OpenVPN, IPsec, WireGuard, SINA

VPN-Lösungen in der Praxis

Karsten Neß

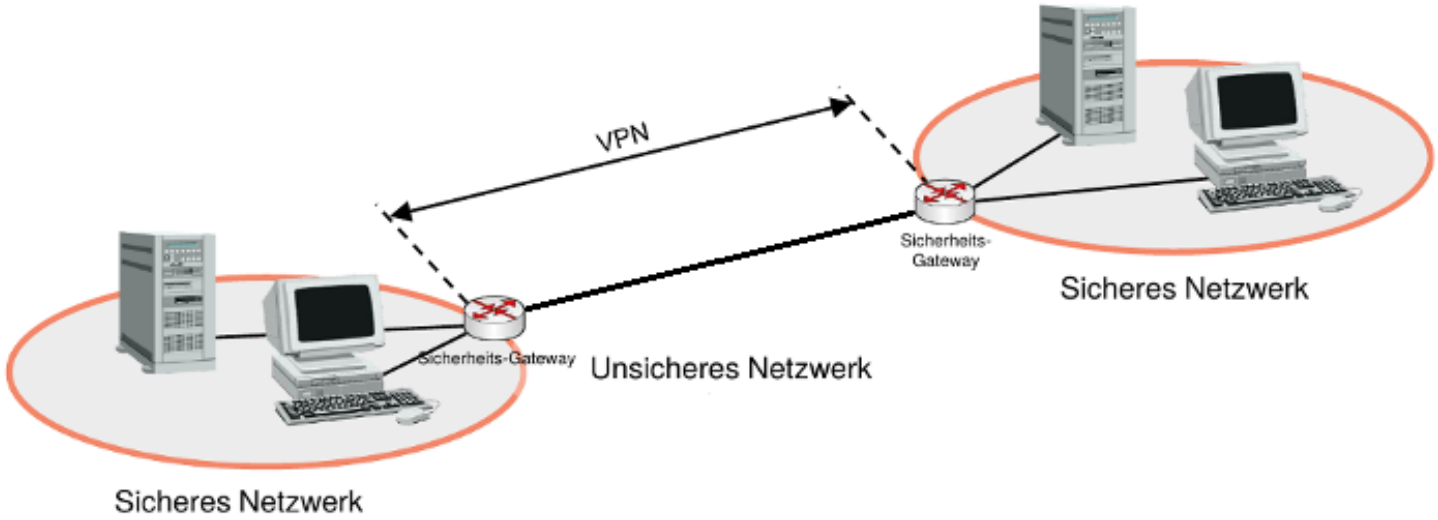
secunet Security Networks AG

SLAC 2018

Table of Contents

Sinnvolle Anwendungen für VPNs

Vertrauenswürdige Endpunkte über unsichere Netzwerke verbinden:



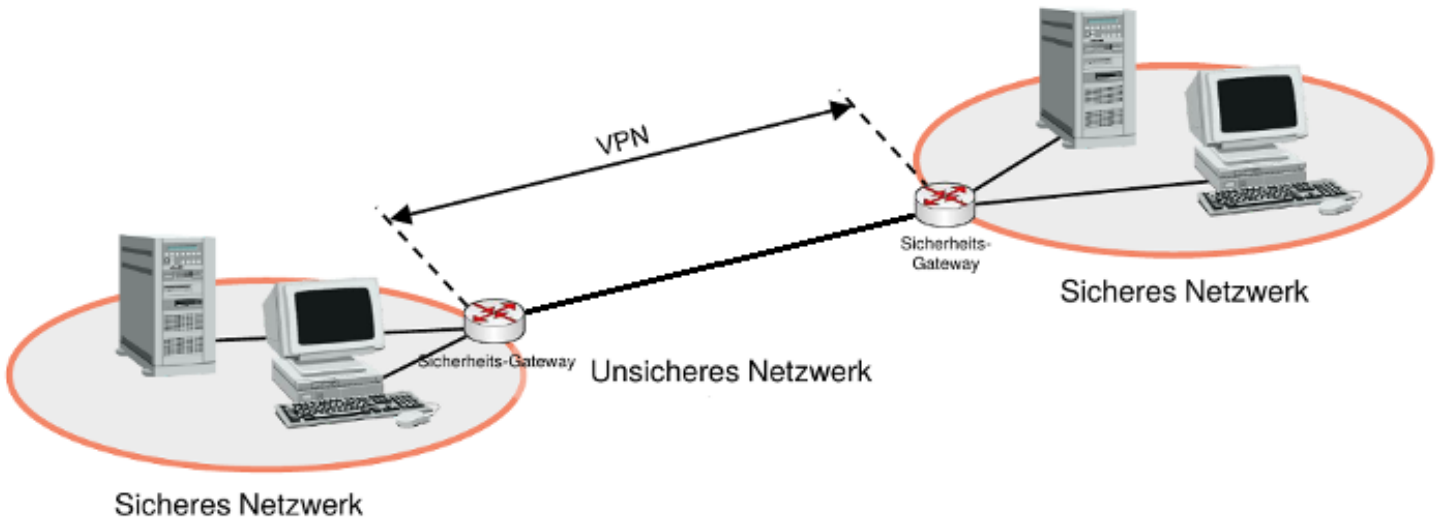
Site-to-Site-VPN: Verbindung zwischen zwei vertrauenswürdigen Netzen

