

UDM-PRO-SE | DSL-Modem erreichen

Was wollen wir?

Auf das an der UDM-PRO-SE angeschlossene Modem zugreifen.

Warum wollen wir das?

Das Modem hat nur ein Netzwerk-Interface (wie z.B Vigor 130), es stehen keine normalen LAN-Ports mehr zu Verfügung oder sollen nicht verschwendet werden.

Dennoch wäre ein gelegentlicher Zugriff erwünscht, um sich z.B die (V)DSL-Verbindungsdatenanzuschauen oder Updates durchzuführen.

Und wie geht das genau?

Voraussetzungen:

- Grundkenntnisse auf der Linux Shell
- Modem Konfiguration
- einen (privaten) IP-Bereich der NIRGENDWO intern benutzt wird.
Das Beispiel hier benutzt: 10.99.99.0/28
10.99.99.1/28 -> IP des Modem
10.99.99.2/28 -> IP des der UDM
- UDM-PRO-SE getestet mit der 2.5.11, es spricht aber nichts dagegen, dass es auch auf den andere UMD oder DR funktioniert. Da müssen aber ggf. Pfade, Netzwerkschnittellen UND Start-Scripte angepasst werden.

Step1: Modem Konfiguration

Am Beispiel eines DrayTek Vigor 130, IP Adresse und Maske setzen und fertig.

rk Configuration

le

ress

10.99.99.1

Mask

255.255.255

Step2:

auf der UDM-PRO-DE ist das "/data" Verzeichnis der Ort, der soweit bekannt, nicht von Updates überschrieben wird (auf der UDM-PRO ist zur Zeit /mnt/data/)

sofern noch nicht geschehen hier ein „myScripts“ Verzeichnis anlegen (`mkdir /data/myScripts`) wo dann alles reinkommt.

Dort dann eine `accessModem.sh` erstellen mit folgenden Inhalt.

Bash: `/data/myScripts/accessModem.sh`

```
#!/bin/bash
```

```
####
##          accessModem.sh          v1.3          for          UDM-PRO-SE
##
## Add a subinterface to the WAN Port of the UDM, also add (p)NAT
## rule to access Modem without need for further Modem Config.
##
##                                     Setup:
## copy to /data/myScripts/accessModem.sh (persistant path may differe on NON PRO-SE)
## change Variable INTERFACE, LOCALE_IP, REMOTE_IP or SCRIPTPATH if needed.
##
##      INTERFACE:      (for      UDM-PRO-SE      other      may      difference)
## eth7      -> PORT 8 (special WAN port on the 8 Port switch on the SE)
##      eth8          ->      PORT      9      (normal      WAN      Port)
##      eth9          ->      Port10      (upper      SFP      Port)
##      eth10         ->      Port11      (lower      SFP      Port)
##
##      LOCALE_IP: the IP and MASK for the interface on the UDM
##      REMOTE_IP: the IP of the Modem
## (reminder, not use any ip network range that use on LAN side)
##
##                                     Use:
## /data/myScripts/acceshssModem.sh start -> Add the Interface and iptable rule
## /data/myScripts/acceshssModem.sh stop  -> Remove Interface and iptable rule
##
##                                     Changelog:
##      V1.0          ->          2022.10.02          -->          Initial
## V1.1 -> 2022.10.17 change iptables static path to "whereis" Variable to cover FW2.x & 3.x
## V1.2 -> 2022.10.20 change to macvlan interface as additional ip get removed by UNFI
##
## V1.3 -> 2023.09.20 change harcoded Script path for logfile to Variable
##      replace "whereis iptables.." to a which call to make it more easy
##      credits for both to noexpand (thanks for point it to me)
##
##      2023,          by          Gierig,          free          to          use
```

```
INTERFACE="eth8"
LOCALE_IP="10.99.99.2/28"
REMOTE_IP="10.99.99.1"
```

```
SCRIPTPATH="/data/myScripts"
SCRIPTLOG="log.txt"
```

```
####      typical      no      change      need      below      this      line      #####
```

```
DATE=$(date +%F_%H:%M:%S)
IPT=$(which iptables)

case "$1" in
    start)
        /sbin/ip link add name ${INTERFACE}.modem link ${INTERFACE} type macvlan
        /sbin/ip addr add ${LOCALE_IP} dev ${INTERFACE}.modem
        /sbin/ip link set dev ${INTERFACE}.modem up

        $IPT -t nat -I POSTROUTING 1 -o ${INTERFACE}.modem -d ${REMOTE_IP} -j MASQUERADE
        echo "$DATE -> accessModem -> Start" >> ${SCRIPTPATH}/${SCRIPTLOG}
        ;;

    stop)
        /sbin/ip addr del ${LOCALE_IP} dev ${INTERFACE}.modem
        /sbin/ip link set dev ${INTERFACE}.modem down

        $IPT -t nat -I POSTROUTING 1 -o ${INTERFACE}.modem -d ${REMOTE_IP} -j MASQUERADE
        echo "$DATE -> accessModem -> Stop" >> ${SCRIPTPATH}/${SCRIPTLOG}
        ;;
esac
```

