /root/
tar zxvf openvpn-2.1_beta14.tar.gz
cd openvpn-2.1_beta14
./configure
make
make install
```

ci dedichiamo ora alle bridge utils

```
cd /root/
tar zxvf bridge-utils-1.1.tar.gz
cd bridge-utils-1.1
```

```
./configure
make
make install
```

Ora l'eseguibile openvpn si trova installato nel seguente path:

```
/usr/local/sbin/openvpn
```

L'installazione NON crea la directory contente i files di configurazione. Occorre provvedere a mano.

```
mkdir /etc/openvpn
```

2) INSTALLAZIONE SU WINDOWS

L'installazione di software per la piattaforma windows e' sempre molto semplice. Occorrono 3 software :

1) openvpn-2.1_beta14-install.exe
2) openvpn-2.0.7-gui-1.0.3-install.exe
3) openvpn-gui-1.0.3-it.exe

Eseguiamo un doppio clic su openvpn-2.1_beta14-install.exe ed accettiamo tutto quello che ci viene proposto. Eseguiamo un doppio clic su openvpn-2.0.7-gui-1.0.3-install.exe ed accettiamo tutto quello che ci viene proposto. Adesso ci rechiamo nella directory

```
c:\programmi\openvpn\bin
```

e cancelliamo il file

```
openvpn-gui.exe.
```

Al suo posto copiamo il file

```
openvpn-gui-1.0.3-it.exe
```

Ovviamente dobbiamo rinominarlo in openvpn-gui.exe .
A questo punto abbiamo tutto il software installato. Possiamo utilizzarlo sia in modalita' server sia in modalita' client. I files di configurazione saranno locati in

```
c:\programmi\openvpn\config\
```

Inoltre un nuovo servizio sara' presente in

```
impostazioni --> pannello di controllo --> strumenti di
                amministrazione --> servizi -- > openvpn service.
```

3) CONFIGURAZIONE SERVER VPN + CLIENT VPN PER COLLEGAMENTO PUNTO A PUNTO

In questo esempio lavoriamo in rete locale, con due elaboratori correttamente formattati, installati e la rete perfettamente funzionante. I due pc si chiamano [con molta fantasia]

```
server
client
```

Ci rechiamo sul pc " server ", generiamo la chiave occorrente col comando

```
openvpn --genkey --secret chiave.txt
```

La sintassi e' valida sia per linux che per windows. Posizioniamo la chiave nella direcotry contenente i files di configurazione.
Adesso creiamo il file di configurazione vero e proprio, che chiameremo " server.conf " Attenzione: Windows richiede obbligatoriamente che i files di configurazione abbiano l'estensione finale " .ovpn ". Quindi il nome giusto utilizzando windows diventera' " server.ovpn "
Un buon esempio potrebbe essere il seguente:

```
dev tap
secret chiave.txt # correggere il path a seconda del SO
ping 10
verb 1
mute 10
ifconfig 10.0.1.1 255.255.255.252
lport 5002
```

Assegniamo cosi' al server un ulteriore indirizzo ip oltre a quello gia' assegnato, e precisamente 10.0.1.1 .

Adesso startiamo il servizio con la sintassi adeguata al SO utilizzato. Per linux occorre in comando

```
openvpn --config /etc/openvpn/server.conf --daemon
```

Invece con windows occorre recarsi nel pannello di controllo, --> strumenti di amministrazione --> servizi -- > openvpn service ed avviare il servizio.

Ci rechiamo ora sul client. Copiamo la chiave " chiave.txt " nella directory contente i files di configurazione , e creiamo il files di configurazione vero e proprio il cui nome sara' " client.con " per linux e " client.ovpn " per windows. Un esempio di tale file potrebbe essere il seguente:

```
remote    ---> INSERIRE L'IDIRIZO IP DEL SERVER
rport 5002
dev tap
ifconfig 10.0.1.2 255.255.255.252
secret  chiave.txt   # correggere il path a seconda del SO
ping 10
verb 1
mute 10
```

Accertiamoci di avere assegnato al client un indirizzo ip differente rispetto al server. Ora occorre avviare openvpn. (Sul lato server openvpn e' gia' in esecuaione ed attende connessioni.) . La sintassi e' la seguente utilizzando linux

```
openvpn --config /etc/openvpn/server.conf --daemon
```

Mentre per windows occorre cliccare sulla piccola icona presente in basso a destra, vicino all'orologio , e scegliere la voce " connetti ".

Abbiamo finito.

4) CONFIGURAZIONE SERVER LINUX PER COLLEGAMENTO CON MOLTI CLIENT

Installiamo da sorgenti openvpn come al punto 1) .
Creiamo la direcotory contenente i files di configurazione

```
mkdir /etc/openvpn
```

Per pura comodita' copiamo all'interno della direcotry appenza creata tutti i files che ci occorrono.

