

DSL-Router und Masquerading mit SuSE 7.3 (Kernel2.4 und iptables)

Konfiguration der internen Netzwerkkarte (Beispiel !!!)

IP: 192.168.0.1 / 255.255.255.0
Nameserverlist: 194.25.2.132 / 145.253.2.11 → /etc/resolv.conf
Domainsearchlist: nacamar.de → /etc/resolv.conf

Konfiguration der externen Netzwerkkarte (an der das DSL-Modemhängt)

IP: 172.16.0.1 / 255.255.0.0

Eintragungen in der /etc/route.conf

```
# <Internes IP Netzwerk> <dummy Gateway> <subnet> <interne Netzwerkkarte>
    192.168.0.0                0.0.0.0        255.255.255.0        eth0

# <ExternesIPNetzwerk> <dummy Gateway> <subnet> <externeNetzwerkkarte>
    172.16.0.0                0.0.0.0        255.255.0.0         eth1

#      Kein Default-Gateway in /etc/route.conf eintragen !!
```

Neustart des Netzwerks:

rcnetwork restart (oder: /etc/init.d/network restart)

Neu einlesen der Routingtable:

rcroute restart (oder /etc/init.d/route restart)

Yast/Administration/Konfigurationsdatei verändern (rc.config)

IP_Dynip yes
IP_Forward yes
start_smpppd yes

Änderungen in der /etc/pppoe.conf

User= ATM@t-online.de
(AnschlusskennungT-onlinenummerMitbenutzernummer)
Demand= yes
DNS= 194.25.2.132 #(Telekomnameserver)
Interface= eth1 # Netzwerkkarte mit Zugang zum DSL.Modem

Start des SMPPPD:

rcsmpppd restart

Information:

Scripte liegen in /etc/ppp

Patches aufspielen für `smpppd.rpm` und `yast2-config-adsl.rpm`

<ftp://ftp.suse.com/pub/suse/i386/update/7.3/n1/smpppd.rpm>

<ftp://ftp.suse.com/pub/suse/i386/update/7.3/yast1/yast2-config-adsl.rpm>

Folgendes Script schreiben (Masquerading ohne spezielle Firewallregeln !):

```
#!/bin/sh
echo"setting firewall rules"
set IPTABLES=/usr/sbin/iptables
#-----
#Standard Policy und flush → verbiete alles
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUTDROP
$IPTABLES -F
#flush aller chains (Tabelle Filter löschen, lokale Prozesse erlauben)
$IPTABLES -t nat -F
#flush aller chains (Tabelle NAT)
$IPTABLES -X          #Löschen aller user definierten Filter
#MASQUERADING
#Maskiere ppp0 (externesInterface)

$IPTABLES -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
#anstatt ppp0 (DSLbzw.normalesModem) kann auch ipp0 (ISDN) oder eth1
# (Router mit Standleitung an eth1) eingetragen sein!!!
```

Dieses Script muss nun noch mit `chown root scriptname` root zugewiesen werden und mit `chmod 700 scriptname` ausführbar gemacht werden. Damit das Script automatisch beim Start ausgeführt wird, muss noch ein Link im entsprechenden Runlevel erzeugt werden.

```
ln -s scriptname/etc/init.d/rc3.d/S99Masqueradingstart
```

(Namensgebung `Sxxyyyyyyyyy`:

`xx`=Reihenfolge, `99yyyyyy`=frei wählbarer Name für Link)

PS:Scriptname steht Synonym für den kompletten Pfad+Dateiname z.B.: `/home/anton/machwas`

