Mikrotik RouterOS

guides for Mikrotik's RouterOS

- Using AAISP's L2TP tunnel service for a second LAN in RouterOS 7.8

Using AAISP's L2TP tunnel service for a second LAN in RouterOS 7.8

Note:
This guide was originally written in March 2023, and is rather hacky...
A new one is coming soon though!

Why?

Before I had my FTTP internet line installed, I had to share a Virgin Media 'Business' connection with housemates.
but running internet-facing services on a connection shared with people who aren't you is A BAD IDEA, security wise...
(and rather rude!)

Plus, Virgin Media is *crap*, when you're a nerd. Even their supposed 'Business' plan...
No static IPv4, no IPv6 at all, CGNAT, horrendous packet shaping, and awful routing/peering on the open internet.

So, I signed up for Andrews & Arnold's L2TP Tunnel service.
It works as a kind of VPN over your existing connection, but gives you a real, un-faffed-with connection, a static IPv4 address, and a /64 of v6!

Sweet!

But there was a problem. I couldn't find a decent guide anywhere on how to set up a second LAN with all its internet traffic going over an L2TP tunnel.

So, I figured it out, and wrote one!
There's most definitely a better way to do this, but hey, it works!

Prerequisites:

> This guide assumes the following things, as this is what I was working with

- you're signed up for AAISP's L2TP tunnel service
- your router is running RouterOS 7.8
- your router is already configured with one WAN (in my case, the Virgin connection), one LAN, and a working IPv4 firewall/NAT setup
- your existing network and WAN are IPv4-only
- you're reasonably comfortable using the RouterOS CLI from inside Winbox
- you have a second device or ethernet port (and cable!) for testing

Disclaimer:

> **Follow this guide at your own risk!**
> **I am not responsible for any mistakes, explosions, bricked routers, or anything else!**

That said, I have done my best to ensure this guide is as accurate as possible, and these instructions did work for me!

(if I did make a mistake though, please contact me, and I'll fix it!)

However, I will not provide any support, so be warned!

1: Create the tunnel

To start off, we'll be creating a new **L2TP Client** interface, using the details provided by AAISP.

Open a new Terminal inside Winbox, then run the following;

```
/interface l2tp-client
add allow-fast-path=yes connect-to=90.155.53.19 disabled=no \
    keepalive-timeout=disabled name=AAISP profile=default user=YOURUSERNAME password=YOURPASSWORD
```

> Make sure to replace "YOURUSERNAME" and "YOURPASSWORD" with the correct details!

After this has been set up, you should now see a new L2TP client interface in the Winbox interface list, called **AAISP.**

Make sure its status is "connected", if it is, you're good to go!

2: Set up a new LAN bridge

Next, we'll be setting up a new LAN bridge called **LAN2**, which will be completely seperate from your existing LAN.

```
/interface bridge port
add bridge=LAN2 interface=ether3
```

> I've used **ether3** as the physical port here, but you can use any port you like.
> (or even a VLAN, but I won't cover that here)

After setting this up, you should see a new bridge interface called **LAN2** in the interface list.

4: Set up an IPv4 address pool for LAN2

Now, we need to configure a range of IPv4 addresses for our new **LAN2** network to use.

```
/ip pool
add name=LAN2 ranges=192.168.60.100-192.168.60.199
```

> I've used the subnet **192.168.60.0/24**, and a DHCP range between **192.168.60.100-192.168.60.199** here.
> (but again, you can choose a different subnet if you'd like!)

5: Add an IPv4 address to the LAN2 bridge

Now we've chosen our subnet, we need to assign an address to our new LAN2 bridge interface.

```
/ip address
add address=192.168.60.1/24 interface=LAN2 network=192.168.60.0
```

>

> Here, I've used**192.168.60.1**, but if you've chosen a different subnet, use an address which matches that!

6: Set up DHCP for LAN2

For our clients to automatically be assigned an IPv4 address in our chosen subnet, we need to configure a DHCP server.

```
/ip dhcp-server network
add address=192.168.60.0/24 dns-server=9.9.9.9 gateway=192.168.60.1 netmask=24
```

> I'm using **Quad9 (9.9.9.9)** as my DNS server here, but you can use any you like.
> (I do recommend Quad9 though, it's brilliant :))

7: Add new WAN and LAN interfaces to the appropriate interface lists

We now need to add our new interfaces to the appropriate interface lists, to make sure our existing firewall rules work properly with them.

