

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



34 ICMP

Ohne ICMP würde das Internet nicht funktionieren. Häufig ist dieses Protokoll auch für Funktionsstörungen verantwortlich. Teilweise hängt dies mit falsch konfigurierten Firewalls zusammen. Dieses Kapitel zeigt Ihnen, wie Sie das Protokoll richtig filtern.

34.1 ICMP

Die Filterung des ICMP-Protokolls (RFC 792) ist ein recht kompliziertes und umfangreiches Thema, da es recht viele wichtige ICMP-Nachrichten gibt. Das Internet Control and Message Protocol ist in erster Linie für die Übertragung von Fehlermeldungen verantwortlich. Außerdem wird dieses Protokoll von dem `ping`-Befehl und dem `traceroute`-Befehl eingesetzt.

Erfreulicherweise unterstützt die Stateful-Inspection des Linux-Kernels auch das ICMP-Protokoll. Das bedeutet, dass der Kernel erkennen kann, ob eine ICMP-Fehlermeldung sich auf eine existente Verbindung bezieht oder nicht. So ist es möglich, ICMP-Fehlermeldungen nur zuzulassen, wenn sie sich tatsächlich auf eine aufgebaute Verbindung beziehen, und jede andere Form direkt zu verwerfen. Die Fehlermeldungen, die sich auf existente Verbindungen beziehen, werden von dem Linux-Kernel als `RELATED` eingeordnet. Wenn Sie `RELATED`-Pakete zulassen, dann werden auch diese Pakete akzeptiert. Dabei löscht dann der Linux-Kernel auch die Verbindung, auf die sich die Fehlermeldung bezog, aus der Verbindungsliste, denn es trat ja ein Fehler auf.

Eine derart allgemeine Betrachtung ist jedoch in einigen Fällen nicht ausreichend. In Abhängigkeit der Firewall-Richtlinien müssen die ICMP-Nachrichten einzeln betrachtet werden. Das wird im Folgenden gezeigt. Dazu betrachten wir zunächst die einzelnen Fehlernachrichten, um uns dann anschließend einzelne Befehle näher anzusehen, die diese Funktionen nutzen (`ping` und `traceroute`).

Im Einzelnen betrachten wir die folgenden ICMP-Nachrichten:

- `destination-unreachable`
- `destination-unreachable: fragmentation-needed`
- `source-quench`
- `redirect`
- `router-advertisement`

- router-solicitation
- time-exceeded
- parameter-problem
- timestamp-request/timestamp-reply
- address-mask-request/address-mask-reply
- echo-request/echo-reply

Ein grundsätzliches Filtern der Meldungen ist sehr einfach. Wenn Sie einfach alle gültigen ICMP-Meldungen akzeptieren möchten, nutzen Sie einfach folgende Regel:

```
$IPTABLES -A FORWARD -p icmp -m state --state RELATED -j ACCEPT
```



Achtung

Denken Sie daran, dass Sie vielleicht sowieso schon eine Regel in Ihrem Skript haben, die alle `RELATED`-Pakete erlaubt!

Wenn Sie nur bestimmte Meldungen akzeptieren möchten, können Sie folgende Regel verwenden:

```
$IPTABLES -A FORWARD -p icmp --icmp-type destination-unreachable -m state --state RELATED -j ACCEPT
```

Was nun wo und in welcher Richtung Sinn macht, erläutern die nächsten Abschnitte.

34.2 ICMP destination-unreachable

Die ICMP-Fehlermeldung `destination-unreachable` wird immer dann gesendet, wenn ein Ziel nicht erreichbar ist. Dabei kann es mehrere Gründe für diese Meldung geben. Im Folgenden sehen Sie eine Auswahl der häufigsten Gründe:

- Der angesprochene UDP-Port wird von keiner Applikation bedient.
- Der angesprochene Rechner kennt das IP-Protokoll nicht.
- Der angesprochene Rechner reagiert nicht auf eine ARP-Anfrage. Er ist wahrscheinlich ausgeschaltet.
- Das Netzwerk, in dem sich der Zielrechner befindet, ist nicht erreichbar.
- Der Rechner oder das Netzwerk werden durch eine Firewall geschützt. Einige Firewalls senden dann ein `destination-unreachable: network-prohibited`.

- Das Paket ist zu groß, um in das nächste Netzwerk geroutet zu werden. Eine Fragmentierung ist erforderlich, aber nicht erlaubt. Nähere Informationen über diese Meldung finden Sie im nächsten Abschnitt.

Die meisten dieser Fehlermeldungen weisen auf fatale Fehler bei der Verbindung hin. Ein Client, der diese Meldungen erhält, erkennt den Fehler und bricht die Verbindungsversuche ab. Lediglich bei der Meldung `fragmentation-needed` wird ein erneuter Versuch mit einem kleineren Paket unternommen. Für eine Firewall bedeuten alle diese Fehlermeldungen mit Ausnahme der `fragmentation-needed`-Meldung, dass die Verbindung nicht aufrechterhalten werden kann. Die Firewall kann daher die Verbindung aus ihrer Zustandstabelle löschen.

Ein Filtern der ICMP-Meldungen ist sehr einfach möglich. Bereits im letzten Abschnitt habe ich die folgende Regel vorgestellt:

```
$IPTABLES -A FORWARD -p icmp --icmp-type destination-unreachable -m state --state RELATED
-j ACCEPT
```

Diese Regel akzeptiert alle ICMP-Fehlermeldungen vom Typ `destination-unreachable` unabhängig von ihrer Richtung und den IP-Adressen. Es ist lediglich erforderlich, dass die Firewall die Fehlermeldung einer Verbindung zuordnen kann.

Jedoch können diese Pakete auch Informationen über interne Systeme preisgeben. Zum einen kann ein Angreifer erkennen, ob ein System existiert, Protokolle erkannt werden und die entsprechenden Ports offen sind.

Hinweis



So kann ein Angreifer zum Beispiel durch Senden eines ESP- oder AH-Pakets herausfinden, ob das System ein VPN-Gateway ist oder nicht. Normale Systeme antworten mit einer `protocol-unreachable`-Meldung. Ein VPN-Gateway versucht das Paket zu analysieren und zu verarbeiten.

