# Implementare una VPN LAN-to-LAN con OpenVPN tra IpCop 1.4.21 e Zeroshell 2.0 RC3


Il sistema operativo multifunzionale
creato da Fulvio Ricciardi

www.zeroshell.net

(autore: Mario Fabiani – mfabi2003@libero.it )

## Introduzione

IpCop e Zeroshell sono distribuzioni Linux dedicate al routing-firewall particolarmente diffuse ed apprezzate per la loro versatilità ed efficienza, nonché per la possibilità di essere installate su dispositivi economici di tipo embedded con limitate risorse hardware.

Il caso reale su cui è basato questo HowTo prevede un collegamento point-to-point in VPN tra:

A) gateway internet (con indirizzo IP pubblico fisso) costituito da un modem-router ADSL collegato ad un dispositivo PCEngines Alix 2D3 con installato IpCop 1.4.21;

B) postazione remota costituita da un Alix 6F2, provvisto di modem 3g, sul quale è installato Zeroshell 2.0 RC3.

A monte di entrambi i router si trovano una serie di PC e/o dispositivi collegati in rete locale che devono poter comunicare tra loro.

La scelta di Zeroshell per il gateway della postazione remota deriva dalla possibilità di avere, all'interno di un unico dispositivo, le funzionalità di router 3g, firewall e client VPN. In più, dal momento che la postazione è in luogo non presidiato e non facilmente accessibile, l'opzione di net balancing - failover risulta fondamentale permettendo, in presenza di un collegamento in ponte radio già esistente, il backup su rete 3g in caso di guasto del link principale.

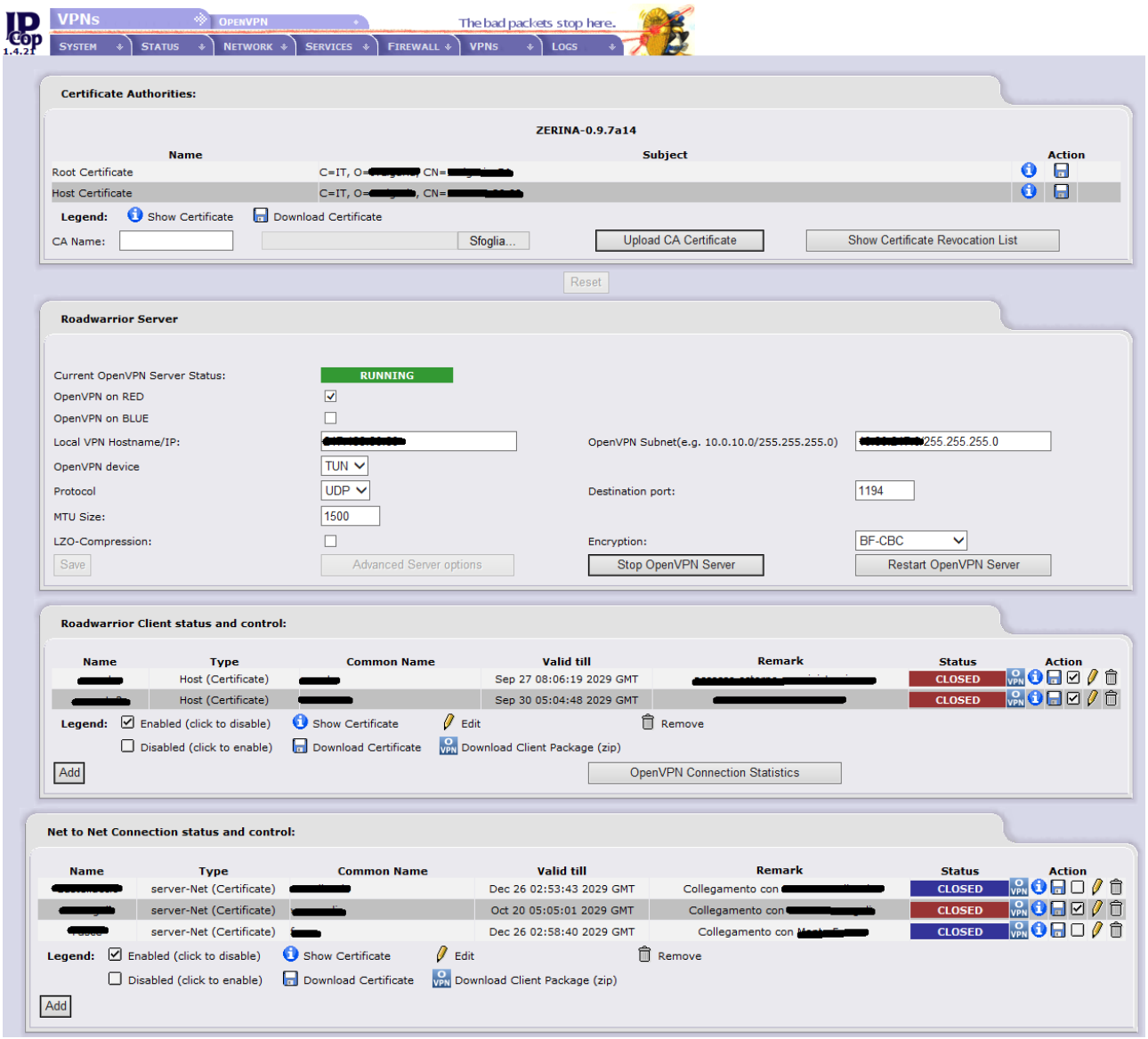
La scelta più ovvia e facile sarebbe stata quella di implementare Zeroshell da entrambi i lati del link. Nel caso specifico però, per ragioni che non mi dilungo a spiegare, è stato deciso di mantenere il gateway IpCop. Si trattava quindi di optare per uno standard VPN riconosciuto da entrambe le distribuzioni e costruire il collegamento operando le necessarie configurazioni, per quanto possibile attraverso le rispettive Web GUI. Non essendo riuscito a trovare in rete alcun tutorial sull'argomento, mi sono deciso a scrivere queste note, sperando che possano essere utili a chi si trovi a dover affrontare situazioni simili. Naturalmente, benché la soluzione proposta sia perfettamente funzionante e in opera ormai da mesi senza particolari problemi, rimane suscettibile di miglioramenti e/o configurazioni alternative.

