

Virtual Private Networks mit OpenVPN

SVEN LEUPOLD

17. Mai 2008

Danksagung

Für die Unterstützung beim Schreiben dieses Artikels danke ich insbesondere:

THOMAS HÄNNI (INDATO GMBH), ROLF SCHMUTZ (NETLABS GMBH), PHILIPP RODRIGUEZ (CLARO)
NICOLE ULRICH, JIRI DVORAK, CHARLES BUECHE UND ALLEN HIER UNGE-
NANNTEN.

Lizenz

Diese Arbeit untersteht der Creative Commons Share Alike License. Für eine Kopie dieser Lizenz gehen Sie unter (a) <http://creativecommons.org/licenses/by-sa/2.5/ch/> oder (b) senden Sie einen Brief an Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA

Inhaltsverzeichnis

1	VPN Grundlagen	4
2	VPN mit OpenVPN	4
3	OpenVPN mit Server/Client Zertifikat im Detail	5
4	Ein kommentiertes Beispiel	7
5	Installation und Konfiguration OpenVPN Server (UNIX)	9
6	Installation und Konfiguration OpenVPN Client (UNIX)	10
7	OpenVPN Client unter Microsoft Windows	12
8	Praktische Erfahrungen mit OpenVPN	13
9	Verschlüsselt gleich sicher?	14
10	Anfragen	14
A	OpenVPN Server Konfiguration	15
B	OpenVPN Client Konfiguration	16
C	Optionale Authentisierung mit LDAP	17

1 VPN Grundlagen

Ein Virtual Private Network (VPN) ist ein logisches Netzwerk, welches ein physisches Netzwerk zum Transport von Daten benutzt. Das physische Netzwerk kann z.B. das Internet oder ein Wireless LAN (WLAN) sein. VPNs „tunneln“ Datenpakete über das Transportnetz. Die Bezeichnung „private“ bezieht sich auf die verwendeten Adressbereiche innerhalb des Tunnels. Neben dem reinen Tunneln durch fremde Netze bieten viele Implementierungen von VPN die Möglichkeit, den Datenstrom zu verschlüsseln. So lassen sich selbst kritische Unternehmensdaten über unsichere Transportnetze übertragen. Typische Anwendungsfälle von VPNs sind daher oft Remote Access via Internet oder sichere Site-To-Site Verbindungen.

VPN Lösungen lassen sich grob in Hard- und Softwarelösungen unterteilen. Viele namhafte Hersteller von Netzwerkequipment bieten spezielle Hardwarelösungen an. Vermehrt trifft man auch auf sogenannte Appliance Lösungen, also fixfertige Geräte, die man nur an den Standorten aufstellt und die bereits alle Funktionen eingebaut haben. Auf der anderen Seite gibt es eine Vielzahl von VPN Softwarelösungen. Hier realisiert eine Software oder ein Betriebssystem die Funktion eines VPN Servers bzw. Clients. Viele neue Software VPN Lösungen forcieren SSL/TLS als Technologie, da sich hier Tunneling und Verschlüsselung kombinieren lassen. Alle VPN Lösungen müssen Methoden enthalten, um einen unautorisierten Zugriff abzuwehren. D.h. es muss geprüft werden, ob die Gegenseite (peer) berechtigt ist, Daten über das VPN zu transportieren. Die sichere Authentisierung ist mindestens so wichtig, wie eine gute Verschlüsselung. Bei der Wahl der optimalen Lösung sollten Kriterien wie Sicherheitsanforderungen, Interoperabilität, Betriebsaufwand und Leistungsfähigkeit in die Entscheidung einfließen.

2 VPN mit OpenVPN

OpenVPN¹ ist eine vielversprechende Open Source Software VPN Lösung basierend auf SSL/TLS. OpenVPN verwendet die bewährte und als sicher eingestufte openssl Library zur Verschlüsselung und Authentisierung. Damit lassen sich höchste Sicherheitsanforderungen realisieren. OpenVPN benutzt standardmässig das robuste User

¹openssl <http://www.openssl.org>

Datagram Protocol (UDP) als Transportschicht. Der Protokoll Overhead ist sehr gering, was sich günstig auf den Datendurchsatz auswirkt.