Teilweise geben die Systeme in den Fehlermeldungen mehr Informationen preis, als dem Administrator lieb ist. So schickten Linux-Kernel der Version 2.0 beliebige Speicherinhalte in den Fehlermeldungen: <http://www.cartel-securite.fr/pbiondi/adv/CARTSA-20030314-icmpleak.txt>. Diese Lücke ist behoben. Jedoch sind mindestens einige Windows-Systeme von ähnlichen Problemen betroffen.

Daher ist es sinnvoll, die Anzahl der Meldungen zu begrenzen. Wenn Sie eine Firewall implementieren, um ein internes Netz beim Zugriff auf das Internet zu schützen, so sollten Sie ICMP-`destination-unreachable`-Meldungen aus dem Internet in Ihr geschütztes Netz erlauben. Tun Sie dies nicht, werden Ihre Clients beim Zugriff auf das Internet nicht über Fehler beim Verbindungsaufbau informiert und werden es mehrfach erneut versuchen, bis ein client-interner Timeout auftritt. Ihre Benutzer werden über diese Verzögerung nicht begeistert sein. Wenn Sie die Fehlermeldun-

gen zulassen, erhält der Client sofort eine entsprechende Nachricht und kann die Information sofort an den Benutzer oder die Applikation weitergeben.

Den Transport von ICMP-destination-unreachable-Meldungen aus Ihrem internen Netz in das Internet ist mit einer Ausnahme (siehe den nächsten Abschnitt 34.3) nicht erforderlich.

Sie können also folgende Regel verwenden:

```
$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -p icmp --icmp-type destination-unreachable
-m state --state RELATED -j ACCEPT
```

Natürlich ist der Transport von ICMP-destination-unreachable-Nachrichten aus der DMZ unter Umständen sinnvoll. Das hängt davon ab, ob Sie den Clients, die auf Ihre Dienste zugreifen, Fehlermeldungen senden möchten oder nicht. Sie können mit der folgenden Regel die destination-unreachable-Meldungen zulassen:

```
$IPTABLES -A FORWARD -i $DMZDEV -o $EXTDEV -p icmp --icmp-type destination-unreachable
-m state --state RELATED -j ACCEPT
```

34.3 ICMP fragmentation-needed

Diese ICMP-Meldung (beziehungsweise ihre falsche Filterung) ist für eine Vielzahl von Problemen im Internet verantwortlich. Schuld ist die Tatsache, dass unterschiedliche Netzwerkmedien unterschiedliche Paketgrößen transportieren können. Die maximale Größe ist die Maximum Transmission Unit (MTU). Befindet sich nun ein Client in einem Netz A mit einer besonders großen MTU und versendet ein Paket an einen Empfänger in einem Netz B mit einer kleineren MTU, so besteht die Gefahr, dass das Paket zu groß für das Netz B ist. Entweder wird das Paket fragmentiert oder es wird verworfen und diese Fehlermeldung erzeugt. Da eine Fragmentierung bei einem Paketverlust mit einem Overhead verbunden ist, führen fast alle modernen Betriebssysteme eine Path Maximum Transmission Unit Discovery (RFC 1191) durch. Sie ermitteln die MTU für den kompletten Pfad und bauen ihre Pakete dann in passender Größe. Die PMTUD ist aber auf diese Fehlermeldung angewiesen. Erreicht die Fehlermeldung nicht ihr Ziel, kann die Verbindung nicht aufgebaut werden. Dummerweise kann der betroffene Client kaum etwas an dieser Tatsache ändern (siehe auch Abschnitt 16.31). Eine genauere technische Betrachtung finden Sie in dem Exkurs MTU und PMTUD (siehe folgenden Exkurs).

Exkurs: MTU und die Path-MTU-Discovery



Da die MTU für unterschiedliche Netzwerkmedien unterschiedliche Werte haben kann, kann es vorkommen, dass ein Rechner so große Pakete erzeugt, dass ein Router auf dem Weg zum Ziel diese Pakete nicht weiterleiten kann. In der Abbildung 34.1 ist ein Token-Ring-Netz und ein Ethernet dargestellt. Die MTU des Token-Ring (16 MBit/s) beträgt 17914 Bytes. Die MTU des Ethernet beträgt

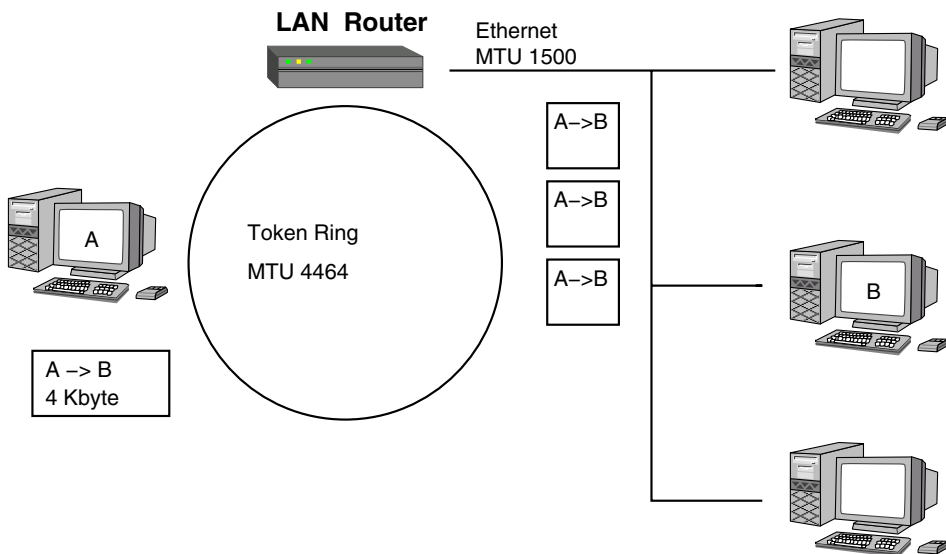


Abbildung 34.1: Im Paket-Switched Network können die Pakete über unterschiedliche Wege das Ziel finden.

