IT-Service

Mit Leidenschaft für unsere Kunden

Netzwerk mit openVPN

Ein Service von

ilexius

perfectly simple. simply perfect!

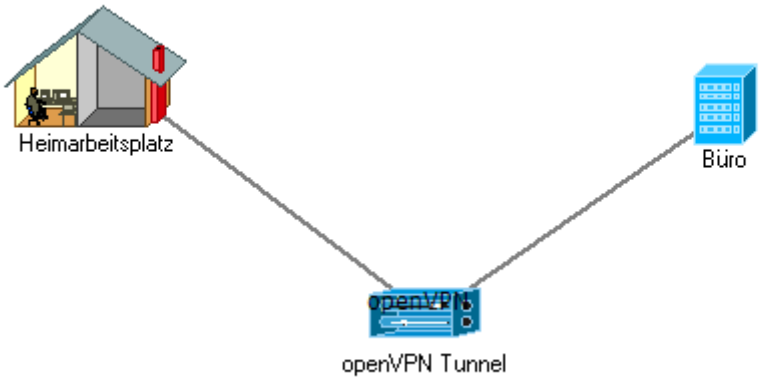
openVPN einrichten

Inhaltsverzeichnis

Die Ausgangslage 3
VPN Certificate Authority (CA) einrichten 3
Das Zertifikat erstellen: Das Root-Certificate der CA und ihren Schlüssel 4
Erstellen des Wurzel-Zertifikates 5
Erstellen des Server-Zertifikates und Schlüssels 5
Erstellen der Client-Zertifikate und Schlüssel 6
Erstellen der Diffie-Hellmann-Parameter* 6
Das VPN mit openVPN einrichten 7
Client-Konfiguration für openVPN 9
Erläuterungen zur Konfigurationsdatei des OpenVPN-Clients 10
Die Firewall-Einstellungen 11
Ansprechpartner zu diesem Artikel: 11

Die Ausgangslage

Eine Firma möchte es den Mitarbeitern ermöglichen, auf das Firmennetzwerk zuzugreifen. Die folgende kleine Skizze verdeutlicht das Szenario:



Die Verbindung muss natürlich verschlüsselt werden, damit niemand an die Daten die über das Internet geschickt werden lesen kann – und als weitere Bedingung sollte die Einrichtung der Verbindung (fast) nichts kosten :-)

Der Heimarbeiter hat eine DSL-Verbindung und der Server der Firma ist über eine feste Adresse im Internet zu erreichen. Der Rechner am Heimarbeitsplatz befindet sich in einem „Netzwerk" mit der IP 192.168.1.0 255.255.255.0 und das Netz in der Firma liegt in folgendem Adressbereich 192.168.2.0 255.255.255.0.

VPN Certificate Authority (CA) einrichten

Das Root-Certificate der CA und ihren Schlüssel

Das erstellen der Zertifikate ist dank vorgefertigter Skripte die im „easy-rsa"-Verzeichnis liegen, schnell erledigt. Kurz in das Verzeichnis gewechselt („/etc/openvpn/easy-rsa") und schon geht es los!

Als erstes wird die VARS-Datei bearbeitet. Hier werden die allgemeinen Einstellungen für die Zertifikate vorgenommen. Die Einstellungsmöglichkeiten sind nachfolgend dargestellt und erläutert.

Achtung

Falls openVPN über RPM, apt-get o.ä. installiert wurde befindet sich das Verzeichnis „easy-rsa" mit den Skripten in /usr/share/openvpn/easy-rsa; von dort am besten nach /etc/openvpn kopieren, sonst werden bei einem Update des Packets alle Einstellungen überschrieben!

VPN Certificate Authority (CA) einrichten

```
# Die Einstellungen für easy-rsa
# Diese Variable muss auf das oberste
# Verzeichnis im easy-rsa-Verzeichnisbaum zeigen
# Dies erledigt pwd ('Print Working Dir') automatisch.
# PWD liefert in diesem Beispiel den Pfad: /etc/openvpn
# somit hat die Variable D den Wert „/etc/openvpn"

export D=`pwd`

# Die Variable sollte auf die Datei openssl.cnf zeigen,
# die im Verzeichnis von easy-rsa
# zu finden ist.

export KEY_CONFIG=$D/openssl.cnf

# Diese Variable gibt das Ziel der zu erstellenden Schlüssel an
# Hier werden also die zu erzeugenden Schlüssel in das Verzeichnis
# /etc/openvpn/keys gespeichert

#
# ACHTUNG: clean-all löscht dieses Verzeichnis per rm -rf
# es werden also alle Dateien in diesem Verzeichnis gelöscht,
# daher sollte es unbedingt richtig konfiguriert sein

export KEY_DIR=$D/keys

# Ausgabe für die rm -rf Warnung

echo ACHTUNG: Das Ausführen von  ./clean-all, führt den
Befehl rm -rf auf das Verzeichnis $KEY_DIR aus!

# Die Variable KEY_SIZE legt die stärker der Verschlüsselung fest
# Durch einen hohe BIT-Angabe ist der Schlüssel sicherer, allerdings
# wird die Erstellung der Diffie Hellmann Parameter länger dauern
# und ebenso der Handshake mit dem Server per TLS.
# Entsprechend wird die Verbindung langsamer.

export KEY_SIZE=1024

# Hier werden die Vorgabefelder für die
# Erstellung des Zertifikates eingestellt.
# Alle Felder müssen Ausgefüllt sein!

export KEY_COUNTRY=DE
```

```
export KEY_PROVINCE=HESSEN
export KEY_CITY=WIESBADEN
export KEY_ORG="ilexius"
export KEY_EMAIL="MeineAdresse@MeinProvider.MeineTLD"
```