Folgendes Script unterstützt Masquerading mit speziellen Firewallregeln !):

```
#!/bin/sh
# In diesem Bsp. stellen ppp0 bzw. eth0 die externe Verbindung dar; eth1 geht zum Intranet
#
echo "setting firewall rules"
set IPTABLES = /usr/sbin/iptables

# -----
# Standard Policy und flush / verbiete alles
$IPTABLES -F INPUT DROP
$IPTABLES -F FORWARD DROP
$IPTABLES -F OUTPUT DROP

$IPTABLES -F          # flush aller chains (Tabelle, Filter löschen, lokale Prozesse erlauben)
$IPTABLES -t nat -F   # flush aller chains (Tabelle NAT)
$IPTABLES -X          # Löschen aller userdefinierten K Filter

echo "done (default policy + flush)"
# -----
# lokale Prozesse
#-----
$IPTABLES -A OUTPUT -o ppp0 -j ACCEPT    #für Standardrouter anstatt ppp0   eth0
$IPTABLES -A OUTPUT -o lo -j ACCEPT      # Akzeptiert alle ausgehenden Pakete von lo
                                           # (interne Prozesse)

$IPTABLES -A INPUT -i lo -j ACCEPT

# Neue Kette mit dem Namen block erstellen (-N), die neue Verbindungen blockt,
# es sei denn, sie kommen von innen (also nicht von eth0  !!)
$IPTABLES -N block
$IPTABLES -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A block -m state --state NEW -i !eth0 -j ACCEPT

$IPTABLES -A block -j DROP

# von INPUT und FORWARD zu block springen
$IPTABLES -A INPUT -j block
$IPTABLES -A FORWARD -j block

# MASQUERADING
# Maskiere ppp0 bzw. eth0 (externes Interface)
$IPTABLES -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

PS:

Dieses Script sollte nur alle von „innen“ (hier eth1) initiierten Verbindungen aufbauen. Von außen kommende Pakete sollten nur durchgelassen werden, wenn sie einen Bezug zu einer bereits bestehenden Verbindung haben. Lokale Prozesse sollten immer über die OUTPUT-Kette Verbindungen aufbauen dürfen.

Ergänzungen falls einige Server, wie z.B. www.otto.de oder www.tschibo.de nicht erreichbar sind:

In den Dateien:

```
/etc/ppp/options  
/etc/ppp/peers/pppoe
```

die Werte für

```
mtu 1492  
mru 1492
```

(kein „=“ Zeichen !!)

setzen !!

In beiden Dateien gleich !!

In beiden Dateien auf 1492 !!! Kein anderer Wert !!

Dann rcsmpppd restart und www.otto.de und www.tschibo.de sind erreichbar !

Wenn die MTU und MRU Modifikation nicht hilft:

(Kopie aus der SuSE-Supportdatenbank: http://sdb.suse.de/de/sdb/html/cg_pmtu2.html)

T-DSL in Zusammenhang mit Routern

Sie verwenden einen zentralen Router der für mehrere Rechner das Masquerading übernimmt. Die Clients können bestimmte Rechner im Internet nicht erreichen. Lokal auf dem Router haben Sie keine Probleme. Als ersten sollten Sie auf einer der Clientmaschinen testen ob wirklich PMTU discovery das Problem bei Ihnen verursacht. Hierzu können Sie an der Kommandozeile das ifconfig Kommando verwenden um die verwendete Paketgröße einzustellen. Beachten Sie daß Sie auf dem Router die Netzwerkkarte an der das T-DSL Modem direkt angeschlossen ist nicht in der MTU verkleinern dürfen.

```
tux@erde:~ $ /sbin/ifconfig  
eth0  Protokoll:Ethernet  Hardware Adresse 00:10:10:00:01:A4  
      inet Adresse:10.10.11.102  Bcast:10.10.255.255  Maske:255.255.0.0  
      UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1  
      RX packets:93589710 errors:0 dropped:0 overruns:0 frame:0  
      TX packets:14879178 errors:0 dropped:0 overruns:0 carrier:0  
      Kollisionen:0 Sendewarteschlangenlänge:100  
      RX bytes:3770027551 (3595.3 Mb)  TX bytes:2994365512 (2855.6 Mb)  
      Interrupt:11 Basisadresse:0xd000
```

Oben fett hervorgehen sehen Sie die Voreinstellung der MTU von 1500 Byte. Probieren Sie nun die MTU zu verkleinern (root rechte erforderlich).