1500 Bytes. Sendet nun der Rechner Eins an den Rechner Zwei 20 Kbyte Daten, so erzeugt er zwei Pakete. Das erste Paket ist etwa 17 Kbyte groß und das zweite etwa 3 Kbyte. Der Router kann diese Pakete jedoch nicht unverändert weitersenden, da die MTU des Ethernet nur 1500 Bytes beträgt. Er fragmentiert die Pakete. Dazu schneidet er zunächst das Paket in kleinere Bestandteile und kopiert jeweils den original IP-Header des Pakets davor. Die Länge dieser Bestandteile muss ein Vielfaches von acht aufweisen. Zusätzlich setzt der Router in allen Fragmenten außer dem letzten das More-Fragments-Bit (MF). Dieses zeigt dem Empfänger an, dass weitere Fragmente folgen. Damit der Empfänger die Fragmente richtig zusammensetzen kann, gibt er noch den Offset des Fragments im Gesamtpaket als Vielfaches von acht an. Die Fragmente werden nun weiter zum Empfänger gesendet, der die Fragmente zusammenbaut und das IP-Paket prozessiert.

Wenn nun aber eines dieser Fragmente verloren geht, wartet der Empfänger vergebens auf dieses Fragment. Nach einem Timeout sendet er eine ICMP-time-exceeded-Fehlermeldung (siehe Abschnitt 34.6). Der Absender sendet das Paket erneut, es wird erneut fragmentiert, und es besteht wieder die Gefahr, dass ein Fragment verloren geht. Das einzelne Fragment kann nicht neu angefordert werden. Der Router hat das Fragment nicht gespeichert, und der Absender weiß nichts von einer Fragmentierung. Bei einem Verlust von 1500 Bytes werden ca. 17.000 Bytes neu übertragen!

Die Path-MTU-Discovery versucht, dieses Problem intelligent zu lösen. Hierzu sendet ein Client, bei dem PMTUD aktiviert ist, alle Pakete mit einem gesetzten Don't-Fragment-Bit (DF). Dieses Bit weist alle Router an, das Paket nicht zu fragmentieren. In unserem Fall kann das Paket jedoch nicht weiter gesendet werden. Der Router verwirft das Paket und sendet eine Fehlermeldung an den Absender. In dieser Fehlermeldung informiert der Router den Absender auch über die MTU des nächsten Hops. Der Absender kann das Paket neu mit einer passenderen Größe bauen. Dieses Paket kann dann weitertransportiert werden. Befindet sich anschließend ein weiteres Netz mit einer noch kleineren MTU auf dem Weg zum Ziel, erfolgt dieser Vorgang erneut.

Sicherlich setzt kaum noch jemand heutzutage Token-Ring-Netze ein. Da fast alle Netze auf Ethernet basieren, könnte man annehmen, dass sich dieses Problem überlebt hat. Jedoch taucht es verstärkt seit der Einführung von DSL-Internetzugängen wieder auf. Die DSL-Anbindung des Rechners an das Modem erfolgt über eine Ethernet-Leitung. Dennoch beträgt die MTU für das IP-Protokoll nicht 1500 Bytes, da das IP-Protokoll noch in ein weiteres Protokoll eingepackt wird. In Deutschland handelt es sich meist um PPPOE. In anderen Ländern wird auch PPTP eingesetzt. Dadurch verringert sich die MTU für das IP-Protokoll um mindestens acht Bytes (PPPOE) auf 1492 Bytes.

In Kombination mit einer Firewall können nun zwei Probleme auftreten:

1. Die Firewall kann fragmentierte Pakete nicht richtig filtern.
2. Die Firewall erlaubt nicht die `fragmentation-needed`-Nachricht.

1. Wenn ein Paketfilter fragmentierte Pakete filtern soll, taucht ein besonderes Problem auf. Die Filterung des ersten Fragments eines HTTP-Pakets ist kein Problem, denn dieses Fragment weist sowohl einen IP- als auch einen TCP-Header auf. Ab dem zweiten Fragment enthält das Fragment aber nur noch einen IP-Header, da der Router lediglich den IP-Header kopiert. Schließlich arbeitet ein Router auch auf der Schicht Drei und nicht auf der Schicht Vier. Ein klassischer Router kann mit einem TCP-Header gar nichts anfangen. Der Paketfilter kann nun ab dem zweiten Fragment nicht mehr entscheiden, ob es sich um ein HTTP- oder ein Telnet-Paket handelt. Soll er das Paket akzeptieren oder verwerfen?

Die ersten Paketfilter hatten eine einfache Lösung parat: Sie filterten das erste Fragment und ließen automatisch alle Fragmente außer dem ersten durch! Erhielt der Empfänger auch das erste

Fragment, so konnte er das Paket defragmentieren und auswerten. Wurde das erste Fragment durch eine Firewall verworfen, konnte der Empfänger das Paket nicht defragmentieren und musste auch alle weiteren Fragmente verwerfen. Leider gab es in der Vergangenheit mehrfach Angriffe (z.B. <http://cert.uni-stuttgart.de/archive/bugtraq/1999/07/msg00232.html>), die diese einfache Lösung umgingen. Daher sammeln heute alle Paketfilter bereits die Fragmente, defragmentieren sie und analysieren erst anschließend das Paket mit ihren Regeln. Wird das Paket akzeptiert, so wird es weitertransportiert und möglicherweise hierzu erneut fragmentiert. Wird das Paket verworfen, so werden auch alle Fragmente verworfen.

2. Bei dem zweiten Problem handelt es sich nicht um ein Sicherheitsproblem, sondern vielmehr um eine Art von Denial-of-Service. In Abbildung 34.2 sehen Sie eine typische Konstellation, wie sie heute im Internet anzutreffen ist.

Der DSL-Anwender greift auf einen Webserver im Internet zu. Dieser Webserver wird durch eine Firewall geschützt, die grundsätzlich keine ICMP-Meldungen von außen erlaubt. Dies ist nicht ungewöhnlich, da bis vor einigen Jahren die Paketfilter kaum in der Lage waren, ICMP-Meldungen sicher zu filtern. Die Firewall-1 des Marktführers CheckPoint konnte in der Version 4 dies von Haus aus nicht.

