Komplexe IPsec- und SSL-VPNs mit Linux

Johannes Hubertz
hubertz-it-consulting GmbH
Theodor-Heuss-Straße 60-66
51149 Köln
johannes@hubertz.de

Zusammenfassung

Verschlüsselung von Netzwerkverbindungen ist trotz wachsender Bedrohungen im Internet noch nicht selbstverständlich, mit OpenVPN und StrongSwan jedoch kostengünstig und sicher realisierbar; die Mechanismen unterscheiden sich wesentlich und haben ihre jeweiligen Vor- und Nachteile. Unternehmen können mit diesen Softwarelösungen vertraulich und zuverlässig Daten zwischen eigenen Standorten austauschen, gesicherten Zugang für Heimarbeitsplätze oder für Handlungsreisende via Internet schaffen und kompatibel zu marktüblichen VPN-Komponenten verschlüsselte Kommunikation mit anderen Unternehmen gestalten. Die Sicht des Administrators auf die jeweilige Lösung steht im Vordergrund, die kryptographische Sicherheit der Software ist bereits an anderen Stellen glaubhaft beschrieben und kann in geeigneter Umgebung vorausgesetzt werden.

1 Virtuelles privates Netzwerk, wie funktioniert das?

Ein reales Netzwerk wird benötigt, um Daten zu transportieren. Sollen die Daten privat bleiben, nutzt Verschlüsselung vor ungewollten Einblicken. Virtuell wird das dann zusätzlich, wenn ein eigenes, unabhängigs Netz über das bestehende, unsichere Netz gestülpt wird. Dies erledigen Router oder VPN-Geräte zumeist dadurch, dass vor die zu transportierenden Datenpakete ein zusätzlicher IP-Header gesetzt wird, Ziel-IP ist dann die jeweilige Gegenstelle. Das so auf die Reise geschickte Paket wird, dort angekommen, um den zusätzlichen Header vermindert und lokal anhand des eigenen Headers zugestellt. Für Absender und Empfänger der Nutzdaten geschehen diese Manipulationen unbemerkt. Selbst aufgrund der grösseren Paketlängen erforderliche Fragmentierungen und Defragmentierungen werden transparent erledigt, vorausgesetzt, die Mechanismen sind sauber programmiert.

Verschiedene Wege führen zum Ziel. Verschlüsselung ist in neueren Linux-Kerneln möglich, ebenso kann ein Anwendungsprogramm selbst die Routinen enthalten oder auch eine Bibliothek. Übliche Algorithmen wie AES oder Blowfish sind auf aktueller Hardware ausreichend performant, um auch viele Benutzer-Sessions gleichzeitig zuzulassen.

Soll Interoperabilität zu VPN-Lösungen anderer Unternehmen und Hersteller erreicht werden, sind SSL-VPNs die schlechtere Wahl. Kein verbindlicher Standard regelt die Methoden, SSL-VPN ist meist nur mit gleichen Geräten an den Enden zu relaisieren. Jedoch wird man nicht für jeden Geschäftspartner eine andere Lösung haben wollen, mehr dazu später. Mit IPsec-VPNs besteht diese Problematik nicht, Interoperabilität war von Beginn an das Hauptziel der auf Linux entwickelten Software, die als Referenz für die Gerätehersteller dient. Der Preis dafür ist eine etwas kompliziertere Administration, diffiziles Entstören und im Normalfall mehr Know-how. Dennoch ist auch ein SSL-VPN fürs Unternehmen unter Umständen eine gute Wahl.

2 SSL-VPN

2.1 SSL

Mitte der 1990'er Jahre entdeckte der Kommerz das Internet. Transaktionen zwischen Käufern und Verkäufern machten schnell vertraulichen Datentransfer erforderlich. Das „WWW“ wandte sich rasant von der initialen Browserversion ab; eine junge, kleine Startup-Firma namens Netscape bestimmte den Trend der Zeit. Eine mit RC2, RC4, also mit schnellen (wenig rechenintensiven) Verschlüsselungsmechanismen versehene HTTP-Version setzte sich unter der Bezeichnung **https** (HyperText Transport Protocol Secured) durch, eine andere Variante namens **shttp** (Secured HyperText Transport Protocol) verschwand bald wieder in der Bedeutungslosigkeit. Netscape nannte es Secured-Socket-Layer in Anlehnung an die Art, wie bzw. an welcher Stelle im Kommunikationsweg programmiert wurde. Socket bezeichnet die Schnittstelle in unixoiden Systemen, die zwischen Anwendung und Systemkern die Übergabe von Parametern und Daten spezifiziert und dadurch einfaches Programmieren nicht nur von Netzwerkfunktionalitäten ermöglicht. Secured Socket meint die zusätzliche Zwischenschicht, die die Verschlüsselungsfunktion zu jeder TCP-Session hinzufügt. Als übliche Abkürzung setzte sich schnell SSL durch, später abgelöst durch TLS, ausgesprochen als Transport-Layer-Security.

Kompatibilität war gefragt, sowohl auf der Server- als auch auf Browserseite. Dies hatte die Offenlegung der Spezifikationen als angenehme Folge. Ein kleines Team in Australien konnte mit diesen Informationen anfangen, die Mechanismen mit offengelegten Quelltexten auch für die Allgemeinheit verfügbar zu machen. Eric A. Young und Tim Hudson setzten mit **ssleay** nicht nur Masstäbe für Programmierqualität, per Mailingliste und vielen Beiträgen anderer wuchs innerhalb weniger Jahre eine professionelle Verschlüsselungsumgebung, die in vielen weiteren Projekten die zentrale Rolle spielt. Für den Webserver Apache wurde zuerst ein Patch veröffentlich, aus dem sich später ein eigenes Modul **mod_ssl** entwickelte. Als ein universell nutzbares Werkzeug (für beliebige TCP-Verbindungen) stellte sich schnell **stunnel** heraus. Andere Anwendungen, z.B. **tinyca und openca** folgten, als **S/MIME** (secured mime) ist TLS heute mit **smtp** bzw. **esmtp** verbandelt.