Site-to-End-VPN: Anbindung ext. Clients an ein vertrauenswürdiges Netz

Multi-Site-VPN: Verbindung zwischen mehreren Netzen und ext. Clients

Sinnvolle Anwendungen für VPNs

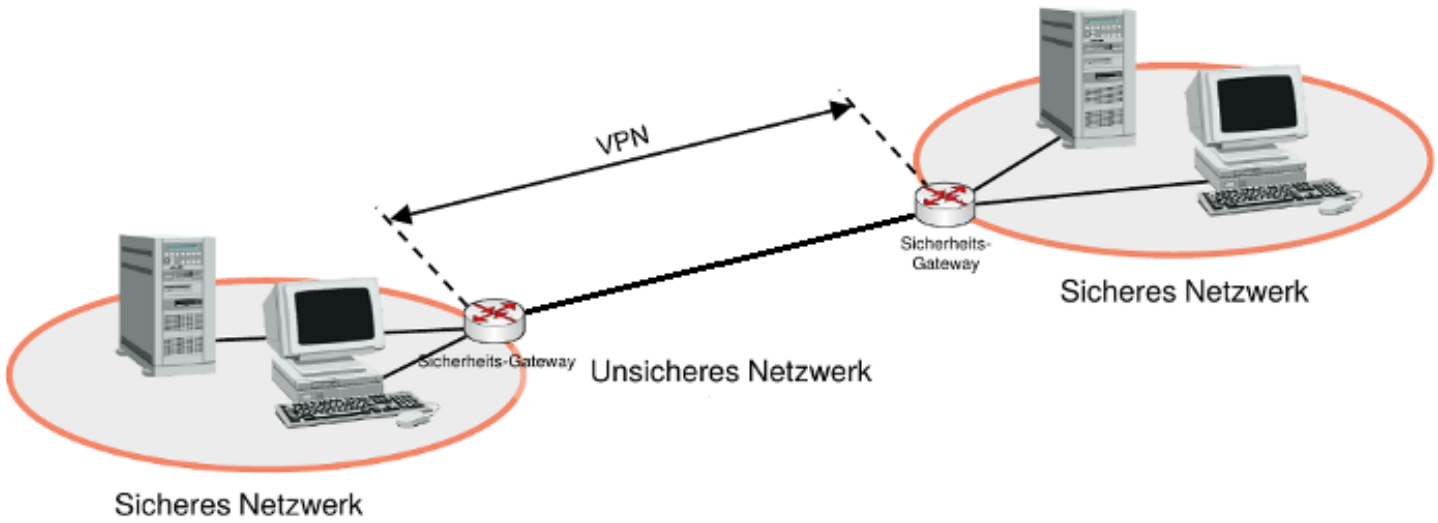
Vertrauenswürdige Endpunkte über unsichere Netzwerke verbinden:



Frage:
Was sind vertrauenswürdigen Bereiche und was ist das unsichere Netz?

Sinnvolle Anwendungen für VPNs

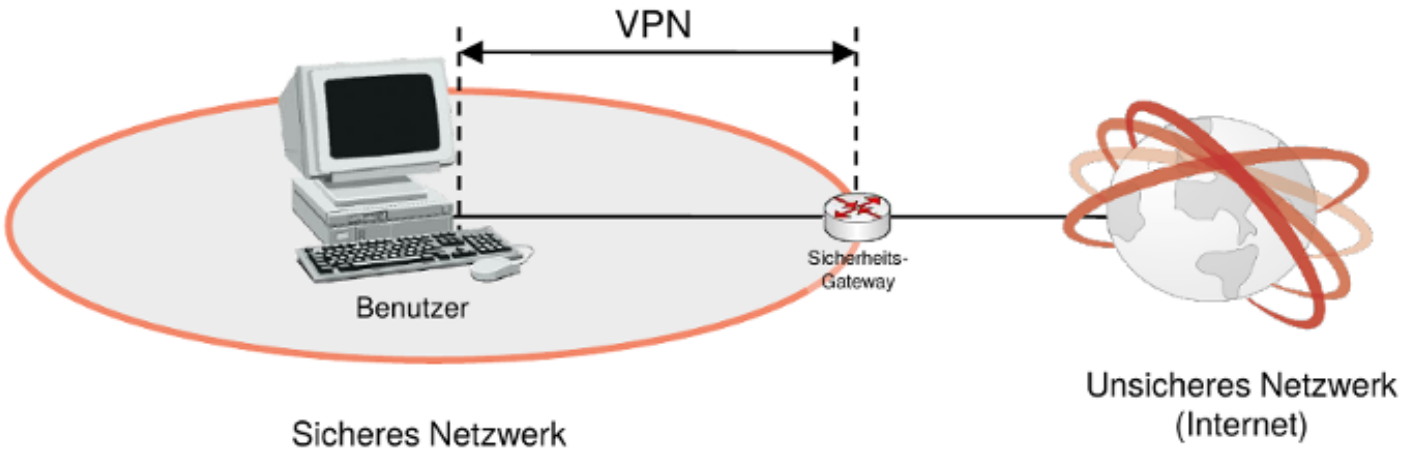
Vertrauenswürdige Endpunkte über unsichere Netzwerke verbinden:



Frage:
Was sind vertrauenswürdigen Bereiche und was ist das unsichere Netz?
Antwort:
Man braucht zuerst ein Sicherheitskonzept für die IT-Infrastruktur.

Reverse-VPN um vertrauenswürdiges Netz zu schützen

Geschützter Zugriff auf unsichere Netzwerke:



VPNs als Billig-Anonymisierungsdienste?

Aus der Werbung eines VPN-Anbieters:

Verbergen Sie Ihre Online-Identität und surfen Sie anonym im Netz!

VPNs als Billig-Anonymisierungsdienste?

Aus der Werbung eines VPN-Anbieters:

Verbergen Sie Ihre Online-Identität und surfen Sie anonym im Netz!

- VPNs anonymisieren lediglich die IP-Adresse eines Internetnutzers. Tracking erfolgt mit Cookies/EverCookies, Fingerprint. . . im Browser.

VPNs als Billig-Anonymisierungsdienste?

Aus der Werbung eines VPN-Anbieters:

Verbergen Sie Ihre Online-Identität und surfen Sie anonym im Netz!

- VPNs anonymisieren lediglich die IP-Adresse eines Internetnutzers. Tracking erfolgt mit Cookies/EverCookies, Fingerprint. . . im Browser.
- Die IP-Anonymisierung von VPNs kann relativ einfach durch Traffic Korrelation oder Traffic Fingerprinting ausgehebelt werden.

VPNs als Billig-Anonymisierungsdienste?

Aus der Werbung eines VPN-Anbieters:

Verbergen Sie Ihre Online-Identität und surfen Sie anonym im Netz!

- VPNs anonymisieren lediglich die IP-Adresse eines Internetnutzers. Tracking erfolgt mit Cookies/EverCookies, Fingerprint. . . im Browser.
- Die IP-Anonymisierung von VPNs kann relativ einfach durch Traffic Korrelation oder Traffic Fingerprinting ausgehebelt werden.
- Ein VPN Betreiber agiert wie ein Internet Zugangsprovider. Vertrauensfrage: VPN-Dienst Onavo spioniert seine Nutzer aus, AnchorFree gibt MAC und EMEI an Werbenetzwerke weiter. . .

