

# IPSEC over L2tp between Debian as server and Mikrotik as client

By

**Ehsan Aminian**

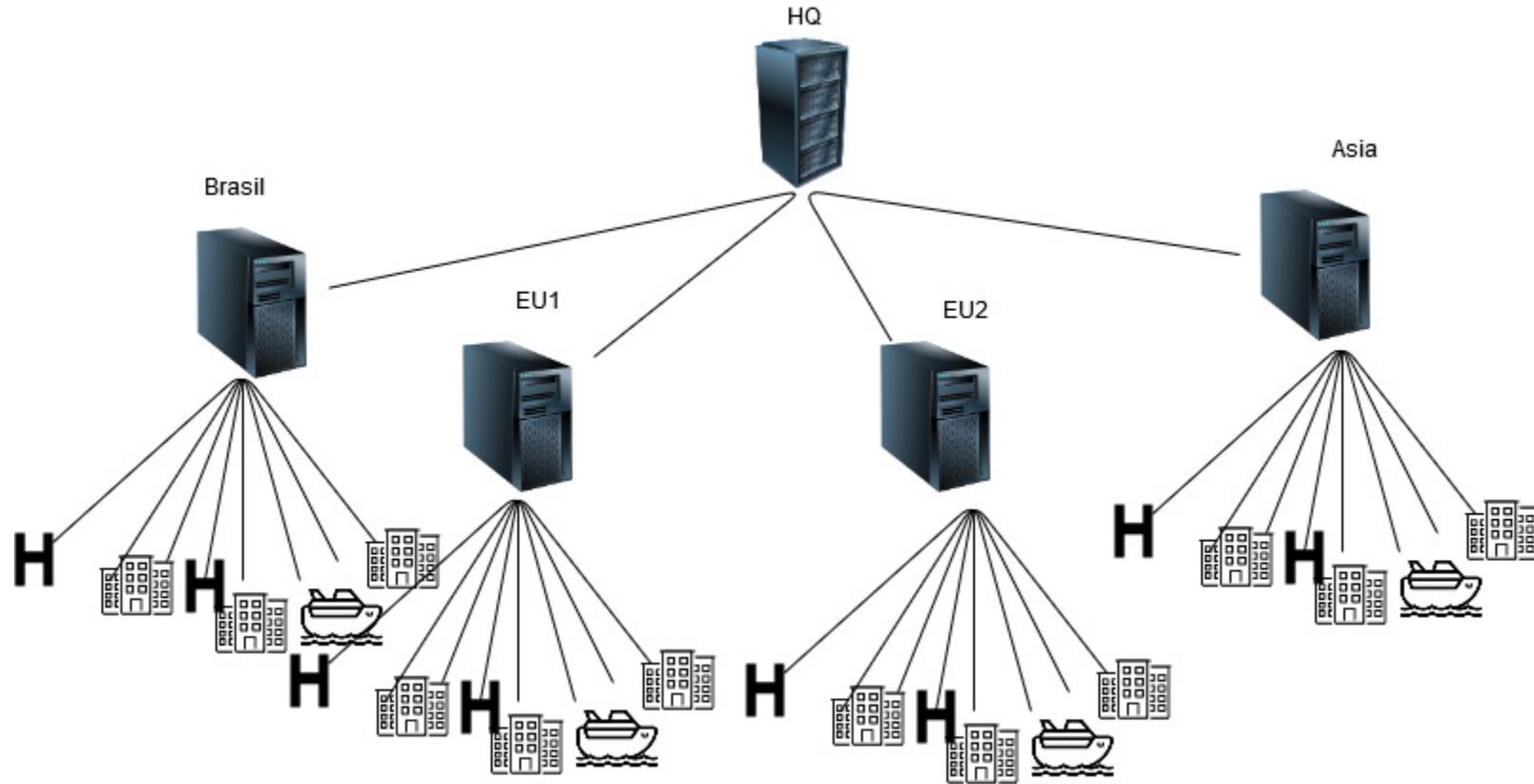
**(Certified Trainer - TR0444, MTCNA,MTCRE,MTCUME,MTCINE,MTCTCE)**