Solange der Webserver nur kleine Datenmengen versendet, tritt kein Problem auf. Sobald der Webserver aber mehr als 1492 Byte große Pakete erzeugt, werden diese von dem DSL-Router auf der Seite des Internet-Service-Providers als zu groß abgelehnt. Dieser Router erzeugt eine `ICMP-fragmentation-needed`-Nachricht und sendet sie an den Webserver. Die Firewall verhindert, dass diese Nachricht den Webserver erreicht. Der Webserver erhält keine Empfangsbestätigung und sendet das Paket nach einiger Zeit erneut. Da er den Grund für den Paketverlust nicht kennt, sendet er das Paket unverändert neu. Dieses neue Paket wird erneut verworfen, und eine Fehlermeldung wird generiert. Nach einiger Zeit wird der Webserver aufgeben und die Verbindung abbrechen. Die Daten werden den Client nicht erreichen.

Dummerweise ist der Client davon betroffen und kann kaum etwas zur Behebung des Problems tun. Als einzige Lösung im Fall des TCP-Protokolls kann er die Maximum Segment Size (MSS) setzen (siehe Abschnitt 16.31).

Die Fehlermeldung `fragmentation-needed` sollten Sie grundsätzlich in allen Richtungen durch Ihre Firewall erlauben. Ansonsten kann es bei der heutigen IT-Infrastruktur

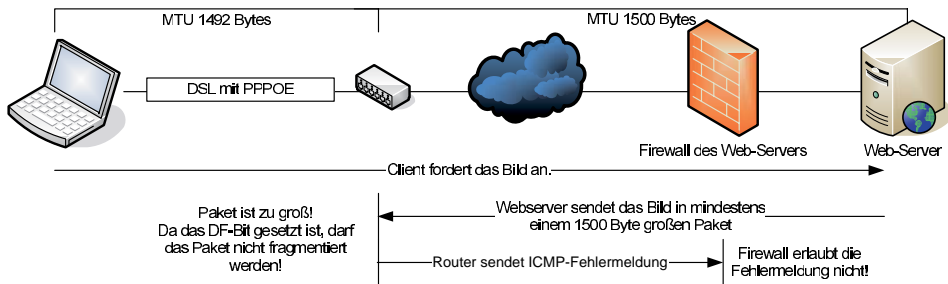


Abbildung 34.2: Ein DSL-Anwender greift auf einen Webserver zu. Der Webserver wird durch eine Firewall geschützt. Die PMTUD funktioniert auf Grund der Firewall-Konfiguration nicht.

mit DSL und VPNs zu Verbindungsproblemen kommen, die Sie und Ihre Kunden betreffen. Am einfachsten erfolgt das mit der folgenden Zeile:

```
$IPTABLES -A FORWARD -p icmp --icmp-type fragmentation-needed -m state --state RELATED,ESTABLISHED -j ACCEPT
```

34.4 ICMP source-quench

Diese Meldung kann ein Router oder ein Rechner versenden, wenn er die empfangenen Pakete nicht schnell genug verarbeiten kann und daher einzelne Pakete aus seiner Empfangswarteschlange unverarbeitet verwerfen muss. Da diese Pakete später erneut gesendet werden müssen und dadurch Verzögerungen auftreten, ist es sinnvoller, den Absender zu informieren, so dass die Pakete dann langsamer gesendet werden.

Heutzutage verwendet kaum noch ein Router diese Meldungen. Source-Quench ist durch moderne Algorithmen abgelöst worden. Auch die meisten TCP-Stacks verfügen über mächtigere Methoden der Flusskontrolle. Jedoch reagieren alle mir bekannten Betriebssysteme noch auf source-quench-Nachrichten. So ist es möglich, mit gefälschten source-quench-Nachrichten existierende Verbindungen zu verlangsamen. Dies kann bis zum Denial-of-Service führen (siehe <http://xforce.iss.net/xforce/xfdb/17429>).

Ein einfaches Werkzeug für die Erzeugung dieser Meldungen ist `tcnps` von Dug Song. Es ist Teil des `dsniff`-Pakets: <http://www.monkey.org/~dugsong/dsniff>.

Sie sollten diese source-quench-Nachrichten nicht durch Ihre Firewall erlauben. Es könnte sinnvoll sein, diese Nachrichten zu protokollieren, so dass Sie erkennen können, ob diese Nachrichten vielleicht doch gültig sind.

```
$IPTABLES -A FORWARD -p icmp --icmp-type source-quench -j LOG --log-prefix "Source-Quench: "
```

34.5 ICMP redirect

Die ICMP-`redirect`-Nachricht erlaubt es einem Router, einen Rechner auf einen besseren Router hinzuweisen. Wann wird eine ICMP-Redirect-Nachricht versandt (Abbildung 5.17 auf Seite 127)?

1. Rechner A sendet ein Paket an Rechner B. Da Rechner B sich in einem anderen Netz befindet, sendet Rechner A das Paket an sein Default-Gateway G1.
2. Das Gateway G1 prüft seine Routing-Tabelle und stellt fest, dass der nächste Hop für das Paket das Gateway G2 ist.
3. Wenn sich der Rechner A, das Gateway G2 und das Gateway G1 in demselben Netz befinden, sendet das Gateway G1 eine Redirect-Nachricht an den Rechner A und teilt ihm mit, dass das Gateway G2 ein besserer Router auf dem Weg zum Ziel Rechner B sei.
4. Außerdem leitet das Gateway G1 das Paket an das Gateway G2 weiter.
5. Der Rechner A trägt das neue Gateway G2 in seiner Routing-Tabelle oder seinem Routing-Cache ein und verwendet nun dieses Gateway, um den Rechner B zu erreichen.