OpenVPN bietet eine Reihe von Authentisierungsfunktionen an. Der Anwender kann, je nach Anforderungen, zwischen gemeinsamen Schlüssel (*shared secret*), Zertifikat und Authentisierungs-Plugin wählen. Kombinationen mehrerer Methoden miteinander sind möglich. Shared Secret Authentisierung ist nur für peer to peer VPN geeignet. Die Lösung skaliert aber nicht für peer to multipeer Topologien. Hier kommen Clients mit X.509 Zertifikaten zum Einsatz. Die Clients müssen ein gültiges Zertifikat vorweisen, um sich am VPN anzumelden.

OpenVPN kann mit Erweiterungen (*Plugins*) flexibel mit anderen Authentisierungsformen ergänzt werden. Beispielsweise kann zusätzlich ein Benutzername und ein Passwort an einem RADIUS oder LDAP Directory Server geprüft werden.

Die komplexen Verschlüsselungs- und Authentisierungsmethoden sind mit OpenVPN vergleichsweise einfach zu verwalten. Eine Konfigurationsdatei steuert die wichtigsten Einstellungen.

Neben dem Tunneling von IP Paketen beherrscht OpenVPN auch Ethernet Bridging. Layer 2 Protokolle (z.B. NetBUI, DHCP) und Broadcasts können so über das VPN transportiert werden. Applikationen, die von diesen Protokollen abhängig sind, können transparent über das VPN benutzt werden.

3 OpenVPN mit Server/Client Zertifikat im Detail

OpenVPN folgt dem klassischen Client/Server Ansatz. Ein OpenVPN Client initiiert eine Verbindung zum OpenVPN Server. Der OpenVPN Server weist sein X.509 Zertifikat vor, welches der Client auf Echtheit prüft. Die Prüfung erfolgt anhand des public keys der ausgebenden *Certificate Authority* (CA). Wurde der OpenVPN Server für authentisch befunden, setzt die 1. Stufe der Client Authentisierung ein. Erfolgt die Authentisierung des Clients mit einem X. 509 Client Zertifikat, prüft der OpenVPN Server das vom Client übermittelte Zertifikat anhand des public keys der ausgebenden CA. Ist das Client Zertifikat gültig und nicht in der Certificate Revocation List (CRL) der CA, ist diese Authentisierungstufe abgeschlossen und die verschlüsselte Verbindung kann beginnen. Standardmässig werden mit SSL/TLS die Sitzungsschlüssel (*session keys*) dynamisch

ausgehandelt. Up- und Downstream verwenden jeweils eigene Keys zum verschlüsseln und signieren.

Wurde ein *shared secret* konfiguriert, ist die Authentisierung mit der Verschlüsselung gekoppelt. Ohne den richtigen Schlüssel kann keine verschlüsselte Verbindung aufgebaut werden. Bei dieser Variante wird ein gemeinsamer Schlüssel (bestehend aus 4 Subkeys) verwendet, um die Datenpakete zu verschlüsseln und zu signieren.

Optional kann jetzt eine weitere Authentisierungsmethode gestartet werden (plugin). Beispielsweise schickt der Client den Benutzername und ein Passwort mit, welches das Plugin prüft und das Ergebnis an den OpenVPN Server zurückgibt. Ist auch diese Authentisierung erfolgreich, darf der Client am VPN teilnehmen.

Der Client bezieht seine private Adresse vom OpenVPN Server und bekommt alle notwendige Einstellungen (DNS, WINS etc.) über den Tunnel mitgeteilt. Als Tunnelendpunkte werden virtuelle Interfaces (TUN/TAP) verwendet. Die Routingtabelle des Clients wird nunmehr so manipuliert, dass alle Pakete durch den Tunnel geschickt werden. Auch serverseitig wird die Kommunikation über ein virtuelles Interface abgehandelt. Die getunnelten IP Pakete werden „ausgepackt“ und anschliessend mit Network Address Translation (NAT) auf eine oder mehrere Adressen (*address pool*) im lokalen Netz übersetzt.

4 Ein kommentiertes Beispiel

Anhand eines Beispiels sollen die Ausführungen unter Kapitel 2 illustriert werden. Eine Firma mit einem Netz am Hauptsitz (192.168.10.0/24) realisiert mit OpenVPN eine Remote Access Lösung über das Internet.