**MikrotikLand (training center)**

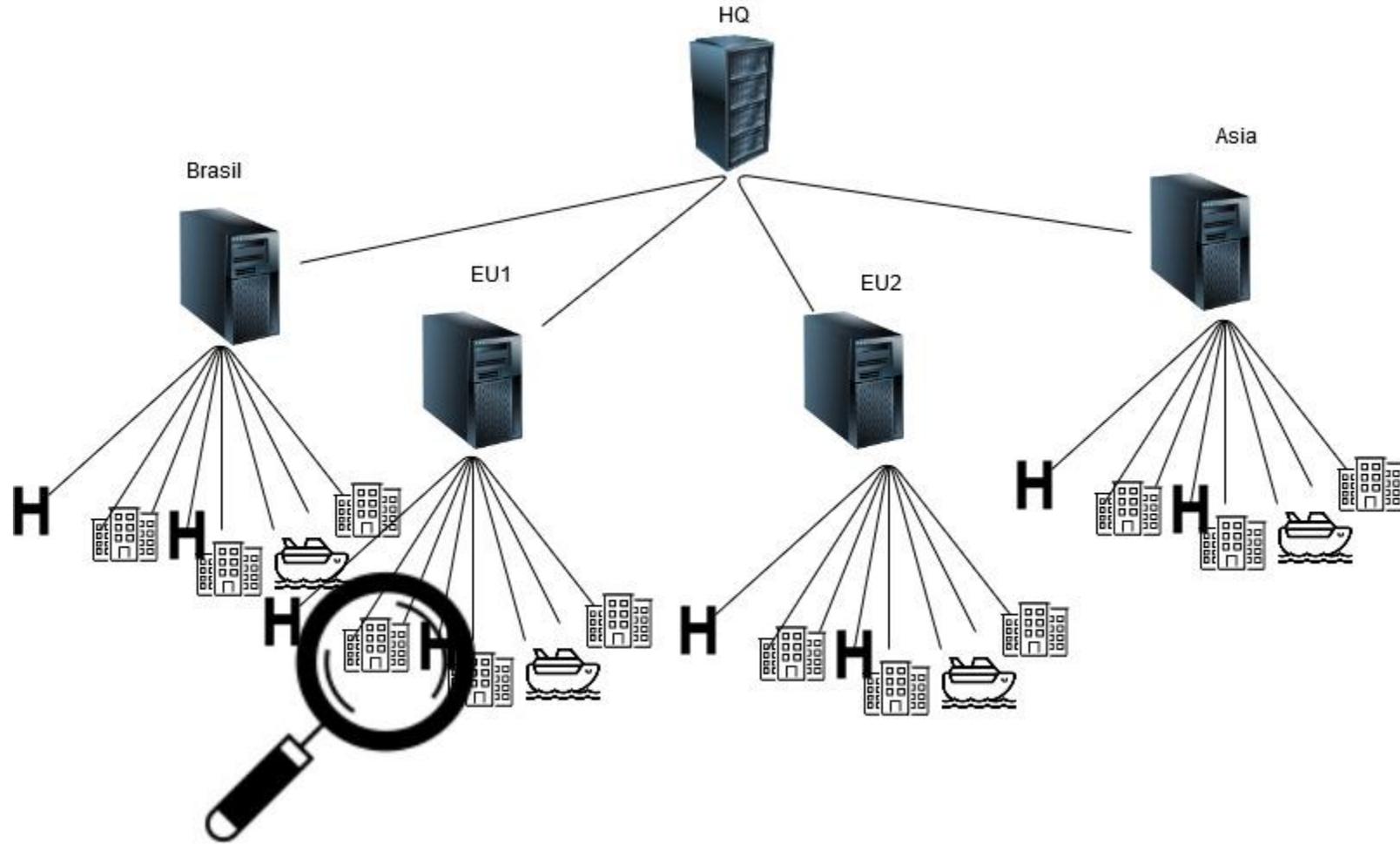
# Outline

- Network Topology
- Limitations
- Solution
- Configurations
- Questions

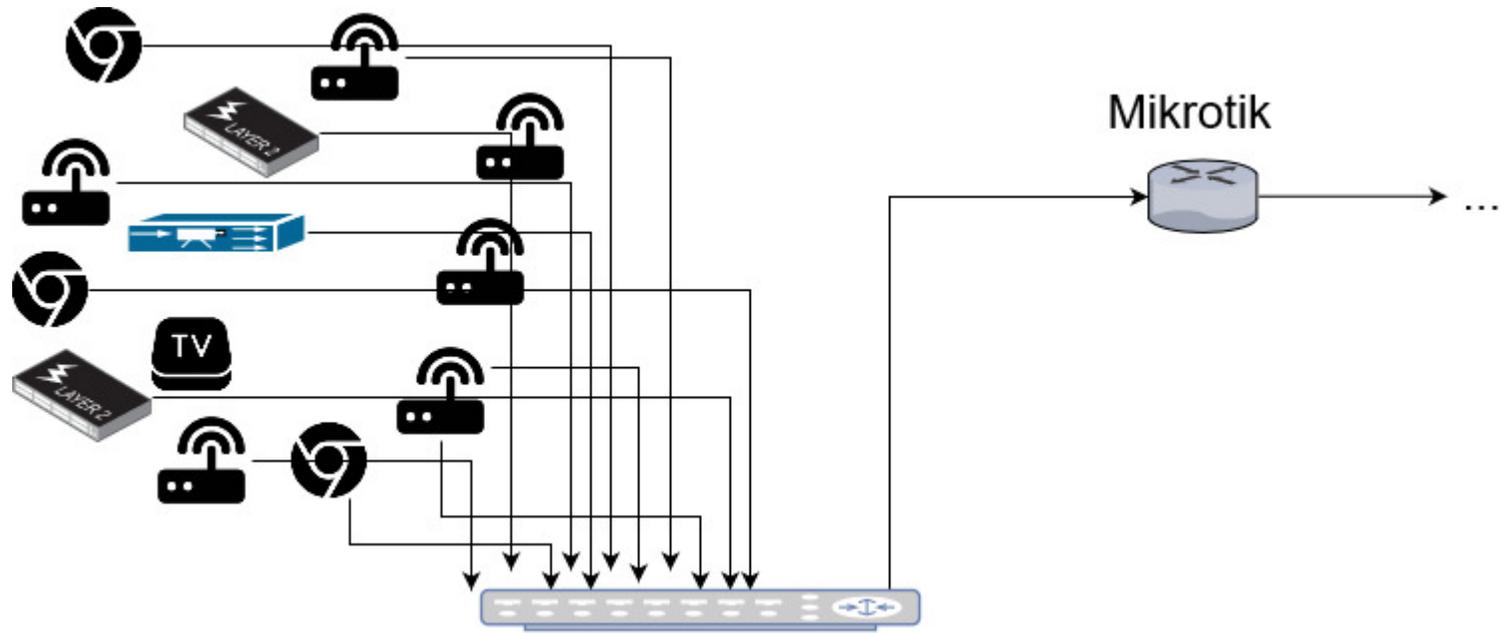
# Network Topology



# Network Topology

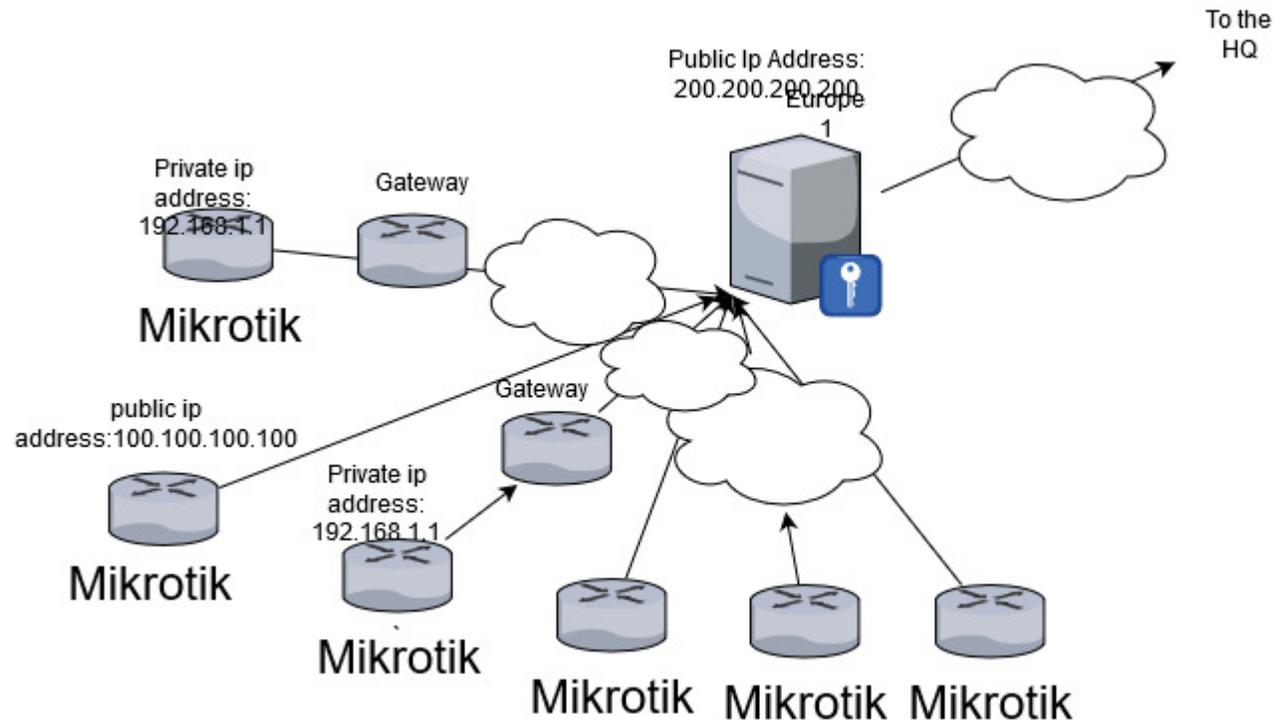


