

OpenVPN mit Data Channel Offload auf pfSense CE 2.9.0

Ole Vautz · 09.10.2026

HU Berlin · IT-Security Workshop

Gliederung

1	Ausgangslage	Engpass, Funktionsweise von DCO, Lösungsoptionen
2	Eigenbau des Moduls	Quellstand, Laboraufbau, Bau und Laden
3	Fallstricke und Patch	Algorithmus, Interface-Typ, Patch
4	Test im Labor	Client-Verbindung, Prüfen ob DCO aktiv ist
5	Risiken und Ausblick	Offene Punkte und nächste Schritte

Ausgangslage

Engpass

OpenVPN verarbeitet jedes Datenpaket im Userspace. Ein einzelner CPU-Kern begrenzt den Durchsatz.

Vorhanden

OpenVPN 2.7.5 in CE 2.9.0 ist schon mit DCO-Unterstützung gebaut.

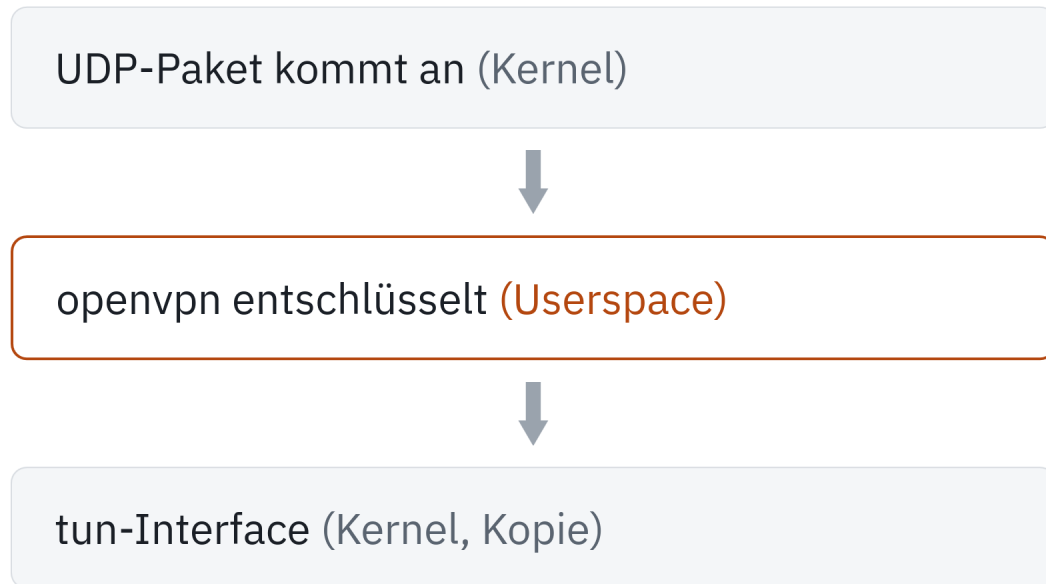
Fehlt

Das Kernelmodul liefert Netgate nur mit pfSense Plus aus.

`/boot/modules/if_ovpn.ko`

Funktionsweise von DCO

Ohne DCO



Mit DCO



Das openvpn-Programm erledigt mit DCO nur noch Handshake, Anmeldung und Schlüsselwechsel.

Lösungsoptionen im Vergleich

Kriterium	pfSense CE + eigenes Modul	OPNsense + DCO	WireGuard
Datenpfad	DCO im Kernel	DCO im Kernel	Kernel (if_wg)
Unterstützung	keine (Eigenbau)	offiziell, „experimental“	offiziell
Bei Updates	Modul neu bauen	gering	gering
Migration	keine	hoch (Neuaufbau)	mittel (Clients neu)
LDAP / MFA	ja	ja	nein
PQ-Handshake	ja (ML-KEM)	ja (ML-KEM)	nur per PSK

Bestimmung des FreeBSD-Quellstands

Die Kernelquellen von pfSense CE sind seit Version 2.8 nicht öffentlich. Gebaut wird daher aus FreeBSD main, mit einem Stand passend zum pfSense-Kernel.