2.2 OpenSSL

Aus **ssleay** entwickelte sich bald OpenSSL als internationale Gemeinschaftsarbeit, nachdem Eric Young seine Mitarbeit an der Entwicklung einstellte. Die Dokumentation wuchs und mehrte den Nutzen des Paketes, welches aus dem Kommandozeilenprogramm openssl und einer dazugehörigen Bibliothek libssl bestand. X.509-Zertifikate wurden damit handhabbar, die schon viel früher in den RFCs (1421-1424, Februar 1993) zu privacy-enhanced-mails Einzug in die Normung der Internet-Kommunikation gehalten hatten. Die Komplexität dieser Mechanismen verhinderte jedoch eine weite Verbreitung. Ein solches Zertifikat besteht aus X.500 Attributen (Name, Adresse, Email, ...), der Signatur des Ausstellers und dem öffentlichen Schlüssel aus einem RSA-Schlüsselpaar. Dieses hat im Idealfall der Nutzer selbst mit dem **keygen**-Tag einer HTML-Seite in seinem Netscape-Browser erzeugt. Der dazugehörige private Schlüssel wird separat in der Zertifikatsdatenbank des Browsers gespeichert. Aussteller ist üblicherweise eine sogenannte Certificate Authority (Zertifizierungsstelle, CA), vergleichbar einem Notar im realen Leben. Später wurde SSL auch im Browser des Marktführers eingebaut, aufgrund der darin fehlenden Funktionalität zum „keygen“-Tag konnten Zertifikate ausschließlich in Form eines PKCS#12-Paketes importiert werden. Diese Binärdatei im PKCS#12-Format enthält das Anwender-Zertifikat, den dazugehörigen privaten Schlüssel und das Zertifikat der signierenden Instanz. Gegen unbefugten Import ist es passwortgeschützt. Selbstverständlich war dieser von der CA importierte private Schlüssel nicht mehr ganz so privat. OpenSSL kann dank der von Dr. S. Henson ab 1997 entwickelten Funktionen ebenfalls PKCS#12-Pakete lesen und schreiben.

Die X.509-Zertifikate können sowohl im PEM- (privacy enhanced mail, ASCII-Klartext, base64-encoded) als auch im DER-Format (data encryption rules, ASN.1, Binär-Format) ver- und bearbeitet werden. In der einfachen Form können ausgestellte Zertifikate in einer flachen Text-Datei verwaltet werden, bis zu einigen tausend Zerti-

```
local x.y.z.u
port 10000
proto tcp
dev tun
ca   /etc/ovpn/cacert.pem
cert /etc/ovpn/vpn-srv.pem
key  /etc/ovpn/vpn-srv-key.pem
crl-verify /etc/ovpn/crl.pem
dh /etc/ovpn/dh2048.pem
server 172.24.0.0 255.255.0.0
ifconfig-pool-persist /etc/ovpn/p1
push route-delay 6 20
push redirect-gateway
push dhcp-option DNS 172.16.0.5
push dhcp-option DNS 172.16.0.3
keepalive 10 120
comp-lzo
max-clients 10000
user nobody
group nogroup
persist-key
persist-tun
status /var/log/ovpn1-status.log
log-append /var/log/ovpn1.log
verb 4
mute 10
reneg-sec 18000
tls-exit
```

Abbildung 1: openvpn.conf

fikaten sicher die schnellste und einfachste Lösung. Der Ersatz durch einen Datenbankanschluss ist vorgesehen und für einige hundertausend Zertifikate sicherlich sinnvoll.

OpenSSL kann auch eine sog. **CRL,** die „Certificate Revocation List“, generieren. Diese ist eine wichtige Voraussetzung, um jemanden auszusperren, obwohl er ein gültiges Zertifikat besitzt. Die CRL besteht im wesentlichen aus den Seriennummern aller derjenigen Zertifikate, die nicht mehr gültig sein sollen, obwohl ihre Gültigkeitsintervalle noch andauern. Diese Liste wird mit dem CA-Zertifikat signiert. Da Zertifikate eine Gültigkeitsdauer in Form von „nicht vor“ und „nicht nach“ Datumswerten mitgegeben ist und eine Kopie nicht vom Orginal zu unterscheiden sind, muß eine Mißbrauchserkennung auf Server- wie auch auf Clientseite möglich sein. Ein neueres Verfahren mit dem gleichen Ziel, jedoch aktuell zum Abfragezeitpunkt nennt sich Online Certificat Status Protocol (OCSP) und wird von aktuellen Clientprogrammen wie z.B. Firefox unterstützt. Auch auf Serverseite ist es integrierbar. Mit diesem Protokoll wird online in Echtzeit bei der Zertifizierungsstelle abgefragt, ob das Zertifikat mit der Sereinnummer XYZ zur Zeit gültig ist. Ein deutlicher Vorteil gegenüber dem CRL-Verfahren ist, dass nichts auf Server oder Browser verteilt werden muß. Im Sinne des Datenschutzes ist es ein wichtiger Nachteil, dass auf der Ausstellerseite ein Nutzungsprofil für jedes einzelne, jemals ausgestellete Zertifikat erstellt werden kann. Dem kann entgegnet werden, dass mit der CRL ebenfalls Profiling betrieben werden kann, das ist korrekt, trifft aber stets nur auf den einzelnen Server zu, nicht aber auf die Zertifizierungsinstanz. Erst bei Zusammenführung *aller* dieser Profile würde sich ein dann allerdings noch aussagekräftigeres Profil ergeben können. Und sollte nicht gerade die Zertifizierungsinstanz das Vertrauen aller geniessen?

2.3 OpenVPN

2004 war es in der Version 1.x schon ein alter Hut, wirklich bekannt geworden ist OpenVPN jedoch erst ab etwa 2005 mit der Version 2.0. Es nutzt libssl, die durch das OpenSSL-Team entwickelt wurde, und kann im Gegensatz zu SSL/TLS auf TCP und UDP aufsetzen. Als Transportvehikel bieter UDP den Vorteil, weniger Overhead zum Nutzdatenstrom hinzuzufügen, TCP lässt sich jedoch einfacher bzw. zuverässiger mit netstat abzählen. Die bekannte Problematik TCP over TCP wurde mit OpenVPN bisher vom Autor nicht beobachtet, allerdings macht sich der größere Overhead durch TCP bei geringen Bandbreiten ($\leq 64\frac{kbit}{s}$) bemerkbar. Den Verschlüsselungsmechanismen ist es jedenfalls gleichgültig, wie der erzeugte Chiffretext transportiert wird; nur die Klartext-Anwendungsprotokolle sorgen für die notwendige Transportverläßlichkeit.

2.3.1 Konfiguration

Die Konfiguration von OpenVPN besteht nur aus wenigen notwendigen und einigen optionalen Zeilen wie z.B. in Abbildung 1 gezeigt; Server und Client unterscheiden sich nur in wenigen Statements. Die Quellen der Software liegen für verschiedene Plattformen in gepackter Form auf `http://openvpn.net` zum Download