# Network Topology



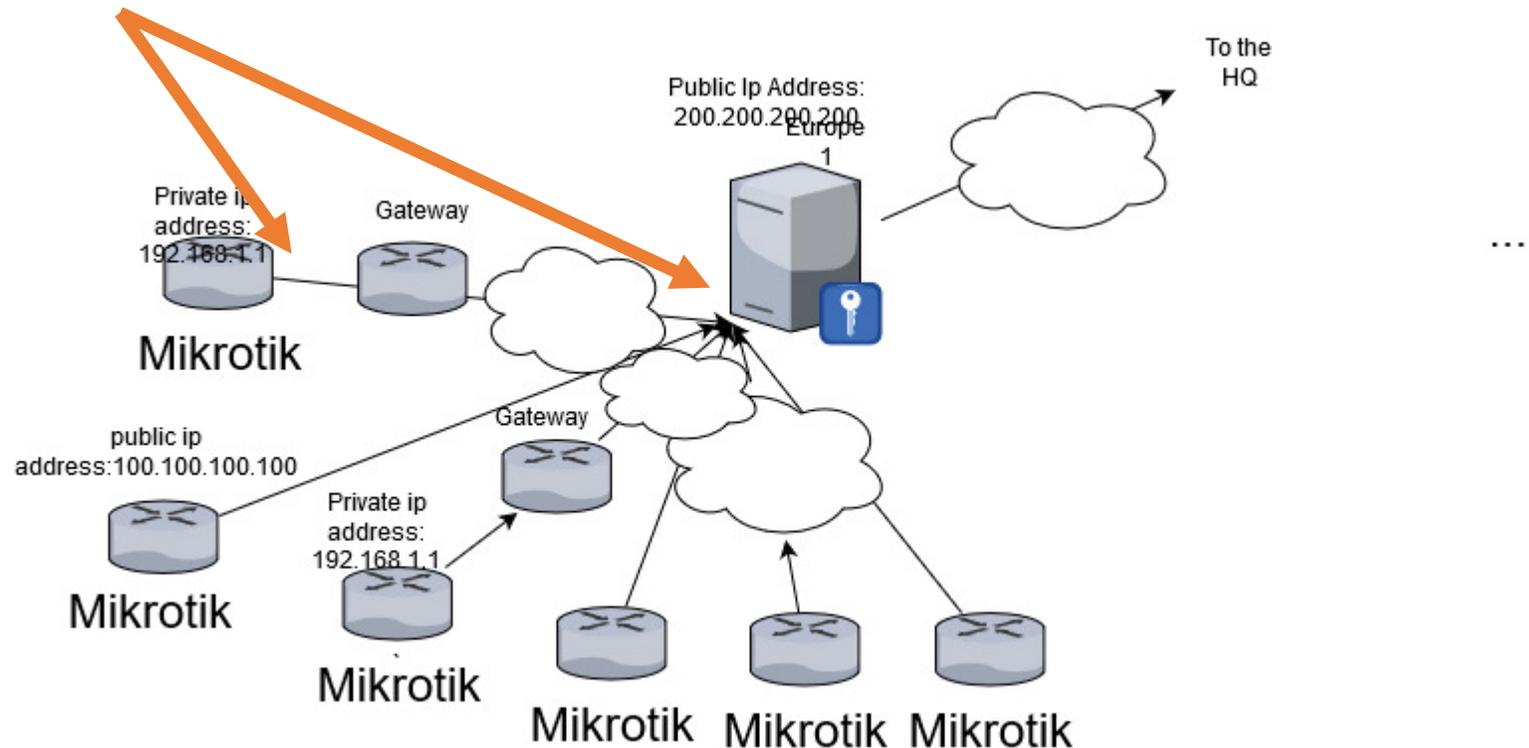
# Limitations

- Each customer has its own IP assigning policy.
- Some of Mikrotik may not have even public IP address



# Solution

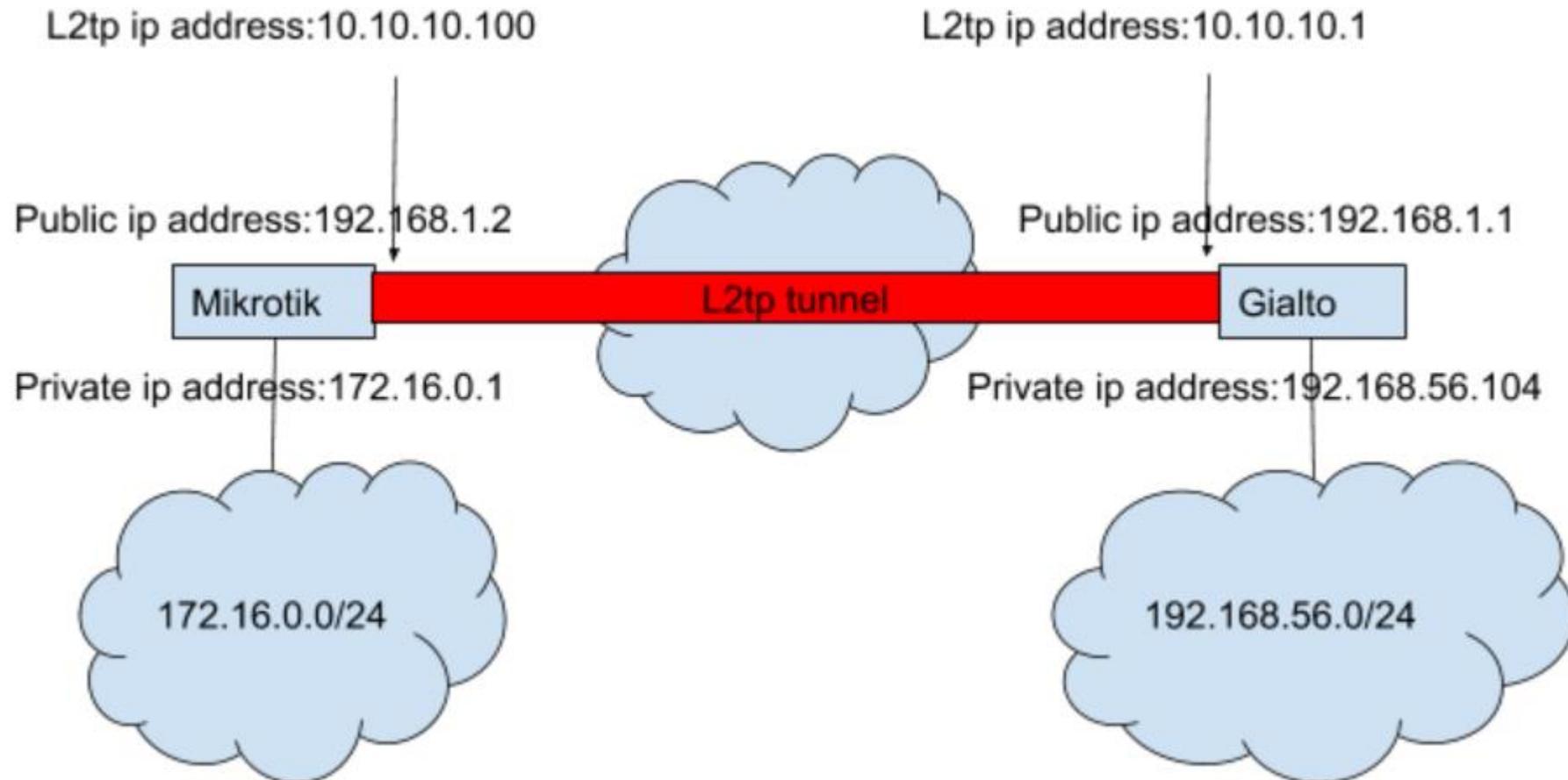
- Layer 2 Tunneling Protocol (**L2TP**) can be used to connect both site of the network