Alles anzeigen

Angepasst werden muss ggf.:

INTERFACE=„eth8“ -> eth8 ist Port 9 auf der UDM-PRO-SE, eth9 währe der obere SFP Port 10, eth10 der untere SFP Port 11

LOCALE_IP="10.99.99.2/28" -> Die Ip und Maske die die UDM bekommt

REMOTE_IP=„10.99.99.1“ -> die IP des Modems

Nach dem da Script ausführbar ist (chmod +x accessModem.sh) kann es mit /data/myScripts/accessModem.sh start gestartet werden

oder mit „stop“ die einstellungen auch wieder entfernt werden.

Mehr ist nicht nötig um dann über die 10.99.99.1 auf das Modem zuzugreifen.

STEP3: AUTO START

Wer nach einem Boot das script nicht Manuel starten möchte fügt noch eine Start Unit hinzu.

(Hier sind dann auch die Meisten unterschiede zu normalen UDM da läuft das booten von Skripten anders ab)

Dazu im gleichen "/data/myScripts/, Verzeichnis eine „accessModem.service“ anlegen:

Code: /data/myScripts/accessModem.service

```
###
##          a      Start      Unit      to      handel      external      accessModem.service
##          2022,      by      Gierig,      free      Modem      Access
##

[Unit]
Description=Modem                        External                        Access
After=network.target

[Service]
Type=simple
RemainAfterExit=yes
ExecStart=/data/myScripts/accessModem.sh      start
ExecStop=/data/myScripts/accessModem.sh      stop

[Install]
WantedBy=multi-user.target
```

Alles anzeigen

mit `ln -s /data/myScripts/accessModem.service /etc/systemd/system/accessModem.service` einen link der datei in system config Verzeichnis anlegen

und mit `systemctl enable accessModem` die start unit dem System bekannt machen.

EXTARS:

```
d, S - static, R - RIP, * - default
1 255.255.255.240 directly co
1 255.255.255.0 via 10.99.99
18/ 255.255.255.255 via 10.99
```

Mit Routen die auf die UDM gehen können interne oder auch externe Ziele erreicht werden. Hier eine route zu meinen 192.168.1.0/24 netz

in dem z.B ein Syslog stehen könnte (man erspart sich irgendwelche Portforwarding regeln) oder die Zweite IP is ein Zeitserver der extern liegt

(offizieller ptb.de Zeitserver) die IP dann auch als NTP quelle im router eintragen und schon hat dieser auch die richtige Uhrzeit...

Anmerkungen:

für die USG gibt es [HIER](#) eine schöne Beschreibung.

Firewall Regeln beachten, auf Gruppen / Vlan / Anschluss basierenden Regeln greifen wahrscheinlich nicht da das "interface" an "Unifi" vorbei eingerichtet ist.

das (p)SNAT mit iptables könnte man sich sparen, die zu Modems degradierten Router (Vigor/Fritz) lassen aber selten eine default route auf dem LAN

Interface zu. Manuelle Routen für die nötigen VLAN wahren aber auch möglich, ist aber extra Pflege auf dem Modem (siehe extras)

Mit dem NAT und ohne Routen hat das Modem aber auch nur max zugriff auf die UDM und kann die Geräte dahinter nicht direkt erreichen

Disclaimer:

Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantie auf Erfolg. Im Falle eines Misserfolges hilft aber die Community hier sicherlich weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden.

Eltern haften für ihre Kinder.

Auswählen: _____

Gültige Software-Version Keine Firmware-Relevanz!