

Ralf Spenneberg

# Linux-Firewalls mit iptables & Co.

Sicherheit mit Kernel 2.4 und 2.6  
für Linux-Server und -Netzwerke



 ADDISON-WESLEY

---

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England  
Don Mills, Ontario • Sydney • Mexico City  
Madrid • Amsterdam

# Teil VI

## Transparente Firewalls







# 29 ProxyARP

Die Verwendung von ProxyARP ist eine der einfachsten Varianten für die transparente Filterung von Paketen in einem Netzwerk. Hierbei wird ein Router in ein vorhandenes Netzwerk eingebracht, ohne dass das Netzwerk neu konfiguriert wird. Auf beiden Seiten des Routers befinden sich Systeme aus demselben Netzwerk. Um dennoch eine Kommunikation zwischen den Systemen zu ermöglichen und ARP-Auflösungen durchzuführen, unterstützt der Router ProxyARP. Eine Filterung ist dann ganz normal mit Iptables möglich.

## 29.1 Wie funktioniert ProxyARP?

Bei dem ProxyARP antwortet ein Rechner auf eine ARP-Anfrage an Stelle (als Proxy) eines anderen Rechners. Eigentlich sehr einfach, aber wann braucht man das und wofür?

Eine häufige Anwendung ist die Einwahl eines Rechners in ein Netzwerk. In der Abbildung 29.1 sehen Sie ein typisches Beispiel.

Der Client wählt sich per Modem in ein Netzwerk ein und verwendet das PPP-Protokoll für die Anmeldung. Über das PPP-Protokoll erhält er eine IP-Adresse, ein Default-Gateway, einen DNS-Server und einen WINS-Server mitgeteilt. In vielen Fällen handelt es sich wie hier bei der IP-Adresse um eine IP-Adresse aus dem

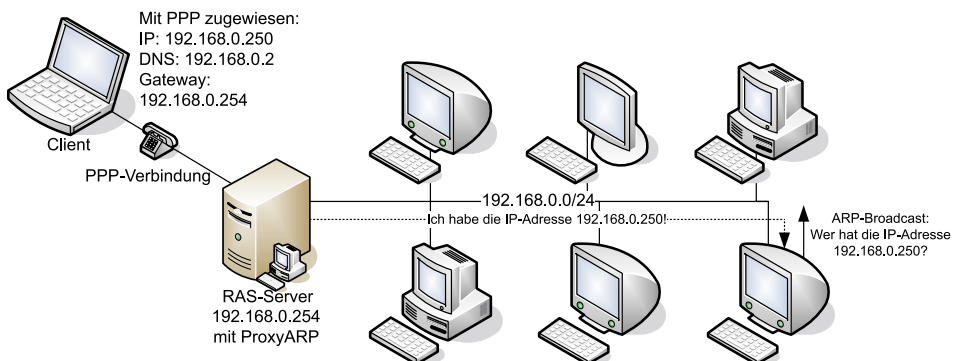


Abbildung 29.1: Bei der Einwahl erhält der Client eine IP-Adresse aus dem internen Netzwerk.

internen Netzwerk. Wenn nun der Client ein Paket an einen Rechner in dem internen Netz schicken möchte, ist dies zunächst kein Problem, da er das Paket nur an den Remote-Access-Service-(RAS-)Server schicken kann. Wenn jedoch der Rechner A aus dem internen Netz ein Paket an den Client schicken möchte, entsteht ein Problem. Der Rechner A in dem internen Netz prüft über seine Netzmaske und die IP-Adresse des Clients, ob sich der Client in seinem eigenen Netz befindet. Da beide Rechner sich in dem Netzwerk 192.168.0.0/24 befinden, sendet er eine ARP-Anfrage für die IP-Adresse 192.168.0.250 aus. ARP-Anfragen werden nur in dem lokalen Netz weitergeleitet. Router wie der RAS-Server arbeiten auf der Schicht (Layer) 3 und leiten diese Pakete der Schicht 2 nicht weiter! Die ARP-Anfrage kann nie den Client erreichen, und der Client kann diese ARP-Anfrage nie beantworten. Der Rechner A erhält keine Antwort auf seine ARP-Anfrage und gibt eine Fehlermeldung aus.