Sicherlich möchten Sie nicht, dass irgendjemand Ihrer Firewall oder einem Rechner in Ihrer DMZ oder Ihrem internen Netzwerk von außen einen besseren Router mitteilen kann. Diese ICMP-Nachricht kann natürlich für Angriffe in Form von Router-Spoofing verwendet werden. Sie sollten diese Nachricht nicht über Ihre Firewall zulassen. Vielleicht ist es sinnvoll, diese Nachrichten zu protokollieren (wenn Sie die Protokolle auch lesen).

```
$IPTABLES -A FORWARD -p icmp --icmp-type redirect -j LOG --log-prefix "Source-Quench: "
```

34.6 ICMP time-exceeded

Diese ICMP-Nachricht zeigt eine Zeitüberschreitung beim Pakettransport an. Bei dem Transport eines Pakets kann es zu zwei Arten der Zeitüberschreitung kommen:

- Bei der Defragmentierung eines Pakets hat der Empfänger nicht alle Fragmente innerhalb seiner vorgeschriebenen Zeit erhalten. Er verwirft das Paket und schickt eine `time-exceeded: ttl-zero-during-reassembly`-Nachricht zurück.
- Bei dem Transport eines Pakets hat das Paket seine Lebensdauer überschritten. Jedes Paket trägt im IP-Header einen Time-to-Live-Wert (TTL). Jeder Router, der das Paket weiterleitet, reduziert diesen Wert um eins. Zusätzlich wird dieser Wert jede Sekunde, die das Paket auf einem Router in der Warteschlange verweilt, um eins reduziert. Dies tritt heute jedoch kaum noch auf. Erreicht der Wert 0, so verwirft der Router das Paket und sendet eine `time-exceeded: ttl-zero-during-transit`-Meldung an den Absender. Hiermit sollen Paket-Irrläufer durch fehlerhafte Routerkonfiguration vermieden werden.

Wenn Sie eine Firewall zum Schutz eines internen Netzes beim Zugriff auf das Internet einsetzen, möchten Sie wahrscheinlich diese Fehlermeldungen in Richtung Ihrer Clients zulassen, damit diese bei Fehlern in der Konfiguration eines Routers von Paketen informiert werden. Fehler bei der Defragmentierung treten heute so gut wie gar nicht mehr auf, da die meisten Betriebssysteme die Path-MTU-Discovery (siehe den Exkurs auf Seite 572) unterstützen.

```
$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -p icmp --icmp-type time-exceeded -m state \
--state RELATED -j ACCEPT
```

34.7 ICMP parameter-problem

Diese Fehlermeldung wird von einem System gesendet, wenn es auf Grund eines Fehlers in dem IP-Header ein Paket nicht verarbeiten konnte. Hierfür gibt es viele verschiedene Gründe. Die Fehlermeldung weist den Absender mit einem Zeiger auf den Grund hin. Es kann jedoch auch sein, dass eine IP-Option im Header für die Zustellung fehlte. Da die Option fehlt, kann nicht auf sie verwiesen werden. Hierfür gibt es dann die besondere Meldung `parameter-problem: required-option-missing`.

Diese Fehlermeldungen treten in modernen Netzen mit den ausgereiften IP-Stacks nur sehr selten auf, so dass ich üblicherweise empfehle, die Meldungen zu protokollieren und zu verwerfen.

```
$IPTABLES -A FORWARD -p icmp --icmp-type parameter-problem -j LOG \
--log-prefix "Parameter-Problem: "
```

34.8 ICMP router-advertisement/router-solicitation

Für das Funktionieren des IP-Protokolls ist es erforderlich, dass jeder Rechner die für ihn zuständigen Router kennt. Üblicherweise werden die Router eines Systems in Konfigurationsdateien festgelegt. Da dies aber einen gewissen manuellen Administrationsaufwand erzeugt, wurde die automatische Routererkennung entwickelt. Diese Routererkennung (Router Discovery, RFC 1256) basiert auf den Router-Advertisement- und Router-Solicitation-Nachrichten. Jeder Router sendet in regelmäßigen Abständen (üblicherweise alle 10 Minuten) ICMP-router-advertisement-Nachrichten aus. Diese Nachrichten enthalten wichtige Informationen über den Router, wie zum Beispiel seine IP-Adresse. So kann der Rechner die Informationen in seiner Routingtabelle eintragen.

Wenn nun ein Rechner neu gestartet wird, erhält er spätestens nach 10 Minuten die notwendigen Routerinformationen. Da dies häufig nicht akzeptabel ist, kann der Rechner auch eine `router-solicitation`-Nachricht senden. Diese weist alle Router in dem lokalen Netzwerk an, jetzt ihre Router-Advertisement-Nachricht zu senden.

Die `router-advertisement`-Nachricht wird an die *All-Devices*-Multicast-Adresse (224.0.0.1) und die `router-solicitation`-Nachricht wird an die *All-Routers*-Multicast-Adresse (224.0.0.2) gesendet.

Sicherlich wollen Sie derartige Nachrichten nicht durch Ihre Firewall erlauben.

34.9 ICMP timestamp-request/timestamp-reply

Mit den ICMP-`timestamp-request`- und `timestamp-reply`-Nachrichten (RFC 792) kann ein Rechner sowohl die Uhrzeit eines anderen Rechners ermitteln als auch die Wegezeit berechnen. Der Absender der `timestamp-request`-Nachricht trägt seine Uhrzeit in Millisekunden seit Mitternacht (UTC) ein (*Originator's Timestamp*). Der Empfänger trägt bei Erhalt der Nachricht seine eigene Uhrzeit in dem *Receive Timestamp*-Feld ein. Die *Transmit Timestamp* trägt er kurz vor dem Versand der `timestamp-reply`-Nachricht ein (Abbildung 34.3).

13 - Request / 14 - Reply	0	Checksum
Identifizier		Sequenznummer
Originator's Timestamp		
Receive Timestamp		
Transmit Timestamp		

Abbildung 34.3: Die Timestamp-Nachrichten

In einer gewissen Weise ähnelt dieser Mechanismus dem Ping (siehe unten), bietet aber mit der absoluten Uhrzeit noch zusätzliche Informationen. Die meisten modernen Betriebssysteme reagieren auf diese Nachrichten, obwohl nur sehr wenige Werkzeuge existieren, die diese Nachrichten erzeugen und auswerten können. Ein allgemein verfügbares Werkzeug ist `Nmap`. Weitere Werkzeuge sind `icmppush` und `sing` (<http://sourceforge.net/projects/sing/>). `Nmap` kann mit der Option `-PP` den Ping nach anderen Systemen im lokalen Netz durchführen.

```
[root@bibo ~]# nmap -PP -sP 192.168.255.0/24
```

```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-07-29 13:23 CEST
Host 192.168.255.1 appears to be up.
MAC Address: 00:13:10:1F:B7:D1 (Unknown)
Host 192.168.255.100 appears to be up.
Host inspiron-8100.opensource-training.de (192.168.255.125) appears to be up.
MAC Address: 00:20:E0:6C:72:1E (Actiontec Electronics)
Nmap finished: 256 IP addresses (3 hosts up) scanned in 2.295 seconds
```

Das Werkzeug `sing` ist noch mächtiger und erlaubt es, jede Art von ICMP-Nachricht zu erzeugen.