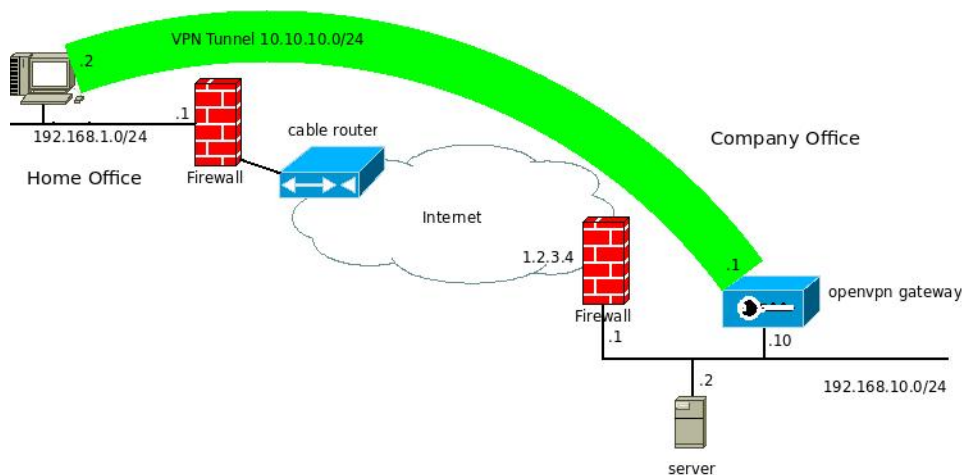


Abbildung 1: Beispiel Netzwerk Topologie

Die Mitarbeiter sind am Heimarbeitsplatz (*home office*) mit ADSL oder Kabel an ihrem lokalen Internet Service Provider (ISP) angeschlossen. Das Heimbüro verwendet den privaten Adressbereich² 192.168.1.0/24. Dieser Adressbereich wird vom ISP nicht reroutet, d.h. eine direkte IP Verbindung zwischen dem *home office* und dem Firmennetz ist nicht möglich. Erst ein OpenVPN Tunnel verbindet die beiden Netze miteinander.

Bevor der Client die VPN Verbindung startet, sieht seine Routingtabelle wie folgt aus:

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

Der Client ist mit einem Ethernet Interface (eth0) am lokalen Netzwerk angeschlossen. Alle Adressen, die nicht im lokalen Netz liegen, werden an den Router 192.168.1.1 (default gateway) geleitet.

²RFC 1918

Die VPN Verbindung wird auf der Kommandozeile oder GUI gestartet.

```
$ /etc/init.d/openvpn start
Enter Auth Username: *****
Enter Auth Password: *****
Enter Private Key Password: *****
```

Nach erfolgreicher Authentisierung und dem Aufbau des Tunnels verändert sich die Routingtabelle des Clients:

Ziel	Router	Genmask	Flags	MSS	Fenster	irtt	Iface
1.2.3.4	192.168.1.1	255.255.255.255	UGH	0	0	0	eth1
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
10.10.10.0	0.0.0.0	255.255.255.0	U	0	0	0	tap0
0.0.0.0	10.10.10.1	0.0.0.0	UG	0	0	0	tap0

Die IP Adresse des OpenVPN Servers (1.2.3.4) wird als statische Route eingefügt. Das Netz 10.10.10.0/24 bildet das logische Netzwerk. Das virtuelle Interface (tap0) bekommt vom OpenVPN Server die Adresse 10.10.10.2 zugewiesen. Die default Route wird auf das virtuelle Interface (10.10.10.1) gelegt, d.h. alle Pakete in nicht-lokale Netze werden über dieses Interface geroutet.

Serverseitig werden alle VPN Verbindungen über ein virtuelles Interface (tun/tap) abgewickelt. Routingtechnisch sind hier keine Änderungen erforderlich, da das VPN Netzwerk (10.10.10.0/24) bereits bekannt ist. Ankommende Pakete aus dem VPN werden mit NAT auf eine interne Adresse 192.168.10.10 übersetzt und schon können die Clients mit Rechnern im internen Netz kommunizieren. Beim Einsatz von iptables als Firewall sieht der NAT Eintrag wie folgt aus:

```
Chain POSTROUTING (policy ACCEPT)
target prot opt source destination
SNAT all -- 10.10.10.0/24 192.168.10.0/24 to:192.168.10.10
```

NAT wird eingesetzt, da im Beispielnetzwerk kein dynamisches Routing konfiguriert ist. Will man ohne NAT arbeiten, muss das VPN Netzwerk auf allen Servern oder auf dem *default gateway* statisch eingetragen werden. Das nachfolgende Listing zeigt den statischen Eintrag der VPN Route auf einem default gateway (Cisco ASA 5505).

```
filch# sh route inside

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 1.2.3.5 to network 0.0.0.0

C    192.168.10.0 255.255.255.0 is directly connected, inside
S    10.10.10.0 255.255.255.0 [1/0] via 192.168.10.2, inside
```