VPNs als Billig-Anonymisierungsdienste?

Aus der Werbung eines VPN-Anbieters:

Verbergen Sie Ihre Online-Identität und surfen Sie anonym im Netz!

- VPNs anonymisieren lediglich die IP-Adresse eines Internetnutzers. Tracking erfolgt mit Cookies/EverCookies, Fingerprint. . . im Browser.
- Die IP-Anonymisierung von VPNs kann relativ einfach durch Traffic Korrelation oder Traffic Fingerprinting ausgehebelt werden.
- Ein VPN Betreiber agiert wie ein Internet Zugangsprovider. Vertrauensfrage: VPN-Dienst Onavo spioniert seine Nutzer aus, AnchorFree gibt MAC und EMEI an Werbenetzwerke weiter. . .
- VPN-Provider folgen den Gesetzen ihres Landes. Daraus kann sich eine Absenkung der Sicherheit ergeben (HideMyAss, PureVPN. . .)

Table of Contents

OSI-Schichtenmodell für Netzwerkkommunikation

7: Anwendung
6: Darstellung
5: Kommunikation

4: Transportschicht (TCP, UDP)
3: Vermittlungschicht (IP)
2: Netzwerkzugang (Ethernet)

1: Bitübertragung

OSI-Schichtenmodell für Netzwerkkommunikation

7: Anwendung
6: Darstellung
5: Kommunikation

4: Transportschicht (VPN mit TLS-Verschlüsselung)
3: Vermittlungschicht (VPN mit IPsec, WireGuard)
2: Netzwerkzugang (VPN mit L2-Encryption)

1: Bitübertragung

Einige VPN-Lösungen

OpenVPN: arbeitet auf OSI Layer 4 mit TCP, verwendet OpenSSL um eine TLS-Tunnel zwischen zwei Endpunkten aufzubauen. Site-to-Site und Site-to-End Verbindungen möglich.

Einige VPN-Lösungen

OpenVPN: arbeitet auf OSI Layer 4 mit TCP, verwendet OpenSSL um eine TLS-Tunnel zwischen zwei Endpunkten aufzubauen. Site-to-Site und Site-to-End Verbindungen möglich.

OpenConnect: arbeitet auf OSI Layer 4 mit UDP, verwendet DTLS-Tunnel zwischen zwei Endpunkten, nur Site-to-End Verbindungen.

Einige VPN-Lösungen

OpenVPN: arbeitet auf OSI Layer 4 mit TCP, verwendet OpenSSL um eine TLS-Tunnel zwischen zwei Endpunkten aufzubauen. Site-to-Site und Site-to-End Verbindungen möglich.

OpenConnect: arbeitet auf OSI Layer 4 mit UDP, verwendet DTLS-Tunnel zwischen zwei Endpunkten, nur Site-to-End Verbindungen.

IPsec: sehr komplexer Standard für VPNs auf OSI Layer 3, inzwischen gibt es vollständige Implementierungen

- IKE v1/v2 für den Schlüsselverwaltung
- AH für die Authentifizierung
- ESP für Verschlüsselung der Daten

Einige VPN-Lösungen

OpenVPN: arbeitet auf OSI Layer 4 mit TCP, verwendet OpenSSL um eine TLS-Tunnel zwischen zwei Endpunkten aufzubauen. Site-to-Site und Site-to-End Verbindungen möglich.

OpenConnect: arbeitet auf OSI Layer 4 mit UDP, verwendet DTLS-Tunnel zwischen zwei Endpunkten, nur Site-to-End Verbindungen.

IPsec: sehr komplexer Standard für VPNs auf OSI Layer 3, inzwischen gibt es vollständige Implementierungen

- IKE v1/v2 für den Schlüsselverwaltung
- AH für die Authentifizierung
- ESP für Verschlüsselung der Daten

WireGuard: Gegenentwurf zu OpenSSL und IPsec (einfach, robust. . .)

SINA L3/L2 in den Sicherheitsstufen S/E/H

Sichere Inter-Netzwerk Architektur:

- Multi-Site VPN-Lösung für mittlere - große Netze, Basis ist IPsec
- Hardware + Software Kombination (SINA Box, SINA Workstation)
- Zertifiziert für *VS-NfD* (S), *VS-Vertraulich* (E) oder *VS-Geheim* (H)

SINA L3/L2 in den Sicherheitsstufen S/E/H

Sichere Inter-Netzwerk Architektur:

- Multi-Site VPN-Lösung für mittlere - große Netze, Basis ist IPsec
- Hardware + Software Kombination (SINA Box, SINA Workstation)
- Zertifiziert für *VS-NfD* (S), *VS-Vertraulich* (E) oder *VS-Geheim* (H)

Features:

- Authentifizierung aller Identitäten über Smartcards (2FA)
- Trennung von Arbeitsumgebung und VPN (Virtualisierung, HW)
- Verschlüsselung aller Daten auf Workstations
- Trusted Network Detection für mobile Workstations
- Separate Surf-Umgebung mit inversem VPN (SafeSurfer, optional)
- Zentrales Management aller Komponenten möglich
- Hochverfügbarkeit, Backup Routing und SOLID

Table of Contents

Mögliche Angriffe auf VPNs

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**
  - 2010 konnte die NSA durch pre-computation Angriff auf den Diffie-Hellman Schlüsseltausch 60% des weltweiten VPN-Traffics on-the-fly entschlüsseln (2015: LogJam Attack publiziert)

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**
  - 2010 konnte die NSA durch pre-computation Angriff auf den Diffie-Hellman Schlüsseltausch 60% des weltweiten VPN-Traffics on-the-fly entschlüsseln (2015: LogJam Attack publiziert)
  - Man-in-the-Middle Angriffe und TLS-Downgrade Angriffe