Sicherlich wollen Sie diese Nachrichten nicht von außen durch Ihre Firewall erlauben. Die Filterung dieser Nachrichten unterscheidet sich von den anderen ICMP-Nachrichten, da es sich hier nicht um eine Fehlermeldung handelt. Man kann fast

von einem Client und einem Server, die miteinander kommunizieren, sprechen. Daher filtert auch Iptables diese Nachrichten anders! Jeder Timestamp-Request wird als neuer Verbindungsaufbau (State: NEW) gewertet. Jeder Timestamp-Reply wird als Teil einer aufgebauten Verbindung gewertet (State: ESTABLISHED). Zusätzlich beendet jeder Timestamp-Reply auch die Verbindung und entfernt sie aus der Tabelle. Die Zuordnung der Reply-Nachrichten zu den entsprechenden Request-Nachrichten erfolgt neben der IP-Adresse über den Identifier und die Sequenznummer der Nachricht.

Möchten Sie die Verwendung dieser Nachrichten für Testzwecke oder für Nmap von innen durch Ihre Firewall nach außen zulassen, können Sie folgende Regel nutzen:

```
IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p icmp --icmp-type timestamp-request -m state \
--state NEW -j ACCEPT
```

Die Funktion dieser Regel setzt natürlich voraus, dass Sie eine allgemeine Regel für die Akzeptanz aller ESTABLISHED-Pakete verwenden.

34.10 ICMP address-mask-request/address-mask-reply

Diese Nachrichten (RFC 950) erlauben es einem Rechner, einen anderen Rechner nach seiner Netzmaske zu fragen. Ursprünglich wurde diese Funktion implementiert, um einem Rechner, der keine Informationen über sein lokales Netz besitzt, die Möglichkeit zu geben, andere Rechner in demselben Netz nach der gültigen Netzmaske zu fragen. Dazu sendet der Rechner eine Address-Mask-Request-Nachricht entweder an die Broadcast-Adresse (255.255.255.255) oder, falls er den lokalen Router kennt, an den Router. Dies ähnelt der Möglichkeit, mit Hilfe der Router-Solicitation-Nachricht den lokalen Router zu ermitteln. Normalerweise werden Address-Mask-Reply-Nachrichten nur von Routern versandt.

Auch diese Nachrichten möchten Sie normalerweise nicht durch eine Firewall in Ihr geschütztes Netz lassen. Vielleicht möchten Sie Werkzeuge wie `ping -mask <ziel>` nutzen. Dann benötigen Sie die folgende Regel:

```
IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p icmp --icmp-type address-mask-request \
-m state --state NEW -j ACCEPT
```

34.11 ICMP echo-request/echo-reply (Ping)

Diese Nachrichten werden Sie sicherlich kennen. Wenn Sie auch nicht die Nachrichten kennen, so kennen Sie zumindest den Befehl, der diese Nachrichten erzeugt: `ping`. Der Ping-Befehl erzeugt eine `echo-request`-Nachricht und sendet diese an den Zielrechner. Dieser beantwortet sie mit einer `echo-reply`-Nachricht. Da hier die Nachrichten wieder in einem scheinbaren Client-Server-Verhältnis stehen, kann Iptables die Nachrichten auch so filtern. Jedes Echo-Request-Paket wird als neue Verbindung erkannt (State: NEW) und eingetragen. Jedes Echo-Reply-Paket wird

der entsprechenden Verbindung zugeordnet (State: ESTABLISHED) und beendet die Verbindung. So können die Echo-Nachrichten sicher von einer Iptables-Firewall gefiltert werden.

Hinweis



Der `ping`-Befehl auf einem Linux-System kann auch einen Broadcast-Ping durchführen. Manche Implementierungen erlauben auch einen Multicast-Ping. Hierbei wird der Echo-Request an die Broadcast- oder Multicast-Adresse geschickt. Sämtliche Rechner, die auf diese Anfrage reagieren (Unix, Router, kein MS Windows), antworten. Da nun aber nur eine Echo-Reply-Antwort von Iptables zugelassen wird und anschließend die Verbindung aus der Zustandstabelle gelöscht wird, ist ein Broadcast-Ping nicht möglich, wenn die Echo-Request- und Echo-Reply-Pakete zustandsorientiert gefiltert werden.

Mit der folgenden Regel erlauben Sie Ihren Benutzern, die Erreichbarkeit von Rechnern im Internet mit dem `ping`-Befehl zu testen:

```
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p icmp --icmp-type echo-request -m state \
--state NEW -j ACCEPT
```

Häufig wird zusätzlich auch die Ping-Funktionalität beim Zugriff auf die DMZ gewünscht. Sie möchten aus Ihrem internen Netz und aus dem Internet prüfen, ob Ihr Webserver noch läuft.

```
$IPTABLES -A FORWARD -o $DMZDEV -p icmp --icmp-type echo-request -m state \
--state NEW -j ACCEPT
```

Hier habe ich auf die Angabe der eingehenden Schnittstelle verzichtet. So wird nun jedes Echo-Request-Paket, das in die DMZ geroutet werden soll, akzeptiert. Dies erfolgt unabhängig davon, ob das Paket von innen oder von außen die Firewall erreicht hat.

Achtung



Gerade bei Ping ist die zustandsorientierte Filterung wichtig. Sie könnten auf die Idee kommen, grundsätzlich jedes Echo-Reply-Paket zu akzeptieren, um so auch Broadcast-Pings zu ermöglichen. Dann öffnen Sie aber möglichen Tunnel-Werkzeugen Tür und Tor. Es existieren eine Reihe von Werkzeugen, die auf der Basis von Echo-Paketen jedes beliebige Protokoll tunneln können. Eines der ersten Werkzeuge war `itunnel` von dem Hacker-Team Teso. Die zustandsorientierte Filterung würde diesen Tunnel nicht erlauben. Der Ping Tunnel (`ptunnel`, <http://www.cs.uit.no/~daniels/PingTunnel/>)

ist sicherlich das mächtigste aktuell verfügbare Werkzeug. Es kann sogar einen Tunnel trotz Anwendung der Zustandsüberwachung aufbauen und betreiben!

