zurück

Projektdokumentation: Virtueller Switch mit Open vSwitch (OVS)

1. Zielsetzung

Aufbau eines virtuellen Switches (ähnlich Extreme Networks) in einer Testumgebung auf einer Linux-VM, um:

- Layer-2-Switching zu simulieren
- VLAN-Tests und später NAC/RADIUS-Integration zu ermöglichen
- möglichst realitätsnahes Verhalten ohne echte Hardware

---

2. Systemumgebung

- Linux-Distribution: (z. B. Ubuntu Server 22.04 / Debian 12)
- VM-Umgebung: (Hypervisor wie Proxmox, VMware, VirtualBox)
- Netzwerkinterfaces der VM:
  - enp2s1: Management-Interface (getrennt!)
  - ens18 bis ens23: Switchports für Clients

---

3. Installation Open vSwitch

- Installation per Paketmanager:

bash

```
sudo apt update
sudo apt install openvswitch-switch
```

- Open vSwitch wird über systemd verwaltet:

bash

```
sudo systemctl enable --now openvswitch-switch
```

---

4. Aufbau der virtuellen Bridge

- Erstellen einer OVS-Bridge (sw0) und Hinzufügen der Ports (ens18 bis ens23):

- Automatisiertes Setup-Skript:

bash

```
#!/bin/bash
set -e

SWITCH_NAME="sw0"
PORTS=("ens18" "ens19" "ens20" "ens21" "ens22" "ens23")

sudo systemctl enable --now openvswitch-switch
sudo ovs-vsctl --if-exists del-br $SWITCH_NAME
sudo ovs-vsctl add-br $SWITCH_NAME

for iface in "${PORTS[@]}"; do
  if ip link show "$iface" > /dev/null 2>&1; then
    echo "→ Port $iface gefunden. Hinzufügen..."
    sudo ip link set $iface up
    sudo ovs-vsctl add-port $SWITCH_NAME $iface
  else
    echo "⚠ Warnung: Interface $iface existiert nicht, wird übersprungen."
  fi
done

sudo ip link set $SWITCH_NAME up
echo "✅ Switch $SWITCH_NAME fertig eingerichtet!"
```

---

5. Fehlerquellen und Lösungen

Problem	Ursache	Lösung
Skript hängt beim Ausführen	Virtuelle NIC (z. B. ens20) defekt oder ohne physisches Backend	Problematische Interfaces aus Skript entfernen oder im Hypervisor entfernen
SSH-Verbindung bricht beim Skriptstart ab	Netzwerkinterfaces werden verändert; temporäre Instabilität	Management-Interface (enp2s1) niemals in die OVS-Bridge integrieren!
CPU-Auslastung geht auf 100 %	Kernel wartet endlos auf Antwort von defektem Interface	Nur funktionierende Interfaces verwenden (ens18, ens19)
timeout funktioniert nicht	Reihenfolge der Befehle (sudo timeout …) war falsch	sudo ip link set … direkt sichern, Interfaces vorher auf Existenz prüfen

---

6. Aktueller Stand

- Switch sw0 erfolgreich erstellt
- Ports ens18, ens19, ens20, ens21, ens22, ens23 eingebunden (nur wenn physisch korrekt)
- Management-Zugang stabil über enp2s1
- SSH-Verbindung bleibt erhalten

- Kein CPU-Overload mehr
- Skript stabilisiert und bereit für Systemd-Autostart

---

7. Geplante nächste Schritte

- Aufbau von VLAN-Tagging (Ports einzeln VLANs zuordnen)
- Optional: Trunk-Ports konfigurieren
- Aufbau von RADIUS/802.1X-Authentifizierung (Integration ins NAC-System)
- LLDP-Aktivierung für Discovery-Simulation wie echte Switche
- Erweiterte Testumgebung mit echten Clients oder VMs

Fazit

- Mit Open vSwitch auf einer schlanken Linux-VM wurde erfolgreich ein virtueller Layer-2-Switch aufgebaut, der sich für Testumgebungen ähnlich einem echten Extreme Networks Switch verhält. Probleme durch defekte Interfaces und SSH-Instabilität wurden erkannt und sauber behoben.

- (Stand: Mai 2025)

Lars Weiß 2025/05/02 19:52