Firewalls unter Linux

ITS-Net-Lin

Sebastian Meisel

2. Januar 2025

1 Einführung

Eine Firewall ist ein sicherheitsrelevantes Netzwerkgerät oder Software, das den Datenverkehr zwischen Netzwerken überwacht und kontrolliert. Unter Linux stehen zahlreiche Tools und Technologien zur Verfügung, um Firewalls zu implementieren. Dieser Text führt in das Konzept einer Stateful Packet Inspection (SPI)-Firewall ein und gibt einen Überblick über gängige Lösungen wie Iptables, Nftables, UFW und Firewalld.

1.1 Das Konzept der SPI-Firewall

Eine SPI-Firewall (Stateful Packet Inspection) überwacht und analysiert den Zustand jeder Verbindung, die durch sie hindurchläuft. Dies bedeutet, dass die Firewall nicht nur die Header-Informationen einzelner Pakete betrachtet, sondern auch den Kontext einer Verbindung beibehält. Dadurch kann sie gezielt entscheiden, welche Pakete erlaubt oder abgelehnt werden sollen. SPI-Firewalls bieten daher eine bessere Kontrolle und Sicherheit im Vergleich zu statischen Paketfiltern.

1.2 Gängige Lösungen unter Linux

1.2.1 Iptables

Iptables ist ein traditionelles Firewall-Tool, das auf dem Netfilter-Framework basiert. Es ermöglicht die Definition von Regeln, um Netzwerkpakete zu filtern und Weiterleitungsrichtlinien festzulegen.

```
1  # Beispiel: Alle eingehenden SSH-Verbindungen erlauben
2  sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

1.2.2 Nftables

Nftables ist der moderne Nachfolger von Iptables und bietet eine effizientere Syntax und bessere Leistung. Es wird ebenfalls vom Netfilter-Framework unterstützt und ersetzt schrittweise Iptables.

1.2.3 Erstellen einer Tabelle

Eine Tabelle wird benötigt, um Regeln und Ketten zu organisieren. Tabellen können verschiedene Protokolle unterstützen (z. B. IPv4, IPv6, ARP).

```
1  # Beispiel: Eine Tabelle für IPv4 anlegen
2  sudo nft 'add␣table␣ip␣filter'
```

!! Wichtig !! Die einfachen Anführungszeichen, sind in diesem Befehl genaugenommen nicht notwendig, aber in anderen. Sie verhindern, dass die Shell bestimmte Zeichen als Befehle interpretiert. Man sollte sich angewöhnen sie immer zu setzen.

1.2.4 Erstellen einer Kette

Innerhalb einer Tabelle werden Ketten definiert, die für eingehenden, ausgehenden oder weitergeleiteten Verkehr verwendet werden können.

```
# Beispiel: Eine Eingabekette erstellen
sudo nft 'add chain ip filter input { type filter hook input priority 0; }'
# Beispiel: Eine Ausgabekette erstellen
sudo nft 'add chain ip filter output { type filter hook output priority 0; }'
# Beispiel: Eine Weiterleitungskette erstellen
sudo nft 'add chain ip filter forward { type filter hook forward priority 0; }'
```

1.2.5 Hinzufügen von Regeln zu einer Kette

Nachdem eine Tabelle und Ketten erstellt wurden, können Regeln hinzugefügt werden.

```
# Beispiel: Eingehende HTTP-Verbindungen in der Eingabekette erlauben
sudo nft 'add rule ip filter input tcp dport 80 accept'
```

Anzeigen und Speichern der Konfiguration Die aktuelle Nftables-Konfiguration kann überprüft und dauerhaft gespeichert werden.

```
# Beispiel: Konfiguration anzeigen
sudo nft list ruleset
# Beispiel: Konfiguration dauerhaft speichern (je nach Distribution)
sudo nft list ruleset > /etc/nftables.conf
```

Mit diesen zusätzlichen Befehlen lässt sich eine vollständige Nftables-Firewall strukturieren und anlegen. Tabellen bieten eine übersichtliche Möglichkeit, Regeln nach Protokoll oder Anwendungszweck zu organisieren.

1.2.6 UFW (Uncomplicated Firewall)

UFW ist ein einfaches Frontend für Iptables und zielt darauf ab, Firewall-Management benutzerfreundlich zu gestalten. Es ist insbesondere bei Ubuntu-Systemen beliebt.

```
# Beispiel: HTTP und HTTPS-Verbindungen erlauben
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
# Beispiel: SSH-Verbindungen deaktivieren
sudo ufw deny 22
```

Damit diese Regeln aktiviert werden, muss `ufw` aktiviert werden:

```
sudo ufw enable
```

Sie können mit `ufw` diese Regel auch leicht wieder deaktivieren:

```
sudo ufw disable
```

1.2.7 Firewalld

Firewalld ist ein dynamisches Firewall-Management-Tool, das die Definition und Verwaltung von Regeln ohne Neustart der Firewall ermöglicht. Es nutzt Zonen, um unterschiedliche Sicherheitsstufen zu implementieren.

```
# Beispiel: SSH in der Zone 'public' erlauben
sudo firewall-cmd --zone=public --add-service=ssh
# Beispiel: HTTP in der Zone 'public' entfernen
sudo firewall-cmd --zone=public --remove-service=http
sudo firewall-cmd --runtime-to-permanent
```