# Solution

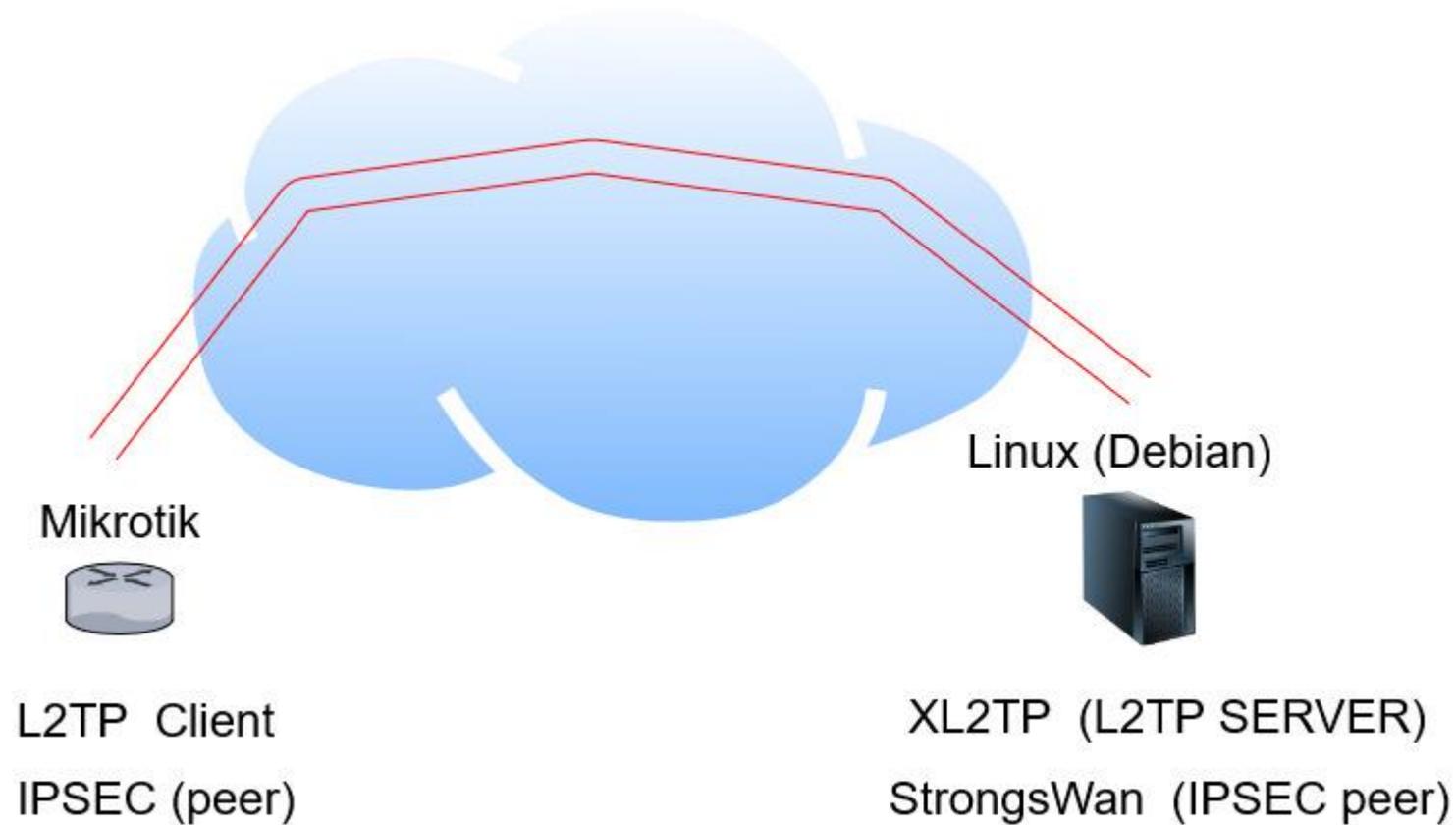
- L2TP Concerns:
  - L2TP is a client server tunnel
    - Regarding to the topology, for establishing L2TP tunnel, we should run L2TP server on the Linux (Debian) and L2TP client on our Mikrotik
  - L2TP does not provide any encryption or confidentiality by itself.
    - **IPSEC** can solve the problem

# Solution



# Configurations

# Services on both sides



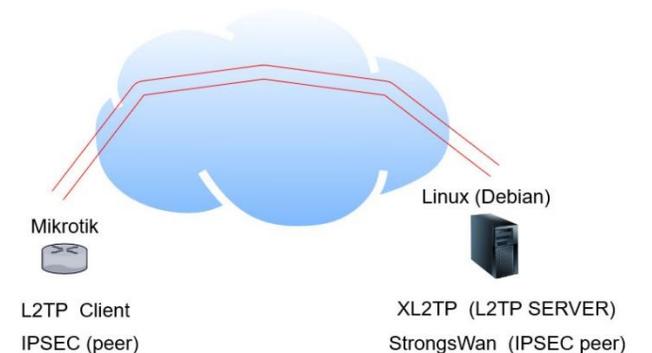
# Server Configuration (XL2TP)

Installation: `sudo apt-get install xl2tpd`

Change the configuration file `/etc/xl2tpd/xl2tpd.conf` as follows:

```
[global]
ipsec saref = no
debug tunnel = no
debug avp = no
debug network = no
debug state = no
access control = no
rand source = dev
port = 1701
auth file = /etc/ppp/chap-secrets
```

```
[lns default]
ip range = 10.10.10.110-10.10.10.150
local ip = 10.10.10.1
require authentication = yes
name = l2tp
pass peer = yes
ppp debug = no
length bit = yes
refuse pap = yes
refuse chap = yes
pppoptfile = /etc/ppp/options.xl2tpd
```