34.12 Traceroute

Traceroute ist keine ICMP-Nachricht. Dennoch habe ich den Befehl hier aufgenommen, da die Antworten der Traceroute-Funktion komplett auf ICMP-Nachrichten basieren. Vielleicht haben Sie sich schon immer gefragt, wie der Traceroute-Befehl funktioniert. Die folgende Ausgabe kennen Sie sicherlich:

```
[spenneb@bibo ~]$ traceroute www.bsd-training.de
traceroute to www.bsd-training.de (81.169.129.12), 30 hops max, 38 byte packets
 1 192.168.255.1 (192.168.255.1) 0.846 ms 0.764 ms 0.711 ms
 2 217.0.116.135 (217.0.116.135) 77.835 ms 46.357 ms 47.983 ms
 3 217.0.73.82 (217.0.73.82) 47.190 ms 46.582 ms 48.130 ms
 4 l-ea1.L.DE.net.DTAG.DE (62.154.89.122) 57.817 ms 57.110 ms 57.894 ms
 5 so-1-1-1-0.lpz2-j2.mcbone.net (62.104.199.89) 59.129 ms 58.560 ms 171.903 ms
 6 LO.blm2-g.mcbone.net (62.104.191.139) 61.900 ms 62.711 ms 60.838 ms
 7 strato-blm1.fdknet.de (194.97.172.146) 60.145 ms 59.459 ms 65.330 ms
 8 81.169.160.38 (81.169.160.38) 57.699 ms 58.237 ms 58.084 ms
 9 81.169.160.158 (81.169.160.158) 57.951 ms 57.246 ms 58.089 ms
10 bsd-training.de (81.169.129.12) 59.158 ms 59.379 ms 58.027 ms
```

Der Traceroute-Befehl ermittelt die Router, über die ein Paket sein Ziel erreicht. Dabei geht das nicht immer so gut wie in diesem Fall. Häufig werden ab einem bestimmten Zeitpunkt Sternchen (* * *) angezeigt. Dann filtert eine Firewall den Verkehr:

```
[spenneb@bibo ~]$ traceroute www.redhat.de
traceroute to www.redhat.de (66.187.229.16), 30 hops max, 38 byte packets
 1 192.168.255.1 (192.168.255.1) 0.871 ms 4.139 ms 0.940 ms
 2 217.0.116.135 (217.0.116.135) 49.293 ms 79.062 ms 47.944 ms
 3 217.0.73.82 (217.0.73.82) 46.832 ms 46.187 ms 48.056 ms
 4 f-ea2.F.DE.net.DTAG.DE (62.154.18.18) 53.053 ms 51.602 ms 52.912 ms
 5 62.156.139.146 (62.156.139.146) 55.010 ms 53.860 ms 55.884 ms
 6 ar1.str.de.colt.net (212.121.151.242) 63.789 ms 70.990 ms 57.906 ms
 7 * * *
 8 * * <Strg-C>
```

Was versendet der Traceroute-Befehl, und warum erhält er eine Antwort von den Routern auf dem Weg zum Ziel? Häufig hört man auch von langjährigen Administratoren Erklärungen, dass das Paket die Router unterwegs aufzeichne oder ähnlichen Unsinn. Im Grunde ist es ganz einfach. Der Befehl versendet ein IP-Paket mit der Zieladresse, die beim Aufruf angegeben wird. Um eine Antwort von den Routern und nicht vom Ziel zu erhalten, modifiziert der Befehl den TTL-Wert im

IP-Header des Pakets. Zunächst sendet der Befehl drei Pakete mit einem TTL-Wert von eins. Der erste Router reduziert den TTL-Wert um eins und muss das Paket verwerfen, da der TTL-Wert null erreicht hat. Zusätzlich sendet er eine ICMP-Time-Exceeded-Fehlermeldung an den Absender. Der Absender erhält also drei Fehlermeldungen von dem Router mit dessen IP-Adresse. Für die IP-Adresse führt der Befehl nun noch eine Rückwärts-Namensauflösung durch (außer Sie geben beim Aufruf des Befehls die Option `-n` an). Der Befehl gibt nun diese Informationen aus:

```
1 192.168.255.1 (192.168.255.1) 0.871 ms 4.139 ms 0.940 ms
```

Nun erzeugt der Befehl drei weitere Pakete, die sich nur in dem TTL-Wert unterscheiden. Dieser ist nun zwei. Der erste Router reduziert den TTL-Wert und routet das Paket weiter an den zweiten Router, der dann das Paket verwirft und nun die Time-Exceeded-Mitteilung mit seiner Absenderadresse verschickt. So erhält der Client die IP-Adresse des zweiten Routers.

Nun stellt sich die Frage, was für ein Paket der Traceroute-Befehl versendet? Ein IP-Paket mit wachsendem TTL-Wert, aber was für ein IP-Paket? Wenn Sie den Windows-`tracert.exe`-Befehl verwenden, handelt es sich um ein ICMP-Echo-Request-Paket. So kann der `tracert.exe`-Befehl leicht ermitteln, wann er das tatsächliche Ziel erreicht hat. Solange der TTL-Wert zu klein ist, um das Ziel zu erreichen, erhält er Time-Exceeded-Meldungen. Sobald das Ziel erreicht wurde, erhält der Befehl eine Echo-Reply-Nachricht.

Der Unix-`traceroute`-Befehl arbeitet anders, auch wenn einige Varianten (z.B. Red-Hat/Fedora) auch mit der Option `-I` ICMP-Echo-Request-Nachrichten verwenden können. Dieser Befehl versendet UDP-Pakete. UDP-Pakete benötigen einen Quell- und einen Zielport. Der Quellport wird zufällig gewählt (> 1023). Der Zielport wird nach einer einfachen Formel gewählt: $\text{BASE} + \text{TTL} - 1$. Der BASE-Wert kann beim Befehl mit der Option `-p` frei gewählt werden. Der Defaultwert ist 33434. Die Zielports liegen also im Bereich 33434-33689. Solange der TTL-Wert zu klein ist, um den Zielrechner zu erhalten, erhält der `traceroute`-Befehl eine Time-Exceeded-Nachricht. Sobald der Zielrechner erreicht wird, basiert die Portwahl auf der Hoffnung, dass auf diesen Ports kein Dienst horcht. Üblicherweise sendet dann ein Betriebssystem eine ICMP-Port-Unreachable-Nachricht zurück. So kann der `traceroute`-Befehl erkennen, dass er sein Ziel erreicht hat.