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**
  - 2010 konnte die NSA durch pre-computation Angriff auf den Diffie-Hellman Schlüsseltausch 60% des weltweiten VPN-Traffics on-the-fly entschlüsseln (2015: LogJam Attack publiziert)
  - Man-in-the-Middle Angriffe und TLS-Downgrade Angriffe
- **Kompromittierung der Authentifizierung**
  - Pre-shared Key Auth. wird häufig geknackt (NSA: HappyDance)

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**
  - 2010 konnte die NSA durch pre-computation Angriff auf den Diffie-Hellman Schlüsseltausch 60% des weltweiten VPN-Traffics on-the-fly entschlüsseln (2015: LogJam Attack publiziert)
  - Man-in-the-Middle Angriffe und TLS-Downgrade Angriffe
- **Kompromittierung der Authentifizierung**
  - Pre-shared Key Auth. wird häufig geknackt (NSA: HappyDance)
- **Modifikation der genutzten Software**
  - Infektion des Betriebssystems mit einem Trojaner (NSA: TAO+NSP)
  - Manipulation der VPN-Software und Kryptomodule

Mögliche Angriffe auf VPNs

- **Angriffe auf die Verschlüsselung**
  - 2010 konnte die NSA durch pre-computation Angriff auf den Diffie-Hellman Schlüsseltausch 60% des weltweiten VPN-Traffics on-the-fly entschlüsseln (2015: LogJam Attack publiziert)
  - Man-in-the-Middle Angriffe und TLS-Downgrade Angriffe
- **Kompromittierung der Authentifizierung**
  - Pre-shared Key Auth. wird häufig geknackt (NSA: HappyDance)
- **Modifikation der genutzten Software**
  - Infektion des Betriebssystems mit einem Trojaner (NSA: TAO+NSP)
  - Manipulation der VPN-Software und Kryptomodule
- **Angriffe von intern** (aus dem vertrauenswürdigen Bereich)
  - Lesenswert: *I was hunting sysadmins!*
    (Interview mit einem ehem. NSA-Mitarbeiter)

Table of Contents

OpenVPN: Cipher Policies

Fähigkeiten der OpenSSL Bibliothek prüfen mit folgenden Kommandos:

```
> openvpn --show-ciphers
> openvpn --show-digests
> openvpn --show-tls
> openvpn --show-curves
```

OpenVPN: Cipher Policies

NSA Suite-B-128 Policy wählen (beim Start oder in Config Datei):
(vgl. IETF RFC 7525, BSI TR-3116-4, NIST Recommendation. . .)

```
> openvpn --tls-version-min 1.2
    --tls-cipher TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256
    --ncp-cipher AES-128-GCM:AES-128-CBC
    --auth SHA256 --dh none --ecdh-curve secp256r1
```

OpenVPN: Cipher Policies

Auswahl der Suite-B-256 Policy (erfordert 30-40% mehr Leistung):

```
> openvpn --tls-version-min 1.2
    --tls-cipher TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
    --ncp-cipher AES-256-GCM:AES-256-CBC
    --auth SHA384 --dh none --ecdh-curve secp384r1
```

OpenVPN: Authentifizierung

OpenVPN Server und Clients authentifizieren sich mit üblicherweise X509v3 Zertifikaten, die von einer CA signiert wurden. OpenVPN kann diese Daten im PEM-Format einlesen oder als PKCS12# Datei.

```
> openvpn ... --ca <datei> --cert <datei> --key <datei>
> openvpn ... --pkcs12 <datei>
```

OpenVPN: Authentifizierung

OpenVPN Server und Clients authentifizieren sich mit üblicherweise X509v3 Zertifikaten, die von einer CA signiert wurden. OpenVPN kann diese Daten im PEM-Format einlesen oder als PKCS12# Datei.

```
> openvpn ... --ca <datei> --cert <datei> --key <datei>
> openvpn ... --pkcs12 <datei>
```

In TLS 1.2 werden Zertifikate der Nutzer für die Authentifizierung unverschlüsselt übertragen (unschön).

OpenVPN: Authentifizierung

OpenVPN Server und Clients authentifizieren sich mit üblicherweise X509v3 Zertifikaten, die von einer CA signiert wurden. OpenVPN kann diese Daten im PEM-Format einlesen oder als PKCS12# Datei.

```
> openvpn ... --ca <datei> --cert <datei> --key <datei>
> openvpn ... --pkcs12 <datei>
```

In TLS 1.2 werden Zertifikate der Nutzer für die Authentifizierung unverschlüsselt übertragen (unschön).

OpenVPN v.2.4 kann den TLS-Handshake zusätzlich verschlüsseln.

1. Pre-shared TA Key erzeugen (unkritisch, auf VPN-Server):

   ```
   > openvpn --genkey --secret ta.key
   ```

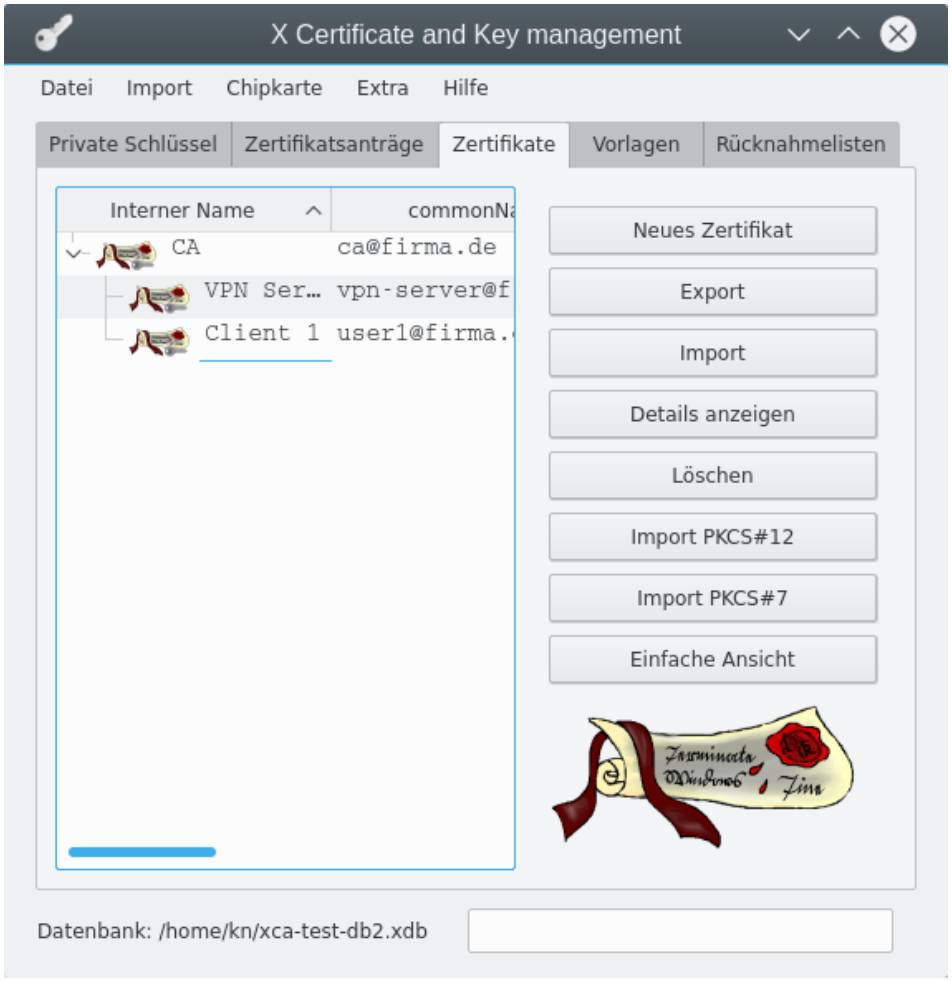
2. TA-Key bei Server und Client verwenden (pre-shared):

