

IPTables — Linux Packet Filtering

Fortgeschrittene Firewall Funktionalität unter Linux

Markus Korzendorfer — Marcus Fritsch[‡]

[‡]Hochschule Furtwangen,
Computer Networking – Fakultät Informatik
Furtwangen

17. Januar 2008

Agenda

- ▶ Einführung
- ▶ Konzepte
- ▶ Konzepte im Detail
- ▶ Other stuff
- ▶ Fazit

Einführung

- ▶ ein packet filter unter vielen
- ▶ Vorgänger waren
 - ipfwadm in Linux 2.0
 - ipchains in Linux 2.2
- ▶ seit Linux Kernel 2.4 gibt es iptables
- ▶ eigentlich nur ein Teil des *netfilter*

Einführung

- ▶ Haupteigenschaften von IPTables
 - IPv6 Unterstützung
 - stateless paket filtering
 - stateful paket filtering
 - alle Arten von NA(P)T
 - Packetmanipulation (mangle)
 - kooperiert mit Traffic Control

Konzepte

▶ Begriffserläuterung

- Tabellen (tables): Enthalten Ketten
- Ketten (chains): Sammlungen von Regeln
- Muster (pattern): Bestimme welche Pakete betroffen sind
- Regeln (rules): Besteht aus Muster und Ziel