```
cp -r openvpn-2.1_beta14/easy-rsa /etc/openvpn/
```

ci rechiamo nella directory contenente gli script necessari per creare le chiavi

```
cd /etc/openvpn/easy-rsa/2.0/
```

editiamo il file " vars " . Le ultime 5 righe devono essere personalizzate. Ecco la versione originale di tali righe :

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
```

adesso dobbiamo lanciare diversi comandi, come da elenco:

```
./vars
./clean-all   (potrebbe essere necessario indicare  espressamente la
 sorgente con  source ./vars  . Il sistema ce lo chiede chiaramente se
 occorre )
openvpn --genkey --secret ta.key
./build-ca  Lo script e' interattivo.
./build-key-server   server .  Lo script e' interattivo , e genera
  la chiave lato server .
./build-key client   . Lo script e' interattivo e genera la chiave
 lato client
./build-dh   . L'esecuzione potrebbe durare qualche attimo.
```

A questo punto all'interno della directory /etc/openvpn/ease-rsa/2.0/keys abbiamo quasi tutti i files che ci occorrono per configurare il nostro server ed un client.
Copiamo tutti questi files appena citati all'interno della directory /etc/openvpn

```
cp openvpn-2.1beta14/easy-rsa/2.0/keys/* /etc/openvpn
```

Il file di configurazione vero e proprio lo prendiamo dalla directory ove sono presenti gli esempi:

```
cp /openvpn/sample-config-files/server.conf /etc/openvpn
```

Editiamo quest'ultimo file inserendo i path corretti dipendentemente dal SO che stiamo utilizzando e salviamo. Utilizzando linux avviamo openvpn con la sintassi

```
openvpn --config /etc/openvpn/server.con --daemon
```

Dal lato client occorre:
1) installare openvpn
2) creare la directory coi files di configurazione
3) copiare le chiavi generate sul server
4) avviare openvpn .

Il sito di openvpn ci propone uno specchieto riepilogativo dei files-chiave da utilizzare; all'url http://openvpn.net/howto.html#pki
Riportiamo tale schema

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

Lo schema considera solamente un server ed un client. Per aggiungere un altro client dobbiamo ripetere la sezione relativa alla generazione della chiave " client " ed aggiungere tali nuovi files nelle direcotry di contententi i files di configurazione, sia sul server che sul nuovo client.

5) DIRETTIVE POSSIBILI ALL'INTERNO DEI FILES DI CONFIGRUAZIONE
l'elenco NON e' esaustivo

client = specifica il ruolo dell'host. In questo caso ? client ?

dev tap
dev tun = queste opzioni sono mutuamente esclusive. Indicano l'utilizzo dell'interfaccia tap oppure dell'interfaccia tun.

dev-node MyTap = questa direttiva e' valida solo per windows. In pratica con tale SO occorre indicare espressamente il nome dell'interfaccia, come indicato nel file openvpn-2.1_beta14/sample-config-files/client.conf

proto tcp
proto udp = queste opzioni sono mutuamente esclusive ed indicaso se utilizzare tcp oppure udp. L'impostazione predefinita indica di utilizzare udp.

remote my-server-1 1194 = indica il server remoto a cui connettersi. Viene accettato un indirizzo ip valido oppure un nome risolvibile tramite dns. E' possibile specificare

piu' di un server remoto in modo da realizzare un
? load balancing ?, come espressamente indicato nel
file
openvpn-2.1_beta14/sample-config-files/client.conf
ove vi e' anche un esempio sull'uso corretto di tale
direttiva

remote-random = Si riferisce sempre al ? load balancing ? . qualora la lista dei server remoti sia a sua volta in remoto, ecco che il server cui openvpn si colleghera' sara' scelto in modo casuale. In pratica la direttiva indica un pc ove e' presente una lista di server vpn.