   ```
   > openvpn ... --tls-crypt ta.key
   ```

OpenVPN: Empfehlungen für PKI

Die PKI für X509v3 Zertifikate ist eine sicherheitskritische Komponente!

- PKI auf abgesichertem Rechner ohne Internetzugang betreiben
- Softwareempfehlung: XCA
- Empfehlung für CA Key (BSI):
  - RSA Key mit 3072 Bit
  - Digest-Algo: SHA256
  - max. 10 Jahre gültig
- Hardware Security Modul für CA Key verwenden (HSM)
- mind. 1 Backup des CA Key und Zertifikat sicher verwahren



OpenVPN: 2-Faktor-Authentifizierung

Weiche 2-Faktor-Authentifizierung (*Besitz* als kopierbare Bytes)

1. Privaten Key der X509v3 Zertifikate mit Passwort schützen.
2. Login auf dem VPN-Server mittels PAM-Auth erzwingen:

```
plugin /usr/share/openvpn/plugin/lib/openvpn-auth-pam.so
```

OpenVPN: 2-Faktor-Authentifizierung

Weiche 2-Faktor-Authentifizierung (*Besitz* als kopierbare Bytes)

1. Privaten Key der X509v3 Zertifikate mit Passwort schützen.
2. Login auf dem VPN-Server mittels PAM-Auth erzwingen:

```
plugin /usr/share/openvpn/plugin/lib/openvpn-auth-pam.so
```

Harte 2-Faktor-Authentifizierung (physikalischer Besitz eines Token)

- Private Schlüssel der Zertifikate auf HSM oder OpenPGP Smartcard

OpenVPN: 2-Faktor-Authentifizierung

Weiche 2-Faktor-Authentifizierung (*Besitz* als kopierbare Bytes)

1. Privaten Key der X509v3 Zertifikate mit Passwort schützen.
2. Login auf dem VPN-Server mittels PAM-Auth erzwingen:

```
plugin /usr/share/openvpn/plugin/lib/openvpn-auth-pam.so
```

Harte 2-Faktor-Authentifizierung (physikalischer Besitz eines Token)

- Private Schlüssel der Zertifikate auf HSM oder OpenPGP Smartcard
- Serialized-ID des Key auf dem HSM ermitteln:

```
> openvpn --show-pkcs11-ids /usr/lib/pkcs11/...
...
```

- Beim Start des OpenVPN Client den PKCS11# Provider laden und die Serialized-ID des Key auf dem HSM angeben:

```
> openvpn ... --pkcs11-providers /usr/lib/pkcs11/...
              --pkcs11-id 'aaaa/bbb/41545F5349474E41....
```

OpenVPN: Nutzer sperren

Mitarbeiter wechseln den Arbeitgeber, Zertifikate werden kompromittiert....

OpenVPN: Nutzer sperren

Mitarbeiter wechseln den Arbeitgeber, Zertifikate werden kompromittiert....

OpenVPN kann eine Certificate Revocation List (CRL) verwenden:

1. In der PKI das Zertifikat des Nutzers sperren.
2. Certificate Revocation List (CRL) exportieren.
3. CRL auf dem VPN-Server importieren:

```
> openvpn ... --crl-verify crl.pem
```

Table of Contents

WireGuard: Installation

- Für Ubuntu gibt es ein PPA-Repository:

```
> sudo add-apt-repository ppa:wireguard/wireguard
> sudo apt update
> sudo apt install wireguard-dkms wireguard-tools
```

- Für Debian ist das Paket *wireguard* aus *unstable* zu instalieren.
- Anleitungen für weitere Linux Distributionen auf der Webseite
- WireGuard für MacOS X installieren:

```
$ brew install wireguard-tools
```

WireGuard: Cipher Policies

Wireguard unterstützt nur einen Satz Krypto-Algorithmen:

- Ed25519 für Public-Key-Authentifizierung
- Curve25519 für Schlüsseltausch mit ECDHE
- ChaCha20-Poly1309 für Verschlüsselung der Daten

WireGuard: Authentifizierung

WireGuard verwendet Public-Key Kryptografie für Authentifizierung.

