nftables

Nachfolger von iptables

Christian Perle

June 3, 2021

Grundstruktur ähnlich iptables

- Benutzt die netfilter-Hooks (PRE_ROUTING, LOCAL_IN, FORWARDING, LOCAL_OUT, POST_ROUTING)
- Tables: Container für Chains
- Chains: Container für Rules
- Rules: Matching von Paketen, abhängig davon passieren Aktionen

Neu gegenüber iptables (1)

- Konfigurationstool im Userspace: nft
  (ersetzt iptables/ip6tables)
- Keine zwingende Trennung von IPv4/IPv6 im Regelwerk
- nft ersetzt auch arptables, ebtables und ipset
- Weitere netfilter-Hooks: NETDEV_INGRESS

Neu gegenüber iptables (2)

- Neue Syntax: Alles in *einem* strukturierten Regelwerk
- Keine Built-In Chains mehr
- Tables und Base Chains müssen erzeugt werden
- Base Chains werden an Hooks gebunden
- Sets, Maps, Verdict Maps, ...

Erste Schritte (1)

- Erzeugen einer Table:
  Family der Table bestimmt die verwendbaren Hooks
  ```
  nft add table inet filter
  ```
- Erzeugen von Base Chains in der Table:
  Type bestimmt die verwendbaren Aktionen
  Hook und Priorität definieren
  ```
  nft add chain inet filter input \
    { type filter hook input \
    priority 0 \; policy accept \; }
  ```

Erste Schritte (2)

- Erzeugen von Rules in Chains:
  `nft add rule inet filter input tcp dport 22 accept`
- Geht das auch schöner?
  nft-Konfigurationsdatei mit nft erzeugen:
  `nft list ruleset > mini.nft`
- Regeln aus Konfigurationsdatei lesen, Option -e (echo) zeigt Syntax der Einzelaktionen
  `nft flush ruleset`
  `nft -e -f mini.nft`

Rules

- Mehr als eine Aktion pro Rule
- Reihenfolge der Aktionen beachten!
  `tcp dport 80 drop log  (Fehler, drop ist final)`
- Wo sind die Counter?
  Counter sind jetzt eine Aktion (optional)
  `tcp dport 80 counter drop`
- Rules löschen: Handles
  `nft list ruleset -a`
  `nft delete rule inet filter output handle X`

Sets (1)

- Container eines definierten Typs
  Adressen, Ports, Interfaces, ...
- Anonyme Sets in Rules (nicht manipulierbar)
- Benannte Sets:
  In Rules verwendbar, Set-Inhalt manipulierbar

```
nft add set inet filter blocked \
  { type inet_service \; }
nft add element inet filter \
  blocked { 80, 8080 }
```

Sets (2)

- Benutzung in einer Rule:

```
nft add rule inet filter output tcp \
  dport @blocked counter drop
```

- Weiteres Set-Element hinzufügen, Rule muss nicht angepasst werden:

```
nft add element inet filter blocked { 443 }
```

Maps

- Bilden Werte eines Typs auf Werte eines anderen Typs ab
- Ähnlich Dictionaries in Programmiersprachen
- Verdict Maps bilden Werte eines Typs auf Aktionen ab

Best Practice/Pitfalls

- Base Chains wenn möglich nach ihrem Hook benennen:
  Macht das Regelwerk besser lesbar
- Mehr als eine Base Chain pro Hook möglich:
  Priorität beachten!

Unter der Haube

- Abarbeiten von Rules im Kernel geschieht durch eine Pseudo-Statemachine
- Wesentlich effizienteres Update des Regelwerks:
  In iptables wurde bei Änderungen eine Table immer komplett neu befüllt und gegen die alte Table ausgetauscht

Geiler Scheiß

- Portknocking mit nftables (ohne Userspace-Daemon)
- Flowtables:
  Pakete mit bekannten Flows bereits am ingress-Hook abgreifen und schnell weiterleiten

Links

- nftables-Wiki:
  https://wiki.nftables.org/wiki-nftables/