Erstellen des Wurzel-Zertifikates

Nachdem die „VARS" Einstellungen angepasst sind, geht es nun endlich ans Erstellen der Zertifikate und Schlüssel. Als erstes muss die PKI vorbereitet werden und das Wurzel-Zertifikat erstellt werden.

Dazu einfach folgende Befehle aufrufen:

```
. ./vars
./clean-all
./build-ca
```

Was machen diese Befehle? Nun der erste **„../vars"** initialisiert die VARS-Datei die eben geändert wurde. Anschließend leert **„./clean-all"** das Verzeichnis „keys", das in der VARS-Datei spezifiziert wurde und **„build-ca"** schließlich erstellt das Zertifikat und der Schlüssel für certificate authority, das root-certificate bzw. das Wurzel-Zertifikat.

Wichtig, beim Erstellen des Zertifikates muss der Common-Name eingegeben werden. Dieser muss einzigartig sein! Der Name könnte beispielsweise „OpenVPN-CA" sein. Der Rest der anzugebenden Parameter kann auf Standard gesetzt bleiben, denn diese Werte wurden ja gerade in die VARS-Datei eingetragen.

Erstellen des Server-Zertifikates und Schlüssels

Jetzt ist der Server dran, der folgende Befehl generiert das Zertifikat und den Schlüssel.

```
./build-key-server server
```

Dabei wird dem Skript der Parameter „server" übergeben, der den Dateinamen für Schlüssel und Zertifikat festlegt. Später werden die Dateien also im Verzeichnis „keys" unter diesem Namen stehen. Wiederum muss unter Common Name ein einzigartiger Name gewählt werden, der bislang nicht verwendet wurde.

Erstellen der Client-Zertifikate und Schlüssel

Nun werden noch schnell die Schlüssel und Zertifikate für die Clients erstellt, mit:

```
./build-key client1
```

Dabei stellt der übergebene Parameter client1 wieder den Namen für das Zertifikat und den Schlüssel da. Wiederum muss unter Common Name ein einzigartiger Name gewählt werden, der bislang nicht verwendet wurde.

Wenn weitere Clients ein Zertifikat und einen Schlüssel erhalten sollen, muss der entsprechende Befehl erneut aufgerufen werden.

Erstellen der Client-Zertifikate und Schlüssel

```
./build-key clientXYZ
```

Erstellen der Diffie-Hellmann-Parameter*

Nun müssen noch die Diffie-Hellmann-Parameter für den Server erstellt werden. Dazu dient der folgende Befehl:

```
./build-dh
```

Der Vorgang dauert einige Zeit - insbesondere bei langsamen PCs und starken Verschlüsselungen kann es etwas länger dauern.

Erstellen der Diffie-Hellmann-Parameter*

***** Mehr zu Diffie-Hellmann unter: http://de.wikipedia.org/wiki/Diffie-Hellman

Dateiname	*Benötigt von*	*Zweck*	*Geheim*
ca.crt	Server und allen Clients	Wurzel-Zertifikat der CA	Nein
ca.key	Ersteller der Zertifikate (meist der Server)	Schlüssel der CA	Ja
Dh[n].pem*	Server	Diffie-Hellman-Parameter	Nein
server.crt	Server	Server Certificate	Nein
server.key	Server	Server Key	Ja
client1.crt	Client	Client1 Certificate	Nein
client1.key	Client	Client1 Key	Ja

* {n} gibt die Stärke der Verschlüsselung an, in unserem Fall 1024 Bit, dh.an der Stelle [n] steht 1024 und der Dateiname sieht dann so aus „dh1024.pem".

Das VPN mit openVPN einrichten

Nun geht es richtig los mit der Einrichtung des eigentlichen openVPNs. Zunächst geht es mit dem Server los, im Anschluss werden wir uns um den (oder die) Clients kümmern.