ProxyARP behebt dieses Problem. Hierbei wird auf dem RAS-Server ProxyARP angeschaltet. Dann beantwortet der RAS-Server die ARP-Anfrage für den Client. Der Rechner A sendet das Paket an den RAS-Server, der nach Betrachtung der Ziel-IP-Adresse das Paket weiter an den Client sendet. Die Verbindung kommt zustande.

Mit dieser Methode können Sie an beliebigen Stellen in einem Netzwerk ohne eine Änderung der Konfiguration (IP-Adressen, Netzmaske und Gateway) Router einfügen. Diese Router können anschließend auch die gerouteten Pakete mit Iptables filtern. Im Folgenden erkläre ich zunächst die Konfiguration des ProxyARP und anschließend die Filterung der Pakete.

## 29.2 ProxyARP-Konfiguration

Die Konfiguration des ProxyARP beschränkt sich auf die aktuellen Kernel 2.4 und 2.6. In den älteren Kernen wurde die ProxyARP-Funktionalität unterschiedlich gehandhabt (<http://www.tldp.org/HOWTO/Proxy-ARP-Subnet/>). Teilweise wird die ProxyARP-Funktionalität auch je nach Hardware unterschiedlich gehandhabt. So verwendet Linux auf der Z-Series von IBM ganz andere Dateien zur Steuerung der ProxyARP-Funktion (<http://www-1.ibm.com/servers/eserver/zseries/library/techpapers/pdf/linux-14mg.pdf>).

Die Konfiguration des ProxyARP unter Linux ist sehr einfach. Beginnen Sie damit, die Netzwerkkarten in Ihrem Router zu konfigurieren. Hierbei ist es durchaus möglich, dass beide Netzwerkkarten identische IP-Adressen erhalten. Sie können aber auch die Netzwerkkarten mit unterschiedlichen IP-Adressen ausstatten:

```
# ip addr ip add 192.168.0.2 dev eth0
# ip addr ip add 192.168.0.2 dev eth1
# ip link set eth0 up
# ip link set eth1 up
```

Nun müssen Sie in der Routing-Tabelle die entsprechenden Routen eintragen. Wenn sich auf der einen Seite des ProxyARP-Routers nur ein Rechner befindet und auf der

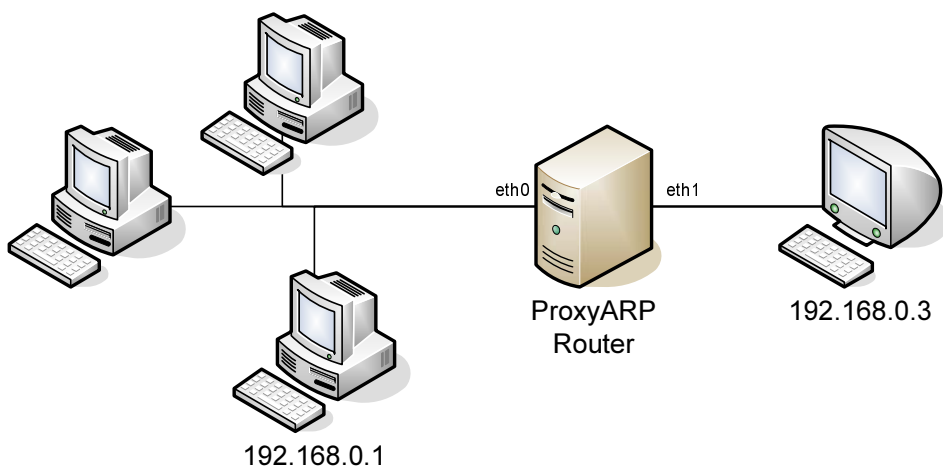


Abbildung 29.2: Ein einfaches Beispiel, in dem sich links und rechts von einem ProxyARP-Router zwei Rechner 192.168.0.1 und 192.168.0.3 befinden.