Zeroshell implementa le VPN LAN-to-LAN utilizzando OpenVPN, standard aperto di grande diffusione, che dà le necessarie garanzie in fatto di sicurezza, interoperabilità e facilità di configurazione. IpCop, d'altra parte, non implementa OpenVPN in modo nativo, ma solo tramite il plug-in Zerina, che per di più permette di configurare le VPN LAN-to-LAN solo nella versione 0.9.7a14, ultima release (alpha) recuperabile in rete. Questa funzionalità, tra l'altro, non è disponibile neanche nella più recente versione 2.0 di IpCop, che si limita ad inglobare OpenVPN in versione Host-to-LAN (Roadwarrior).

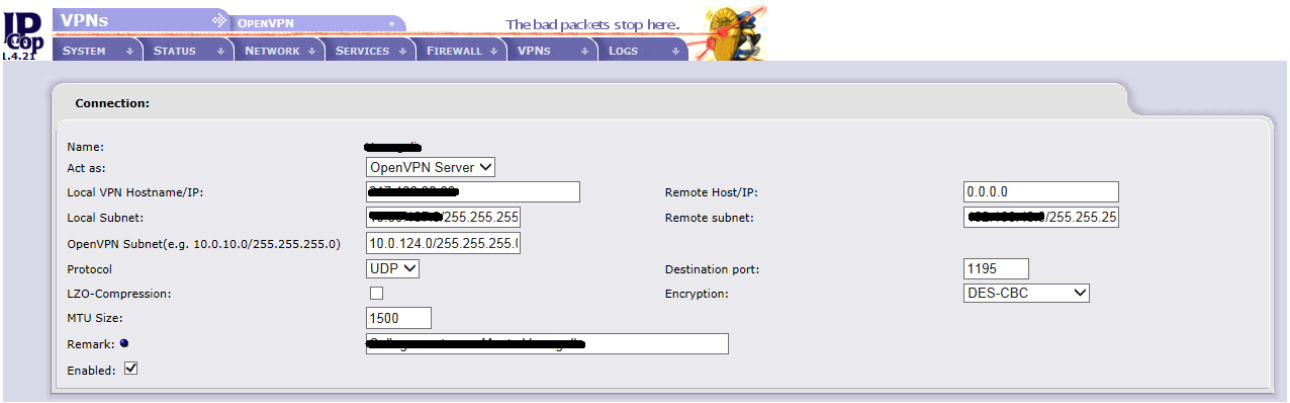
## Configurazione di IpCop

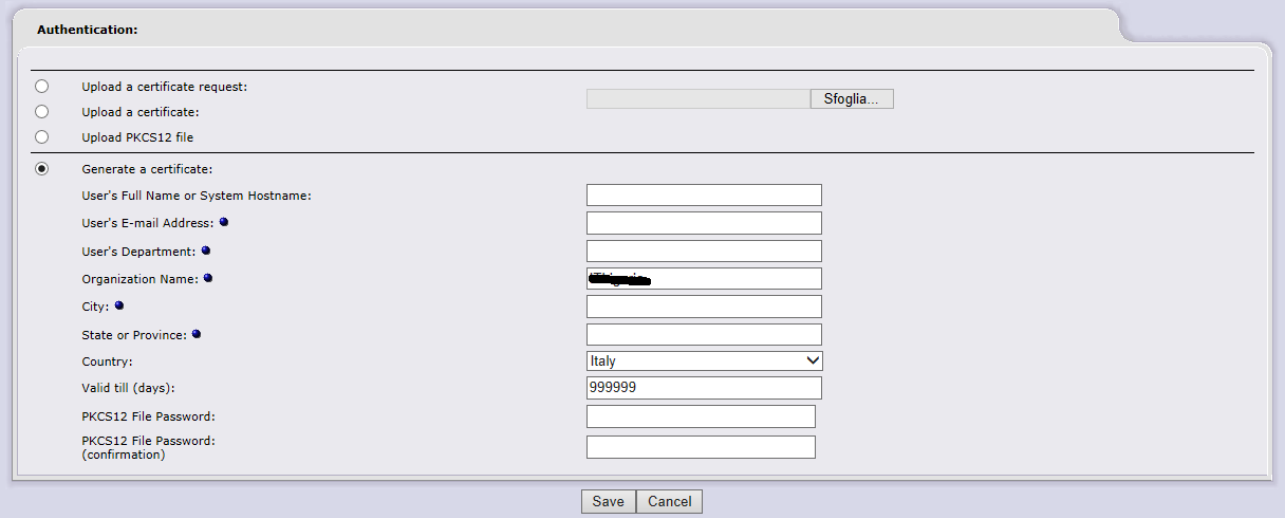
Per quanto riguarda l'installazione e la configurazione del plug-in Zerina su IpCop 1.4, rimando all'apposito tutorial in rete. Se si installa su una versione maggiore della 1.4.18, occorre operare una piccola modifica sull'installer secondo quanto descritto qui. Non sono però riuscito a recuperare nessun HowTo circa l'utilizzo della feature Net-to-Net, altra ragione per la quale mi sono deciso a scrivere queste note.

Al termine dell'installazione e configurazione di base, avremo nel menu *VPNs* dell'interfaccia web di IpCop una voce *OpenVPN* ,che dà accesso a una pagina come la seguente (qui sono presenti connessioni e configurazioni già create, che ovviamente in fase di prima installazione non apparirebbero):