5 Installation und Konfiguration OpenVPN Server (UNIX)

Die Installation des OpenVPN Servers erfordert Administratorrechte auf dem UNIX Betriebssystem. Die Quellen der Software sind frei verfügbar. Nach dem herunterladen der Quellen wird kompiliert und installiert:

```
$ ./configure --prefix=/opt/openvpn
$ make
$ su -c "make install"
```

Alternativ gibt es bereits fertige Pakete für diverse Distributionen. Das Konfigurationsverzeichnis */etc/openvpn* enthält die Serverkonfiguration, Zertifikate und Plugins. Die zentrale Konfigurationsdatei (Appendix A) des OpenVPN Servers *local.conf* beinhaltet alle Einstellungen. Der Server verwendet das virtuelle Interface *tap* und wartet auf dem UDP Port 1194 auf Verbindungen.

```
port 1194
proto udp
dev tap
```

Der private Key, die CA- und Server Zertifikate sind ebenfalls unter */etc/openvpn* abgelegt. Die Angabe der Dateinamen ist relativ zu diesem Verzeichnis. Der private Key muss auf Filesystemebene entsprechend geschützt werden, um unberechtigten Zugriff zu verhindern. (`chmod 400`)

```
ca cacert.pem
cert server.pem
key server.key
crl-verify crl.pem
dh dh1024.pem
```

Das Diffie Hellman Parameter File *dh1024.pem* wird mit openssl erstellt.

```
openssl dhparam -out dh1024.pem 1024
```

OpenVPN erlaubt das Verwenden dynamischer Adressen für Clients und Server. Der Benutzer kann „unterbrechungsfrei“ weiterarbeiten, selbst wenn die IP Adresse wechselt. Der Server prüft den Status der verbundenen Clients im Intervall von 10 Sekunden. Ist der Client für 120 Sekunden unerreichbar, wird die VPN Verbindung terminiert.

```
keepalive 10 120
```

Routing, DNS, WINS und andere Optionen können dem Client mitgeteilt werden. Im Beispiel wird der DNS Server auf 192.168.10.2 gesetzt. Die “redirect-gateway” Option bewirkt, dass der default gateway des Clients umgesetzt und der gesamte Netzwerkverkehr des Clients durch den VPN Tunnel geroutet wird.

```
push "redirect-gateway"
push "dhcp-option DNS 192.168.10.2"
```

Optional kann eine externe Authentisierung konfiguriert werden. Beispielskripts zur Authentisierung werden mit dem Quelltext mitgeliefert. Eine einfache LDAP Authentisierung ist im Appendix B abgebildet. Das Skript dient lediglich dem Verständnis und ist für nicht für den produktiven Einsatz angedacht. Darüber hinaus lässt sich OpenVPN mit beliebigen PAM Modulen koppeln und in bestehende Authentisierungs und Authorisierungsarchitekturen einbinden. Als Beispiel seien RADIUS, LDAP, OPIE und ACE genannt.

```
auth-user-pass-verify /etc/openvpn/knock.sh via-env
```

6 Installation und Konfiguration OpenVPN Client (UNIX)

Die Installation und Konfiguration auf einem UNIX Client ist identisch mit der des OpenVPN Servers. Einzig die Konfigurationsdatei */etc/openvpn/local.conf* ist verschieden. OpenVPN wird angewiesen im Client Modus zu arbeiten. Im Beispiel verbindet der Client Server auf einen Server (1.2.3.4) mit UDP auf Port 1194.

```
client
dev tap
proto udp
remote 1.2.3.4 1194
```

Die Client-und CA Zertifikate sind unter */etc/openvpn* abgelegt.

```
ca cacert.pem
cert client.pem
key client.key
```

Für die Authentisierung mit Benutzername und Passwort muss in der Client Konfiguration folgender Eintrag gemacht werden:

```
auth-user-pass
```

7 OpenVPN Client unter Microsoft Windows

Die Installation unter Windows³ benötigt Administratorrechte. Nach dem Aufruf des Installationsprogramms `openvpn-2.X-install.exe` wird man durch die Installation geleitet. Am Ende der Prozedur steht in der Programmliste ein neuer Eintrag *OpenVPN* zur Verfügung. Unter `C:\Programme\OpenVPN\config` werden die Konfigurationsdatei `local.ovpn` und die Client- und CA Zertifikate abgelegt. Die Konfiguration für Clients ist identisch mit der für UNIX Clients. Als virtuelles Interface wird unter defaultmässig `tap` verwendet. Ein Mischbetrieb von `tun` und `tun` Devices ist jedoch nicht möglich.