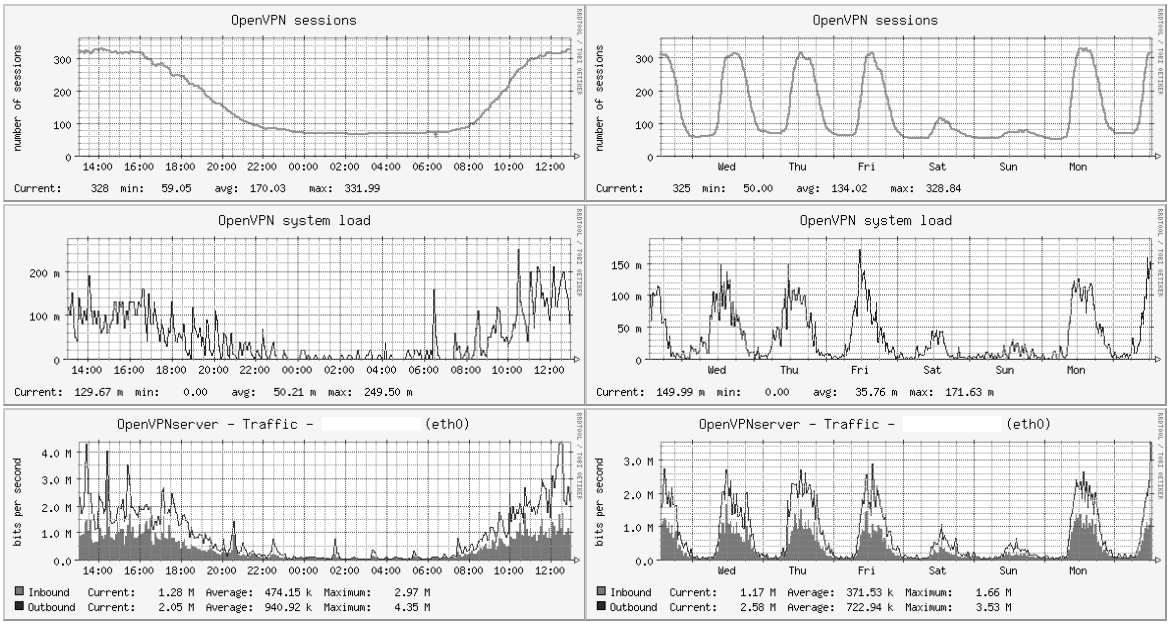

Abbildung 2: OpenVPN Nutzungs- und Lastverhalten

bereit, für proprietäre Systeme auch als ausführbare Installationsroutinen. Bei `http://openvpn.se` ist sogar ein GUI erhältlich für Benutzer, die sich nicht in den Tiefen der Konfiguration verlieren wollen.

Aus verschiedenen Gründen kann es sinnvoll sein, die ausführbaren Programme selbst zu erzeugen, beispielsweise um von den verschiedenen Linux-Distributionen und derer Bibliotheken unabhängig zu sein. Beide Seiten benötigen Zertifikate, jeweils das Eigene mit dem dazugehörigen privaten Schlüssel und das des Ausstellers, um dessen Signatur unter dem des jeweiligen Gegenübers verifizieren zu können.

Zu jedem Client-Zertifikat können auf Serverseite in einem Verzeichnis Skripte und weitere spezifische Einstellungen abgelegt werden. Der Server überprüft bei jedem Sessionaufbau seine CRL (Certificate Revocation List), in der die Seriennummern aller zurückgeweisenen Zertifikate durch die ausstellende Instanz signiert abgelegt sind. Ist die Seriennummer des ankommenden Zertifikates in der CRL vorhanden, kommt keine Verbindung zustande. Auf Serverseite wird mit “local IP-Adresse“ festgelegt, auf welcher IP-Adresse die OpenVPN-Instanz auf ankommende Client-Verbindungswünsche lauscht. Das Pendant “remote “ kann im Client auch mehrfach genannt werden, dann werden die verschiednen Möglichkeiten zyklisch in aufeinanderfolgenden Sessions genutzt.

Mit “ifconfig-pool-persist Datei“ wird dem Server diejenige Datei benannt, in der IP-Adressen der Clients über eine laufende Sitzung hinaus gespeichert werden. So erhält ein Client in mehreren Sitzungen stets die gleiche IP-Nummer, die daher für Firewall-Scripts zur Realisierung unterschiedlicher Rechte der Clients in der Serverlandschaft des Unternehmens genutzt werden kann. Gespeichert wird jeweils der CommonName aus dem Zertifikat und die Netzadresse des /30, welches zu jedem Clientzertifikat aus dem gegebenen Pool entnommen wird. Der Client erhält jeweils die obere nutzbare Adresse aus diesem /30, die untere benutzt er als default Gateway.

2.3.2 Lastverteilung

Mit steigender Anzahl gleichzeitiger Client-Sessions ist mit zunehmender Last auf dem Server zu rechnen. Alle Verschlüsselung findet innerhalb von Benutzterprozessen statt. Bei der ersten Inbetriebnahme für viele Aussendienstler ist logischerweise wenig Erfahrung vorhanden, wie sich diese Last auswirkt. Um im Bedarfsfall möglichst flexibel reagieren zu können, ist eine Verteilung der Last bzw. des Datenverkehrs erstrebenswert.

```
#!/bin/bash
#
NAME=`/bin/cat /etc/hostname`
SERVERS="${SERVERS} 10.9.2.4 "
SERVERS="${SERVERS} 10.9.2.5 "
SERVERS="${SERVERS} 10.9.2.6 "
SERVERS="${SERVERS} 10.9.2.7 "
PORT="10000"
#
while :
do
    clear
    DATE=`date`
    LOAD=`/root/bin/loadavg.pl`
    printf "\n\tOpenVPN sessions on %s\n" $NAME
    printf "\n\tlocaltime:  %s \n\tsystemload: %s\n" "$DATE" "$LOAD"
    netstat -an | grep -v WAIT >/tmp/ocnt
    SUM=`grep :${PORT} /tmp/ocnt | grep -c ESTABLI`
    printf "\n\n\tOpenVPN sessions (all)              : %-5d\n\n" $SUM
    for IP in ${SERVERS}
    do
        NUM=`grep ${IP}:${PORT} /tmp/ocnt | grep -c ESTABLI`
        printf "\tOpenVPN sessions on %-15s: %-5d\n" $i $NUM
    done
    sleep 10
done
```

Abbildung 3: OpenVPN-Sitzungszähler

Dies kann auf einfache Weise erreicht werden: Auf der Serverplattform werden mehrere IP-Adressen aktiv, auf diesen lauscht je ein Serverprozess auf ankommende Client-Sessions. Die Clients werden bei ihrer Installation gleichmässig auf die vorhandenen Server-Adressen verteilt. Sinkt nun die gewünschte Performanz aufgrund der Last, kann einfach durch neue Hardware und Umzug der entsprechenden Anzahl IP-Adressen die Last besser verteilt werden.

Um ewtas über die Last, die auf einem Pentium mit 2GHz und 2GByte RAM entsteht, herauszufinden, kann z.B. mit cacti (`http://cacti.net`) Datenmaterial gesammelt und gleich graphisch aufbereitet dargestellt werden. Die Ergebnisse in Abbildung 2 deuten mindestens in die Richtung, dass der Server deutlich mehr Sessions verkraften kann als aktuell benötigt werden. Der Server beherbergt 6 OpenVPN-Instanzen, ein wichtiges Werkzeug zur Beobachtung ist ein einfaches Shellskript wie in Abbildung 3 gezeigt. Es zählt regelmässig die aktuelle Anzahl Sitzungen pro Serverinstanz und zeigt das Ergebnis.