- Ziele (targets): Entscheiden was mit dem Paket geschieht

▶ Regelverarbeitung

Konzepte im Detail, tables & chains

▶ Standardtables

- mangle: Manipulation des headers
- nat: Adressübersetzung
- filter: Paketprüfung

▶ Chains

- Sind Sammlungen von Regeln
- Es existieren fünf Typen von Standardchains
 - Prerouting
 - Forward
 - Input
 - Output
 - Postrouting

Konzepte im Detail, tables & chains

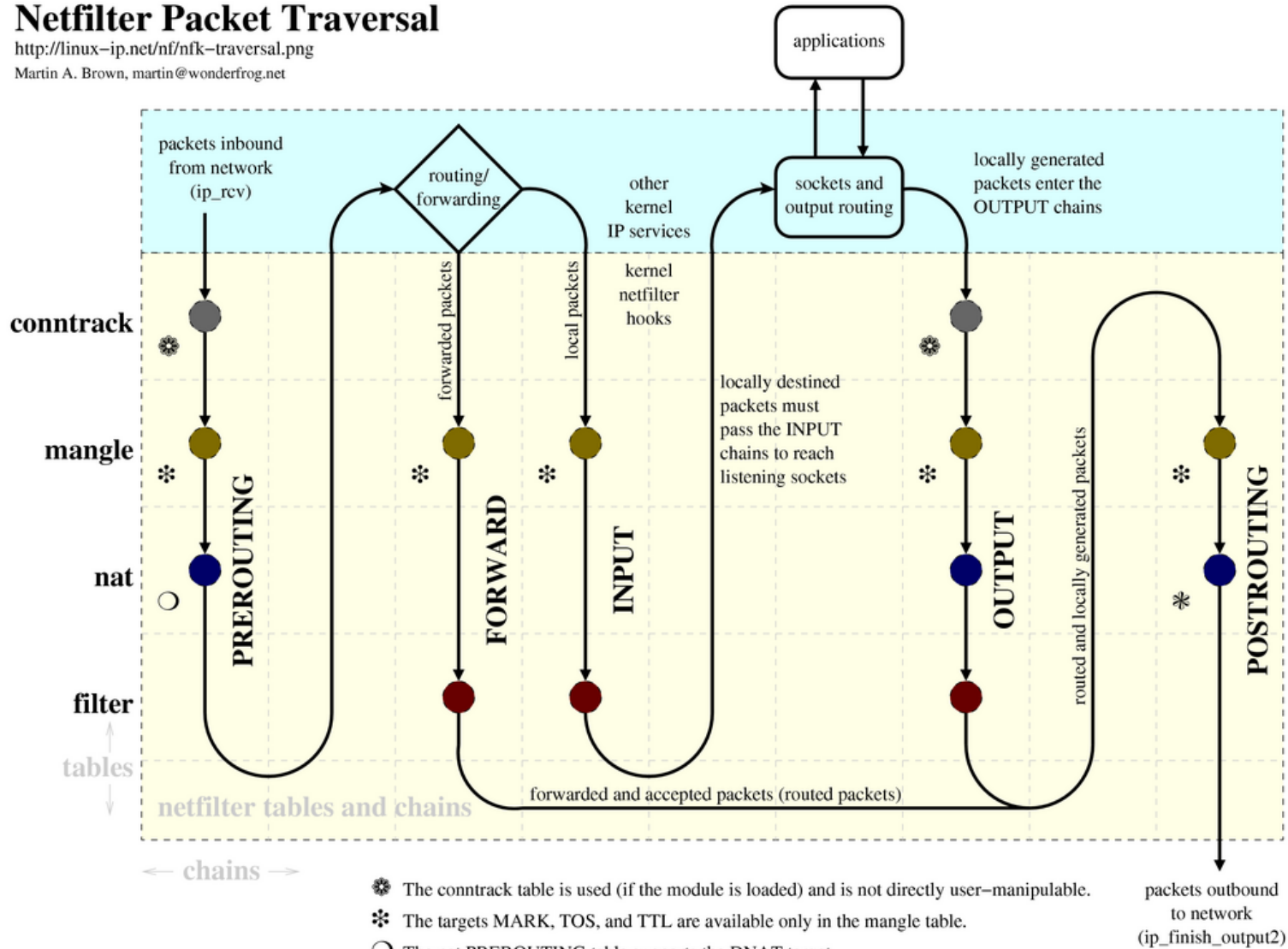
- ▶ Input, Output und Forward chains besitzen immer eine Standardregel
- ▶ In Prerouting und Postrouting können nur Pakete manipuliert (mangle und/oder NAT), nicht jedoch gefiltert werden
- ▶ für fest vorgegebene Chains kann man Policies definieren
- ▶ benutzerdefinierte Chains auch möglich, jedoch keine Policies
- ▶ Jeder Standardtable besitzt mehrere Ketten
 - mangle: enthält alle Chains
 - nat: enthält Prerouting, Output und Postrouting
 - filter: enthält Forward, Input und Output

