

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



35 IPv6

Das IPv6-Protokoll wird bisher kaum eingesetzt. Dennoch unterstützt Iptables bereits das IPv6-Protokoll. Während die Linux-Kernel bis einschließlich 2.6.13 das IPv6-Protokoll nur wenig unterstützen, wird ab dem Kernel 2.6.14 eine wesentliche Änderung eintreten. Bei den älteren Kernen unterstützt das Connection Tracking noch nicht das IPv6-Protokoll. Hierfür ist immer ein Patch des USAGI-Projekts erforderlich gewesen, da das Connection Tracking nur das IPv4-Protokoll unterstützt. Um nicht das ganze Connection Tracking für das IPv6-Protokoll erneut schreiben zu müssen, hat das Netfilter-Team diese Funktionalität komplett überarbeitet und neu geschrieben (siehe Kapitel 27), so dass nun auch bald das IPv6-Protokoll komplett unterstützt wird.

Aktuell ist dies aber leider noch nicht der Fall.

35.1 Filterung mit ip6tables

Der Befehl `ip6tables` funktioniert identisch zum Befehl `iptables`. Ihnen stehen leicht andere Tests und Ziele zur Verfügung. Eine große Anzahl der in Patch-O-Matic verfügbaren Patches sind auch für IPv6 verfügbar. Lediglich alle Funktionen, die auf dem Connection Tracking aufbauen, stehen nicht für IPv6 zur Verfügung. Dies wird sich aber in nächster Zukunft ändern. Dann können Sie auch diese Funktionen nutzen.

Im Moment haben Sie die folgenden Ziele im Kernel 2.6.14 zur Verfügung:

- `ACCEPT`, `DROP`, `RETURN` und `QUEUE`, da es sich hier um fest eingebaute Ziele handelt. Alle weiteren Ziele werden über ladbare Kernelmodule realisiert.
- `LOG`, `MARK`, `NFQUEUE`, `REJECT`, `ROUTE` und `TRACE` stehen, wie für das IPv4-Protokoll, zur Verfügung.
- `HL`: Dies ist ein Ziel, das nur für das IPv6-Protokoll zur Verfügung steht. Es wird weiter unten erläutert.

Diese Ziele können Sie in den drei Tabellen `raw`, `filter` und `mangle` einsetzen. Die Tabelle `nat` steht nicht zur Verfügung, da es (noch) keine Unterstützung für Connection Tracking und Network Address Translation gibt.

Um die Pakete zu prüfen, können Sie auf die folgenden Tests zurückgreifen:

- `-p, --protocol, -s, --source, -d, --destination, -i, --in-interface, -o, --out-interface` haben dieselbe Funktion wie bei IPv4.
- `-p icmpv6 --icmpv6-type`: Um das spezielle ICMPv6-Protokoll zu unterstützen, haben Sie hier einen eigenen Test.
- Erweiterungen, die identisch den Iptables-Erweiterungen funktionieren, sind:
 - `-m ah`
 - `-m esp`
 - `-m length`
 - `-m limit`
 - `-m mac`
 - `-m mark`
 - `-m multiport`
 - `-m owner`
 - `-m physdev`
- Neu hinzugekommen für das Protokoll IPv6 sind:
 - `-m dst`: Hiermit können Sie Optionen im IPv6-Destination-Header prüfen.
 - `-m eui64`: Dies prüft, ob der EUI64-Teil einer autokonfigurierten IPv6-Adresse stimmt.
 - `-m hbh`: Dies prüft den IPv6-Hop-by-Hop-Header.
 - `-m hl`: Dieser Test prüft das Hop-Limit-Feld im IPv6-Header.
 - `-m ipv6header`: Hiermit können Sie Optionen im IPv6-Header prüfen.
 - `-m rt`: Hiermit prüfen Sie den IPv6-Routing-Header.

35.2 Neue IPv6-Targets

35.2.1 HL

Mit diesem Ziel können Sie das IPv6-Hoplimit-Feld modifizieren. Der Hoplimit ist mit dem TTL-Feld des IPv4-Protokolls vergleichbar. Dieses Target darf nur in der Mangle-Tabelle eingesetzt werden. Die Verwendung des Targets ist gefährlich, da eine Erhöhung des Hoplimits Routing-Loops erzeugen kann.

Das Target hat die folgenden Optionen:

- `--hl-set <hops>`: Hiermit setzen Sie den Wert absolut.
- `--hl-dec <wert>`: Hiermit reduzieren Sie das Hoplimit um den angegebenen Wert.
- `--hl-inc <wert>`: Hiermit erhöhen Sie das Hoplimit um den angegebenen Wert.

35.3 Neue IPv6-Matches

35.3.1 dst

Mit diesem Test können Sie den IPv6-Destination-Header prüfen. Der Test hat zwei zusätzliche Optionen:

- `--dst-len <länge>`: Dies prüft die totale Länge des Headers.
- `--dst-opts <TYPE>[:<LEN>], []`: Hiermit prüfen Sie einzelne Optionen und ihre Länge.

35.3.2 eui64

Wenn die IPv6-Adresse per Autokonfiguration zugewiesen wurde, enthalten die niedrigen 64 Bit der IP-Adresse die MAC-Adresse der Netzwerkkarte. Dieser Test prüft, ob das tatsächlich der Fall ist.

35.3.3 hbh

Hiermit können Sie die IPv6-Hop-by-Hop-Optionen testen. Die Option `--hbh-len <länge>` prüft die totale Länge, während Sie mit `--hbh-opts <Type>[:<länge>]` die einzelnen Optionen und ihre Länge prüfen können.

35.3.4 hl

Dieses Modul prüft das Hoplimit-Feld. Sie können prüfen, ob der Wert mit einem bestimmten Wert übereinstimmt, kleiner oder größer ist. Dieser Test ist mit dem `ttl`-Test vergleichbar.

- `--hl-eq <hops>`: Gleich.
- `--hl-lt <hops>`: Kleiner.
- `--hl-gt <hops>`: Größer.

35.3.5 ipv6header

Mit diesem Test können Sie die Optionen im IPv6-Header testen. Mit der Option `--header <headers>` können Sie die Header angeben, die in dem Paket enthalten oder nicht (!) enthalten sein müssen. Dabei müssen Sie alle Header angeben, damit der Test zutrifft. Wenn Sie nur einen Teil angeben möchten, müssen Sie zusätzlich die Option `--soft` definieren.

Sie können die folgenden Header testen: `hop, dst, route, frag, auth, esp, none, proto`.

35.4 rt

Dieser Test prüft den Routing-Header. Hierfür haben Sie die folgenden Optionen:

- `--rt-type <typ>`: Hiermit prüfen Sie den numerischen Typ des Routing-Headers.
- `--rt-segyleft <num>[<num>]`: Dies prüft das Segments-Left-Feld.
- `--rt-len <länge>`: Hiermit prüfen Sie die Länge des Headers.
- `--rt-0-res`: Dies prüft, ob das reservierte Feld gesetzt ist.
- `--rt-0-addr <ip>[,<ip>]`: Dies prüft, ob im reservierten Feld (Typ 0) eine Adresse vorhanden ist.
- `--rt-0-not-strict`: Die letzte Option muss alle IP-Adressen im Routing-Header definieren. Wenn Sie lediglich eine prüfen möchten, müssen Sie zusätzlich diese Option angeben.