1. Auf jedem Peer einen Schlüsselpaar generieren:

```
# wg genkey > private.key
# chmod 700 private.key
# wg pubkey < private.key
```

2. Die öffentlichen Schlüssel zwischen allen Peers austauschen.

WireGuard: Server starten und Peers definieren

Full-Text-Adventure im Terminal:

```
# ip link add dev wg0 Wireguard
# ip address add dev wg0 192.168.111.1/24
# wg set wg0 private-key <datei>
# ip link set wg0 up
# wg set peer <pubkey-c2> allowed-ips 192.168.111.2/32
# wg set peer <pubkey-c3> allowed-ips 192.168.111.3/32
...
# sysctl -w net.ipv4.ip_forward=1
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

WireGuard: Server starten und Peers definieren

Full-Text-Adventure im Terminal:

```
# ip link add dev wg0 Wireguard
# ip address add dev wg0 192.168.111.1/24
# wg set wg0 private-key <datei>
# ip link set wg0 up
# wg set peer <pubkey-c2> allowed-ips 192.168.111.2/32
# wg set peer <pubkey-c3> allowed-ips 192.168.111.3/32
...
# sysctl -w net.ipv4.ip_forward=1
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Konfiguration speichern:

```
# wg showconf wg0 > /etc/wireguard/wg0.conf
```

Konfiguration laden:

```
# wg setconf wg0
```

WireGuard: Client starten und mit Server verbinden

Full-Text-Adventure im Terminal:

```
# ip link add dev wg0 Wireguard
# ip address add dev wg0 192.168.111.2/24
# wg set wg0 private-key <datei>
# ip link set wg0 up
# wg set peer <pubkey-server> allowed-ips 0.0.0.0/0
        endpoint <server-ip>:<wg-port>
```

WireGuard: Client starten und mit Server verbinden

Full-Text-Adventure im Terminal:

```
# ip link add dev wg0 Wireguard
# ip address add dev wg0 192.168.111.2/24
# wg set wg0 private-key <datei>
# ip link set wg0 up
# wg set peer <pubkey-server> allowed-ips 0.0.0.0/0
          endpoint <server-ip>:<wg-port>
```

Konfiguration speichern:

```
# wg showconf wg0 > /etc/wireguard/wg0.conf
```

Wireguard starten, Peers konf., DNS und Routen setzen usw.:

```
# wg-quick up wg0
```

Table of Contents

IPsec ist ein sehr komplexer und flexibler Standard

- IKE v.1 (veraltet) und IKE v.2 (charon) für den Schlüsseltausch.
- Authentifizierung erfolgt auf zwei Ebenen:
  1. Auth. des Gerätes (X509v3, OpenPGP, via TPM-2.0)
  2. Auth. des Nutzers (RADIUS, XAuth, Smartcard, SSH...)
- Für die Verschlüsselung (AH, ESP) gibt es zwei Modi:
  1. Transport-Modus: nur Payload wird verschlüsselt, IP-Header bleibt.
  2. Tunnel-Modus: Payload und IP-Header werden verschlüsselt, zusätzliche IP-Header für die Endpunkte werden eingefügt.
- ...

Komponenten eines IPsec Netzwerkes müssen identisch arbeiten!
Die Konfiguration muss dokumentiert, aktuell und nachvollziehbar sein.

strongSwan: Konfigurationsdateien

strongSwan (Linux) verwendet folgende Konfigurationsdateien:

1. Definition der IPsec Verbindungen in */etc/ipsec.conf*

```
conn %default
    ...
conn <vpn-1>
    ...
conn <vpn-2>
    ...
```

2. Privaten Schlüssel und Auth.-Daten in */etc/ipsec.secrets*
3. Laden und Konfiguration von Modulen in */etc/strongswan.conf*

```
charon {
  load_modular = yes
  plugins { include strongswan.d/charon/*.conf }
}
```

IPsec Crypto Policy Suite-B-128

Konfiguration in */etc/ipsec.conf* (Linux):

```
conn %default
    ikelifetime = 60m
    keylife = 20m
    rekeymargin = 3m
    keyingtries = 1
    keyexchange = ikev2

    ike = aes128gcm16-prfsha256-ecp256
    esp = aes128gcm16-ecp256

    leftauth = rsa-2048-sha256-ecdsa-256-sha256
    rightauth = rsa-2048-sha256-ecdsa-256-sha256
```

IPsec Crypto Policy Suite-B-128

Konfiguration in der PowerShell (Win10):

```
PS C:\> Add-VPNConnection -Name "myvpn"
         -ServerAddress 1.2.3.4 -TunnelType "Ikev2"
         -EncryptionLevel "Required" ...

PS C:\> Set-VpnConnectionIPsecConfiguration
         -ConnectionName "myvpn" -DHGroup ECP256
         -AuthenticationTransformConstants SHA256128
         -CipherTransformConstants GCMAES128
         -EncryptionMethod AES128 -PfsGroup ECP256
         -IntegrityCheckMethod SHA256 ...
```

IPsec Crypto Policy Suite-B-256

Konfiguration in */etc/ipsec.conf* (Linux):

```
conn %default
    ikelifetime = 60m
    keylife = 20m
    rekeymargin = 3m
    keyingtries = 1
    keyexchange = ikev2

    ike = aes256gcm16-prfsha384-ecp384
    esp = aes256gcm16-ecp384

    leftauth = rsa-3072-sha256-ecdsa-384-sha256
    rightauth = rsa-3072-sha256-ecdsa-384-sha256
```

IPsec Crypto Policy Post-Quantum (NTRU, BLISS-B)

Konfiguration in */etc/ipsec.conf* (Linux, strongSwan 5.3.0+):

```
conn %default
    ikelifetime = 60m
    keylife = 20m
    rekeymargin = 3m
    keyingtries = 1
    keyexchange = ikev2

    ike = chacha20poly1305-prfsha256-ntru256!
    esp = chacha20poly1305-ntru256!

    leftauth = ntru256-bliss
    rightauth = ntru256-bliss
```

IPsec Crypto Policy Post-Quantum (NTRU, BLISS-B)

Konfiguration in */etc/ipsec.conf* (Linux, strongSwan 5.3.0+):

```
conn %default
    ikelifetime = 60m
    keylife = 20m
    rekeymargin = 3m
    keyingtries = 1
    keyexchange = ikev2

    ike = chacha20poly1305-prfsha256-ntru256!
    esp = chacha20poly1305-ntru256!

    leftauth = ntru256-bliss
    rightauth = ntru256-bliss
```

CCS'17: *To BLISS-B or not to be - Attacking strongSwan's Implementation of Post-Quantum Signatures* (30. Nov. 2017)

IPsec Crypto Policy Suite-A

Suite-A Kryptografie kann man mit IPsec nutzen (modulares Konzept).

IPsec Crypto Policy Suite-A

Suite-A Kryptografie kann man mit IPsec nutzen (modulares Konzept).