# Server Configuration (XL2TP)

## Adding Users and secret for L2TP clients

You have to edit `/etc/ppp/chap-secrets` file like the following:

```
# Secrets for authentication using CHAP
# client  server  secret      IP addresses
  ali      *      123        10.10.10.100
```

Above file shows this credential:

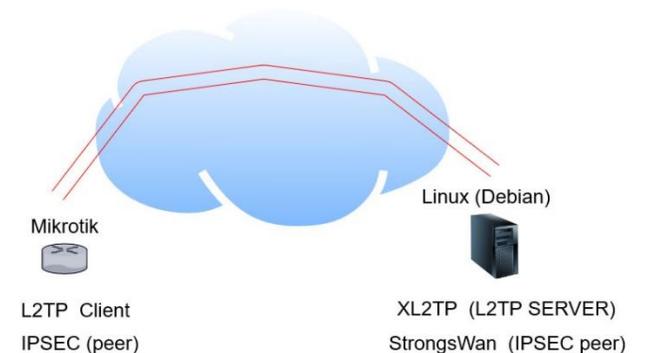
username: ali

Password: 123

Ip address:10.10.10.100

Run the below command to start XL2TP:

`XL2TP` or `XL2TP -D` (to start it in debug mode)

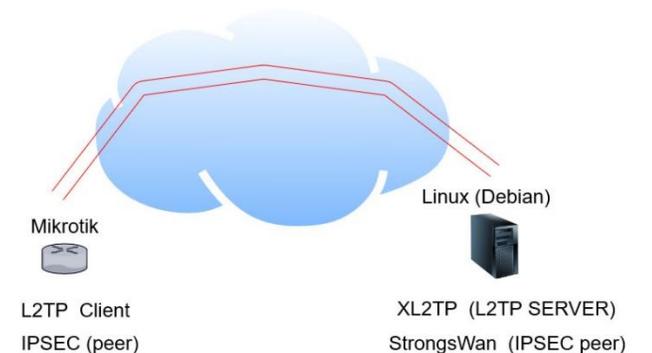


# Server Configuration (IPSEC)

Installation: `sudo apt-get install strongswan`

Edit the file `/etc/ipsec.conf`:

```
config setup
conn vpnserver
    type=tunnel
    authby=secret
    rekey=no
    keyingtries=3
    left=10.10.10.1
    leftsubnet=192.168.56.0/24
    rightsubnet=172.16.0.0/24
    leftid=10.10.10.1
    right=10.10.10.100
    auto=add
    esp=aes128-sha1-modp1536
    aggressive = no
    ike=aes128-sha1-modp1536
    ikelifetime = 3h
```



# Server Configuration (IPSEC)

## IPSec Password:

Add the following line to this file **`/etc/ipsec.secrets`** :

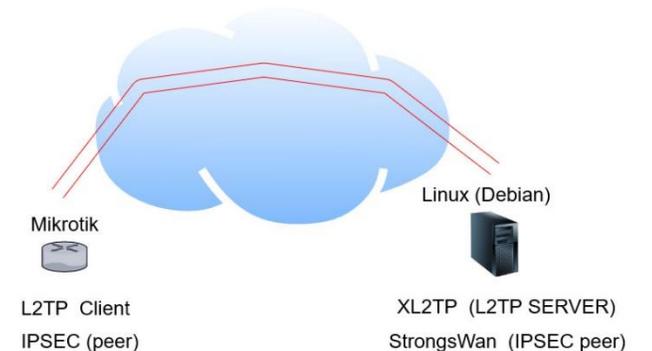
```
%any %any : PSK "123"
```

Run this command to start ipsec:

```
ipsec start
```

To check if everything is going well, check the status by:

```
ipsec statusall
```



# Some Extra work

This Linux command shows the policies and states of IPsec tunnel.

```
ip xfrm state
```

```
ip xfrm policy
```

## Firewall configuration:

You need to accept packet from your **l2tp** clients. (L2tp is port 1701)

You can see if you receive something in L2tp interface

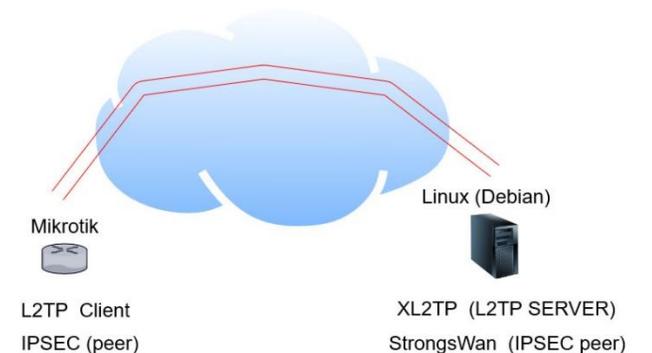
```
tcpdump -i eth0 'port 1701'
```

```
tcpdump -i ppp0
```

## How to deny all l2tp without IPSEC encryption from Mikrotik client?

```
iptables -A INPUT -p udp --dport 1701 -m policy --dir in --pol ipsec -j ACCEPT
```