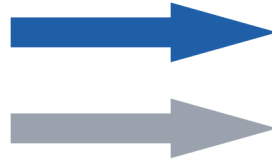
Merkmal	Wert
pfSense-Kernel	16.0-CURRENT, d8e3138ecf52, 17.08.2026
__FreeBSD_version	1600018
Wert in FreeBSD main	30.04. bis 19.06.2026
Gebauter Stand	7386c0675c8 (letzter Stand mit 1600018)
Abweichende Option	NETGRAPH ETF (für if_ovpn ohne Bedeutung)

Laboraufbau

Build-VM

FreeBSD 15.1-RELEASE

Baut das Modul, dient danach als
OpenVPN-Client mit DCO



Test-VM

pfSense CE 2.9.0

Liefert die Kerneldaten, lädt das Modul,
betreibt den OpenVPN-Server

Bau und Laden des Moduls

1 Kerneldaten auslesen

```
uname -a · kern.osreldate · kern.conftxt
```

2 Quellstand wählen

```
git checkout 7386c0675c8
```

3 Mit pfSense-Konfiguration bauen

```
make buildkernel KERNCONF=PFSENSE  
MODULES_OVERRIDE=if_ovpn
```

4 Auf pfSense laden

```
kldload if_ovpn
```

Fallstrick 1: Algorithmus

Ursache

Steht ein CBC-Verfahren unter den gewählten Algorithmen oder als Fallback, schaltet OpenVPN DCO ab. Die Oberfläche setzt als Fallback standardmäßig AES-256-CBC.

Lösung

Gewählte Algorithmen und Fallback-Algorithmus müssen beide DCO-tauglich sein: AES-GCM oder CHACHA20-POLY1305.

Note: `--data-ciphers-fallback` with cipher 'AES-256-CBC' disables data channel offload.

Fallstrick 2: Interface-Typ

Was pfSense CE tut

Legt vor dem Start ein klassisches tun-Interface an, benennt es in ovpnN um und schreibt dev-node in die Konfiguration.

Was DCO braucht

Ein if_ovpn-Interface. OpenVPN übernimmt aber das vorhandene tun-Interface und schickt ihm DCO-Befehle.

```
dco_set_ifmode: failed to set ifmode=00008002: Invalid argument (errno=22)
```

Defekter Zustand: OpenVPN läuft, überträgt aber keine Daten.

Handstart ohne pfSense geht; jeder Neustart über pfSense bringt den Fehler zurück.

Patch für openvpn.inc

Situation	Interface	Konfiguration
Modul nicht geladen	tun	mit disable-dco
Modul geladen, Instanz DCO-tauglich	if_ovpn	DCO aktiv
disable-dco in Custom options	tun	mit disable-dco
Instanz nicht tauglich (CBC, TCP ...)	tun	mit disable-dco

Eine Änderung an /etc/inc/openvpn.inc, eingespielt über System → Patches.

Build-VM als OpenVPN-Client

1 Pakete und Modul

```
pkg install openvpn · kldload if_ovpn
```

2 Schlüssel holen

CA und TLS-Key per scp von der pfSense, Client-Zertifikat und Schlüssel
aus der Oberfläche exportieren

3 Verbinden

```
openvpn --config client.conf
```

4 Testen

```
ping 10.8.0.1
```

Prüfen, ob DCO aktiv ist

1 Log-Level erhöhen

Server bearbeiten → Verbosity level: 4

2 OpenVPN-Log prüfen

```
grep -iE 'dco|offload' /var/log/openvpn.log
```

oder Status → System Logs → OpenVPN

3 Interface prüfen

```
ifconfig ovpn1 | grep groups
```

DCO aktiv

```
disable_dco = DISABLED  
ovpn-dco device [ovpn1] opened  
groups: openvpn
```

DCO nicht aktiv

```
disable_dco = ENABLED  
... disables data channel offload  
groups: tun openvpn
```

Risiken und Ausblick

Risiken

- Netgates unveröffentlichte Kerneländerungen sind nicht prüfbar
- Nach jedem Update: Modul neu bauen, Patch prüfen
- Kein Support durch Netgate

für die Zukunft

- Lasttest mit iperf3, mit und ohne DCO
- Langzeit-Stabilität
- Ladeskript mit Versionsprüfung
- Verhalten mit CARP/HA

Danke für Ihre Aufmerksamkeit.

Noch Fragen?