packet traversal

Netfilter Packet Traversal

<http://linux-ip.net/nf/nfk-traversal.png>

Martin A. Brown, martin@wonderfrog.net



cf. <http://www.docum.org/qos/kptd/>

cf. http://open-source.arkoon.net/kernel/kernel_net.png

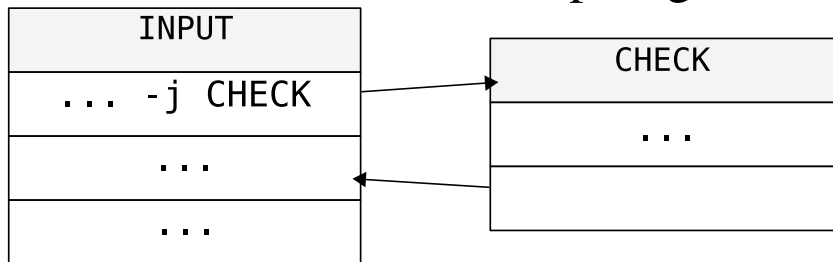
cf. <http://iptables-tutorial.frozentux.net/>

Regeln und Ziele

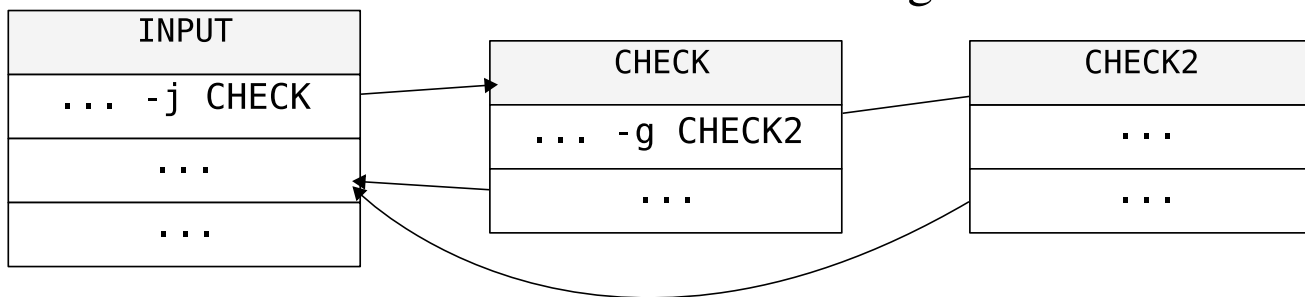
- ▶ definiert durch *Tabelle*, *Kette*, *Match* und *Ziel*
- ▶ jede *Regel* muss ein *Ziel* referenzieren
- ▶ falls kein *Match* vorhanden ist wird implizit jedes paket matchen
- ▶ das *Ziel* wird angesprungen wenn das *Match* zutrifft
- ▶ ein *Ziel* ist
 - entweder ein 'einfaches' *Ziel*
 - oder eine weitere, benutzerdefinierte *Kette*

Regelabarbeitung

- ▶ Regeln werden strikt linear abgearbeitet
- ▶ 'Programmfluss' mittels `--jump` und `--goto`
 - `--jump` oder `-j` springt eine Kette oder ein Ziel an
 - `return` an die Stelle des Absprungs + 1



- `--goto` oder `-g` springt eine benutzerdefinierte Kette an
- `return` an die Kette, welche die aktuelle aufgerufen hat



Iptables, das eingemachte (Module)

▶ state

- implementiert eine entscheidung aufgrund von Verbindungsstati
- mögliche Stati sind:
 - ESTABLISHED erlaubt auf bereits bestehende verbindungen zu matchen
 - RELATED erlaubt auf 'zugehörige' Pakete zu matchen, z.B. icmp meldungen zu bekannten verbindungen.
 - INVALID das Paket ist ungueltig

Iptables, das eingemachte (Module)

▶ limit

- erlaubt matchen auf ein paket limit
- z.B. nuetzlich zur Abwehr von SYN-Flood attacken

▶ owner

- matchen auf den besitzer eines pakets
- nur fuer lokal generierte pakete
- z.B. darf ein apache nicht nach 'draussen' verbinden
- aber; eingeschraenkt, sid/uid/cmd auf SMP broken
- uid bei suid-prozessen sowieso nutzlos

Iptables, das eingemachte (Module)

▶ ulog

- erlaubt das loggen zum userspace
- sinn und zweck: usermode tools koennen mit den paketen arbeiten
 - traffic accounting
 - statistische auswertung
 - (irre) ab in die datenbank damit
- kommunikation ueber das netlink interface
- libraries libnfnetlink als low level library
- libnfulog als high level netfilter netlink library fuer ulog

NA(P)T

- ▶ ausschliesslich in der table nat
- ▶ targetsets
- ▶ DNAT
 - nur in den Ketten PREROUTING und OUTPUT
 - 'klassisches' NA(P)T
 - aendert ziel adress (und/oder port) eines pakets
 - und aller nachfolgenden pakete der verbindung
- ▶ REDIRECT
 - nur in den Ketten PREROUTING und OUTPUT
 - ein 'durchlaufendes' paket an sich selbst schicken, z.B. fuer transparent proxy