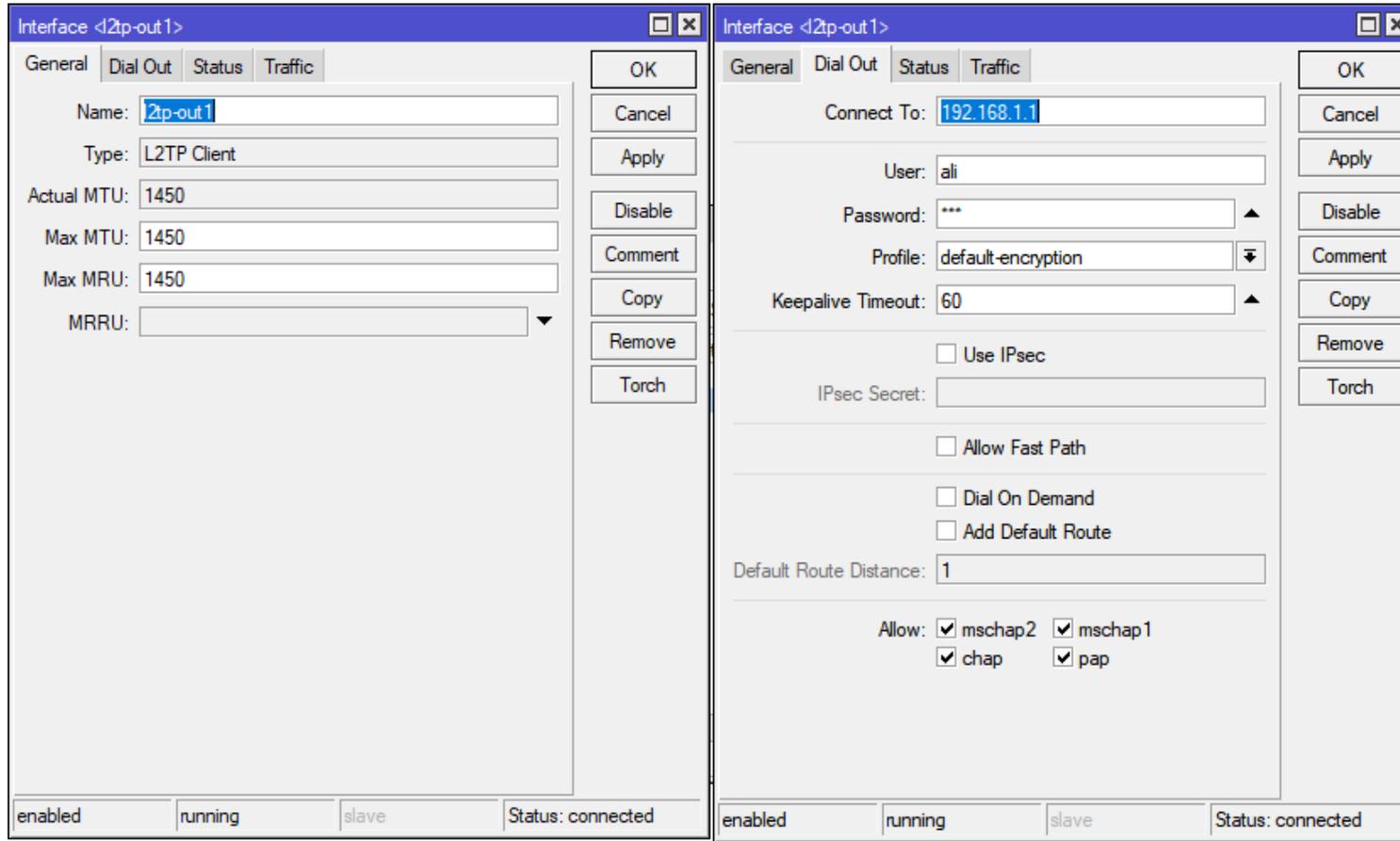
```
iptables -A INPUT -p udp --dport 1701 -j DROP
```



# Mikrotik Configuration (L2TP)

Add a l2tpclient in ppp section with the following configuration:

(Don't forget that user and password should be the same as those in /etc/ppp/chap-secrets)



The image shows two screenshots of the Mikrotik WinBox configuration interface for an L2TP client interface named 'l2tp-out1'.

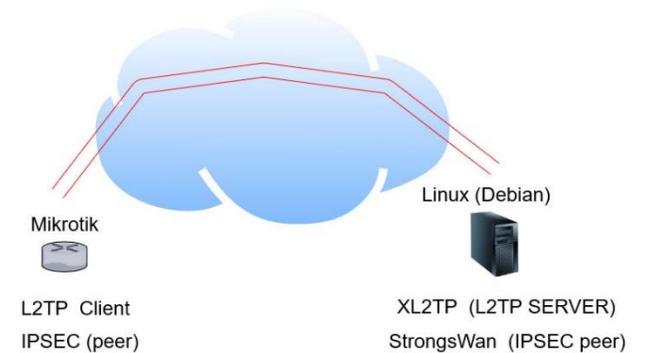
**Left Screenshot (General Tab):**

- Name: l2tp-out1
- Type: L2TP Client
- Actual MTU: 1450
- Max MTU: 1450
- Max MRU: 1450
- MRRU: (empty)

**Right Screenshot (Dial Out Tab):**

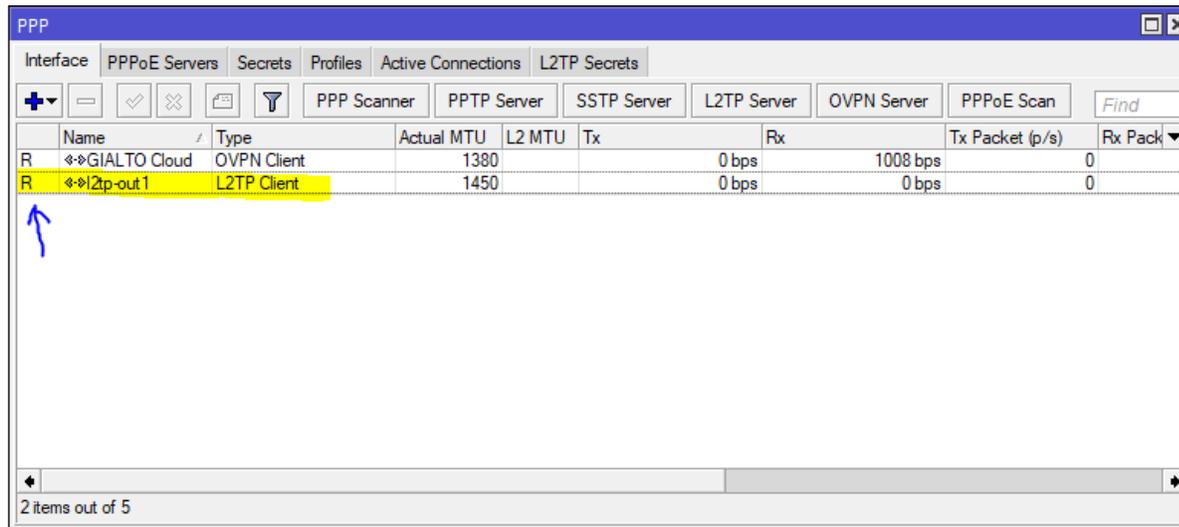
- Connect To: 192.168.1.1
- User: ali
- Password: \*\*\*
- Profile: default-encryption
- Keepalive Timeout: 60
- Use IPsec
- IPsec Secret: (empty)
- Allow Fast Path
- Dial On Demand
- Add Default Route
- Default Route Distance: 1
- Allow:  mschap2,  mschap1,  chap,  pap

Both windows show the interface status as 'enabled', 'running', 'slave', and 'Status: connected'.

