Cipe – VPN in einem IP-Tunnel

- Kurzakte –

Server:

Installation	
• Installation des Pakets cipe (Verschlüsselter IP ueber UDP Tunnel) aus der Paketgruppe Productivity/Networking/Security /usr/sbin/ciped-cb • auf **beiden** Routern installieren	
Konfiguration	
/etc/cipe/options	• Hauptkonfigurationsdatei • Erstellen aus mitgelieferter samples.options • Einstellung der internen und externen IP-Adressen
psaux|md5sum	• Erzeugen eines statischen keys • muss an den Gegenstellen gleich sein
/etc/modules.conf	• folgende Einstellungen müssen enthalten sein : alias cipcb0 cipcb alias cipcb1 cipcb options cipcb cipe_debug=0 • immer, wenn die Schnittstelle cipcb0 aktiviert wird, wird das Kernelmodul cipcb verwendet
/etc/cipe/ip-up	• beim Start des virtuellen Netzwerks wird dieses Skript aktiviert • Anpassung der Routing-Einstellungen, wobei die offizielle Adresse der Gegenseite als Host für das lokale CIPE-Interface eingetragen wird
/etc/cipe/ip-down	• Skript wird beim Deaktivieren der virtuellen Schnittstelle gestartet
Start als Stand-Alone-Dienst	
/etc/init.d/cipe rccipe (nur bei SuSE)	• start | stop | restart | status - startet manuell
insserv cipe (nur SuSE)	• Einfügen des Startskriptes in den Standard-Runlevel • alternativ über den Runlevel-Editor im Yast
Funktionskontrolle	
ifconfig cipb0	• Betrachten der virtuellen Schnittstelle
route	• Sichten der eingetragenen Routen
Dokumentation	
http://www.cipe.org	

Clients:

auf jedem Rechner der lokalen Netze muss der VPN-Router als Gateway eingetragen werden

/etc/cipe/samples.options

```
# Originaldatei SuSE
# Surprise, this file allows comments (but only on a line by themselves)

# This is probably the minimal set of options that has to be set

# Without a "device" line, the device is picked dynamically

# the peer's IP address
ptpaddr       6.5.4.3       # lokale IP-Adresse der Gegenstelle (z. B. 192.168.10.1)
# our CIPE device's IP address
ipaddr        6.7.8.9       # eigene lokale IP-Adresse (z. B. 192.168.10.2)
# my UDP address. Note: if you set port 0 here, the system will pick
# one and tell it to you via the ip-up script. Same holds for IP 0.0.0.0.
me            bigred.inka.de:6789   # eigen externe Adresse (z. B. user1.dyndns.org)
# ...and the UDP address we connect to. Of course no wildcards here.
peer          blackforest.inka.de:6543      # Adresse der Gegenstelle (z. B. user2.dyndns.org)
# The static key. Keep this file secret!
# The key is 128 bits in hexadecimal notation.
key           3248fd20adf9c00ccf9ecc2393bbb3e4    # einige Zeichen ändern!!!  (auf beiden Seiten gleich)
```

/etc/cipe/samples.options

```
# !!!  Testdatei für den Router in Hamburg !!!
# Für Router der Gegenseite müssen ip-Adressen getauscht werden!

# Surprise, this file allows comments (but only on a line by themselves)

# This is probably the minimal set of options that has to be set

# Without a "device" line, the device is picked dynamically
# virtuelle Netzwerkkarte für den Datenversand
device  cipcb0  # virtuelle Netzwerkkarte für den Datenversand

# the peer's IP address
# (lokale Adresse des Routers der Gegenseite)
ptpaddr       192.168.18.1

# our CIPE device's IP address
# (lokale Adresse des Routers)
ipaddr        192.168.17.1

# my UDP address. Note: if you set port 0 here, the system will pick
# one and tell it to you via the ip-up script. Same holds for IP 0.0.0.0.
# me <eigene externe IP>:<UDP-Port>
me            200.0.24.1:9999

# ...and the UDP address we connect to. Of course no wildcards here.
# peer <externe IP der Gegenseite>:<UDP-Port>
peer          200.0.4.2:9999

# Ausschalten der Datenverschlüsselung
# nokey       yes

# The static key. Keep this file secret!
# The key is 128 bits in hexadecimal notation.
# Erstellen eines Schlüssels mit psaux|md5sum
key           3248fd20adf9c00ccf9ecc2393bbb3e4

# Protokollierung aller Aktionen zu Testzwecken
# debug       yes
```

1,1 Top

/etc/cipe/ip-up

```
#!/bin/sh
# ip-up <interface> <myaddr> <daemon-pid> <local> <remote> <arg>

# Arguments:
#  $1 interface     the CIPE interface
#  $2 myaddr        our UDP address
#  $3 daemon-pid    the daemon's process ID
#  $4 local         IP address of our CIPE device
#  $5 remote        IP address of the remote CIPE device
#  $6 arg           argument supplied via options

# Purposes for this script: set up routes, set up proxy-arps, etc.
# start daemons, logging...

umask 022
PATH=/sbin:/bin:/usr/sbin:/usr/bin

case `uname -r` in
2.0*)
  # Under Linux 2.0, a minimal route to the remote CIPE is needed.
  # 2.1 and later sets this one by itself.
  route add -host $5 dev $1
  ;;
esac

# If this becomes our default route...
#route add default gw $5

# just a logging example
now=`date "+%b %d %T"`
echo "$now UP   $*" >> /var/log/cipe.log

# Create/update PID file. Note: PKCIPE needs this.
echo "$3 $1" >/var/run/cipe/${6:-$1}.pid

# Trigger the key exchange procedure, useful when we're using SOCKS
# This _must_ run delayed and in the background
#(sleep 10; ping -c5 $5) &

# If the system runs gated, tell it what has happened
#gdc interface

# The following are just ideas for further consideration

# Interconnect two 10. subnets through the Internet!
# Assuming $4 is in 10.1 and $5 in 10.2
#route add -net 10.2.0.0 netmask 255.255.0.0 gw $5

# Proxy-ARP the peer's address on eth0
#arp -i eth0 -Ds $5 eth0 pub

# Evil tricks department: masquerade the CIPE peer's /24 network to our IP
#NA=`expr $5 : '\([0-9]*\.[0-9]*\.[0-9]*\.\)'`
#ipfwadm -F -a accept -m -b -S $NA.0/24 -D 0.0.0.0/0
# the usual way for this would be a case selection on $5 or $6, however

exit 0
```