Der Einsatz von Suite-A Kryptografie ist nur sinnvoll,
wenn das Gesamtkonzept der IT darauf abgestimmt ist.

- Standard UEFI-BIOS mit Intel Management Engine?
- Secure Boot Schlüsselmaterial von Microsoft?
- ...

IPsec Crypto Policy Suite-A

Suite-A Kryptografie kann man mit IPsec nutzen (modulares Konzept).

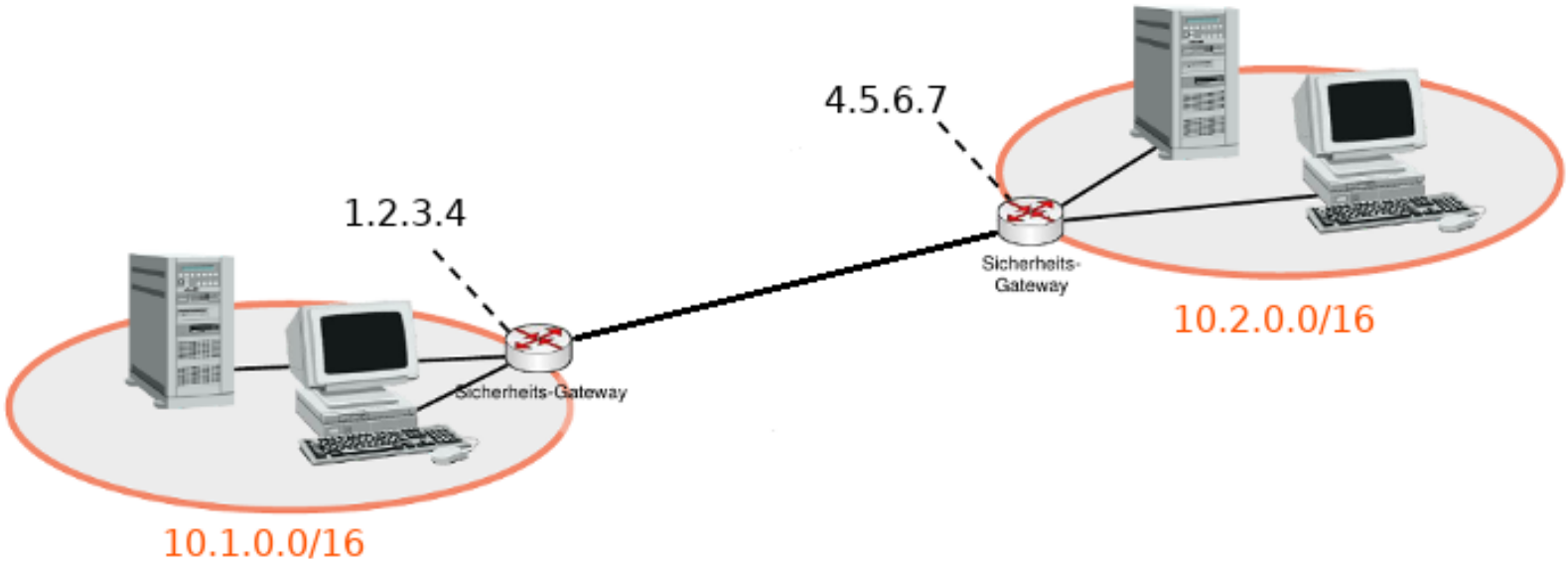
Der Einsatz von Suite-A Kryptografie ist nur sinnvoll,
wenn das Gesamtkonzept der IT darauf abgestimmt ist.

- Standard UEFI-BIOS mit Intel Management Engine?
- Secure Boot Schlüsselmaterial von Microsoft?
- ...

Wenn Ihre Kommunikation als *Geheim* klassifiziert ist, dann kann die **secunet** Security Networks AG eine Lösung liefern.

Beispiel 1: Verbindung Server-2-Server

Zwei vertrauenswürdige Netzwerke sollen verbunden werden:



- Authentifizierung der VPN-Server erfolgt mit selbstsig. Zertifikaten
- Zertifikate sind pre-shared auf der Gegenstelle vorhanden (keine CA)
- Authentifizierung der Nutzer wird nicht genutzt

Beispiel 1: Verbindung Server-2-Server

Konfiguration der Verbindung in */etc/ipsec.conf*

```
conn serverA-zu-serverB
  left= 1.2.3.4
  leftcert = serverA.pem
  leftsendcert = never
  leftsubnet = 10.1.0.0/16
  leftfirewall = yes

  right = 5.6.7.8
  rightcert = serverB.pem
  rightsubnet=10.2.0.0/16
  auto = add
```

Beispiel 1: Verbindung Server-2-Server

Konfiguration der Verbindung in */etc/ipsec.conf*

```
conn serverA-zu-serverB
  left= 1.2.3.4
  leftcert = serverA.pem
  leftsendcert = never
  leftsubnet = 10.1.0.0/16
  leftfirewall = yes

  right = 5.6.7.8
  rightcert = serverB.pem
  rightsubnet=10.2.0.0/16
  auto = add
```

```
conn serverB-zu-serverA
  left= 5.6.7.8
  leftcert = serverB.pem
  leftsendcert = never
  leftsubnet = 10.2.0.0/16
  leftfirewall = yes

  right = 1.2.3.4
  rightcert = serverA.pem
  rightsubnet=10.1.0.0/16
  auto = add
```

Beispiel 1: Verbindung Server-2-Server

Konfiguration der Verbindung in */etc/ipsec.conf*

```
conn serverA-zu-serverB
  left= 1.2.3.4
  leftcert = serverA.pem
  leftsendcert = never
  leftsubnet = 10.1.0.0/16
  leftfirewall = yes
  right = 5.6.7.8
  rightcert = serverB.pem
  rightsubnet=10.2.0.0/16
  auto = add
```

```
conn serverB-zu-serverA
  left= 5.6.7.8
  leftcert = serverB.pem
  leftsendcert = never
  leftsubnet = 10.2.0.0/16
  leftfirewall = yes
  right = 1.2.3.4
  rightcert = serverA.pem
  rightsubnet=10.1.0.0/16
  auto = add
```

