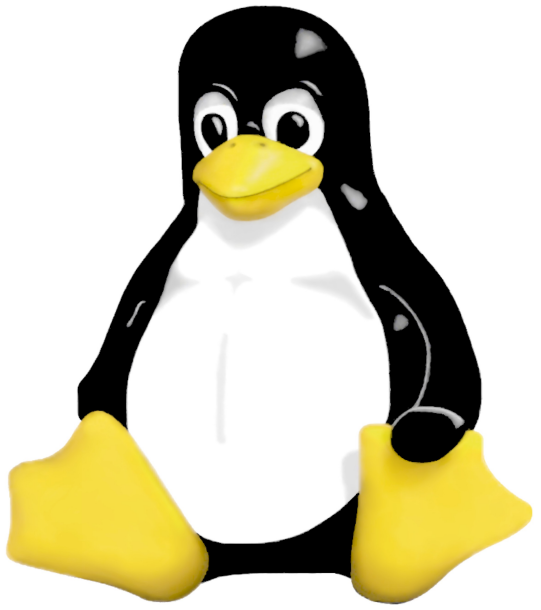
OpenVpn Tunnel sicuro e gratuito, il coltellino svizzero delle vpn



Whois lan

- ✔ 19871016 Giuseppe Marocchio – lan
- ✔ SysAdmin, network mantainer
- ✔ Gentoo user, Debian abuser
- ✔ Homepage http://www.giuseppe-marocchio.com

Introduzione

- ✔ Concetto di Vpn
- ✔ Cenni di crittografia
- ✔ OpenVpn vs IpSec
- ✔ OpenVpn come point-to-point
- ✔ OpenVpn come point-to-multi-point
- ✔ Generare una CA per i certificati di OpenVpn
- ✔ Tun vs Tap (rountig vs bridge)

Vpn

- ✔ È un tunnel site-to-site
- ✔ Simula una rete virtuale, solitamente eth o ppp
- ✔ È lo strumento per eccellenza per estendere

trammite internet o altre reti la propria LAN in

totale sicurezza, versatilità e risparmio economico

Crittografia (1)

- ✔ Cifratura simmetrica
- ✔ Cifratura assimmetrica
- ✔ Checksum crittografico
- ✔ Firma digitale

Crittografia (2)

Cifratura simmetrica -> confidenzialità

✔ Per avere confidenzialità dobbiamo poter cifrare i dati

✔ la cifratura simmetrica usa UNA sola chiave per cifrare e decifrare i dati

✔ Gli algoritmi di cifratura simmetrica sono molto veloci

esempio: des, 3des, blowfish, AES, rc5,rc6

Crittografia (3)

Checksum crittografico -> integrità

✔ i dati devono arrivare a destinazione completi e non compromessi

✔ il checksum crittografico è il risultato a lunghezza fissa di una funzione matematica NON reversibile

✔ un checksum può essere visto come il “sommario unico ” del nostro messaggio; una qualsiasi modifica al messaggio deve generare un checksum diverso

alcuni esempi: Md-5, sha-1

Crittografia (4)