anderen Seite der Rest des Netzwerks (siehe Abbildung 29.2), dann können Sie die Routen folgendermaßen setzen:

```
# ip route add 192.168.0.3/32 dev eth1
# ip route add 192.168.0.0/24 dev eth0
```

Nun müssen Sie lediglich für jede Netzwerkkarte, die ProxyARP unterstützen soll, dies zunächst anschalten. Dies erfolgt in dem `/proc`-Verzeichnis. Dort existiert für jede Netzwerkkarte ein Eintrag `/proc/sys/net/ipv4/conf/ethX/proxy_arp`. Tragen Sie hier eine Eins ein, um die Funktion anzuschalten:

```
# echo "1" > /proc/sys/net/ipv4/conf/eth0/proxy_arp
# echo "1" > /proc/sys/net/ipv4/conf/eth1/proxy_arp
```

### Tipp



Wenn Sie eine Option für alle Netzwerkkarten aktivieren möchten, genügt es auch, die Option in `/proc/sys/net/ipv4/conf/all` anzuschalten.

Anschließend müssen Sie noch die IP-Weiterleitung aktivieren und eine Kommunikation sollte nun zwischen 192.168.0.1 und 192.168.0.3 möglich sein.

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

ProxyARP ist eine sehr einfache Möglichkeit, um in einem Netzwerk nachträglich ohne Änderung der Konfiguration einen Router einzuführen. Ganz transparent ist jedoch dieser Router nicht. Er wird, wie jeder andere Router, den TTL-Wert der weitergeleiteten Pakete um eins heruntersetzen. Ansonsten ist aber auf der Schicht 3 (IP) keine weitere Modifikation der Pakete erkennbar.

## 29.3 Filterung mit Iptables

Die Filterung mit Iptables ist beim Einsatz von ProxyARP sehr einfach. Da es sich beim ProxyARP-System um einen Router handelt, können Sie ganz normal die Pakete filtern. Die Pakete durchlaufen, wie auf einem normalen Router, die Ketten PREROUTING (Tabellen Mangle und NAT), FORWARD (Tabellen Mangle und Filter) und POSTROUTING (Tabellen Mangle und NAT). Sie müssen nur aufpassen, wenn Sie in Ihren Regeln IP-Adressen angeben, dass Sie darauf achten, dass auf beiden Seiten der Firewall IP-Adressen aus demselben Netzwerk vorhanden sind.

Handelt es sich beim Rechner 192.168.0.3 aus Abbildung 29.2 um einen MySQL-Datenbankserver, den Sie zusätzlich mit einer Firewall vor dem lokalen Netz 192.168.0.0/24 schützen möchten, dann könnten Sie die folgenden Regeln verwenden:

```
MYSQL_SRV=192.168.0.3
DIENSTE="mysql,ssh"
LANDEV=eth0 # An dieser Schnittstelle ist das restliche LAN angeschlossen
MYSQLDEV=eth1 # Hier ist der MySQL-Server angeschlossen
```

```
$IPTABLES -P FORWARD DROP
```

```
$IPTABLES -A FORWARD -i $LANDEV -o $MYSQLDEV -d $MYSQL_SRV
-p tcp -m multiport --dport $DIENSTE -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

## 29.4 Fazit

So können Sie mit einfachen Mitteln auch den Verkehr in einem Netz ohne Änderungen der Adressen, Netzmasken und Default-Gateways filtern.

Um jedoch tatsächlich die Firewall ohne IP-Adressen zu betreiben, benötigen Sie den Bridge-Modus. Der nächste Abschnitt erläutert diesen und die Anwendung von Iptables.