```
root@erde:~ # /sbin/ifconfig eth0 mtu 1400
```

Sollte Sie nun die problematischen Server erreichen können so können Sie für diesen Fall folgende Veränderungen vornehmen:

Möglichkeit 1:

Mit iptables

Wenn Sie iptables verwenden genügt es wenn Sie lediglich den T-DSL Router umkonfigurieren. Eine explizite Konfiguration der Clientmaschinen ist unnötig. Um iptables verwenden zu können muß auf Ihrem Router ein Linux Kernel Version 2.4 installiert sein.

Geben Sie auf dem Router als Benutzer root folgendes Kommando ein:

```
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Dieses Kommando bewirkt daß die "MSS" Option die beim Verbindungsaufbau zwischen dem Internet-Server und einer Clientmaschine Ihres Netzwerkes ausgetauscht wird vom Router "umgeschrieben" wird.

Möglichkeit 2:

Umkonfiguration der Clientmaschinen

Falls Sie auf den Router keinen administrativen Zugriff haben können Sie auch die Clientmaschinen umkonfigurieren und die MSS für eine bestimmte Route festlegen.

1. Verkleinern Sie die angebotene Segmentgröße mit der MSS Option des route Kommandos.

Dazu können Sie z.B. die Datei /etc/route.conf wie folgt verändern:

```
Steht bei Ihnen z.B.  default    10.10.0.8    0.0.0.0  eth0  
                    setzen Sie ans Zeilenende einfach  
                    default    10.10.0.8    0.0.0.0  eth0 mss 1400
```

um zum Beispiel die Segmentgröße auf 1400 Byte zu setzen (Normal: 1448 Byte). Vorteil: LAN Verbindungen erhalten die volle Paketgröße. Nur TCP Verbindungen zu entfernten Rechnern bekommen kleinere Paketgrößen vorgeschlagen. Nachteil: Bei falschen lokalem Setup gibt es evtl. trotzdem Problem (z.B. eine lokale, falsch konfigurierte Firewall). Funktioniert nicht für UDP (z.B. Realaudio, Netmeeting, Real-Video verwenden UDP).

2. Verkleinern Sie die MTU der Netzwerkkarte. Dies können Sie z.B. in /etc/rc.config erledigen.
Suchen Sie nach der Konfiguration des Netzwerkkinterfaces in dieser Datei, die beispielsweise so aussehen kann:

```
IFCONFIG_0="10.10.11.102 broadcast 10.10.255.255 netmask 255.255.0.0"  
und setzen Sie  
IFCONFIG_0="10.10.11.102 broadcast 10.10.255.255 netmask 255.255.0.0 mtu 1400"
```

ein um die MTU auf 1400 Byte zu begrenzen.

Vorteil: Funktioniert sehr zuverlässig da auch die MSS Option richtig vorgeschlagen wird. Funktioniert auch für UDP Pakete.

Nachteil: Alle Pakete werden kleiner. Damit wird der Overhead durch Header größer und damit der Durchsatz im lokalen Netz ein wenig geringer. Ist eher die "Holzhammermethode".

1. Eine Variation der 1. Methode: Setzen Sie eine Route auf den/die "problematischen" Rechner den Sie nicht erreichen können mit einer verringerten Segmentgröße. Um zum Beispiel eine Route auf den Web-Server (213.95.15.200) der SuSE Linux AG mit einer MSS von 1400 Byte zu setzen nehmen Sie in /etc/route.conf folgende Zeile auf.

```
213.95.15.200 10.10.0.8 255.255.255.255 eth0 mss 1400
```

oder -temporär - an der Kommandozeile (root rechte erforderlich!)

```
route add -host 213.95.15.200 gw 10.10.0.8 eth0 mss 1400
```

