# Server-Konfiguration

**–verify-x509-name name type** Accept connections only if a host's X.509 name is equal to **name.** The remote host must also pass all other tests of verification.

Which X.509 name is compared to **name** depends on the setting of type. **type** can be "subject" to match the complete subject DN (default), "name" to match a subject RDN or "name-prefix" to match a subject RDN prefix. Which RDN is verified as name depends on the **–x509-username-field** option. But it defaults to the common name (CN), e.g. a certificate with a subject DN "C=KG, ST=NA, L=Bishkek, CN=Server-1" would be matched by:

**–verify-x509-name 'C=KG, ST=NA, L=Bishkek, CN=Server-1'** and **–verify-x509-name Server-1 name** or you could use **–verify-x509-name Server- name-prefix** if you want a client to only accept connections to "Server-1", "Server-2", etc.

**–verify-x509-name** is a useful replacement for the **–tls-verify** option to verify the remote host, because **–verify-x509-name** works in a **–chroot** environment without any dependencies.

Using a name prefix is a useful alternative to managing a CRL (Certificate Revocation List) on the client, since it allows the client to refuse all certificates except for those associated with designated servers.

**NOTE:** Test against a name prefix only when you are using OpenVPN with a custom CA certificate that is under your control. Never use this option with type "name-prefix" when your client certificates are signed by a third party, such as a commercial web CA

# Management Console

Die Management Konsole läuft auf localhost und ist über P. 7505 erreichbar.

```
root@server6:/etc/openvpn/staticclients# telnet localhost 7505
```

Beenden mit quit.

```
INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info
help
Management Interface for OpenVPN 2.3.10 x86_64-pc-linux-gnu [SSL (OpenSSL)]
[LZO] [EPOLL] [PKCS11] [MH] [IPv6] built on Feb  2 2016
Commands:
auth-retry t           : Auth failure retry mode (none,interact,nointeract).
bytecount n            : Show bytes in/out, update every n secs (0=off).
echo [on|off] [N|all]  : Like log, but only show messages in echo buffer.
exit|quit              : Close management session.
forget-passwords       : Forget passwords entered so far.
help                   : Print this message.
hold [on|off|release]  : Set/show hold flag to on/off state, or
                         release current hold and start tunnel.
```

```
kill cn                 : Kill the client instance(s) having common name cn.
kill IP:port            : Kill the client instance connecting from IP:port.
load-stats              : Show glsobal server load stats.
log [on|off] [N|all]    : Turn on/off realtime log display
                          + show last N lines or 'all' for entire history.
mute [n]                : Set log mute level to n, or show level if n is
absent.
needok type action      : Enter confirmation for NEED-OK request of 'type',
                          where action = 'ok' or 'cancel'.
needstr type action     : Enter confirmation for NEED-STR request of 'type',
                          where action is reply string.
net                     : (Windows only) Show network info and routing table.
password type p         : Enter password p for a queried OpenVPN password.
remote type [host port] : Override remote directive, type=ACCEPT|MOD|SKIP.
proxy type [host port flags] : Enter dynamic proxy server info.
pid                     : Show process ID of the current OpenVPN process.
pkcs11-id-count         : Get number of available PKCS#11 identities.
pkcs11-id-get index     : Get PKCS#11 identity at index.
client-auth CID KID     : Authenticate client-id/key-id CID/KID (MULTILINE)
client-auth-nt CID KID  : Authenticate client-id/key-id CID/KID
client-deny CID KID R [CR] : Deny auth client-id/key-id CID/KID with log
reason
                             text R and optional client reason text CR
client-kill CID [M]     : Kill client instance CID with message M
(def=RESTART)
env-filter [level]      : Set env-var filter level
client-pf CID           : Define packet filter for client CID (MULTILINE)
rsa-sig                 : Enter an RSA signature in response to>RSA_SIGN
challenge
                          Enter signature base64 on subsequent lines followed
by END
signal s                : Send signal s to daemon,
                          s = SIGHUP|SIGTERM|SIGUSR1|SIGUSR2.
state [on|off] [N|all] : Like log, but show state history.
status [n]              : Show current daemon status info using format #n.
test n                  : Produce n lines of output for testing/debugging.
username type u         : Enter username u for a queried OpenVPN username.
verb [n]                : Set log verbosity level to n, or show if n is
absent.
version                 : Show current version number.
```

# Debugging auf OpenVPN Client Seite (Linux)

```
journalctl -fu NetworkManager
```

# Client IPs fest zuweisen

In die *.conf kommt eine neue Direktive:

```
client-config-dir /etc/openvpn/staticclients
```

In diesem Verzeichnis für jeden Client einen Datei openvpn_dvsdnet_[name] legen. Diese enthält die IP Adresse und die Netzmaske des Clients:

```
ifconfig-push 192.168.50.16 255.255.255.0
```

OpenVPN liest diese Datei beim Connect zusätzlich ein, aber DNS und Gateway kommen weiterhin über die zentralen push Kommandos. Ggf kann man auch ein Client-spezifisches Push anhängen, siehe dazu http://michlstechblog.info/blog/openvpn-set-a-static-ip-address-for-a-client/

Quelle: https://github.com/OpenVPN/easy-rsa

# EASYRSA: CA einrichten

```
./easyrsa init-pki
./easyrsa build-ca
```

DH erzeugen

```
./easyrsa gen-dh
```

# Zertifikate erzeugen

Signing Request (CSR) erzeugen, mit **nopass** = Key **ohne** Passwort

```
./easyrsa gen-req EntityName

./easyrsa gen-req EntityName nopass
```

danach signieren mit

```
./easyrsa sign-req server EntityName

./easyrsa sign-req client EntityName
```

*server* und *client* bestimmt, ob es ein Server oder Client Zertifikat ist.

## Achtung bei OpenVPN

Der Client sollte den im OpenVPN Zertifikat angegebenen Common Name prüfen. Server prüft seinerseits den Zertifikatstyp des Clients (RFC3280):

```
# Verification of certs
# Details: https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
# old method (Name/name-prefix from CN field)
# verify-x509-name locutus.netzwissen.local name
# new method from RFC3280: type of certificate must be client
remote-cert-eku "TLS Web Client Authentication"
```

# Zertifikate zurückziehen

```
./easyrsa revoke server EntityName
```

Danach mit easyrsa gen-crl die zurückgezogenen zertifikate in die crl aufnehmen.

pki/index.txt zeigt, welche Zertifikate zurückkgezogen wurden.

# Inhalte kontrollieren

#### CSR

```
openssl req -in www2.netzwissen.de.csr -text -noout
```

#### Zertifikat

```
openssl x509 -in certificate.crt -text -noout
```