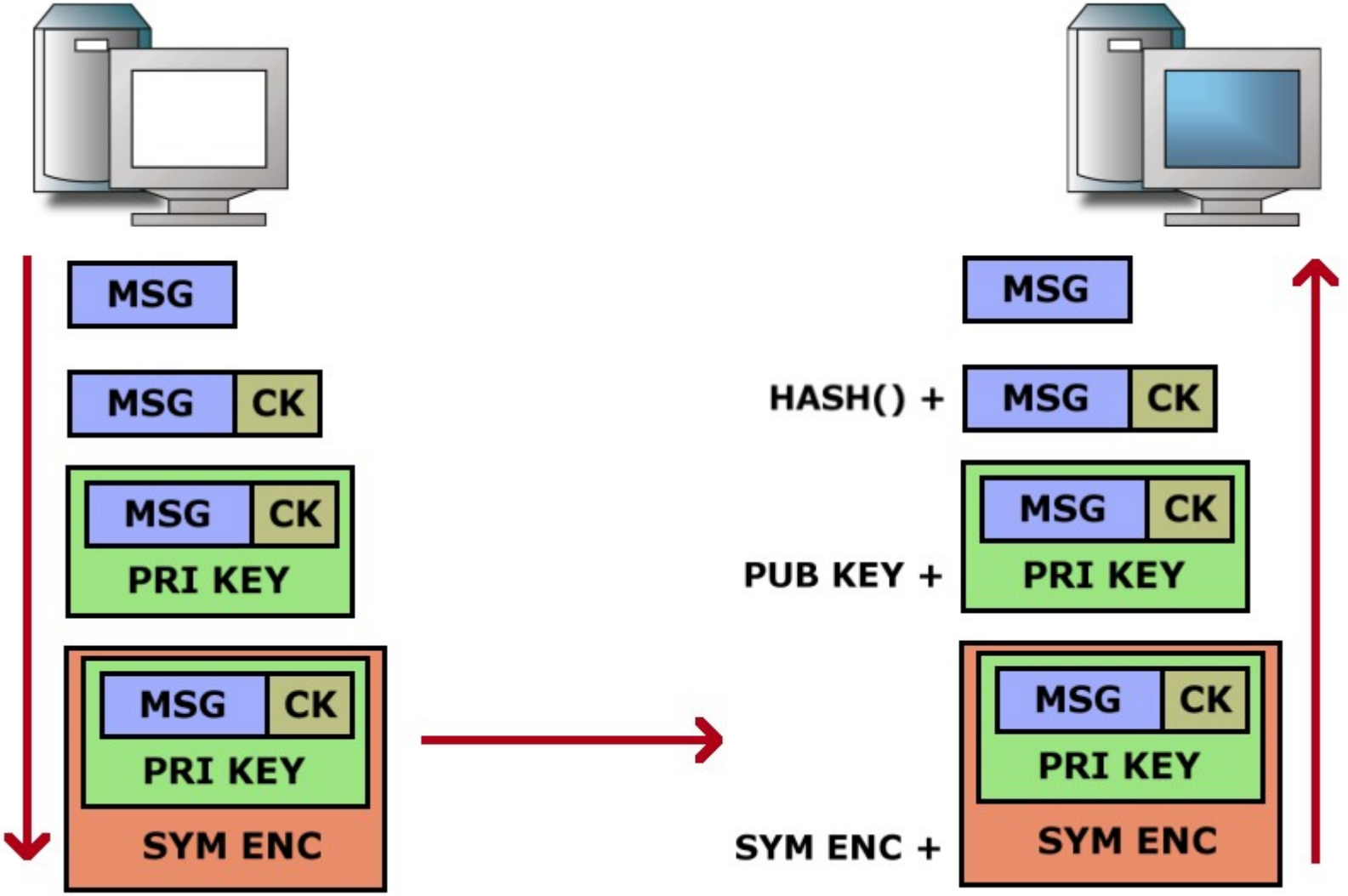
Cifratura asimmetrica, firma digitale -> confidenzialità, responsabilità

✔ dobbiamo essere sicuri che dall'altra parte ci sia davvero chi ci aspettiamo

✔ la crittografia asimmetrica è basata su 2 chiavi, una pubblica una privata. Con quella pubblica posso solo criptare. Con quella privata posso decriptare il messaggio

✔ per autenticare il mittente è sufficente un messaggio criptato con la sua chiave privata
la crittografia a chiave pubblica rendere sicuro l'hand-shaking anche se le parti non si sono mai incontrate fisicamente

Crittografia (5)