2.3.3 Betrieb

Nach mehr als anderthalb Jahren ständiger Beobachtung stellt sich das Gefühl ein, eine gute Sache zu betreiben. OpenVPN skaliert nicht nur ausgezeichnet, es ist auch absolut stabil im Betrieb. Ausfallzeiten sind bisher ausschließlich auf manuelle Eingriffe zurückzuführen, um einem Zertifikat eine bestimmte, andere IP-Adresse fest einzustellen. Grund hierfür war nie OpenVPN, sondern nur an feste IP-Adressen gekoppelte Rechte, die mit iptables eingestellt bzw. reglementiert werden. Sinnvollerweise werden solche Durchgriffsrechte auf Netze größer /30 vergeben, natürlich sind aber auch Einzelregulierungen möglich.

Diese Änderungen sind für die aktiven Benutzer weitestgehend schmerzfrei, da die Clientsoftware ab dem Moment, in dem die Verbindung serverseitig abgebaut wird, diese sofort neu initiiert. Durch die Tunnel stattfindender Webtraffic oder bspw. LotusNotes-Zugriff verkraftet diese kurzen Unterbrechungen, so dass der Benutzer meist nur ein Rucken seines Datentransfers bemerkt, wenn er es überhaupt registriert.

2.3.4 Entstörung

Manches Mal treten in solchen Installationen Fehler auf, z.B. wenn am anderen Ende Releasewechsel stattfinden. Das in Stufen seiner Geschwätzigkeit einstellbare Logging in Verbindung mit der Status-Datei von OpenVPN auf der Serverseite ist i.A. aussagekräftig genug, um der Ursache auf die Spur zu kommen. Auf der Seite der Clients ist ebenfalls Logging in verschiedenen Stufen möglich. Der Benutzer muß über die Zugriffsrechte verfügen, dies auch lesen zu können. Im Zweifel ist es jedoch einfacher, Quell- und Zieladresse mit

tcpdump zu sniffern, wenngleich verschlüsselter Verkehr nicht gerne mitgelesen werden mag. Die Tatsache, auf welchen Ports überhaupt Daten ausgetauscht werden, hilft schon einiges weiter. Zusätzlich kann auf dem internen Interface des OpenVPN-Gateways der Klartext ebenfalls mitgeschnüffelt werden; falls dieser nicht zusätzlich verschlüsselt ist, kann er wirksam entstört werden. Ansonsten helfen sicher auch die Logs der genutzten internen Server, um den Ursachen der gesuchten Fehler auf die Schliche zu kommen.

2.3.5 Interoperabel?

Nur selten sind kommerzielle SSL-VPN-Lösungen kompatibel zu anderen, zumeist wird in den Appliances auf Web-Oberflächen zur Administration zurückgegriffen. Dies stellt grundsätzlich einen zusätzlichen Angriffspunkt dar, die Einstellmöglichkeiten bleiben i.A. begrenzt. Oft ist üblich, mit externen DNS-Dienstleistern wie bspw. `dyndns.org` die jeweilige Gegenstelle zu finden. Dies hat den Vorteil, auch mit dynamisch zugewiesenen Adressen zu funktionieren. Inwieweit jedoch DNS im Zusammenhang mit Sicherheitsfunktionen eine Rolle spielen sollte, mag jeder selbst entscheiden; mindestens ist anzuzweifeln, ob DNS überhaupt jemals sicher sein kann. Auf jeden Fall ist es relativ leicht zu fälschen. Auch die aktive Einbindung einer Unternehmens-PKI wird meist nicht unterstützt, selbst das Im- und Exportiern von Zertifikaten ist nicht immer möglich. Der Begriff CRL taucht in machen Geräten gar nicht erst auf. Inwieweit dann von Sicherheit dieser Geräte überhaupt gesprochen werden darf, hängt wesentlich vom jeweiligen Verständnis des Prinzips „security by obscurity“ ab.

Zumindest in einem Aspekt ist OpenVPN diesen kommerziellen Lösungen ähnlich: Es ist auch nicht zu anderen SSL-VPN-Lösungen kompatibel. Allerdings sind alle anderen Randbedingungen so, dass, verständnisvolle Administration vorausgesetzt, es eine sichere Unternehmenskommunikation ermöglichen kann. Durch die offengelegten Quelltexte ist jedem jederzeit möglich, die Funktionalität nicht nur nachzuvollziehen, sondern bei Bedarf auch an seine eigenen Bedürfnisse anzupassen. Mit vorhandener eigener Infrastruktur kann auch durch OpenVPN-Tunnel mit VoIP-Softphones sicher und vertraulich telefoniert werden. Von Skype kann in diesem Zusammenhang jedoch aufgrund des möglicherweise vorherrschenden Wunsches nach Sicherheit nur abgeraten werden.

3 IPsec-VPN

Ursprünglich wurde IPsec für die nächste Version des Internet-Portokolls definiert, welches in einigen dutzend RFC spezifiziert ist. Wesentlicher Grund der vollständigen Integration kryptographischer Sicherheiten in IPv6 ist deren Abwesenheit in der immer noch aktuellen Version IPv4. Der Pakettransport kann mit „authenticated header“ gegen Fälschungen der Absende und Empfänger-Adressen abgesichert werden, indem die Header aller IP-Pakete kryptographisch signiert werden. Inhalte werden mit der Methode „encapsulated payload “ mit gängigen Krypto-Algorithmen verschlüsselt und so gegen ungewollte Einblicke geschützt. Der Schlüsselaustausch für diese Massnahmen ist ebenfalls in RFC genormt.

Alle diese Schutzmaßnahmen wurden nach ihrer Spezifikation als so gut bewertet, dass eine Rückportierung auf die aktuell genutzte Protokoll-Version 4 durchgeführt wurde. Hierbei wurden verschiedene Netzwerk-Hardware-Hersteller wie Cisco, Nortel und einige andere sich schnell darüber einig, dass ihre Produkte untereinander kompatibel sein müssen, um eine Marktdurchdringung zu erreichen. Ein Konsortium, getragen von verschiedenen Herstellern, betrieb die Implementierung in Kanada öffentlich, d.h. die Quelltexte entstanden auf RedHat-Linux und wurden als OpenSource im Internet zum Download angeboten.

Herstellerunabhängige Entwickler unterstützen ebenfalls dieses Projekt, so stellte z.B. Prof. Andreas Steffen aus der Schweiz seinen X.509-Patch der Öffentlichkeit zur Verfügung.

3.1 Geschichte

Als das Herstellerkonsortium das Projekt FreeSwan im Jahr 2003 beendete, geschah dies nicht aufgrund veränderter Marktsituation oder anderer Marketingstrategien. Die Zielvorgabe – die Referenz einer IPsec-Implementierung auf IPv4 zu schaffen – war erreicht, gegen diese können Hersteller ihre eigenen Implementierungen testen und kompatibel gestalten, ohne Produkte von Mitbewerbern intenstiv untersuchen zu müssen. Dieses als Reverse-Engineering bekannte Vorgehen, ist nicht nur in der Europäischen Union üblicherweise durch Gesetze und Kleingedrucktes in den Begleitpapieren der Geräte verboten, sondern auch durch die nun vorhandene und gut funktionierende Referenz überflüssig.