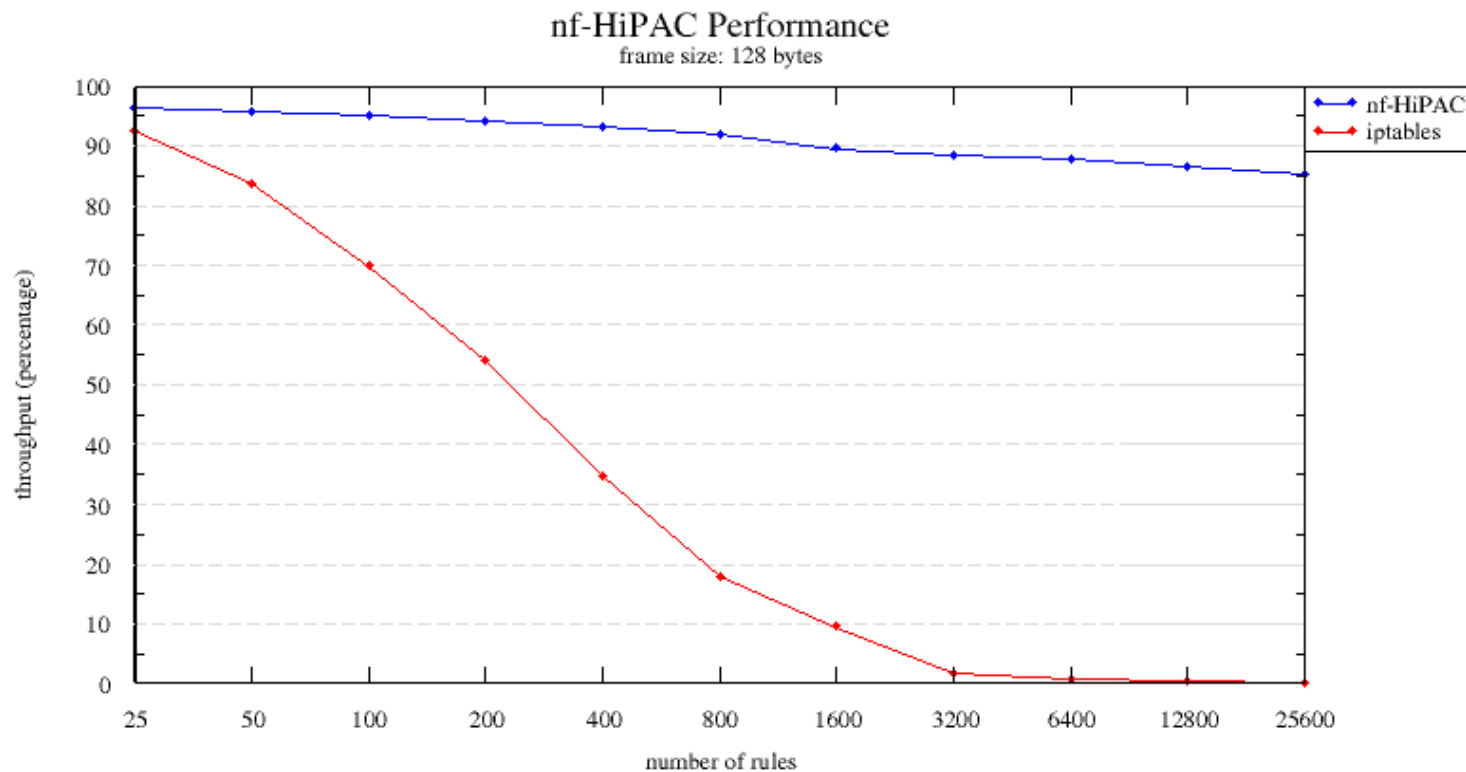
NA(P)T

▶ MASQUERADE

- nur in der PREROUTING Kette gueltig
- Maskieren von privaten adressen fuer IPv4, z.B. bei dial-up verbindugen oder VPNs
- Kann den quell port remappen
 - innerhalb eines spezifizierten bereichs
 - zufaellig, seit Kernel 2.6.21 und iptables 1.3.8

Performance

- ▶ Für kleine Regelwerke grundsätzlich gut, aber;
- ▶ Abarbeitung der Regeln strikt linear
- ▶ Umfangreiche Regelwerke werden schnell langsam
- ▶ Alternative: nf-hipac



Performance

- ▶ Implementiert hochperformanten packet filter
- ▶ Implementiert iptables kompatibles interface
- ▶ aber: noch kein IPv6 support
- ▶ aber: wird nicht sehr aktiv gepflegt; letztes release am 12. Oktober 2005
- ▶ letztes offizieller patch gegen linux 2.6.14 (31. Januar 2006, Version 2.6.14.7)

Fazit

- ▶ nur so sicher wie die Konfiguration
- ▶ Konfigurationsfehler könnten fatale Folgen haben
- ▶ Ausreichender Schutz auf Paketebene
- ▶ Höhere Instanzen werden nicht unterstützt