Hintergrund

Das "Hängenbleiben" der Verbindung ist nichts T-DSL spezifisches, sondern durch die kleinere zulässige MTU von T-DSL (durch PPPOE) bedingt. Die zulässige MTU kann auch verkleinert sein wenn Sie manche kommerziellen Einwahlrouter oder Windows als Router verwenden und andere Ethernet Frames statt normalen Ethernet II Frames im LAN verwenden. Ebengleiches gilt für IP-over-IP oder CIPE Tunnel. Die Netzwerkpakete die über diese Verbindungen geschickt werden müssen also ein bißchen kleiner sein als überall sonst üblich. Kommt ein zu großes Paket an, so würde normalerweise passieren: Beim Eintritt des Paketes in die Strecke mit kleinerer MTU wird vom Router der die Netzwerkschnittstelle mit der kleineren MTU hat, transparent fragmentiert. Das Netzwerkpaket wird von diesem Router in mehrere, kleinere IP Pakete zerlegt die dann klein genug sind. Der Zielrechner setzt diese IP Fragmente wieder zusammen. Diese transparente Fragmentierung ist eine enorme Belastung für die Backbone-Router des Internets die Ihre Pakete fragmentieren müssen (Im Falle von T-DSL wäre das z.B. der Router der Telekom). Um die Kosten für die Internet Infrastruktur zu senken wird bei modernen Betriebssystemen diese Belastung der Infrastruktur durch ein spezielles Flag in jedem IP Paket, das sog. "DF" (Don't fragment) Bit vermieden. Bei Linux wird dies seit Kernel Version 2.4 eingesetzt. Das ganze nennt sich Path MTU discovery und ist in RFC 1191 beschrieben. Die Router die so ein großes Packet normalerweise fragmentieren müssten senden bei gesetztem DF bit im IP Paket eine ICMP (Internet Control message protocol)-Nachricht "ICMP: destination unreachable: need to fragment" an den Absender des großen Paketes zurück. Sogar mit Vorschlag iner neuen MTU. Das ursprüngliche Paket wird vom Router verworfen. Der absendende Rechner empfängt die ICMP Nachricht und verkleinert das Paket um es dann nochmals zu verschicken. Kommt dieses ICMP Paket beim Absender nicht an, so bleibt die Verbindung hängen. Leider gibt es einige Betreiber von wichtigen Servern die davorliegende "Firewalls" falsch konfiguriert haben und diese ICMP Pakete wegwerfen. Das ist die eigentliche Ursache des Fehlers. Die Verbindung bleibt scheinbar hängen, da beide Seiten denken es wäre lediglich ein Packet verloren gegangen und eine erneute Übertragung des unveränderten Paketes probieren - die aber natürlich wieder fehlschlägt.

Glossar

Frame: Mit Frame bezeichnet man den Rahmen eines Netzwerkpaketes. Damit die an der Datenübertragung beteiligten Rechner wissen woher und wohin ein Netzwerkpaket gehen soll und was da eigentlich drin ist, werden die Daten "verpackt". Diese "Verpackung" wird als Rahmen (engl. frame) bezeichnet und besteht bei Ethernet aus Ziel- und Quelladresse und dem Typ der enthaltenen Daten. Bei manchen Typen ist auch noch die Paketlänge gespeichert. MSS: Maximum segment size. Die maximal zulässige Größe eines Segments. Dies ist eine Option einer TCP Datenverbindung die die zulässige Paketgröße beim Verbindungsaufbau festlegt. Die MSS kann für eine Route mit dem route Kommando festgelegt werden. Die MSS ist eng mit der MTU verwandt. MRU: Maximum receive unit. Diese Einstellmöglichkeit haben Sie nur für PPP Verbindungen wo die Paketgröße beim Verbindungsaufbau ausgehandelt wird. Es gilt sinngemäß das gleiche wie für die MSS und die MTU. MTU: Maximum transmit unit. Die maximal zulässige Größe eines Netzwerkpaketes in Byte. Normalerweise darf ein (Ethernet-) Netzwerkpaket maximal 1514 Byte groß sein. Daraus ergibt dann die für TCP/IP nutzbare MTU von 1500 Byte). Bei bestimmten Verfahren (Tunnel, PPPOE) wird die zulässige Größe kleiner, da ein Paket in einem anderen eingepackt wird. Bei der Verwendung von PPPOE darf ein Paket nur noch 1492 (Standard Ethernet II Frames), bzw. 1490 Byte groß sein. PPPOE PPP over ethernet (Point to point protocol over ethernet) ist eine Erfindung die es der deutschen Telekom ermöglicht die bestehende Einwahlstruktur für Modem und ISDN Verbindungen auch für das nicht verbindungsorientierte DSL verwenden zu können. Beachten Sie die Informationsseite der Telekom hierzu.