Da Stillstand gerne mit Rückschritt verwechselt wird, enstanden durch die Ankündigung der Fertigstellung von FreeSwan gleich zwei Nachfolgeprojekte: OpenSwan und StrongSwan. Letzteres wird vom Autor des X.509-Patches und seinem Team an der Eidgenössischen Technischen Hochschule in Rapperswil gepflegt. Mittlerweile ist es in der Version 4 erschienen, Updates erscheinen regelmässig und bei Bedarf. Die Software unterstützt auch das neuere Schlüsselaustauschverfahren IKEv2, welches gegenüber seiner Vorgängerversion deutlich schneller funktioniert und mit nur 2 Paketwechseln auskommt.

Um StrongSwan zu nutzen, ist nicht viel an Voraussetzungen zu schaffen: Erstens wird ein Vanilla Kernel-Source (2.4.x oder 2.6.x) benötigt, der sauber compiliert und danach auch auf der zu verwendenden Hardware läuft. Zweitens der StrongSwan-Patch, zu beziehen von `http://strongswan.org`, wird neben dem Quell-Verzeichnis des Kernels ausgepackt, dort führt ein Aufruf auf Shell-Ebene **make menugo** in das Konfigurationsmenü des Kernels, vermehrt um die Einstellmöglichkeiten eben diesen Patches. Ist alles fertig eingestellt und wird das Menü verlassen, beginnt skriptgesteuert der Compilerlauf, an seinem Ende sollte ein fertiger, nun etwas größerer Kernel entstanden sein; außerdem werden einige Userland-Programme zur Steuerung der Funktionalität erzeugt.

Sind die Ergebnisse auf dem Zielsystem angekommen (Niemand will auf einem IPsec-Gateway einen C-Compiler haben, oder?), kann schon mit der Konfiguration begonnen werden. Für den ersten Wurf reicht der bevorzugte Editor, wenns jedoch mehr und mehr verschiedene oder gleichartige Konfigurationsschnipsel werden, ist vielleicht ein skriptgesteuerter Ansatz zu bevorzugen.

3.2 Konfiguration

Um mehrere, gleichartige Firmenstandorte zu verbinden, ist zuerst eine geeignete, flexible Architektur festzulegen. Jeder Standort kann beispielsweise so wie in Abbildung Nr. 4 gezeigt ist, dargestellt werden. Nicht an jedem Standort müssen alle Komponenten tatsächlich realisiert sein, aber ein einheitliches Anschlußschema wie auch Adressierungsschema ist von Vorteil: CIDR bietet sich an, Netzwerk adäquat zu beschreiben. Abhängig von der Anzahl Mitarbeiter (oder anderen, geeigneteren) Maßstäben kann mit gutem Willen immer eine Netzmaske gefunden werden, die für alle Standorte passt. RFC1918-Adressen sind im internen Bereich selbstverständlich, wenn es auch noch Firmen geben mag, die offizielle Adressen aufgrund des ehemals zugewiesenen Überflusses oder auch aus Unkenntnis intern nutzen. Ein `/20` mag im Folgenden als Beispiel dienen und kann durch sinnvolle Aufteilung in kleinere, am Ort verschieden genutzte Netze für ein Unternehmen ausreichen. Variable Netzmasken schaffen so Struktur, das Gateway kann als Firewall auch gleich den IP-Verkehr zwischen Servern und Anwendern regulieren.

Die Abbildung Nr. 5 zeigt die Informationen, die mindestens vorhanden sein müssen, um ein IPsec-VPN „ans Laufen“ zu bringen. Um daraus eine funtionierende Konfiguration zu erzeugen, ist ein konstanter Vorspann, gefolgt von Sequenzen für jede IPsec-Verbindung in der Datei **ipsec.conf** und eine separate Datei **ipsec.secrets** nötig. Letztere enthält zu jedem IP-Paar aus der zweiten Spalte der Steuerdatei einen sog. „preshared key“(**PSK**). Diese PSK können aus **/dev/random** generiert werden, im normalen Betrieb eines Linuxsystems sollte hierzu unvorhersagbarer Zufall in ausreichender Menge vorhanden sein.

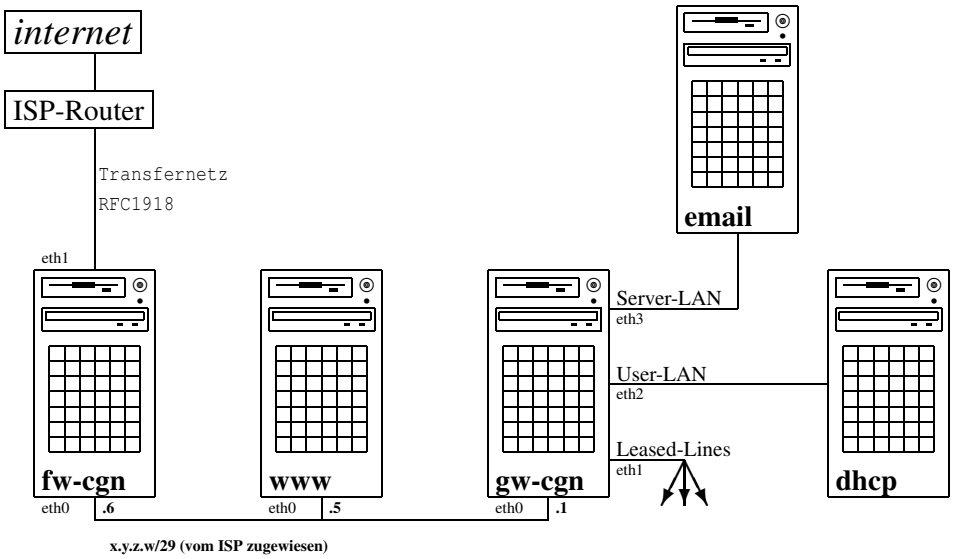

Abbildung 4: Typischer Firmen-Standort

Adressen aus 172.22.0.0/24 sind durch die jeweiligen externen ISP-Adressen zu ersetzen.

```
#loc.   gateway         next-hop        internal net
bln     172.22.0.41     172.22.0.46     10.11.48.0/21
cgn     172.22.0.25     172.22.0.30     10.11.40.0/21
nyc     172.22.0.65     172.22.0.70     10.11.4.0/21
sdy     172.22.0.17     172.22.0.22     10.0.0.0/8
kap     172.22.0.9      172.22.0.14     10.11.56.0/21
tok     172.22.0.1      172.22.0.6      10.11.16.0/21
to2     172.22.0.1      172.22.0.6      10.11.80.0/21
```