I'm using the list **WAN** for internet-facing interfaces, and **LAN** for my internal networks.

```
/interface list member
add interface=AAISP list=WAN
add interface=LAN2 list=LAN
```

Once this has been configured, plug a handy second device into your router's **ether3** port.

You should now be assigned an IPv4 address from your chosen range, and be able to access the internet using your existing WAN connection!
If that works, unplug it again, then move on to the next step.

8: Set up a new routing table for LAN2

Next, we will be configuring a **new routing table for LAN2**, seperate from our existing/main one.

This will let us apply routing rules to only LAN2, so we can direct only LAN2's internet traffic down our tunnel.

```
/routing table
add disabled=no fib name=LAN2
```

9: Add IPv4 default route for LAN2

Now we have a new routing table for LAN2, we need to set the default gateway on it to use our **AAISP** tunnel.

Without this, our router won't know where to send traffic from LAN2 destined for the internet!

```
/ip route
add dst-address=0.0.0.0/0 gateway=AAISP pref-src="" routing-table=LAN2 \
    distance=1 scope=30 target-scope=10 suppress-hw-offload=no disabled=no
```

10: Route all IPv4 internet traffic from LAN2 down the L2TP tunnel

Next, to allow IPv4 internet access on LAN2, we need to use our new **LAN2** routing table instead of our main one.

```
/routing rule
add action=lookup-only-in-table disabled=no src-address=192.168.60.0/24 table=LAN2
```

This rule tells our router to use the **LAN2** routing table for all traffic originating from the **192.168.60.0/24** subnet.

> If you've used a different subnet, be sure to replace **192.168.60.0/24** here with yourchosen subnet!

11: Test IPv4 internet access!

To test that everything we've done so far works properly, plug your handy second device back into the **ether3** port on your router.

It should then be assigned a local IPv4 address from your DHCP range, and have access to the internet.

Then, open a web browser, and go to speedtest.net.

You should now see your ISP listed as **AAISP**, and your new external IPv4 address!

Run the speed test to make double sure it works properly, and if it does, your tunnel is now configured properly for IPv4!

But wait? What about IPv6?

Brilliant observation, Watson!

We still need to configure our router to allow IPv6 access over the tunnel as well, so let's configure that now :)

12: Add IPv6 default route for LAN2

This is almost the same as Step 9, but instead of IPv4, it's providing a default route to the internet for IPv6 traffic from LAN2.

```
/ipv6 route
add disabled=no distance=1 dst-address=::/0 gateway=AAISP routing-table=LAN2 \
    scope=30 target-scope=10
```

13: Add an IPv6 address to the LAN2 bridge

Now, we need to add an IPv6 address to our **LAN2** bridge interface.

You'll need to use the IPv6 prefix supplied by AAISP, which can be found by logging into control.aa.net.uk, and looking at the **IP Addresses** section under **Broadband Circuit Details**.

Once you've found that, (**it should start with 2001:, and end with ::**) add it to the **LAN2** bridge interface.

```
/ipv6 address
add address=YOUR-IPV6-ADDRESS interface=LAN2
```

> Make sure to replace YOUR-IPV6-ADDRESS with your actual IPv6 address!

14: Set the IPv6 DNS server

For IPv6 clients to resolve domain names, we need to advertise an IPv6 DNS server using Neighbour Discovery.

```
/ipv6 nd
set [ find default=yes ] dns=2620:fe::fe interface=LAN2
```

> I've used **Quad9** again here, but you can use any IPv6 DNS server!

15: Test IPv6 internet access!

You're almost done, but now we need to test that IPv6 works!

To do this, you'll need to unplug then reconnect your handy second device to your router's **ether3** port, then wait around 30-60sec for an IPv6 address to be automatically assigned.

Then, visit test-ipv6.com in a web browser, and you should be rewarded with a perfect 10/10 score! It will show both your public IPv4 address and your public IPv6 address too :)

15: Set up the IPv6 Firewall

Finally, we need to make sure your firewall is configured for IPv6, as well as for IPv4.
This ensures your connected devices remain safe on the internet.