Unter wird der OpenVPN Client per rechte Mausklick auf das `local.ovpn` Symbol gestartet. Die Angabe des Benutzernamens, des Passworts und der Passphrase werden in einem command Fenster gemacht. Alternativ steht auch eine GUI basierende Variante zur Verfügung, auf die hier aber nicht speziell eingegangen wird.⁴

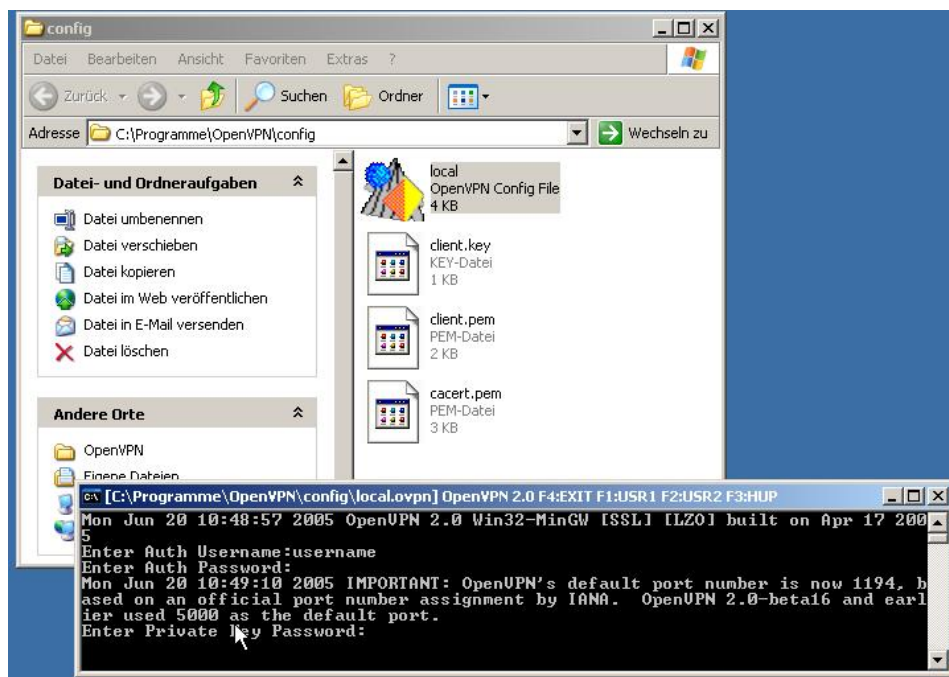


Abbildung 2: OpenVPN Client unter Microsoft

³Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

⁴<http://openvpn.se/>

8 Praktische Erfahrungen mit OpenVPN

Die VPN Lösung mit OpenVPN hat sich in der Praxis bewährt. Einzig an Bandbreite sollte nicht gespart werden. Die stärkste Performancebremse ist eine zu geringe upload Bandbreite bei asymmetrischen Anbindungen (z.B. ADSL/Kabel). Ab Werten von 256 kbps lässt sich bereits flüssig arbeiten. Tabelle 1 vergleicht den Durchsatz beim Kopieren einer 10MB grossen Datei über verschiedene Methoden. Der Client ist asymmetrisch mit 400/2000kbps (Kabel) und der Server mit 640/1536kbps (ADSL) am Internet angeschlossen. Der geringe Protokoll Overhead von OpenVPN ist deutlich ersichtlich. Die Abweichung der Transferzeiten zwischen getunnelter und ungetunnelter Übertragung ist minimal. Der unter Kapitel 4 beschriebene Setup eignet sich hervorragend zum Browsen von Informationen und für Remote Sessions. (ssh, console, VNC, X11 etc.) Will man hingegen ein Dokument direkt mit dem Textverarbeitungsprogramm auf dem Share öffnen, stösst man schnell an die Grenzen. Besser fährt man, wenn die Datei temporär auf eine lokale Ablage kopiert wird. Nach dem Bearbeiten wird die Datei auf den Share zurückkopiert.