Tralasciando la configurazione del *Roadwarrior server* e dei conseguenti client, la sezione che ci interessa è quella più in basso, ovvero *Net-to-Net connection status and control*, che permette appunto di creare una connessione tra due reti. Cliccando su *Add*, dopo aver selezionato *Connection Type: Net-to-Net Virtual Private Network* apparirà una finestra di configurazione come la seguente:






Vediamo le diverse voci disponibili.

**Sezione *Connection*:**

*Name*: possiamo scegliere quello che più ci aggrada.

*Act as*: è possibile scegliere se il dispositivo locale agirà da server o da client. In questo caso, essendo il nostro il gateway principale, lasceremo selezionato *OpenVPN Server*.

*Local VPN Hostname/IP*: qui va indicato l'indirizzo esterno (pubblico) del gateway. Se il nostro provider ci ha fornito un indirizzo fisso, va scritto qui. Altrimenti è possibile (ma non testato) indicare un hostname fornito da un qualsiasi provider DDNS.

*Remote Host/IP*: stesso discorso della voce precedente ma per il gateway remoto. Nel nostro caso, visto che si tratta di una connessione 3g con indirizzo dinamico, possiamo indicare semplicemente 0.0.0.0, che istruisce il server ad accettare connessioni da qualsiasi indirizzo IP.

*Local subnet/remote subnet*: Qui possiamo indicare le sottoreti nelle quali risiedono i dispositivi a monte dei due gateway. Questo serve a OpenVPN per configurare le route statiche necessarie a far comunicare le due reti.