Abbildung 5: Steuerdatei für die IPsec-Konfiguration

```
version 2
config setup
        interfaces="%defaultroute"
        klipsdebug=none
        plutodebug=none
        uniqueids=yes
        nat_traversal=yes
#
conn bln-cgn
        auto=start
        authby=secret
        keyingtries=0
        disablearrivalcheck=no
        left=172.22.0.41
        leftnexthop=172.22.0.46
        leftsubnet=10.11.48.0/21
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.40.0/21
#
conn bln-nyc
        auto=start
        authby=secret
        keyingtries=0
        disablearrivalcheck=no
        left=172.22.0.41
        ...
```

Abbildung 6: IPsec-Konfiguration (Anfang)

```
#!/bin/bash
if [ -f /etc/ipsec.secrets.new ] ; then
        if [ -f /etc/ipsec.conf.const ] ; then
                /bin/cat /etc/ipsec.conf.const >> /etc/ipsec.conf
        fi
        mv /etc/ipsec.secrets.new /etc/ipsec.secrets
        /etc/init.d/ipsec restart
fi
```

Abbildung 7: cron-job

```
conn l2tp
        auto=add
        authby=rsasig
        keylife=28800
        keyingtries=5
        ikelifetime=8h
        rekeymargin=200m
        disablearrivalcheck=no
        left=%any
        leftrsasigkey=%cert
        rightrsasigkey=%cert
        rightid="C=DE, ST=Germany, O=hubertz-it-consulting GmbH, OU=
                        IPSec-gateways, CN=gw-kk"
        rightcert=/etc/ipsec.d/gw-kk.der
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.41.253/32
```

Abbildung 8: Aussendienstler-Tunnel

Sinnvolleweise werden die Konfigurationsdateien und PSK nicht auf den IPsec-Gateways, sondern auf einem ausschließlich zur Netzadministration genutzten PC generiert und anschließend verteilt. Danach werden diese neuen Konfigurationen durch Eingabe eines Kommandos duch ssh aktiviert, hierbei müssen die alten IPsec-Tunnel abgebaut und dann die neuen zuerst verhandelt und dann aktiviert werden. Ein **cron-job** kann dies gut entkoppelt von jeder Terminalsession erledigen. Die darin benutzte Datei `/etc/ipsec.conf.const` wird, wenn vorhanden, an die neue, generierte Konfigurationsdatei angehangen. Dieser Anhang besteht aus genau den Tunneldefinitionen, die an nur diesem Standort nötig sind, um bspw. standortspezifischen Projekten die notwendige Verbindung zu jeweiligen Geschäftspartnern zu ermöglichen. Auslandsverbindungen zu Mutter-/Tochterfirmen sind vielleicht auch nur an einem Standort des Unternehmens erwünscht. Und Aussendienstmitarbeiter wollen vielleicht aus der Ferne an ihre Ressourcen im internen Netz? Letzteres wird mit dem Konfigurationsschnipsel in Abbildung 8 ereicht, ein X.509-Zertifikat, etwas Software von `vpn-dialer.sf.net` (geschrieben von Thomas Kriener), und die richtige Konfiguration lassen Aussendienstler und Projektmitarbeiter eben genau da arbeiten, wo es für die Arbeit bestmöglich geschehen kann. Selbstvertändlich können hier auch andere, spezielle Konfigrationen mit in diesem Part untegebracht werden. Verschiedene Geschäftspartner haben üblicherweise verschiedene Firewall- und IPsec-Geräte im Einsatz, hier kann StrongSwan seine Stärken ausspielen. Einige Beispiel für die Kompatibilität mit kommerziellen Geräten seien hier aufgeführt:

- Nortel
- Checkpoint Firewall One
- Checkpoint NG R55 Build 127
- Sonicwall

Die gezeigten Konfigurationsschnipsel sind abgesehen von IP-Adressen aus laufenden Geräten entnommen. Bei der Werdung dieser Konfigurationen nicht zu vermeidende Erfahrungen liessen diese so als funktionierende, wirksame Konfigurationen zurück; Gegenbeispiele sind jederzeit willkommen. Dennoch mögen wenige Kommentare hierzu im Folgenden erlaubt sein.

Bei der Checkpoint Release 55 Build 127 ist z.B. wichtig, pfs auf „no“ und die Einstellungen der verschiedenen Timer so wie im Beispiel vorzunehmen. Wahrscheinlich treten aber beim ersten Verbindungstest zu jener Check-

```
conn to-nortel
        auto=start
        type=tunnel
        authby=secret
        compress=no
        esp=3des-md5
        ike=3des-md5-modp1024
        auth=esp
        pfs=no
        left=ip-of-his-gateway
        leftsubnet=10.137.61.0/27
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.40.0/21
```

Abbildung 9: Tunnel zu Nortel IPsec-Device

```
conn to-firewall-one
        auto=start
        type=tunnel
        authby=secret
        auth=esp
        esp=aes256-sha1
        pfs=yes
        keyexchange=ike
        keyingtries=0
        keylife=60m
        ikelifetime=24h
        disablearrivalcheck=no
        left=ip-of-his-gateway
        leftsubnet=192.168.20.0/24
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.40.0/21
```

Abbildung 10: Tunnel zu Firewall One

```
conn to-checkpoint
        auto=start
        type=tunnel
        authby=secret
        auth=esp
        pfs=no
        keyexchange=ike
        keyingtries=0
        keylife=120m
        ikelifetime=1h
        disablearrivalcheck=no
        left=ip-of-his-gateway
        leftsubnet=10.100.111.205/32
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.40.0/21
```

Abbildung 11: Tunnel zu Checkpoint NG R55 Build 127

```
conn to-sonicwall
        auto=start
        type=tunnel
        authby=secret
        auth=esp
        pfs=yes
        keyexchange=ike
        keyingtries=0
        keylife=8h
        ikelifetime=1h
        disablearrivalcheck=no
        left=ip-of-his-gateway
        leftsubnet=192.168.33.32/32
        right=172.22.0.25
        rightnexthop=172.22.0.30
        rightsubnet=10.11.40.0/21
```