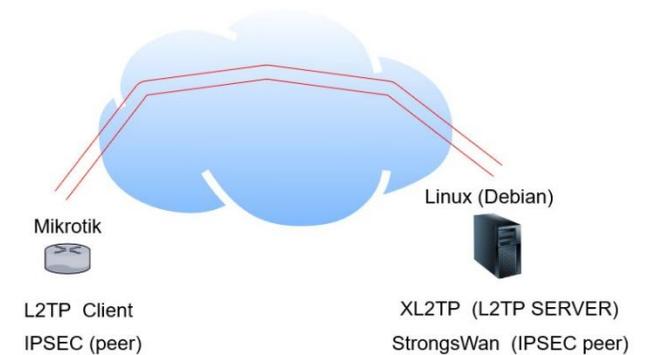


# Mikrotik Configuration (L2TP)

Check if the connection is established or not  
(you should see R in front of the connection in **ppp -> interface section**)



	Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Pack
R	↔GIALTO Cloud	OVPN Client	1380		0 bps	1008 bps	0	
R	↔l2tp-out1	L2TP Client	1450		0 bps	0 bps	0	



# Mikrotik Configuration (IPSEC)

Now, define a new IPSEC manually and set the following configurations:

**IPSEC => Policy => Add**

IPsec Policy <172.16.0.0/24:0>192.168.56.0/24:0

General Action Status

Src. Address: 172.16.0.0/24

Src. Port: [dropdown]

Dst. Address: 192.168.56.0/24

Dst. Port: [dropdown]

Protocol: 255 (all)

Template

OK Cancel Apply Disable Comment Copy Remove

enabled Template Active

IPsec Policy <172.16.0.0/24:0>192.168.56.0/24:0

General Action Status

Action: encrypt

Level: require

IPsec Protocols: esp

Tunnel

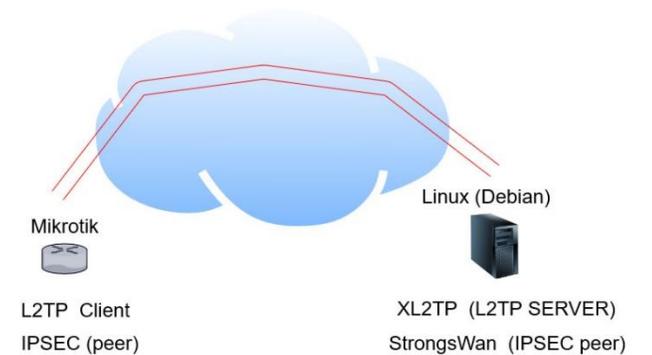
SA Src. Address: 10.10.10.100

SA Dst. Address: 10.10.10.1

Proposal: default

OK Cancel Apply Disable Comment Copy Remove

enabled Template Active



# Mikrotik Configuration (IPSEC)

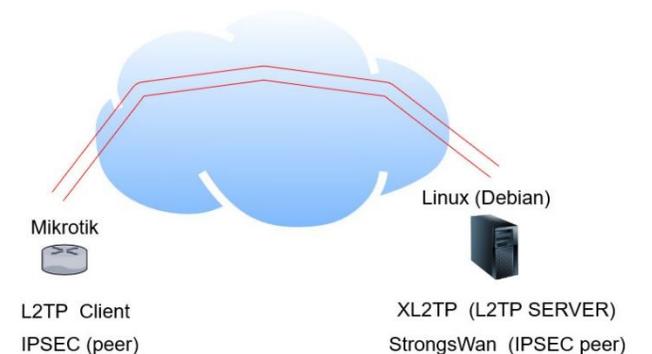
Then in the **proposal** tab, choose the following:

The screenshot shows the 'IPsec Policy Proposal <default>' window. The 'Name' field is set to 'default'. Under 'Auth. Algorithms', 'sha1' is selected. Under 'Encr. Algorithms', 'des' and 'aes-128 cbc' are selected. The 'Lifetime' is set to '03:00:00' and the 'PFS Group' is set to 'modp1536'. The window is currently in the 'enabled' state.

Auth. Algorithms	Encr. Algorithms
<input type="checkbox"/> md5	<input checked="" type="checkbox"/> des
<input type="checkbox"/> null	<input checked="" type="checkbox"/> aes-128 cbc
<input type="checkbox"/> sha512	<input type="checkbox"/> aes-256 cbc
<input type="checkbox"/> 3des	<input type="checkbox"/> twofish
<input type="checkbox"/> aes-192 cbc	<input type="checkbox"/> camellia-128
<input type="checkbox"/> blowfish	<input type="checkbox"/> camellia-192
<input type="checkbox"/> camellia-128	<input type="checkbox"/> camellia-256
<input type="checkbox"/> camellia-256	<input type="checkbox"/> aes-128 ctr
<input type="checkbox"/> aes-192 ctr	<input type="checkbox"/> aes-256 ctr
<input type="checkbox"/> aes-128 gcm	<input type="checkbox"/> aes-192 gcm
<input type="checkbox"/> aes-256 gcm	

Leave IPSEC **group** unchanged as:

The screenshot shows the 'IPsec Group <default>' window. The 'Name' field is set to 'default'. The window contains buttons for 'OK', 'Cancel', 'Apply', 'Copy', and 'Remove'. A list at the bottom shows 'default'.



# Mikrotik Configuration (IPSEC)

Then add a peer (IPSEC=> peer) like this:

IPsec Peer <10.10.10.1>

General Advanced

Address: 10.10.10.1

Port: [dropdown]

Local Address: 10.10.10.100

Profile: default

Auth. Method: pre shared key

Exchange Mode: main

Passive

Secret: \*\*\*

OK Cancel Apply Disable Comment Copy Remove

enabled responder

Unsafe configuration, suggestion to use certificates

IPsec Peer <10.10.10.1>

General Advanced

Policy Template Group: default

Notrack Chain: [dropdown]

Send Initial Contact

My ID Type: auto

Mode Configuration: [dropdown]

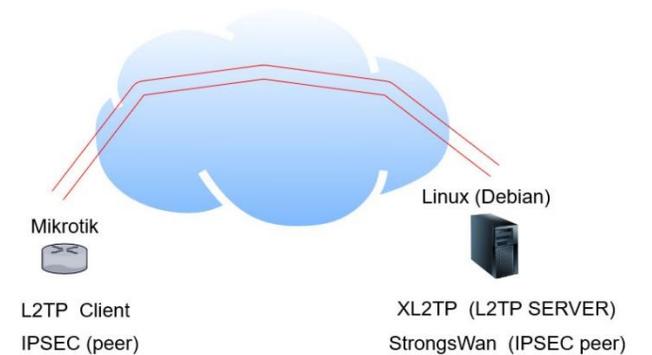
Generate Policy: no

Compatibility Options:  skip peer id validation

OK Cancel Apply Disable Comment Copy Remove

enabled responder

Unsafe configuration, suggestion to use certificates



# Mikrotik Configuration (IPSEC)

Define peer **profile** as below:

IPsec Peer Profile <default>

Name:

Hash Algorithms:

Encryption Algorithm:

<input type="checkbox"/> des	<input type="checkbox"/> 3des
<input checked="" type="checkbox"/> aes-128	<input type="checkbox"/> aes-192
<input type="checkbox"/> aes-256	<input type="checkbox"/> blowfish
<input type="checkbox"/> camellia-128	<input type="checkbox"/> camellia-192
<input type="checkbox"/> camellia-256	

DH Group:

<input type="checkbox"/> modp768	<input checked="" type="checkbox"/> modp1024
<input type="checkbox"/> ec2n155	<input type="checkbox"/> ec2n185
<input checked="" type="checkbox"/> modp1536	<input type="checkbox"/> modp2048
<input type="checkbox"/> modp3072	<input type="checkbox"/> modp4096
<input type="checkbox"/> modp6144	<input type="checkbox"/> modp8192
<input type="checkbox"/> ecp256	<input type="checkbox"/> ecp384
<input type="checkbox"/> ecp521	