*OpenVPN subnet*: il tipo di device VPN che Zerina-IpCop propone per il collegamento LAN-to-LAN è il TUN, che prevede l'assegnazione di una sottorete IP "interna" al link VPN (su indirizzi privati) che viene proposta in automatico, ma può essere naturalmente modificata. In questo caso avremo 10.0.124.1 per l'endpoint locale e 10.0.124.2 per l'endpoint remoto.

*Protocol/Destination Port/LZO-Compression/Encryption/MTU Size*: parametri di OpenVPN di cui si dovrà tenere conto in sede di configurazione del client: nel mio caso ho selezionato protocollo UDP, porta 1195 (anziché 1194 già impegnata con il server Roadwarrior), nessuna compressione, crittografia con algoritmo DES-CBC (per alleggerire al massimo il carico CPU), MTU 1500.

*Remark*: eventuale commento descrittivo

Consiglio di non editare i parametri *Advanced*, in quanto l'interfaccia di configurazione tramite GUI non sembra faccia il suo dovere, probabilmente a causa della versione immatura del plug-in. In ogni caso si tratta di setting che possono essere lasciati nella loro configurazione di default.

**Sezione *Authentication*:**

Zerina implementa l'autenticazione esclusivamente tramite certificati X509, quindi è necessario caricare o creare un certificato da associare al client appena configurato. Ciò è possibile solo se preventivamente è stata abilitata una Certification Authority e sono presenti i relativi certificati Root e Host nella pagina principale di OpenVpn. Nel nostro caso, trattandosi di un collegamento VPN all'interno della stessa organizzazione, è stata creata una CA locale. Per

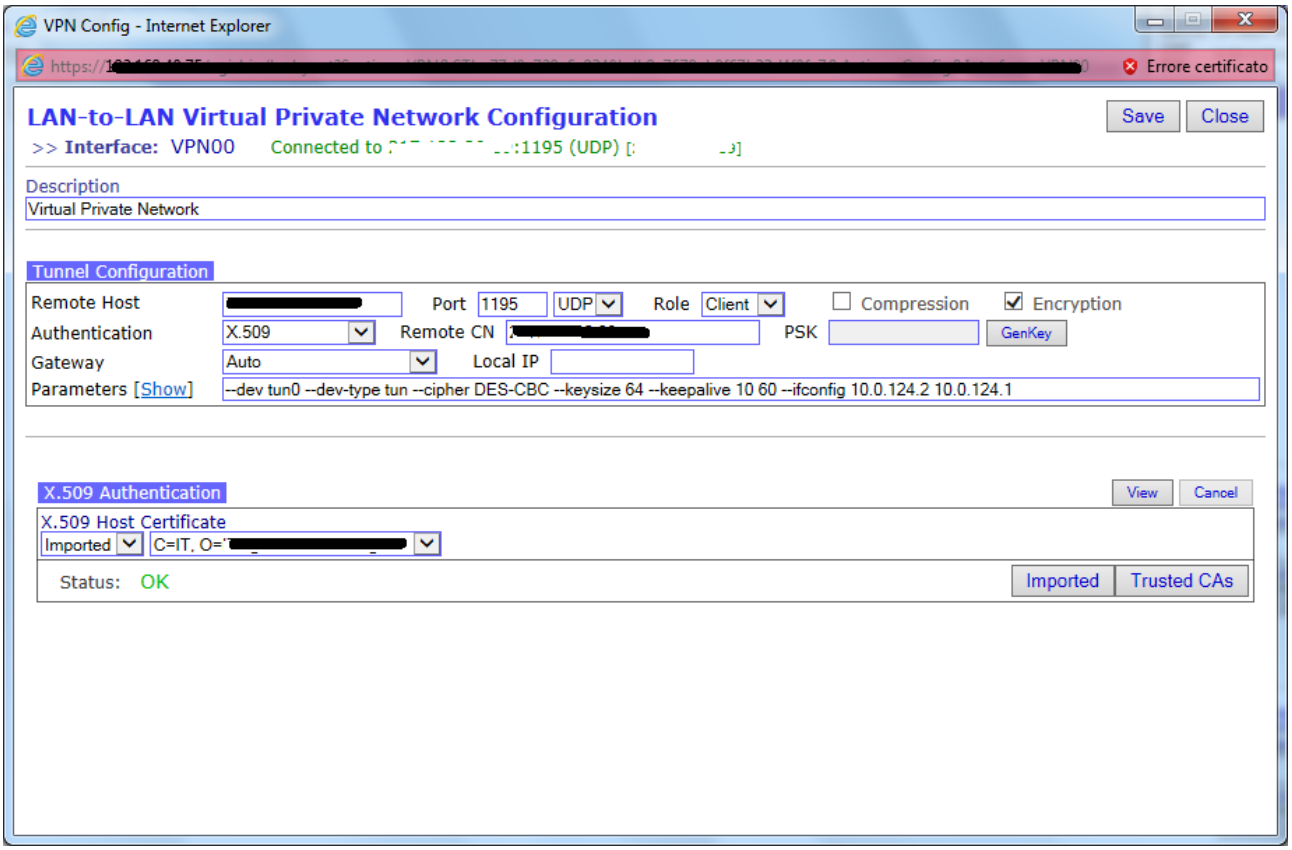
emettere un certificato è sufficiente popolare i campi relativi al nome, paese, validità e password. Infine fare clic su Save.