Abbildung 12: Tunnel zu Sonicwall

point Firewall trotzdem Fehler auf. Mit dem dortigen Administrator sind ja nicht nur vorab PSK und beteiligte IP-Adressen abzusprechen, sondern insbesondere auch die Tunnel-Inhalte, also `left-` und `rightsubnet`. Alle diese Parameter werden üblicherweise in Checkpoints Produkten über ein GUI dem Gerät bekanntgemacht. Der Admin vertraut natürlich darauf, alles richtig gemacht zu haben. Aus den StrongSwan-Logs ergibt sich zumeist jedoch eine kleine Fehlermeldung in der Art wie „`NO_PROPOSAL_CHOSEN`". Tiefere Enstörung ist also gefragt. Das Kommando `ipsec whack -help` gibt einen kurzen Überblick, wie man am Besten seine Festplatte mit überflüssigen Meldungen vollschreiben kann. Mit dem Namen der IPsec-Session kann dann gezielt die Information ins Log geschrieben werden, die gerade gesucht werden; `ipsec whack -name to-checkpoint -debug-all` erzeugt ausreichend Lesestoff für einen Abend, selbst wenn schon nach wenigen Minuten das gegenteilige Kommando `ipsec whack -name to-checkpoint -debug-none` gegeben wird. Schaut man lange genug in die so erzeugten Logs, findet man ziemlich sicher die Ursache der Fehlermeldung: Die andere Seite verhandelt nicht das, was abgesprochen wurde. Sondern, wenn z.B. nur genau ein interner Rechner auf der anderen Seite erreichbar sein sollte, also ein `/32`, so erscheint in den hiesigen Logs unvermutet das Netz, in dem jener Rechner angesiedelt ist, z.B. ein `/22`. Der neugierige Logdateileser erfährt auf seine Frage, dies sei doch das interne Netz und dazu sei eine Route im Gerät drin. Also müsse das doch richtig sein. Sobald dann die hiesige Konfiguration entsprechend angepasst ist, funktionieren Tunnel und Kommunikation. Fazit: Traue nie einem Gerät, dessen Quellen nicht lesbar sind. Oder sollte vermutet werden, dies sei die einzige Ungereimtheit in jener teuren Anlage? Das sich am Kerkhoffschen Prinzip seit 1883 nichts ändert, hat sich offensichtlich noch nicht herumgesprochen. Und das darunter nicht nur der Algorithmus, sondern sehr wohl auch die Implementierung fällt, sollte nicht erst durch Bruce Scheiers Bücher bekannt sein. Lediglich die Geheimhaltung der Schlüssel sollten zur Sicherheit der Verschlüsselung beitragen, die Methode sollte allgemein diskutabel sein.

Andere Devices haben andere Einschränkungen, z.B. mögen nicht alle Geräte Sonderzeichen in den PSKs. Die Logs mit variabel einstellbarer Geschwätzigkeit sind stets eine sehr wertvolle Hilfe, auf der gegenüberliegenden Seite selbst unbekannte Geräte mit StrongSwan zu funktionierenden IPsec-Sessions zu bewegen.

3.3 Betrieb

Im Laufe der Zeit wird in größeren Unternehmen ab und zu mal umstrukturiert, Vorstände wechseln, Internetprovider wechseln, Kollegen und Geräte ebenfalls. Jedoch sind die Gründe, IP-Nummern zu ändern, nur selten technischer Art. Wenn ein gut funktionierendes internes Netz wegen golfspielender Geschäftsführer geändert werden muß, kommen schon mal Zweifel ob der Sinnhaftigkeit auf. Andererseits stellt es, mit gut funktionierenden StrongSwan-Konfigurationen am ISP A im laufenden Betrieb hin zu ISP B zu wechseln, zusätzlich versehen mit dem Hemmschuh, bei ISP B nur an einem Standort einen Internet-Anschluß und ansonsten MPLS-VPNs zu haben, schon eine gewisse Herausforderung an den Netzwerkadministrator dar. Die Migration der Steuerdatei in Abbildung Nr. 5 hin zu der Version in Abbildung Nr. 13 stellt eine mögliche Lösung dar. Sie stellt auch durch das MPLS-VPN, welches gemäss seiner RFC-Spezifikation nicht zwingend Verschlüsselung bietet, sicher, dass aller Verkehr innerhalb des Unternehmens auf den WAN-Strecken verschlüsselt ist. Allerdings muß der Verkehr zu Geschäftspartnern und Aussendienstlern auf einem zusätzlichen IPsec-Gateway separiert werden, da das MPLS-VPN keinen Weg nach außen zulässt.

3.4 Hochverfügbarer Betrieb

Je wichtiger verschlüsselter Datenverkehr zu Geschäftspartnern und Aussendienstlern im eigenen Unternehmen wird, umso kritischer werden Ausfallzeiten bewertet. Der Internetzugang, Web- und Email werden als unternehmenskritische Anwendungen eingestuft mit der Folge, dass ausreichend Mittel zur Verfügung stehen, Hochverfügbarkeit durch Redundanz herzustellen.

Für den eigentlichen Internetzugang stehen altbewährte Mittel wie Peering mit eigenem AS zur Verfügung. Allerdings ist diese Art von Redundanz relativ kostenintensiv und wird nicht für alle Standorte die optimale

```
# loc.  gateway          next-Hop         subnet
bln     172.22.0.41      172.22.0.46      10.11.48.0/21
cgn     172.22.0.25      172.22.0.30      10.11.40.0/21
nyc     172.22.0.65      172.22.0.70      10.11.4.0/22
sdy     172.22.0.17      172.22.0.22      10.0.0.0/8
kap     172.22.0.9       172.22.0.14      10.11.56.0/21
tok     172.22.0.1       172.22.0.6       10.11.16.0/21
to2     172.22.0.1       172.22.0.6       10.11.80.0/21
# inet4all via cgn
I01     172.22.0.25      172.22.0.30      0.0.0.0/1
I02     172.22.0.25      172.22.0.30      128.0.0.0/3
I03     172.22.0.25      172.22.0.30      160.0.0.0/5
I04     172.22.0.25      172.22.0.30      168.0.0.0/6
I05     172.22.0.25      172.22.0.30      172.0.0.0/12
### !!! never open next line or gateways will be lost !!!!
### !!!Ixx 172.22.0.25  172.22.0.30      172.16.0.0/12 !!!
I06     172.22.0.25      172.22.0.30      172.32.0.0/11
I07     172.22.0.25      172.22.0.30      172.64.0.0/10
I08     172.22.0.25      172.22.0.30      172.128.0.0/9
I09     172.22.0.25      172.22.0.30      173.0.0.0/8
I10     172.22.0.25      172.22.0.30      174.0.0.0/7
I11     172.22.0.25      172.22.0.30      176.0.0.0/4
I12     172.22.0.25      172.22.0.30      192.0.0.0/3
```

Abbildung 13: Steuerdatei für die migrierte IPsec-Konfiguration

Lösung darstellen, weil diese nicht alle gleichermassen kritisch sind. Speziell für die IPsec-Tunnel ist eine einfachere Lösung denkbar, die zwar auch zwei oder mehr ISP pro wichtigem Standort benötigt, jedoch den verschlüsselten Verkehr nicht durch Wechsel der Routen auf den IPsec-Gateways beeinflusst: Man nehme pro ISP einen zusätzlichen Router, der IPIP-Tunnel und dynamisches Routing mit z.B. OSPF beherrscht und kann so Redundanz fürs interne Netz zu sehr geringen zusätzlichen Kosten schaffen. Abbildung Nr. 14 zeigt das Prinzip.