resolv-retry infinite = indica ad openvpn di risolvere in maniera continuativa il server remoto. Cio ' puo' essere utile utilizzando un portatile

nobind = indica espressamente di non utilizzare bind

user nobody
group nobody = impostando tali direttive (valide solo per linux e *nix) ecco che la vpn viene attivata assegnandole i privilegi dell'utente nobody e del gruppo nobody. Cio' aumenta ulteriormente la sicurezza

persist-key
persist-tun = indicando tali direttive ecco che in caso di riavvio del sistema openvpn cerca di preservare lo stesso medesimo stato esistente prima dell'interruzione

http-proxy-retry
http-proxy [proxy server] [proxy port #] = indicare queste direttive se l'accesso alla rete esterna avviene obbligatoriamente tramite un proxy

mute-replay-warnings = in caso di reti wireless e qualora si sospetti l'abbondanza di pacchetti duplicati , occorre impostare questa direttiva

ca ca.crt
cert client.crt
key client.key = indicano la posizione dei certificati occorrenti al funzionamento di openvpn. Per creare tali certificati occorre eseguire gli script presenti in openvpn-2.1_beta14/easy-rsa/ Qui troviamo ulteriori directory in base al SO da noi utilizzato. E' preferibile indicare il path di tali certificati in maniera estesa ; esempio

ca /etc/openvpn/config/key/ca.crt

I files di configurazione di windows richiedono il doppio back-slah , esempio:

ca c:\\programmi\\openvpn\\config\\ca.crt

ns-cert-type server = riporto in inglese poiche' non sono certo dell'esatta
traduzione:
To use this feature, you will need to generate
your server certificates with the nsCertType
field set to "server". The build-key-server
script in the easy-rsa folder will do this.

tls-auth ta.key 1 = qualora il server utilizzi la tsl-auth key ecco che
dobbiamo utilizzare qst direttiva. I client
imposteranno a ? 1 ? , i server imposteranno a ? 0 ?

cipher x = imposta il livello cdi crittografia. I valori possibili di ? x ?
sono :
cipher BF-CBC # Blowfish
cipher AES-128-CBC # AES
cipher DES-EDE3-CBC # Triple-DES
ovviamente occorre indicare lo stesso livello di crittografia
sul server e sui clients

comp-lzo = indicare di utilizzare la compressione

verb 3 = livello di verbosita'. I valori possibili sono da 0 a 9 , dove 0
indica una verbosita' nulla e 9 indica la verbosita' massima

mute 20 = indica il numero massimo di messaggi uguali da inviare al log.
In questo caso, qualora avessimo 40 messaggi uguali tra loro,
solamente in primi 20 verranno inviati al log.

local a.b.c.d = questa direttiva si utilizza nel file di configurazione del
server vpn ed indica su quale indirizzo ip il demone vpn
deve restare in ascolto. Tale direttiva e' opzionale ed in
sua assenza il demone vpn rimarra' in ascolto su tutte le
interfacce

port 1194 = indica su quale porta il demone vpn deve rimanere in ascolto.
Dal file openvpn-2.1_beta14/sample-config-files/tls-home.conf :

```
# OpenVPN 2.0 uses UDP port 1194 by default
# (official port assignment by iana.org 11/04).
# OpenVPN 1.x uses UDP port 5000 by default.
# Each OpenVPN tunnel must use
# a different port number.
# lport or rport can be used
# to denote different ports
# for local and remote.
; port 1194
```

dh dh1024.pem = riporto dall'inglese : # Diffie hellman parameters.
Generate your own with:
openssl dhparam -out dh1024.pem 1024
Substitute 2048 for 1024 if you are using
2048 bit keys.

server 10.8.0.0 255.255.255.0 = configura l'host per l'utilizzo server,

assegnandogli espressamente l'indirizzo ip
indicato. Gli altri ip utili (vista la metmask)
saranno disponibili per i client. Questa
direttiva non dove essere utilizza in caso di
utilizzo del " bridging ethernet " . la pagina
man di openvpn e' molto esaustiva al riguardo

ifconfig-pool-persist ipp.txt = nel file ipp.txt ? che openvpn crea da solo ?
vengono memorizzate le associazioni
client < -- > indirizzo ip. In caso di riavvio
dei sistemi ecco che openvpn cerchera' di
connettere gli host remoti in base a quanto
indicato in tale file

server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100 =
viene espressamente indicato a openvpn di utilizzare
la modalita' bridge. Sara' compito
dell'amministratore l'impostazione di tale bridge.
Una buona spiegazione la si trova nel file
openvpn-2.1_beta14/sample-config-files/server.conf

push "route 192.168.10.0 255.255.255.0"
push "route 192.168.20.0 255.255.255.0" = dall'inglese :

```
# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server
```