Transport	Transferzeit in s
ftp 100Mbps switched	1/1
scp 100Mbps switched	1/1
ftp direkt	206/155
ftp über openvpn	217/164
scp direkt	204/157
scp über openvpn	214/167

Tabelle 1: Übertragungszeiten 10MB File

9 Verschlüsselt gleich sicher?

Das Thema Sicherheit spielt beim Einsatz von VPN Techniken eine grosse Rolle. Der Einsatz von starker Verschlüsselung und Client Zertifikaten allein bietet noch keinen absoluten Schutz. Sicherheit beginnt und endet beim Benutzer. Entfernt ein VPN Benutzer die Passphrase seines *private key* oder speichert Passwörter im Klartext, ist die Sicherheit potentiell gefährdet. Im Fall eines Identitätsdiebstahls, kann auf dem OpenVPN Server das Zertifikat des Clients widerrufen werden. Dazu wird das revozierte Zertifikat in der *certificate revocation list* (CRL) eingetragen. OpenVPN prüft die Liste und sperrt den Zugang für diese Clients.

Allgemein gilt, dass Firmen, die VPN Lösungen einsetzen, ihre Mitarbeiter auf das Thema Sicherheit sensibilisieren und entsprechend anleiten sollten.

Der Einsatz eines Firewalls ist unbedingt empfehlenswert. Das VPN schleust Datenverkehr von aussen direkt ins Firmennetz. Eine firmenweite Sicherheitspolicy regelt genau, welche Art von Kommunikation zugelassen wird. Eine spezielle Remote Access Zone bzw. DMZ ist sinnvoll. Virens Scanner sollten die Integrität der VPN Clients überprüfen, um auszuschliessen, dass andere Systeme via VPN infiziert werden.

10 Anfragen

Haben Sie Fragen oder Anregungen zu diesem Artikel? Kontaktieren Sie uns unter info@netnea.com.

A OpenVPN Server Konfiguration

```
#
# Example Corp. OpenVPN configuration file for
# office using SSL/TLS mode and RSA certificates/keys.
#
# '#' or ';' may be used to delimit comments.
dev tap

proto udp
port 1194

mode server
tls-server

ifconfig 10.10.10.1 255.255.255.0
ifconfig-pool 10.10.10.2 10.10.10.254 255.255.255.0
ifconfig-pool-persist ipp.txt

push "dhcp-option DNS 192.168.10.2"
push "route-gateway 10.10.10.1"
push "redirect-gateway"

tun-mtu 1500
tun-mtu-extra 32
mssfix 1450

dh dh1024.pem
ca cacert.pem
cert server.pem
key server.key
crl-verify crl.pem

user openvpn
group openvpn

comp-lzo

# Verbosity level.
# 0 -- quiet except for fatal errors.
# 1 -- mostly quiet, but display non-fatal network errors.
# 3 -- medium output, good for normal operation.
# 9 -- verbose, good for troubleshooting
verb 3

log /var/log/openvpn.log

status /var/log/openvpn-status.log

keepalive 10 120

auth-user-pass-verify /etc/openvpn/knock.sh via-env
```

B OpenVPN Client Konfiguration

```
#####
# Example Corp. client-side           #
# for connecting to multi-client server. #
# On Windows you might want to rename this #
# file so it has a .ovpn extension      #
#####
; this openvpn acts as client
client
; we use tap device due to Windows limitations
dev tap
; we use UDP (default)
proto udp
; this is the openvpn server name (public name + port)
remote 1.2.3.4 1194
; retry dns lookup of server name infinite
resolve-retry infinite
; do not bind to a special port number
nobind
; persistent settings over reboots
persist-key^M
persist-tun
; our CA certificate file
ca cacert.pem
; our client certificate file
cert client.pem
; our client key file
key client.key
; we do verify server cert
ns-cert-type server
; we use LZO compression
comp-lzo
; verbosity level is set here
verb 3
; we do log
log /var/log/openvpn.log
; optional authentication
auth-user-pass
```

C Optionale Authentisierung mit LDAP

```
#!/bin/sh
#
# a simple ldap auth script for openvpn
#
LDAP_HOST=192.168.10.2
#
# check nach leeren username/password oder [Aa]nonymous
#
if [ "$username" = \"{a}nonymous" || "$username" = \"{a}nonymous" || -z "$username" || -z "$password" ] ; then
    exit 1;
fi
#
# testen, ob bind mit credentials funktioniert - ungueltige Zeichen werden zu _
#
ldapwhoami -x -h $LDAP_HOST -D uid=$username,ou=users,dc=example,dc=com -w $password
#
if [ "$?" = "0" ]; then
    exit 0;
else
    exit 1;
fi
exit 1;
```