Proposal Check:

Lifetime:

Lifebytes:

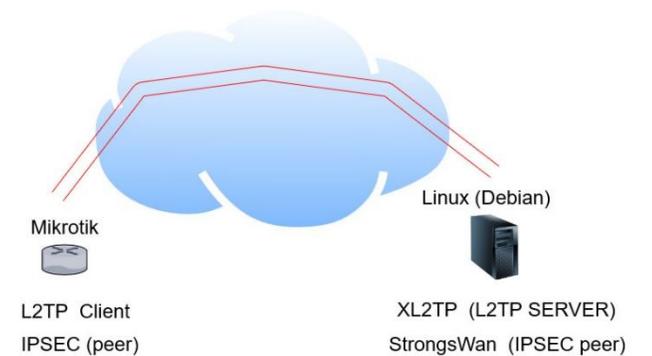
NAT Traversal

DPD Interval:  s

DPD Maximum Failures:

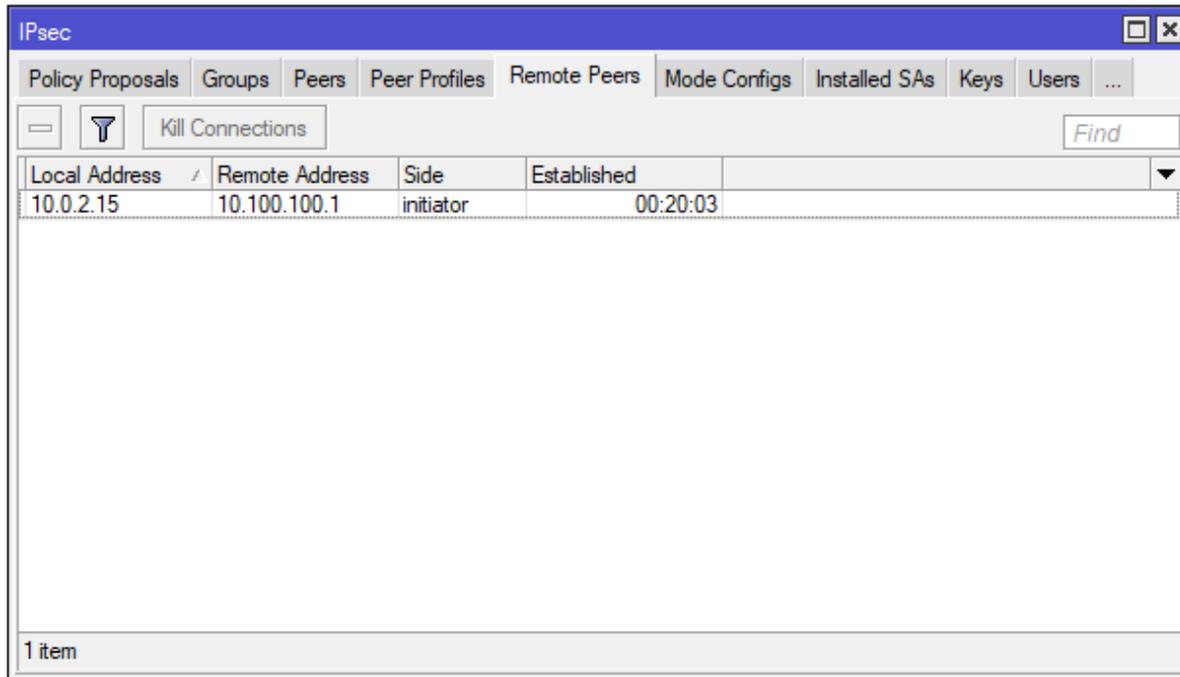
default

OK  
Cancel  
Apply  
Copy  
Remove



# Mikrotik Configuration (IPSEC)

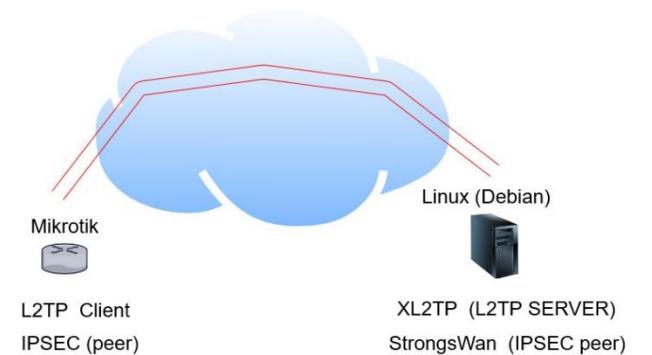
If IPSEC tunnel is established, you should see the following in **IP->IPSEC\_Remote Peers**



The screenshot shows the Mikrotik WinBox interface for the IPsec configuration. The 'Remote Peers' tab is selected, displaying a table with one established peer. The table has columns for Local Address, Remote Address, Side, and Established. The data row shows a local address of 10.0.2.15, a remote address of 10.100.100.1, a side of 'initiator', and an established time of 00:20:03. The interface also includes a 'Kill Connections' button and a search field.

Local Address	Remote Address	Side	Established
10.0.2.15	10.100.100.1	initiator	00:20:03

1 item



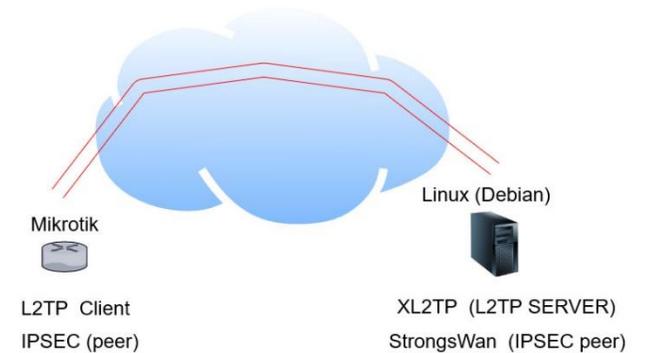
# Mikrotik Configuration (IPSEC)

And the following in **IPSEC=>Policy**:

#	Src. Address	Src. Port	Dst. Address	Dst. Port	Protocol	Action	Level	Tunnel	PH2 State
0	::/0		::/0			255 (all) encrypt			
1	172.16.0.0/24		192.168.56.0/24			255 (all) encrypt	require	yes	established

Also you should see in **IPSEC=>INSTALLED SAs**:

SPI	Src. Address	Dst. Address	Auth. Algorithm	Encr. Algorithm	Encr. Key Size	Current Bytes
E d98e643	10.10.10.1	10.10.10.100	sha1	aes cbc	128	132076
E c6724a80	10.10.10.100	10.10.10.1	sha1	aes cbc	128	132076



**Questions?**

**End..**