A questo punto avremo disponibile una connessione server Net-to-Net la cui descrizione sarà presente nell'elenco in fondo alla pagina principale *OpenVPN*.

Tramite l'apposita icona con il simbolo del floppy disk, presente in fondo alle voci dell'elenco, possiamo effettuare il download del certificato e della relativa chiave privata (in formato PKCS12 – estensione .p12) relativo al client appena creato. Allo stesso modo occorrerà scaricare il certificato (in formato PEM) relativo all'Autorità di certificazione creata in precedenza (*Root certificate*, in cima alla pagina *OpenVPN*)

## Configurazione di Zeroshell

La creazione di una VPN Lan-to-Lan OpenVPN in Zeroshell avviene cliccando sul pulsante *New VPN* nella pagina *Setup-Network*. Il box di configurazione, alla fine dell'impostazione, sarà il seguente:




Come si vede, la configurazione si effettua in parte impostando i setting proposti dalla Web GUI, in parte inserendo una serie di parametri aggiuntivi nel campo *Parameters*. Questo in quanto alcune delle impostazioni standard del client OpenVPN di Zeroshell non sono compatibili con quelle del server IpCop. Le modifiche indicate sono quelle che hanno permesso di ottenere una situazione funzionante e stabile, dopo una serie di tentativi di collegamento effettuati operando con diversi setting sia da un lato che dall'altro del link. La stringa dei parametri aggiuntivi è la seguente:

*--dev tun0 --dev-type tun --cipher DES-CBC --keysize 64 --keepalive 10 60 --ifconfig 10.0.124.2 10.0.124.1*

Vediamo nel dettaglio il significato dei singoli setting.

La differenza principale nella gestione OpenVPN LAN-to-LAN tra Zeroshell e IpCop sta nel fatto che quest'ultimo utilizza device di tipo TUN anziché TAP. Ne deriva che, di base, la VPN deve essere di tipo "routed" con tutti i pro e i

contro di una simile soluzione. Avendo sperimentato senza successo la modifica del device type su IpCop, alla fine la scelta è ricaduta sull'impostazione del dispositivo TUN su Zeroshell, cosa che è possibile semplicemente inserendo le direttive:

*–dev tun0 –dev-type tun*

**N.B.** la modifica del device in modo corretto, in realtà, comporrebbe la riscrittura degli script di gestione della VPN sostituendo le occorrenze di "TAP" con "TUN" e la conseguente creazione di uno startup script di avvio che carichi le impostazioni a ogni eventuale riavvio del gateway. Tuttavia la soluzione proposta ha dimostrato di funzionare egregiamente e ha il vantaggio, oltre che di essere molto più semplice, di applicarsi solo alla VPN specifica.

Viene poi impostato l'algoritmo di cifratura su DES-CBC con chiave a 64 bit:

*--cipher DES-CBC –keysize 64*

La scelta di un algoritmo così "leggero" è stata effettuata per diminuire il carico di CPU sui gateway date esigenze di sicurezza poco stringenti, ma nessuno ovviamente vieta di adottare soluzioni diverse o di mantenere l'algoritmo di default che per OpenVPN è il BF-CBC.

*--keepalive 10 60*

Il keep-alive è un meccanismo di verifica della connessione che permette di far ripartire il tunnel nel caso l'endpoint remoto non risponda a dei tentativi di ping effettuati a cadenza regolare. L'impostazione 10-60 (un ping ogni 10 secondi – timeout 60 secondi) è quella di default prevista da IpCop.