OpenVpn vs IpSec

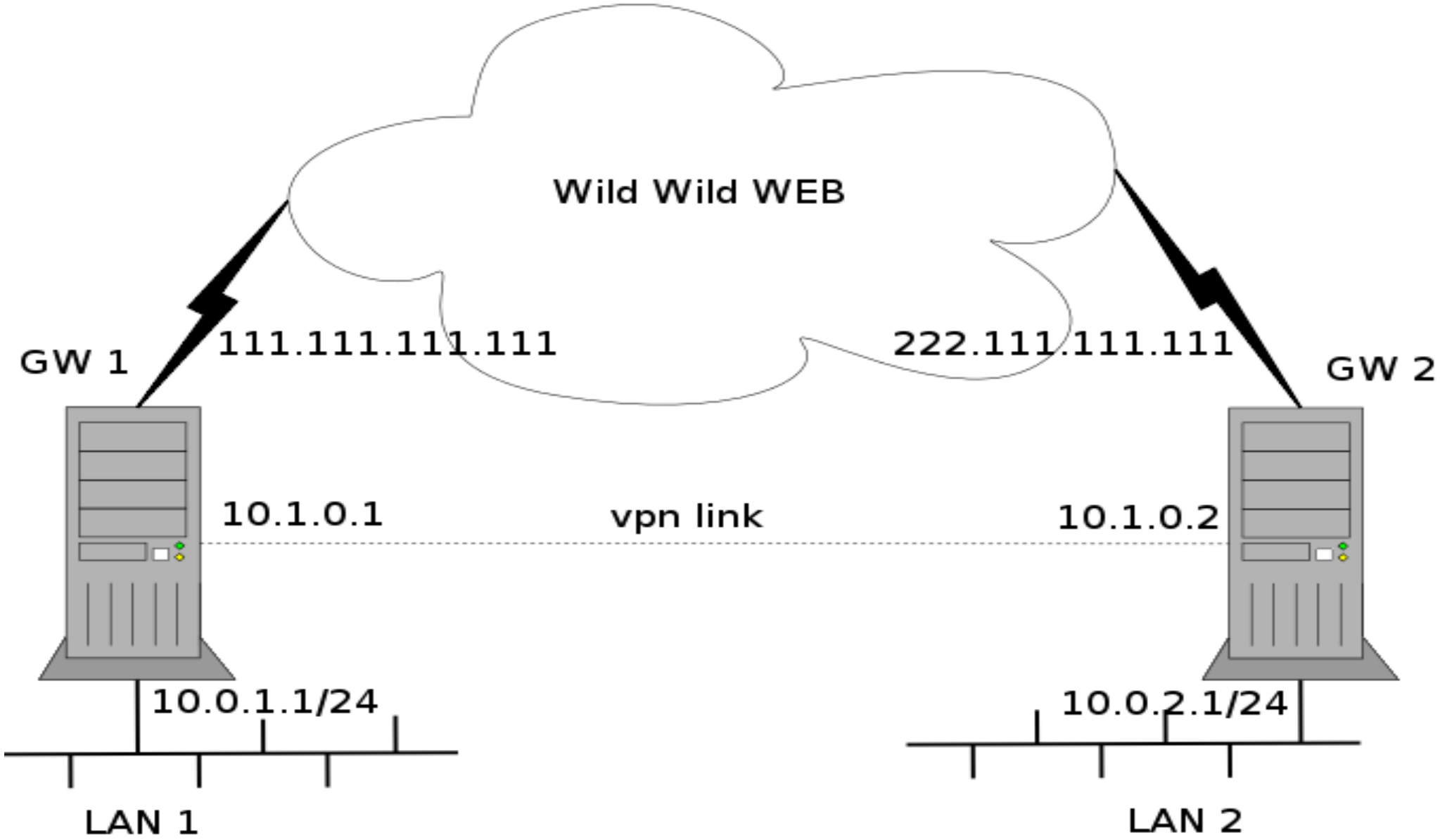
In breve

- ✔ IpSec standard dal 1995, Openvpn non ancora .

- ✔ Ipsec lavora interamente in kernel space, tuttavia è dotato di alcuni programmi di gestione user space

- ✔ OpenVpn utilizza un driver per interfacce virtuali tun/tap (tun.ko) lato kernel e un programma (demone) per instaurare il tunnel criptato

Vpn lan-lan


Wild Wild WEB
GW 1
111.111.111.111
222.111.111.111
GW 2
10.1.0.1
vpn link
10.1.0.2
10.0.1.1/24
10.0.2.1/24
LAN 1
LAN 2

Vpn lan-lan (2)

Gw1 config file

```
dev tap
proto udp-server
local 111.111.111.111
secret /etc/openvpn/key.server
ping-restart 20
verb 4
ifconfig 10.1.0.1 255.255.255.252
lport 5050
```

✔ Non scordiamoci di generare la chiave crittografica!

```
openvpn --genkey --secret /etc/openvpn/key.server
```

Vpn lan-lan (3)

Gw2 config file

```
dev tap
proto udp-server
remote 111.111.111.111
secret /etc/openvpn/key.server
ping-restart 20
verb 4
ifconfig 10.1.0.2 255.255.255.252
rport 5050
```

✔ Non scordiamoci di copiare la chiave condivisa da gw1 a gw2!