Der IPsec-Verkehr wird durch IPIP-Tunnel geleitet, die Ziele sind bspw. durch OSPF mehrfach erreichbar, so dass der Ausfall eines ISP zu verkraften ist. Die Ausfallzeiten beschränken sich auf die Konvergenzzeiten des Routingprotokolls, mit Standardeinstellungen sollte 30-35 Sekunden nach einem Ausfall der Verkehr wieder fliessen. Bei Wiedereintritt des ausgefallenen IPIP-Tunnels ist kein Verlust zu bemerken. Die zusätzliche Router können auch mit Linux und Quagga realisiert sein, proprietäre Hardware funktioniert meist aber „out of the box“ und trägt ja auch an dieser Stelle zur Vertraulichkeit des internen Netzes nichts bei.

4 Ausblick

StrongSwan in der Version 2 als Nachfolger von FreeSwan stellt eine stabile, kryptographisch zufriedenstellende Softwarelösung dar, die ausreichende Vertraulichkeit herstellt. In Verbindung mit Linux und seinen Bordmitteln zur Absicherung (iptables) vor Fremdeinwirkung kann performant und ausfallsicher damit ein Unternehmensübergreifendes VPN fürs alltägliche Geschäft wie auch zur Nutzung mit VoIP hergestellt werden.

Neuerungen im Bereich VPNs kommen zwar fast täglich auf den Markt, jedoch ist Vorsicht angeraten. Bruce Schneier, amerikanischer Krypto-Experte, schreibt hierzu in seinem Buch **Secrets and Lies:**[1] „Jeder, der eine eigene kryptographische Grundfunktion erstellt, ist entweder ein Genie oder ein Narr. Angesichts des Genie/Narr-Verhältnisses stehen die Chancen nicht gut.“

Die aktuelle StrongSwan Version 4 ist von diesem schönen Spruch natürlich nicht getroffen, ob jedoch neu herausgegebene Software bereits für die produktive Unternehmenskommunikation benutzt werden kann, muß jedes Unternehmen selbst entscheiden. StrongSwan bietet als Neuerung nun IKE in der Version 2, d.h. eine mit nun 4 gegenüber vorher 7 Paketen deutlich schnellere Verhandlung der Parameter. Außerdem ist mittels eines Vermittlungsservers doppeltes NAT möglich, d.h. zwei Clients je hinter einem NAT-Gateway tauschen mit der Vermittlungsstelle die Parameter und anschließend untereinander direkt die verschlüsselten Inhalte. Dies kommt

[1] ISBN 3-89864-302-6, S.110

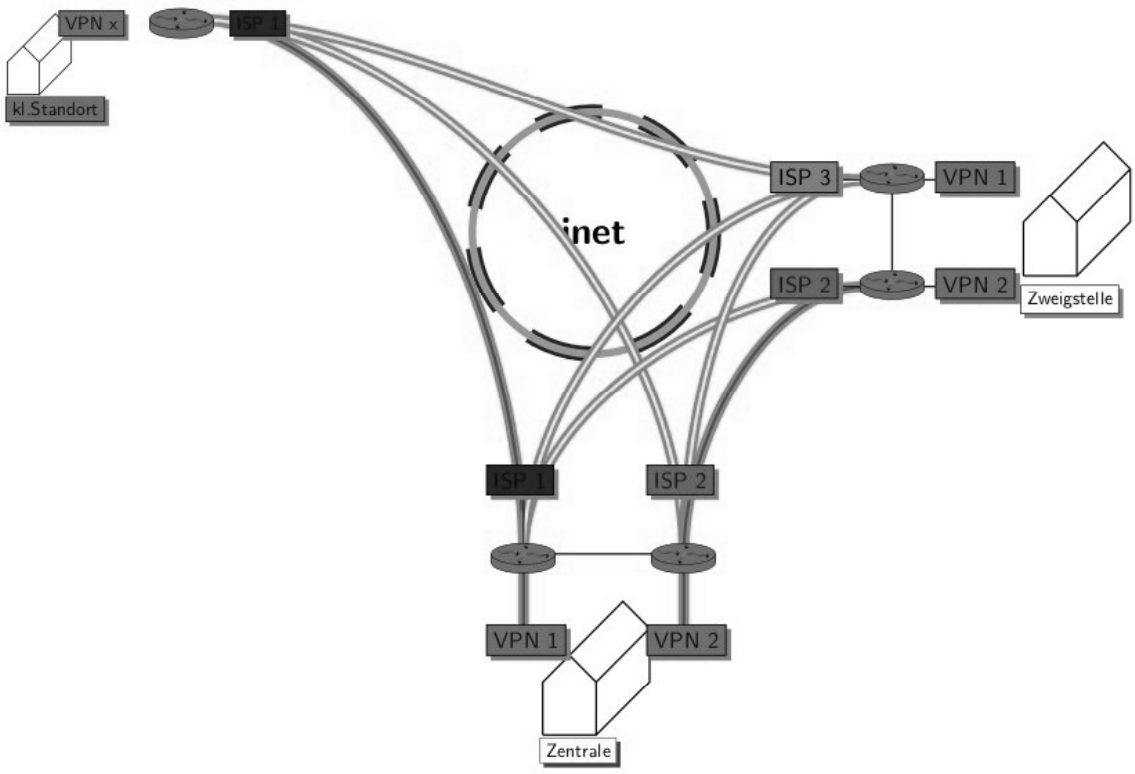

Abbildung 14: Hochverfügbares VPN

unmittelbar aktuellen VoIP-Lösungen zugute, bei denen Vertraulichkeit üblicherweise nicht enthalten ist. Proprietäre Lösungen wie z.B. Skype funktionieren zwar, lassen aber Vertraulichkeit aufgrund des verwendeten Prinzips „security by obscurity“ vollständig vermissen. Im Unternehmensumfeld sollte eine verantwortungsvolle Risikoabschätzung auch diese Art von Ungereimtheiten ans Tageslicht bringen.

5 Schlussbemerkungen

Nach nun mehr als 7 Jahren Betrieb, teilweise mit bis zu 35 Geräten gleichzeitig, die mit FreeSwan oder StrongSwan zufriedenstellend untereinander, aber auch mit verschiedensten proprietäten Gegenstellen kommunizieren, bleibt zu sagen, dass mit dieser Software nur positive Erfahrungen gemacht werden konnten. Es ist zwar manchmal mühsam, die Logdateien zu lesen, aber immer lehrreich. Wenn eine Verbindung fertig konfiguriert ist, hat man üblicherweise nie wieder Ärger mit der Sache. Sehr ähnlich, wenn auch nicht so langfristig, sind die Erfahrungen mit OpenVPN. Um Aussendienstler einfach ans Unternehmen anzuschliessen, ist es schnell, unkompliziert und betriebssicher; in jedem Fall ist es deutlich einfacher als die vormals preferierte Lösung mit L2TP durch IPsec. Damit kommt es gleichermassen dem Budget wie auch dem Administrator entgegen.