In Winbox, navigate to **IPv6 > Firewall**, and you should be greeted with the default IPv6 firewall rules.

However, if you see none, configure the defaults from Mikrotik, using this:

```
/ipv6 firewall {
  address-list add list=bad_ipv6 address=::/128 comment="defconf: unspecified address"
  address-list add list=bad_ipv6 address=::1 comment="defconf: lo"
  address-list add list=bad_ipv6 address=fec0::/10 comment="defconf: site-local"
  address-list add list=bad_ipv6 address=::ffff:0:0/96 comment="defconf: ipv4-mapped"
  address-list add list=bad_ipv6 address=::/96 comment="defconf: ipv4 compat"
  address-list add list=bad_ipv6 address=100::/64 comment="defconf: discard only "
  address-list add list=bad_ipv6 address=2001:db8::/32 comment="defconf: documentation"
  address-list add list=bad_ipv6 address=2001:10::/28 comment="defconf: ORCHID"
  address-list add list=bad_ipv6 address=3ffe::/16 comment="defconf: 6bone"
  address-list add list=bad_ipv6 address=::224.0.0.0/100 comment="defconf: other"
  address-list add list=bad_ipv6 address=::127.0.0.0/104 comment="defconf: other"
  address-list add list=bad_ipv6 address=::/104 comment="defconf: other"
  address-list add list=bad_ipv6 address=::255.0.0.0/104 comment="defconf: other"
  filter add chain=input action=accept connection-state=established,related,untracked comment="defconf:
accept established,related,untracked"
  filter add chain=input action=drop connection-state=invalid comment="defconf: drop invalid"
  filter add chain=input action=accept protocol=icmpv6 comment="defconf: accept ICMPv6"
  filter add chain=input action=accept protocol=udp port=33434-33534 comment="defconf: accept UDP
traceroute"
  filter add chain=input action=accept protocol=udp dst-port=546 src-address=fe80::/10 comment="defconf:
accept DHCPv6-Client prefix delegation."
  filter add chain=input action=accept protocol=udp dst-port=500,4500 comment="defconf: accept IKE"
  filter add chain=input action=accept protocol=ipsec-ah comment="defconf: accept ipsec AH"
  filter add chain=input action=accept protocol=ipsec-esp comment="defconf: accept ipsec ESP"
```

```
  filter add chain=input action=accept ipsec-policy=in,ipsec comment="defconf: accept all that matches ipsec policy"
  filter add chain=input action=drop in-interface-list=!LAN comment="defconf: drop everything else not coming from LAN"
  filter add chain=forward action=accept connection-state=established,related,untracked comment="defconf: accept established,related,untracked"
  filter add chain=forward action=drop connection-state=invalid comment="defconf: drop invalid"
  filter add chain=forward action=drop src-address-list=bad_ipv6 comment="defconf: drop packets with bad src ipv6"
  filter add chain=forward action=drop dst-address-list=bad_ipv6 comment="defconf: drop packets with bad dst ipv6"
  filter add chain=forward action=drop protocol=icmpv6 hop-limit=equal:1 comment="defconf: rfc4890 drop hop-limit=1"
  filter add chain=forward action=accept protocol=icmpv6 comment="defconf: accept ICMPv6"
  filter add chain=forward action=accept protocol=139 comment="defconf: accept HIP"
  filter add chain=forward action=accept protocol=udp dst-port=500,4500 comment="defconf: accept IKE"
  filter add chain=forward action=accept protocol=ipsec-ah comment="defconf: accept ipsec AH"
  filter add chain=forward action=accept protocol=ipsec-esp comment="defconf: accept ipsec ESP"
  filter add chain=forward action=accept ipsec-policy=in,ipsec comment="defconf: accept all that matches ipsec policy"
  filter add chain=forward action=drop in-interface-list=!LAN comment="defconf: drop everything else not coming from LAN"
}
```

16: You're done!

If you've followed this properly, you should now have a working, fully separate LAN2, with full IPv4 and IPv6 internet access!

Go and make yourself a nice brew, you deserve it :)

You can now connect **ether3** to a network switch, and use it for all your clients, servers, and more!

I hope this was useful :)