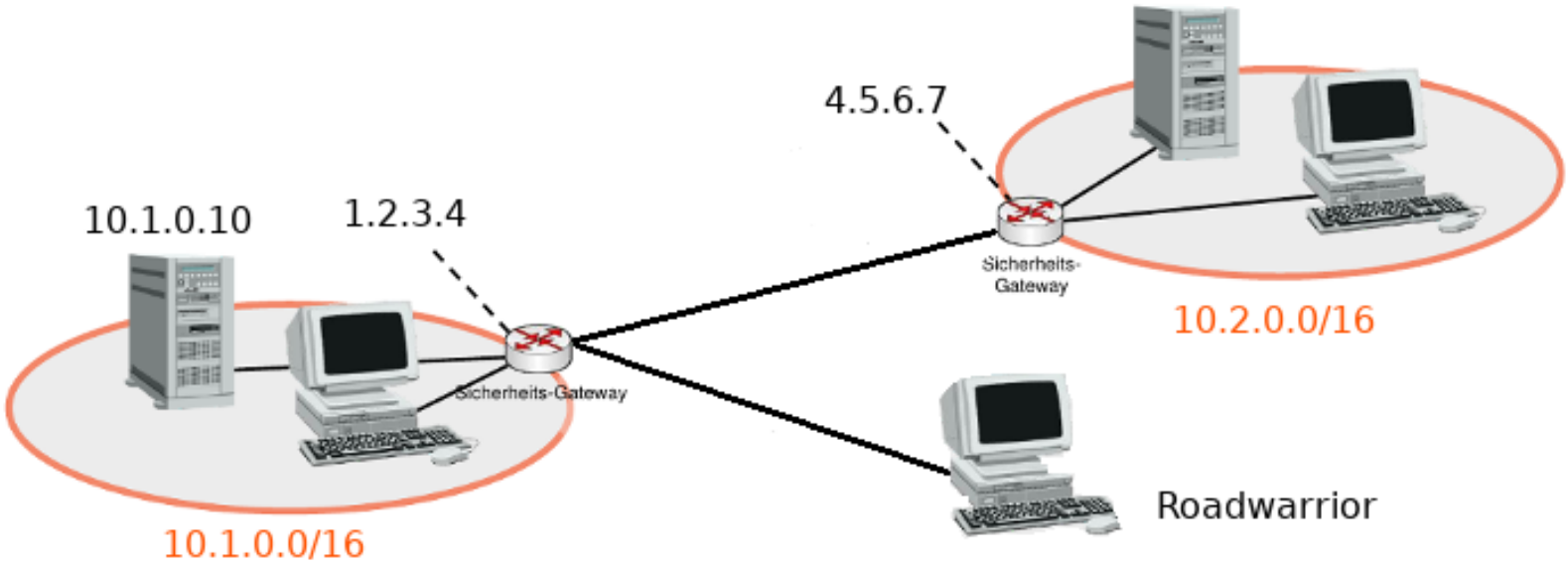
Private Keys werden in */etc/ipsec.secrets* konfiguriert:

```
: RSA serverA.key
```

```
: RSA serverB.key
```

Beispiel 2: Anbindung externer Clients

Ext. Clients (roadwarrior) sollen auf einen internen Server zugreifen:



- Roadwarrior sollen auf den Server 10.1.0.10 zugreifen können.
- Authentifizierung erfolgt mit X509v3 Zertifikaten (PKI-basiert).
- Für den Zugriff via VPN erhalten die Clients eine IP 10.3.0.0/24.

Beispiel 2: Anbindung externer Clients

Konfiguration der Verbindung in */etc/ipsec.conf*

```
conn serverA-zu-rw
  left= 1.2.3.4
  leftcert = serverC.pem
  leftid = @s1.firma.tld
  leftsubnet = 10.1.0.10/32
  leftfirewall = yes

  right = %any
  rightid = *@firma.tld
  rightsourceip = 10.3.0.0/24

  auto = add
  type = transport
```

Beispiel 2: Anbindung externer Clients

Konfiguration der Verbindung in */etc/ipsec.conf*

```
conn serverA-zu-rw
  left= 1.2.3.4
  leftcert = serverC.pem
  leftid = @s1.firma.tld
  leftsubnet = 10.1.0.10/32
  leftfirewall = yes

  right = %any
  rightid = *@firma.tld
  rightsourceip = 10.3.0.0/24

  auto = add
  type = transport
```

```
conn rw-max-mustermann
  left= %any
  leftcert = max-cert.pem
  leftid = @max.firma.tld
  leftsourceip = %config
  leftfirewall = yes

  right = 1.2.3.4
  rightid = @s1.firma.tld
  rightsubnet = 10.1.0.10/32

  auto = add
  type = transport
```

Beispiel 2: Anbindung externer Clients

Gesamten Traffic der Clients durch das VPN zur Firma schicken,
um Sicherheitsrichtlinien zu enforcen (Data Leak Prevention u.ä):

```
conn serverA-zu-rw
  ...
  leftsubnet = 0.0.0.0/0
  ...
```

```
conn rw-max-mustermann
  ...
  rightsubnet = 0.0.0.0/0
  ...
```

Beispiel 2: Anbindung externer Clients

Gesamten Traffic der Clients durch das VPN zur Firma schicken,
um Sicherheitsrichtlinien zu enforcen (Data Leak Prevention u.ä):

```
conn serverA-zu-rw
  ...
  leftsubnet = 0.0.0.0/0
  ...
```

```
conn rw-max-mustermann
  ...
  rightsubnet = 0.0.0.0/0
  ...
```

Validierung der Zertifikate:

- CA-Zertifikate in */etc/ipsec.d/cacerts* speichern
- Certificate Revocation Lists (CRL) in */etc/ipsec.d/crls* ablegen
  (oder via HTTP- bzw. LDAP-Uri abrufen, via OCSP validieren...)

strongSwan: weitere Beispiele

Weitere 135 Beispielkonfigurationen für IPsec mit IKE v.2 findet man im Wiki auf der strongSwan Webseite: https://www.strongswan.org

strongSwan: weitere Beispiele

Weitere 135 Beispielkonfigurationen für IPsec mit IKE v.2 findet man im Wiki auf der strongSwan Webseite: https://www.strongswan.org

Ich danke für Ihre Aufmerksamkeit.