Server-Konfiguration für openVPN

```
# Dies ist die Konfigurationsdatei für den Server

# Das Geräte für den Tunnel festlegen und das Protokoll bestimmen

dev tun
proto udp

# Routing festlegen:

route 192.168.4.1 255.255.255.252 192.168.1.1


# Servereinstellungen bestimmen
# der Server bekommt den gesamten Adressraum des
# 192.168.4.0/24
# Netzebereiches als mögliche Adressen für Clients angegeben

server 192.168.4.0 255.255.255.0
```

Das VPN mit openVPN einrichten

```
mode server


# Push veranlasst die Änderung des Routings beim Client. Dieser
muss
# auf „pull" eingestellt sein, damit er die Anweisungen akzeptiert.
push "route 192.168.1.0 255.255.255.0"


# Verschluesselung einstellen
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key
dh /etc/openvpn/easy-rsa/keys/dh1024.pem
cipher AES-128-CBC
tls-server


# die 0 am Ende der nächsten Zeile erfordert als Komplemten für die
# Richtung der Identifikation beim Client die 1


tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0

#Kompression
comp-lzo

# Diese Einstellung legt fest, dass der Server ein Ping-ähnliches
Signal # alle 10 Sekunden sendet und die Verbindung gilt als
abgebrochen,
# wenn nach 60 Sekunden keine Antwort gekommen ist.

Keepalive 10 60


#Logs

status openvpn.log
verb 6


# Key und Zertifikat werden im Speicher behalten, damit stellen wir
# sicher, dass die Dateien auch nach der Änderung von Benutzer und
# Gruppe noch gelesen werden können.
persist-key
persist-tun
```

```
# Maximum Transfer Unit
tun-mtu 1452
```

Client-Konfiguration für openVPN

Der Client soll als einzelner Rechner von überall mit dem openVPN-Server Verbindung aufnehmen können. Unter Windows werden die Einstellungen im Verzeichnis "C:\Programme\OpenVPN\config" vorgenommen. Dort erstellt man eine Datei mit der Endung „ovpn".Anschließend kann die Datei mit dem „Editor" oder „Wordpad" bearbeitet werden.

Die Konfigurationsdatei dazu sieht wie folgt aus:

```
# Dies ist die Konfigurationsdatei des Clients.

# Dem Client wird vom Server automatisch eine IP zugewiesen

# Ziel des Tunnels, also der VPN-Server, als Beispiel hier die
DynDNS-
# Adresse. Selbstverständlich kann man hier auch eine IP-Adresse
o.ä.
# angeben

remote meineAdresse.dyndns.org

# Legt das Gerät für den Tunnel fest und den Port (Port kann auch
# entfallen, dann wird der Standard-Port 1194 verwendet)
# dev tun legt einen IP-Tunnel mit Routing fest
# dev tap Tunnel mit Ethernet-Bridging

dev tun
port 1194


# der Rechner wird als client angemeldet und erhält vom Server eine
IP
# Einstellungen beim Server beachten!

client

# Festlegen des Protokolls

proto udp

#wegen dyndns float einfügen, sonst kann es bei einem
```

Client-Konfiguration für openVPN

```
# Verbindungsreset zu Problemen kommen.
#float erlaubt validierte Pakete auch von wechselnden IP-Adressen

float

#client konfigurieren
tls-client

# Festelgen, dass der Client PUSH-Anweisungen des Servers
akzeptiert

pull

#Verschlüsselung

tls-auth C:\\Programme\\OpenVPN\\easy-rsa\\key\\ta.key
1
ca C:\\Programme\\OpenVPN\\easy-rsa\\key\\ca.crt
cert C:\\Programme\\OpenVPN\\easy-rsa\\key\\client1.crt
key C:\\Programme\\OpenVPN\\easy-rsa\\key\\client1.key
cipher AES-128-CBC

# Kompression

comp-lzo

# Maximum Transfer Unit MUSS auf Server und Client identisch sein!
tun-mtu 1452
```

Erläuterungen zur Konfigurationsdatei des OpenVPN-Clients

Der Client verbindet sich von einem beliebigen Adresse zum Server. Er erhält durch „remote"-Angabe das Ziel mit dem er sich verbinden soll.

Als Geräte wird wie beim Server tun gewählt (weil es sonst nicht funktioniert ;-)) und der Port sowie das Protokoll identisch festgelegt. Nun wird dem Client explizit gesagt, dass er client ist, damit „weiß" der Client wie er sich gegenüber dem Server verhalten muss. Er bezieht automatisch eine IP-Adresse und ist bei der Authetifizierung der entsprechende Client usw.

Eine weitere Einstellung die beim Client festgelegt ist, ist die „float" Option. Sie stellt sicher, dass der Client validierte Packete vom Server auch dann akzeptiert, wenn dieser seine IP-Adresse geändert hat. Folglich ist diese Option nur dann notwendig, wenn bei der Verbindung die Möglichkeit besteht, dass die IP-Adresse sich verändert.

Die Firewall-Einstellungen

Nun muss noch die Firewall freigeschalet werden, sofern eine aktiviert ist :-)

Die Firewall-Einstellungen

```
# Protokoll ist udp und der Port 1194, sonst muss hier angepasst
# werden

iptables -A INPUT -p udp --dport 1194 -j ACCEPT

# Erlaube Pakete der TUN/TAP Schnittstelle.
iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT
```

damit akzeptiert die Firewall nun alle Pakete für alle tun-Geräte (daher das „+“-Zeichen) . Zur Sicherheit sollten je nach Bedarf Einschränkungen eingeführt werden!

Ansprechpartner zu diesem Artikel:

ilexius GmbH
Herr Sebastian Koch
Herr Thomas Schlüter
info@ilexius.de

Weitere Anleitungen zu verschiedenen Themen finden sich unter:
www.ilexius.de