Vpn lan-lan (4)

Vpn up!

```
Gw2# ping 10.1.0.1 -c 2
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
64 bytes from 10.1.0.1:icmp_seq=1 ttl=64 time=1.95 ms
64 bytes from 10.1.0.1:icmp_seq=2 ttl=64 time=2.23 ms

--- 10.1.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss,
time 999ms
rtt min/avg/max/mdev = 1.950/2.090/2.230/0.140 ms
```

✔ La Vpn è attiva, con ifconfig possimo vedere quindi l'interfaccia virtuale tap0

Vpn lan-lan (5)

Traceroute?

```
Pc-lan1# tracepath 10.0.2.2
 1:  pc-lan1.lan (10.0.1.2)  0.108ms pmtu 1500
 1:  gw1.lan (10.0.1.1)      1.406ms
 2:  gw2.lan (10.1.0.2)      5.394ms
 3:  pc-lan2.lan (10.0.2.2)  15.059ms reached

Resume: pmtu 1500 hops 3 back 3
```

Funziona! Ora (dopo aver inserito le opportune rotte statiche) 2 host delle 2 lan riescono a vedersi tra loro in maniera perfettamente sicura, l'unico limite è la banda verso internet che è spesso soggetta a oscillazioni

Vpn lan-host

✔ Una vpn lan to single host, è simile, se non uguale all'esempio precedente. Basta solo inserire una rotta statica all'avvio della vpn!

route add -net 10.0.1.0/24 gw 10.1.0.2

✔ E' un operazione automatizzabile utilizzando il comando “up” di openvpn.

✔ Il comando “up” permette di eseguire uno script personalizzato all'avvio della vpn

Vpn punto multipunto

✔ Una vpn punto multipunto non banale quanto una vpn site-site

✔ Una vpn punto-multipunto sott'intende l'autenticazione dei vari utenti che contemporaneamente devono potersi collegare

✔ Gestire la comunicazione tra i vari client multipunto

✔ E' necessaria una CA per la generazione dei certificati

Certificati

Easy-RSA

`cd /etc/openvpn/easy-rsa`
modificare vars
impostare `KEY_CONFIG`
impostare `KEY_DIR`

✔ Eseguire `./vars` per impostare i valori delle variabili eseguire `./clean-all`

✔ Ricordare che i file .key sono segreti, mentre i file .crt e .csr non sono critici.

Certificati (2)

Costruiamo la CA:

- ./build.ca

nella directory KEY_DIR avremo ora l'accoppiata ca.crt e ca.key

- ./build-dh

Certificati (3)

Generiamo ora i certificati per il client.

```
./build-req client
```

I campi possono essere impostati come si vuole, con l'accortezza che il commonname dev'essere univoco tra tutti, di solito il nome del client.
Il certificato dev'essere ora firmato dall'autorità:

```
./sign-req client
```

In KEY_DIR si troveranno le chiavi appena create, client.crt e client.key
Per semplificare il tutto:

```
./build-key client
```

Simple Server

server.conf

```
port 1194
proto udp
dev tap
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key #
This
file should be kept secret
dh /etc/openvpn/easy-rsa/keys/dh1024.pem
server 10.0.0.0 255.255.255.0
client-config-dir ccd
route 10.0.0.0 255.255.255.252
```

Simple Server (2)

```
client-to-client
keepalive 10 120
comp-lzo
user nobody
group nogroup
persist-key
persist-tun
status openvpn-status.log
log openvpn.log
log-append openvpn.log
verb 3
route-gateway 10.0.0.1
```

Start Server

✔ È ora quindi possibile avviare il server come demone `openvpn –config server.conf`

✔ Ricordiamoci che possiamo specificare un ip statico per un client direttamente dal server trammite l'apposito script in ccd/Nome-client o una qualsiasi altra operazione (Rotta statica ecc..)

Tun vs Tap

Tun o Tap ? Questi sconosciuti

✔ tun, vpn in routing, crea un interfaccia Tun0
es:

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00
          inet addr:10.2.0.1  P-t-P:10.2.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500
Metric:1
```

✔ tap, interfaccia per vpn bridging su layer ethernet, molto utile quando si deve usare qualche protocollo non-ip o si deve usare la vpn in bridge a qualche altra interfaccia eth

Moduli per openvpn

Openvpn ha la possibilità di integrarsi con altri sistemi di autenticazione, (con l'accoppiata utente-password) tuttavia non esistono moduli già prefatti, scrivere un modulo è semplice.

✔ È possibile scriverlo trammite un programma esterno in qualsiasi linguaggio

oppure è possibile scriverlo direttamente in C e caricarlo dinamicamente all'avvio del demone.

✔ Esistono dei moduli semplici di esempio nella directory dei sorgenti di openVpn

Domande?

Bene, andiamo a pranzo ;)

Credits

Per alcune slide mi sono ispirato ad altre slide di amici, si ringraziano (in random order) Florin Iamandi
ed Enrico Cherubini per avermi concesso l'utilizzo di alcuni loro testi

Potete trovare queste slide su **http://www.giuseppe-marocchio.com**