N2N Interface an eine Brücke kleben

Beschreibung:

Der sinn eines layer 2 Tunnels ist das der Ganze Netzwerkverkehr geroutet wird, also kein ip sondern ethernet frame.
Damit aber unser vpn interface auch irgendwo raus kann und nicht nur im tunnel gefangen ist, erstellen wir eine Brücke mit einem normalen interface, also das was irgendwo raus kann und unser n2n0 interface mit dran.
Das auf beiden seiten. und schon haben wir Sinnbildlich ein Netzwerkkabel von A nach B gelegt, übers Internet natürlich.

Inbetriebnahme

Das Paket bridge-utils haben wir schon installiert.

Auch hier packen wir das wieder in unsere interfaces Datei.
Wir erstellen eine neue bridge und packen an diese bridge unseren verbindungsaufbau und wenn das unser Server ist auch den supernode.
Denn erst wenn die Bridge fertig ist erstellen wir den tunnel und assignen ihn. Wir können den n2n0 Tunnel nicht sofort in der config mit dran packen, denn er existiert noch nicht, aber die Netzwerkkarte schon.

Beispiel interfaces:

Erläuterung:
Ein bridge ist wie eine Netzwerkkarte.
hier bekommt sie unsere Adresse die die Netzwerkkarte sonst hätte samt gateway netmask dns
Unter bridged ports wer hätte es gedacht stehen nachher die devices. Unsere Netzwerkkarte und das n2n0 Device
Wir tragen aber nur die echte Netzwerkkarte ein.
Die anderen Befehle kennen wir schon, ist das device up dann supernode erstellen, sollte es der supernode server sein.

(hier ist es der Fall)
im pre-down den supernode wieder killen

im post-up die edge verbindung herstellen
und nun kommt das neue, nachdem die edge verbindung hergestellt wurde, mittels brctl die n2n0 device zur bridge hinzufügen
im predown das interface von der brücke wieder entfernen.
Dann zu guter letzt edge killen.
Schon haben wir unsere n2n0 device an eine brücke gepackt mit einem anderen netzwerkinterface

```
auto eth0
iface eth0 inet manual

auto br0
iface br0 inet static

    address 167.235.xxx.xxx
    netmask 255.255.255.255
    gateway 172.31.1.1
    dns-nameservers 8.8.8.8 8.8.4.4

    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0

    post-up nohup /usr/bin/supernode -l 5555 > /var/log/supernode.log 2>&1 &
    pre-down /usr/bin/pkill supernode

    post-up /usr/sbin/edge -r -d n2n0 -c "layer2" -k "7473535ghbfdsAq!" -a 10.10.2.1 -l "167.235.xxx.xxx:5555" -f
> /va>
    post-up /bin/sh -c "brctl addif br0 n2n0"
    pre-down /bin/sh -c "brctl delif br0 n2n0"
    pre-down /usr/bin/pkill edge
```

Hinweis: Wenn das n2n0 interface an eine Brücke hängt, ist das pingen auf den internen adressen nicht mehr möglich, außer man hängt dei internen Adressen an die bridge.
bei einem layer 2 tunnel interessieren uns die ips auch nicht.
Denn eijn layer 2 tunnel kann gleichgestellt werden wie ein switch.
Die netzwerkports die mit der Brücke verbunden sind , sind die buchsen vom switch.

Würde man eine dritte verbindung noch von irgendwo anders aufbauen, können alle 3 Standorte miteinander kommunizieren als wären sie an einem switch.

Fertig.

Wekzeuge zum Überprüfen des status:

arping:

Augabe:

Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 21.255ms
Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 20.561ms
Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 3.325ms
Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 4.039ms
Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 2.805ms
Unicast reply from 78.47.xxx.xxx [FA:B6:CE:7E:64:D5] 4.669ms

wie wir sehen ist das die MAC Adresse des ersten Adapters, aber wir haben das doch eigentlich auf enps19 gelegt,
denn enp6s19 ist doch mit dem netzwerkadpter des wir nennen es mal vswitch PC verbunden.
jep aber das arp kommt vom vswitch enp6s20 an das enp6s19 im server an, da aber alle interfaces doert erreichbar sind, gibt es die mac des ersten wieder.
Aber wir sehen das arp findet zur richtigen maschine.
Würden wir aber zum beispiel ein vlan auf de enp6s19 packen,
würde das arp immer noch funktionieren, allerdings die Pakete kommen nicht mehr an enp6s20 an.

brctl show

Ausgabe vom vswitch

bridge name□bridge id□□STP enabled□interfaces
br0□□8000.3e9283b89594□no□□enp6s20
□□□□□□□   n2n0

Hiermit können wir überprüfen ob auch beide Netzwerkadapter an die Brücke geheftet sind

trace route

Ausgabe:

traceroute to 78.47.xxx.xxx (78.47.xxx.xxx), 30 hops max, 60 byte packets
 1 fritz.box (192.168.178.1) 18.739 ms 18.697 ms 24.687 ms
 2 85.16.121.38 (85.16.121.38) 26.205 ms 27.366 ms 27.396 ms
 3 bbrt-ol-0-90730207-ae26.ewe-ip-backbone.de (85.16.251.206) 27.341 ms 27.333 ms 27.324 ms
 4 bbrt-ol-0-1-90730201-ae4.ewe-ip-backbone.de (80.228.90.49) 27.317 ms 27.310 ms 27.303 ms
 5 bbrt-ffm-0-23730205-ae11.ewe-ip-backbone.de (212.6.114.38) 35.071 ms 35.064 ms 35.057 ms
 6 decix-gw.hetzner.com (80.81.192.164) 35.048 ms 10.851 ms 10.819 ms
 7 core12.nbg1.hetzner.com (213.239.252.26) 19.413 ms 19.395 ms 28.227 ms
 8 * * *
 9 spine2.cloud1.nbg1.hetzner.com (213.133.112.198) 19.348 ms spine1.cloud1.nbg1.hetzner.com
(213.133.112.194) 20.421 ms spine2.cloud1.nbg1.hetzner.com (213.133.112.198) 20.412 ms
10 * * *
11 12205.your-cloud.host (116.203.161.48) 17.632 ms 22.230 ms 22.212 ms
12 static.115.118.47.78.clients.your-server.de (78.47.xxx.xxx) 35.191 ms 33.650 ms 33.603 ms

Version #5
Erstellt: 29 Oktober 2023 14:11:43 von Admin
Zuletzt aktualisiert: 29 Oktober 2023 14:40:06 von Admin