Wenn Sie nun die Traceroute-Funktionalität von innen über Ihre Firewall erlauben möchten, müssen Sie folgende Regeln verwenden:

```
# Akzeptiere Time-Exceeded-Nachrichten von außen. Für Windows und Unix
# erforderlich
$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -p icmp --icmp-type time-exceeded -m state \
  --state RELATED -j ACCEPT
```

```
# Für Windows-tracert.exe erlaube echo-request-Pakete
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p icmp --icmp-type echo-request -m state \
  --state NEW -j ACCEPT
```

```
# Die folgende Regel akzeptiert die Antworten. Diese Regel ist in den meisten
# Skripten bereits vorhanden
$IPTABLES -A FORWARD -m state --state ESTABLISHED -j ACCEPT

# Für Unix-traceroute erlaube UDP-Pakete
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p udp --dport 33434:33689 -m state \
  --state NEW -j ACCEPT

# Erlaube Port-Unreachable-Nachrichten
# Dies ist meistens bereits der Fall, da von außen meistens alle Dest-Unreach
# Nachrichten akzeptiert werden
$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -p icmp --icmp-type port-unreachable -m state \
  --state RELATED -j ACCEPT
```

Die Firewall selbst wird nun in den Ausgaben der Traceroute-Befehle nicht auftauchen, da sie selbst nicht das Recht hat, Time-Exceeded-Nachrichten zu versenden. Statt der Firewall werden drei Sterne in der Ausgabe erscheinen. Wenn Sie möchten, dass auch die Firewall angezeigt wird, können Sie die folgende Regel hinzufügen:

```
$IPTABLES -A OUTPUT -o $INTDEV -p icmp --icmp-type time-exceeded -m state \
  --state RELATED -j ACCEPT
```

Diese Regel erlaubt es der Firewall, das lokal erzeugte Time-Exceeded-Paket nach innen zu versenden.

34.13 Optimierung der ICMP-Regeln

Wenn Sie das ICMP-Protokoll so detailliert filtern möchten, wie ich es in diesem Kapitel beschrieben habe, ist es sinnvoll, die Regeln irgendwie ein wenig zu optimieren. Hierfür bieten sich die benutzerdefinierten Ketten an (siehe Abschnitt 9.3). Im Folgenden demonstriere ich das am Beispiel der ICMP-Pakete der FORWARD-Kette. Zunächst müssen Sie sicherstellen, dass auch Ihre ICMP-Regeln die Pakete tatsächlich betrachten können. Hierfür ist es wichtig, dass diese Regeln früh genug in der FORWARD-Kette betrachtet werden. Am einfachsten erstellen Sie zu Beginn Ihrer FORWARD-Kette die folgenden beiden Regeln:

```
$IPTABLES -A FORWARD -m state --state ESTABLISHED -j ACCEPT
$IPTABLES -A FORWARD -p icmp -j MY_ICMP
```

Die erste Regel akzeptiert sämtliche Pakete, die zu aufgebauten Verbindungen gehören. Dies sollte immer in jeder Kette die erste Regel sein, da die meisten Pakete in diese Kategorie gehören. Bei einer einfachen DNS-Anfrage ist es jedes zweite Paket, bei einer TCP-Verbindung sind es alle ab dem zweiten Paket. Das können schnell

auch mal mehrere hundert Pakete sein. Da Sie entscheiden, welche Verbindung aufgebaut werden darf (State: NEW), ist dies auch sicher.

Anschließend prüfen Sie in der zweiten Regel, ob es sich um ein ICMP-Paket handelt, und springen dann in die benutzerdefinierte Kette MY_ICMP. Diese müssen Sie nun anlegen und mit Regeln füllen. Wenn Sie meinem Rat folgen, werden Sie alle Destination-Unreachable-Meldungen von außen, alle Fragmentation-Needed-Meldungen, Time-Exceeded-Meldungen von außen (für Traceroute) und Echo-Request-Pakete von innen (für den Ping) zulassen. Ihre Regeln könnten dann folgendermaßen aussehen:

```
$IPTABLES -N MY_ICMP

# Destination-Unreachable von außen
$IPTABLES -A MY_ICMP -i $EXTDEV -p icmp --icmp-type destination-unreachable -m state \
--state RELATED -j ACCEPT

# Fragmentation-Needed
$IPTABLES -A MY_ICMP -p icmp --icmp-type fragmentation-needed -m state --state RELATED \
-j ACCEPT

# Time-Exceeded von außen für traceroute
$IPTABLES -A MY_ICMP -i $EXTDEV -o $INTDEV -p icmp --icmp-type time-exceeded -m state \
--state RELATED -j ACCEPT

# Echo-Request von innen für ping
$IPTABLES -A MY_ICMP -i $INTDEV -o $EXTDEV -p icmp --icmp-type echo-request -m state \
--state NEW -j ACCEPT

# Protokolliere Parameter-Problem und Source-Quench
$IPTABLES -A MY_ICMP -p icmp --icmp-type source-quench -j LOG --log-prefix "Source-Quench: "
$IPTABLES -A MY_ICMP -p icmp --icmp-type parameter-problem -j LOG \
--log-prefix "Parameter-Problem: "

# Alle weiteren ICMP-Nachrichten werden unterdrückt
$IPTABLES -A MY_ICMP -j DROP
```

Die letzte Regel verwirft alle ICMP-Pakete, die im Vorfeld nicht akzeptiert wurden. Damit ist sichergestellt, dass am Ende dieser benutzerdefinierten Kette auch kein Rücksprung in die FORWARD-Kette erfolgt und dort möglicherweise noch Pakete akzeptiert werden oder einfach nur Prozessorleistung für unnötige Tests vergeudet wird.