;client-config-dir ccd
;route 192.168.40.128 255.255.255.248 = dall'inglese :

```
# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.
```

push "redirect-gateway" = indica ai client di utilizzare il pc ove openvpn e'
in esecuzione in modalita' server , anche come
gateway.

push "dhcp-option DNS 10.8.0.1"
push "dhcp-option WINS 10.8.0.1" = indica espressamente di utilizzare il pc
ove openvpn e' in esecuzione in modalita'
server anche come dns e wins.

Maggiori informazioni su :
http://openvpn.net/faq.html#dhcpcaveats

duplicate-cn = questa direttiva consente a molti client di utilizzare le stesse chiavi. In pratica, generando una sola chiave e copiandola pari pari su molti client, ecco che essi potranno collegarsi ad un medesimo server openvpn. Il file di configurazione server.conf sconsiglia a chiare note questo comportamento

keepalive 10 120 = indicando questa direttiva diciamo ad openvpn di eseguire un ping ogni 10 secondi verso l'altro capo della vpn. Qualora per 120 secondi non si ottiene alcuna risposta, il demone vpn considera caduta la linea.

```
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
```

= indica il livello di crittografia utilizzato. Lo stesso livello di crittografia deve essere indicato sia sui client che sul server openvpn

max-clients 100 = indica quanti client possono ottenere accesso simultaneo ad un unico server openvpn

client-to-client = con questa direttiva ecco che i client possono vedersi tra di loro. Il comportamento predefinito e' esattamente l'opposto: i client vedono solo il server.

persist-key
persist-tun = dall'inglese :

```
# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade
```

```
log         openvpn.log
log-append  openvpn.log
```

= come comportamento predefini openvpn invia i log al servizio syslog. Nei sistemi windows quando openvpn e'in esecuzione come servizio, i log vengono indirizzati nella directory \Programmi\OpenVPN\log
Utilizzando l'opzione ? log ? otteniamo la cancellazione del file di log ad ogni riavvio del servizio. Utilizzando l'opzione ? log-append? otteniamo che il file di log non venga cancellato ad ogni riavvio del servizio. I log vengono cosi' accodati

verb 3 = indica il livello di verbosita'. 0 equivale a nessun log (tranne errori fatali 9 equivale al massimo dei messaggi d'informazione

mute 20 = indica il numero massimo di messaggi identici inviati al log. In pratica , utilizzando ? mute 20 ? , qualora vengano prodotti 40 righe di log uguali tra loro, solo le prime 20 verranno registrate, le altre

saranno scartate

6) AVVIO AUTOMATICO DI OPENVPN ALL'ACCENSIONE DEL SISTEMA.

Nei sistemi linux ove openvpn e' stato installato tramite pacchetto, uno script di avvio e' stato posizionato in /etc/init.d (oppure nella posizione esatta per la distro in oggetto) . Tale script controlla la presenza di un file .conf nella directory /etc/openvpn e se trova tale file avvia il servizio.
Con slacware possiamo utilizzare il file /etc/rc.d/rc.local
Tale file e' l'ultimo ad essere elaborato daghli script di avvio, e l'amministratore puo' inserirvi tutti i comandi opportuni. Essi saranno eseguiti, e successivamente si potra' procedere al login.
Nei sistemi windows invece viene aggiunto un nuovo servizio, accessibile da pannello di controllo, strumenti di amministrazione , servizi.
Tale servizio e' impostato a off in maniera predefinita. Impostandolo a on, ad ogni riavvio il sistema cerca un file con estenzione .ovpn nella direcotry c:\programmi\openvpn\config , e se lo trova lancia il servizio.
Qualora fossero presenti piu' di un file ecco che verranno lanciate tante istanze di openvp quanti sono i files di configurazione presenti.
DA: http://openvpn.net/howto.html#start

7) ALTRI ASPETTI

Sarebbe interessante analizzare l'utilizzo delle bridge utils , tramite le quali e' possibile accedere alla lan " dietro " al server. Al momento questo non mi occorre, quindi soprassiedo.
Tutti coloro i quali intendono aggiungere altre info al presente documento o rimediare ad inesattezze possono contattarmi all'indirizzo mail

syspkq@email.it

Veronese Claudio