Infine viene settato il parametro

*--ifconfig 10.0.124.2 10.0.124.1*

che istruisce il client a stabilire un tunnel le cui estremità avranno come indirizzi interni quelli indicati, in sintonia con quanto configurato su IpCop ma a indirizzi invertiti (primo indirizzo = endpoint locale, secondo indirizzo = endpoint remoto).

Per quanto riguarda invece i parametri impostabili direttamente dalla GUI, essi dovranno rispecchiare necessariamente quanto impostato sul server IpCop, ovvero:

*Remote Host* e *Remote CN* : indirizzo IP pubblico del server OpenVPN.

*Port*: 1195-UDP

*Role*: Client

*Compression* : deselezionato

*Encryption*: selezionato

*Authentication*: X509

Il resto può essere lasciato come da default.

A questo punto, avendo optato per l'autenticazione con certificati X509, occorrerà nella sezione apposita *X509 Authentication* selezionare un certificato valido per la macchina locale. Tale certificato dovrà essere quello emesso per il client dalla CA a suo tempo configurata sul gateway IpCop. A questo scopo si deve istruire Zeroshell a riconoscere la suddetta CA tra le Authority "trusted", operazione possibile tramite la voce *X509 CA – Trusted CAs* del menu Web. Sarà sufficiente selezionare il file .PEM precedentemente esportato da IpCop nella casella *Import* e cliccare l'omonimo pulsante. La CA di IpCop apparirà quindi nella lista delle autorità "di fiducia".

Per il certificato da assegnare al nostro host client le cose sono un pochino più complicate, dato che IpCop produce per i client certificati di tipo PKCS#12 (che contengono anche la chiave privata), mentre Zeroshell è in grado di importare solo nel formato PEM: occorrerà quindi effettuare una conversione del formato dei certificati per renderli compatibili.

Per fare ciò possiamo utilizzare OpenSSL, toolkit per la gestione dei protocolli SSL e TLS che fornisce utili strumenti per la codifica e conversione dei certificati. OpenSSL è disponibile sia per Linux che come tool a linea di comando per Windows.

Avendo a disposizione i certificati in formato PKCS#12 (e supponendo che il certificato si chiami pippo.p12), la prima cosa da fare sarà sfruttare la libreria crittografica pkfs12 di OpenSSL per estrarre il certificato SSL e la chiave privata, tramite i comandi:

openssl pkcs12 -in pippo.p12 -out pippo.key -nocerts –nodes

openssl pkcs12 -in pippo.p12 -out pippo.pem -nokeys –clcerts

a questo punto, utilizzando la libreria rsa, occorrerà decodificare la chiave pippo.key con il comando:

openssl rsa -in pippo.key –out pippo_s.key

verrà richiesta ad un certo punto la password di protezione della chiave, impostata a suo tempo durante la generazione del certificato su IpCop.

Alla fine ci ritroveremo, insieme a pippo.p12, due file pippo.pem e pippo_s.key, nel formato giusto per essere importati all'interno di Zeroshell. Attraverso il menu *X509CA-Imported* selezioneremo i file rispettivamente nelle caselle *Certificate* e *Key* e cliccando su *Import* avremo il certificato del client nella lista dei certificati importati.

A questo punto, nella configurazione della VPN LAN-to-LAN, sezione *X509 Authentication*, selezioneremo *Imported* nel menu a tendina delle CA disponibili, e il certificato client appena importato nel successivo menu relativo ai certificati.

Una volta salvata la configurazione, la VPN è pronta a funzionare. Sarà sufficiente abilitare i device di rete da entrambi i lati del collegamento. Se otterremo la classica scritta in verde *Connected to…* nell'interfaccia di *Setup-Network* di Zeroshell, avremo raggiunto l'obiettivo di far comunicare tra loro i due gateway. La GUI di IpCop invece continuerà a segnalare la connessione come *CLOSED*, probabilmente a causa di un piccolo bug nell'interfaccia di Zerina. Un rapido sguardo al log di OpenVPN permetterà comunque di constatare l'avvenuta connessione.

Per far parlare le due reti a monte dei gateway occorrerà infine configurare opportunamente le route statiche per indirizzare sulla VPN le connessioni tra le relative sottoreti IP. Dal lato IpCop, se è stata indicata la sottorete locale nella configurazione del server OpenVPN, la cosa avverrà in automatico. Dal lato Zeroshell, si dovrà inserire manualmente la route nel menu *Router* – pulsante *Routing Table.* Nell'ipotesi che la sottorete a monte del gateway IpCop sia la 192.168.1.0/24, la route da aggiungere sarà la seguente: