

VPN-Anbindung von Windows-Clients an FreeS/WAN

Wolfgang Barth¹

Version 1.1
2004-04-22

¹wob@swobspace.de

Zusammenfassung

Dieses Dokument beschreibt die Anbindung von Windows-Clients (Win2k, WinXP) via IPsec an ein VPN-Gateway unter Linux mit FreeS/WAN in zwei Varianten. Zum einen wird die native IPsec-Fähigkeit des Windows-Clients genutzt, die allerdings am Windows-Client einige Konfigurationsarbeit erfordert. Hier wird das Tool von Marcus Müller eingesetzt, mit dem sich die erforderlichen Einstellungen auf Windowsseite sehr einfach und FreeS/WAN-like durchführen läßt (<http://vpn.ebootis.de>).

Die zweite Variante – IPsec in Kombination mit L2TP – ist die von Microsoft bevorzugte Version. Sie ermöglicht eine sehr einfache Konfiguration am Windows-Client, setzt aber am VPN-Gateway die zusätzliche Konfiguration eines L2TP-Daemons und des PPP-Daemons voraus.

Alle in diesem Dokument enthaltenen Programme, Darstellungen und Informationen wurden sorgfältig zusammengestellt und geprüft. Trotzdem sind Fehler nie ganz auszuschließen. Der Autor übernimmt daher keinerlei Garantie für eine fehlerfreie Funktionalität, die sich aus der Konfiguration nach den hier gemachten Angaben ergibt.

Alle Warennamen, Handelsbezeichnungen und Gebrauchsnamen sind eingetragene Warenzeichen der jeweiligen Eigentümer. Der Begriff Windows wird generisch benutzt für die Betriebssysteme von Microsoft. Microsoft, Windows, Windows 2000, Windows XP und andere Namen sind Marken und/oder eingetragene Warenzeichen von Microsoft.

Eine Wiedergabe von Warennamen, Handelsbezeichnungen und Gebrauchsnamen hier auch ohne besondere Kennzeichnung berechtigt nicht zur freien Verwendung derselben.

©Alle Rechte bei Wolfgang Barth. Das Dokument kann für den eigenen, auch firmeninternen Gebrauch (etwa im Intranet) weitergegeben werden, wenn es unverändert bleibt und ein vollständiger, eindeutiger Quellenhinweis auf die Urheberschaft hinweist.

History

2004-04-18 Version 1.1: Erweitert um IPsec/L2TP

2003-12 Erste Version, basierend auf IPSEC.EXE von Markus Müller

Inhaltsverzeichnis

1	Einführung	5
1.1	Natives IPsec oder IPsec/L2TP?	6
1.2	Konfigurationsschema	8
1.3	Mit Windows ans Internet: Safety first!	11
1.4	FreeS/WAN, Openswan und strongSwan	13
2	Konfiguration des Gateways	14
2.1	Zertifikate	14
2.2	FreeS/WAN-Konfiguration	16
2.3	Setup bei Verbindungsauf-/abbau des PPPD	19
3	Konfiguration des Windows-Clients (IPSEC.EXE)	21
3.1	Installation der Hilfstools	21
3.2	Import des Zertifikates	23
3.3	Lokale Konfiguration: ipsec.conf	23
3.4	Verbindung starten	24
4	IPsec/L2TP	27
4.1	L2TP-Protokollstack	27
4.2	Windows-Client: private Verbindung über VPN	28
4.3	Konfiguration des VPN-Gateways	34
4.3.1	FreeS/WAN-Konfiguration	36
4.3.2	L2TP-Konfiguration	37
4.3.3	Firewalling	39
4.4	Troubleshooting	40

A	Generierung von X.509-Zertifikaten	41
A.1	Certifikation Authority	42
A.2	Zertifikat erstellen	43
A.3	Zertifikat nach Windows exportieren	44
B	Zertifikate unter Windows	45
C	Dynamisches DNS für das VPN-Gateway	52
D	Nützliche Werkzeuge für die Windows-Clients	55
E	Links	56

Kapitel 1

Einführung

Öffentliche Netze sind unsicher. Wenn sich diese Aussage auf das Internet als „das öffentliche Netz“ schlechthin bezieht, gibt es wohl kaum irgendwelche Widersprüche. Regelmäßige Virenattacken sorgen dafür, daß die „Gefahr Internet“ nicht so schnell aus dem Bewußtsein verschwindet.

Leider muß man auch die inzwischen sehr weit verbreiteten Funk-Netzwerke, auch bekannt unter dem Begriff „Wireless Lan“ oder kurz: WLAN, zu den öffentlichen Netzwerken zählen. Bisherige Verschlüsselungsmechanismen sind nicht ausreichend, und sehr oft werden sogar die einfachsten Sicherungsmaßnahmen nicht umgesetzt. Stichproben in deutschen Großstädten zeigen seit fast zwei Jahren, daß mehr als 60% aller WLANs unzureichend oder völlig ungesichert betrieben werden. Die Stichproben sind zwar nicht repräsentativ, erschreckend aber ist trotzdem, daß trotz einer anhaltenden Diskussion in den Medien nur wenig Aufwand betrieben wird, diese Funknetzwerke abzusichern. In diesem Sinne muß man auch WLANs als – ungewollte, aber real existierende – öffentliche Netzwerke betrachten.

In jedem Netzwerk – unabhängig davon, ob es sich um ein Firmennetzwerk oder ein privates Netzwerk handelt – stellt sich irgend wann einmal die Frage, ob man nicht von außen Zugriff auf das Netzwerk nehmen könnte. Früher übliche Einwahlen per Modem oder ISDN sind nicht nur teuer (Verbindungsgebühren), sondern auch nicht mehr zeitgemäß. Ein Zugriff über das Internet muß her. Allerdings ist das Internet als öffentliches Netzwerk prinzipiell unsicher, der Zugang muß besonders abgesichert werden.

Ähnlich gelagert ist der Zugriff per Funklan, sei es vom Firmengelände aus, oder zuhause einfach von der Terasse oder Balkon (wer sitzt bei schönem Wetter schon gerne drinnen ;-). Wie oben bereits angedeutet, muß man Funklans ebenfalls wie öffentliche, unsichere Netzwerke behandeln.

Die gängigste Methode, einen sicheren Zugang auf ein internes Netzwerk über öffentliche Netze zu gewährleisten, ist ein „Virtual Private Network“ VPN. Die Daten werden eingepackt und verschlüsselt übertragen, um einen Fremdzugriff zu verhindern.

Gegenstand dieses Dokumentes ist der Einsatz des Protokolls IPsec, das nach wie vor als die sicherste Methode für VPNs gilt. Als Gateway wird ein VPN-Router unter Linux mit FreeS/WAN eingesetzt, für die Clientseite wird die Konfiguration für Windows (Win2k und WinXP) beschrieben.

Grundkenntnisse in Firewalling unter Linux sowie Grundkenntnisse in der Konfiguration von FreeS/WAN werden vorausgesetzt, sie sind nicht Gegenstand des vorliegenden Dokumentes. Wer sich noch nicht mit Firewalling unter Linux und der Konfiguration von FreeS/WAN beschäftigt hat, sei auf „Das Firewall Buch“, 2. Auflage 2003, SuSE Press vom gleichen Autor verwiesen. Dort wird auch der Einsatz von Linux-Clients für ein VPN-Gateway beschrieben.

1.1 Natives IPsec oder IPsec/L2TP?

Aus Sicht des Linux-Gateways ist natives IPsec die einfachere Variante. Am Linux-Gateway richtet man ganz normal den Zugang für Roadwarrior in Kombination mit Zertifikaten ein. Auf der Windowsseite ist ebenfalls eine vergleichbare Konfiguration erforderlich, die Dank des Tools von Marcus Müller die einem die aufwendige Konfiguration von Sicherheitsrichtlinien abnimmt, man erstellt einfach eine Konfigurationsdatei, die sich sehr stark an der FreeS/WAN-Syntax orientiert.

Die IPsec/L2TP-Variante ist zwar auf Windowsseite einfacher, da der Client sozusagen zum Betriebssystem gehört, man konfiguriert einfach einen Internet-Zugang über ein privates Netzwerk und nimmt dann noch ein paar einfache Einstellungen vor. Am Gateway dagegen ist zusätzlich ein L2TP-Daemon und der PPP-Daemon erforderlich. IPsec dient hier nur als Transport-Protokoll, das L2TP verschlüsselt über das Internet an das Gateway weiterleitet. Über L2TP wird wiederum das PPP-Protokoll transportiert, weil sich damit Dinge wie Authentifikation (Userspezifisch), Zuweisung von IP-Adressen, DNS- und WINS-Servern erledigen lassen.

Die nachstehende Übersicht von Vor- und Nachteilen bezieht sich auf das IPsec-Tool von Markus Müller bei nativem IPsec, beziehungsweise auf den Einsatz des l2tpd-Daemons (www.l2tpd.org) bei IPsec/L2TP und stellt nur eine Zusammenfassung dar. Ein sehr ausführlicher Vergleich findet sich unter:

<http://www.jacco2.dds.nl/networking/freeswan-l2tp.html>

Vorteile bei nativem IPsec

- FreeS/WAN-like Konfiguration am Windows-Client.
- Außer FreeS/WAN keine aufwendige Konfiguration notwendig.
- Kein Protokoll-Overhead

Nachteile bei nativem IPsec

- Umständliche Einwahl: erst RAS-Verbindung aufbauen, dann das IPsec-Tool starten, dann einen Ping auf die Gegenstelle, damit der Tunnel aufgebaut wird.
- Keine virtuelle IP-Adresse, Client erhält die dynamische IP des Providers (schwierigere Paketfilterung).
- Nicht NAT-T fähig.

Vorteile von IPsec/L2TP

- Eingebauter Client bei Win2k/XP, für andere Microsoft-Betriebssysteme gibt es einen kostenlosen L2TP-Client von Microsoft.
- Userfriendly Dialin: Verbindungsauf-/abbau über einen Vorgang (RAS-Verbindung wird automatisch mit auf- und abgebaut).
- Einfache Konfiguration über Wizard (Internet-Verbindung über privates Netzwerk).
- Virtuelle IP-Adresse zuweisbar (sogar userabhängig!)
- Unterstützt NAT-T

Nachteile von IPsec/L2TP

- Protokoll-Overhead: IP->IPsec->L2TP->PPP, dadurch langsamer.
- erhöhter Konfigurationsaufwand am VPN-Gateway.

Letztenendes muß jeder die Entscheidung für sich selbst treffen. Für IPsec/L2TP spricht der „eingebaute Windowsclient“ und die Möglichkeit, virtuelle IP-Adressen zuzuweisen.

1.2 Konfigurationsschema

Die Konfiguration, auf die sich das Dokument bezieht, stellt Abbildung 1.1 dar. Das VPN-Gateway besitzt drei Netzwerkinterfaces: `eth0` zum lokalen Netzwerk hin, `ppp0` als DSL-Anschluß zum Internet, und `eth1` für ein WLAN. Implizit bedeutet das, daß der DSL-Anschluß über `eth2` läuft, wenn beide VPN-Anbindungen gleichzeitig zum Einsatz kommen. Wer nur DSL einsetzt, wird `ppp0` über `eth1` abbilden, und wer nur WLAN anbindet, wird ebenfalls nur `eth1` verwenden. In jedem Falle kann so die Konfiguration auf dem VPN-Gateway 1:1 übernommen werden, unabhängig davon, ob man nur WLAN, nur DSL oder beides einsetzt.

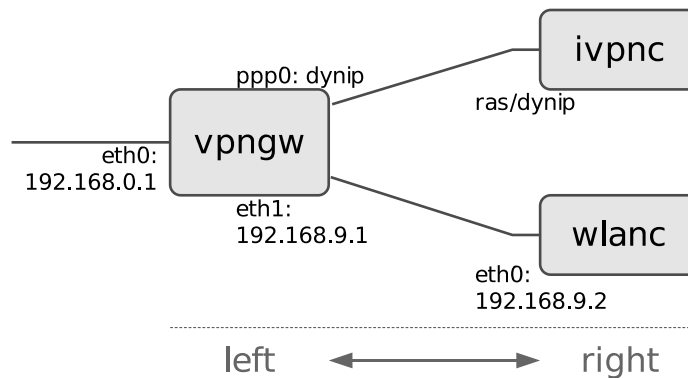


Abbildung 1.1: Konfigurationsschema für die Anbindung Internet- und WLAN-VPN-Clients

Für die Konfiguration von Gateway und Clients wird die eigene Seite und die fremde Seite in *left* und *right* unterschieden. Die Konvention hier: *left* ist immer das VPN-Gateway, *right* sind immer die Clients, unabhängig davon, ob man die Konfigurationsdaten auf der Client-, oder auf der Gateway-Seite betrachtet.

□ vpngw

Das VPN-Gateway läuft unter Linux mit FreeS/WAN. Hierfür ist ein aktueller Kernel 2.4.x erforderlich ($\geq 2.4.23$). Der Kernel muß für FreeS/WAN gepatcht werden, dabei kommt FreeS/WAN in der Version 2.0x ($\geq 2.0.4$) zum Einsatz (www.freeswan.org). Ab FreeS/WAN Version 2.04 gibt es bereits Unterstützung für die Kernelserie 2.6, die aber im Moment noch (vor dem Erscheinen des 2.6.0) als experimentell bezeichnet werden muß.

Die Fähigkeit, X.509-Zertifikate zu verwenden, die für die Anbindung von Windows-Clients erforderlich ist, bringt FreeS/WAN ebenfalls nicht automatisch mit. Hierzu gibt es Patches, die auf FreeS/WAN angewendet werden müssen, bevor der eigentliche Kernel gepatcht wird (www.strongsec.com). Wie die Patches in den Kernel kommen, wird in „Das Firewall Buch“ von Wolfgang Barth, 2. Auflage, SuSE Press, beschrieben. Außerdem finden sich Informationen hierüber auf den Webseiten der entsprechenden Projekte. Sie sind im Anhang E zusammengestellt.

Das VPN-Gateway hat direkte Verbindung zum Internet bzw. zu einem WLAN, es sollte daher selbstverständlich sein, das Gateway über geeignete Paketfilterung abzusichern und das System zu härten. Darauf sollten nur unbedingt notwendige Dienste laufen (etwa `sshd`), alles andere ist abzuschalten.

– Lokales Netzwerk: `eth0`

IP-Adresse	192.168.0.1
Netzwerk	192.168.0.0/255.255.255.0
Broadcast	192.168.0.255

– Internet: `ppp0` (DSL)

IP-Adresse	dynamisch
IPsec-Interface	<code>%defaultroute(=ipsec0)</code>

Die Adressvergabe bei einfachem DSL ohne statische IP erfolgt dynamisch, am einfachsten ist es daher, für die FreeS/WAN-Konfiguration `%defaultroute` für die Identifikation des richtigen Interfaces und auch der aktuellen IP-Adresse einzusetzen.

– WLAN: `eth1`

IP-Adresse	192.168.9.1
Netzwerk	192.168.9.0/255.255.255.0
Broadcast	192.168.9.255
IPsec-Interface	<code>ipsecl=eth1</code>

Beim der WLAN-Anbindung wird ein privater IP-Adressbereich verwendet. Das Interface `eth1` bekommt eine feste IP-Adresse zugewiesen, für die Clients `wlanc` kann man sich überlegen, ob man dort die IP-Adressen fest vergibt, oder ob man im WLAN einen DHCP-Server einsetzt.

□ `ivpnc`

Der Internet-VPN-Client `ivpnc` wird über RAS angebunden. Dabei wird angenommen, daß `ivpnc` direkt die dynamische IP-Adresse be-

kommt, sich also kein Router mit NAT zwischen dem Client und dem Internet befindet.

Für einen zwischengeschalteten NAT-Router ist eine *NAT Traversal* Funktion von FreeS/WAN erforderlich. NAT-T ein eigener, zusätzlicher Patch von FreeS/WAN, der im Augenblick leider nicht für die FreeS/WAN-Serie 2.0x vorliegt. Wer auf NAT-T angewiesen ist, muß hierfür FreeS/WAN 1.99 mit dem passenden Patch einsetzen. Die Konfiguration von FreeS/WAN 1.99 weicht von der in diesem Dokument vorgestellten Konfiguration an einigen Stellen ab.

□ wlanc

Die Konfiguration des WLAN-Clients `wlanc` ist relativ einfach. Man sorgt dafür, daß die WLAN-Netzwerkkarte eine IP-Adresse aus dem verwendeten Bereich erhält (normale Windows-Netzwerkkonfiguration).

In der hier verwendeten Konfiguration ist das VPN-Gateway der einzige Rechner zwischen internen Netzwerk und den öffentlichen Netzen. Wo immer möglich sollte man eine mindestens zweistufige Lösung einsetzen. Besser wäre sogar noch eine dreistufige Lösung, die vor allem dann sehr sinnvoll ist, wenn man über eine Standleitung mit festem IP-Adressbereich an das Internet angebunden ist und man dem VPN-Gateway eine eigene, feste IP-Adresse zuteilen kann.

Eine zwei- beziehungsweise dreistufige Lösung ändert prinzipiell nichts an der VPN-Konfiguration, nur die Firewall-Regeln müssen angepaßt werden. Da es hier ausschließlich um die VPN-Konfiguration, nicht um das vollständige Firewalling geht, sei für die mehrstufige Lösung auf das bereits genannte „Das Firewall Buch“ 2. Auflage, verwiesen.

Das VPN-Gateway kann nicht nur ein selbstinstallierter Linux-Firewall sein, sondern genauso gut eine Linux-basierte fertige Lösung, etwa ein CD-basierter Firewall (*HSPfire::, SuSE Firewall on CD*) oder eine sogenannte Appliance (zum Beispiel *Benhur II* von Pyramid). Fertige Lösungen bieten meistens vorkonfigurierte Schemas oder eine graphische Administrationsoberfläche.

Sofern die fertige Lösung ein Administrationsinterface mitliefert, sollte man unbedingt dieses verwenden, da die Konfigurationsdaten oft datenbankbasiert gespeichert und die Konfigurationsfiles daraus automatisch generiert werden. Manuelle Änderungen an Konfigurationsfiles können bei fertigen Lösungen beim nächsten Aufruf des Admin-Interfaces wieder überschrieben werden.

1.3 Mit Windows ans Internet: Safety first!

Bei den ersten Gehversuchen hat der Autor selbst unangenehme Erfahrungen mit der Einwahl von Windows 2000 ins Internet gemacht. Der Rechner, eine nackte Win2k-Installation mit Service-Pack 2, dient ausschließlich zu Testzwecken. Die Einwahl erfolgte via RAS/ISDN über eine Call-by-Call Nummer ins Internet. Keine 30 sec. nach der Einwahl erfolgte bereits der Absturz von `svchost.exe`, ein Programm, das sich mit um die RAS-Verbindung kümmert. Die RAS-Verbindung ist dann weder zu trennen (nur physikalisches „Stöpsel raus“ hilft hier noch), noch zu konfigurieren. Neuboot ist angesagt, und spätestens nach 30 sec. beginnt das Spiel von vorne.

Die Ursache: einer der Würmer, die den RPC-Bug unter einigen Betriebssystemen ausnutzen, versucht eine Infektion über Port 135. An manchen Tagen kann man an einem ganz normalen DSL-Anschluß (natürlich mit einer Linux-Firewall gesichert) unmittelbar nach der Einwahl mehr als 30 Hits pro Minute (!) feststellen, alles Internet-Nutzer, die sich nicht um die Absicherung ihres Systems gekümmert haben.

Daher sollte man beim Windows-Client einige Spielregeln beachten, bevor dieser die Verbindung zum Internet aufnimmt:

❑ *Personal Firewall verwenden.*

Eine restriktive Handhabung der erlaubten Verbindungen über eine Personal Firewall ist unbedingt notwendig, um direkte Netzwerkangriffe zu vermeiden. Es tauchen immer neue Exploits zu bisher unbekanntem Sicherheitslücken auf, denen aber allen gemeinsam ist, daß sie versuchen, eine Netzwerkverbindung zum lokalen Client aufzubauen, bevor die Infiltration erfolgt. Unterbindet man zunächst alle Verbindungen, insbesondere diejenigen, die von Außen zum Client aufgebaut werden sollen, hat man eine reale Chance, auch unbekannte, neue Würmer (die netzwerkbasierete Angriffe starten) fernzuhalten.

Allerdings ist eine Personal Firewall manchmal unbequem. Gerade bei der Installation neuer Programme oder dem Einsatz bisher noch nicht verwendeter Dienste kann die ständige Fragerei, ob man eine Verbindung auch wirklich erlauben will, nervig sein. Trotzdem: lieber sinnvolle Fragen beantworten und sich etwas Zeit nehmen ist immer noch besser, als sich bequem, aber ungefragt irgendwelche Viecher einzufangen.

❑ *Virens Scanner installieren, regelmäßig aktualisieren und benutzen (!)*

Nicht alle Infektionen eines Rechners geschehen über eine direkte Netzwerkverbindung von außen. Attachements von Emails oder normal über das Web gezogene Dateien könnten verseucht sein. Hier hilft nur ein zusätzlicher Virenschanner, denn eine Personal Firewall ist nicht in der Lage, den Inhalt von Dateien auf schädliche Funktionen zu prüfen. Kombinationsangriffe in der Vergangenheit haben gezeigt, daß manchmal auch ein per Email oder Download eingefangener Virus eine Personal Firewall deaktivieren kann, was dann wiederum für direkte Angriffe aus dem Internet Tür und Tor öffnet.

□ *Alle aktuellen Patches einspielen.*

Unabhängig von den anderen Sicherungsmaßnahmen sollte man immer alle aktuellen, verfügbaren Patches am Client einspielen. Man ist nie davor gefeit, den Virenschanner falsch zu konfigurieren, mal versehentlich abzustellen oder es treten Probleme beim Aktualisieren der Virensignatur auf. Eine Personal Firewall hilft meist sehr viel, manchmal aber auch zuviel, so daß man beim Testen auch schnell mal in die Versuchung kommt, sie kurz abzustellen um auszuprobieren, ob man ein aufgetretenes Problem an der Firewall liegt oder nicht. Manchmal reichen wenige Sekunden ungeschützt, um das System zu infizieren.

Nicht immer durchgehend, aber sehr oft verwenden neue Würmer Exploits, die auf einer älteren, schon länger bekannten Sicherheitslücke beruhen. Nichts ist peinlicher für einen Administrator als eine Infektion eines Systems, die man mit einem bereits 6 Monate alten Patch hätte vermeiden können.

Last but not least sollte man sich darüber im Klaren sein, das trotz aller Sicherheitsmaßnahmen am Client immer noch von dort Gefahr drohen kann, weil die Sicherheitsmaßnahmen dort trotz aller Sorgfalt auch einmal versagen könnten. Das ist leider nie ganz auszuschließen.

Für das VPN-Gateway bedeutet dies: auch durch den relativ sicheren Tunnel nur Verbindungen nach drinnen erlauben, die unbedingt nötig sind. Man sollte generell alles verbieten, und nur die absolut notwendigen Dinge erlauben.

Eine gute Lösung wäre etwa, innerhalb eines Firmennetzwerkes nur auf einen Terminalserver Zugriff zu gewähren, und jeglichen Datenaustausch zwischen Server und Client zu verhindern.

Dort, wo etwa über Netzwerklaufwerk Dateien ausgetauscht werden müssen, kann man einen gesicherten Bereich bereitstellen, zum Beispiel ein gesonderter Server in einem Grenznetz, der von innerhalb des Netzwerkes und via VPN-Tunnel erreichbar ist. Darüber lassen sich dann Daten mit dem VPN-Client austauschen, den direkten Zugriff auf interne Server

könnte man ganz verbieten. Der spezielle Server muß gezielt überwacht und regelmäßig auf Viren, Trojaner und Würmer gescannt werden.

Entsprechende Protokollierung von unerlaubten Verbindungsversuchen am VPN-Gateway helfen, etwaige infizierte Clients schnell zu identifizieren, da viele Würmer, die zur Verbreitung auch das Netzwerk verwenden (also sich nicht ausschließlich auf die Infektion von Dateien verlassen), versuchen permanent wohlbekannte Ports zum Verbindungsaufbau anzusprechen.

Die Darstellung hier ist keineswegs abschließend oder erschöpfend, sondern nur als Appetizer gedacht, um sich der Problematik bewußt zu werden und einmal ausführlich darüber nachzudenken.

In jedem Falle ist es ratsam, einschlägige Security-Informationen einzuholen und – wenn möglich – täglich zu beobachten. Neben den Webseiten der Virenhersteller ist die Security-Seite des Heise-Verlages¹, da dort aus der aktuellen Nachrichtenlage alle Security relevanten Nachrichten übersichtlich zusammengestellt sind. Eine wichtige, aber englische Webseite ist www.securityfocus.com, mit einem fast unerschöpflichen Fundus an aktuellen Nachrichten und sicherheitsrelevanten Artikeln.

1.4 FreeS/WAN, Openswan und strongSwan

Wenn in diesem Dokument von FreeS/WAN die Rede ist, bedeutet das nicht, daß die beiden anderen Entwicklungszweige nicht ebenfalls eingesetzt werden können. Sowohl Openswan 2.0 als auch strongSwan 2.0 basieren auf FreeS/WAN 2.04, sind daher weitestgehend mit FreeS/WAN kompatibel.

Der Autor selbst setzt inzwischen strongSwan ein, weil damit weniger Patches erforderlich sind und strongSwan vom Maintainer der wichtigsten Komponente, den X.509-Patches selbst gepflegt wird. Da X.509 bei Openswan zumindest im Moment unabhängig weiterentwickelt wird, ist nicht auszuschließen, daß sich auch die Konfiguration der Entwicklungszweige auseinander lebt. Was für den eigenen Zweck am ehesten einzusetzen ist, muß jeder für sich selbst entscheiden.

¹<http://www.heise.de/security/>

Kapitel 2

Konfiguration des Gateways

Dieses Kapitel bezieht sich auf die Konfiguration des Gateways für den Einsatz mit dem IPsec-Tool von Markus Müller. Für den Einsatz in Verbindung mit L2TP ist eine leicht abweichende Konfiguration erforderlich. Im Kapitel über den Einsatz von L2TP (Kapitel 4) wird nur auf die abweichende Konfiguration eingegangen, Sie sollten daher beim Einsatz von L2TP diese Kapitel ebenfalls lesen.

2.1 Zertifikate

Die Authentifikation für Windows-Clients basiert auf X.509-Zertifikaten. Solche Zertifikate werden von Windows 2000 und Windows XP nativ unterstützt, sofern man das richtige Import-Format wählt (PKCS 12, *.p12).

Die Generierung der Zertifikate erfolgt über OpenSSL, das bei jeder Linux-Distribution als Paket mitgeliefert wird. Im Anhang A wird die Erstellung von X.509-Zertifikaten und der Export im PKCS 12-Format kurz beschrieben. Eine ausführlichere Darstellung findet sich in „Das Firewall Buch“ von Wolfgang Barth, 2. Auflage, SuSE Press.

Jedes Zertifikat läßt sich über einen „Distinguished Name“ DN identifizieren. Der DN der einzelnen Zertifikate wird in der Konfiguration benötigt. Hier eine Übersicht der verwendeten Distinguished Names:

- *Certification Authority (CA)*

```
C=DE, O=Testfirma, CN=Testfirma CA
```

Zur Unterscheidung bekommt die CA einen Namen im CN, der nicht mit dem Client oder dem Gateway verwechselt werden kann. Der DN der ausstellenden CA wird bei der Konfiguration des Win-Clients benötigt (nicht der DN des Client-Zertifikates!).

❑ *VPN-Gateway*

```
C=DE, O=Testfirma, CN=gateway.testfirma.de
```

Das Gateway erhält im CN-Teil des DN einfach den Hostnamen mit der vollen Domainangabe.

❑ *Client ivpnc*

```
C=DE, O=Testfirma, OU=IVPNC, CN=user1@testfirma.de
```

Hier bekommt jeder User sein eigenes Zertifikat, um dieses gegebenenfalls später gezielt sperren zu können. CN besteht aus der Emailadresse des Users. Um die Nutzung – und damit die Zugriffsrechte später regeln zu können, erhält ein Internet-Nutzer den Zusatz OU=IVPNC im DN, um diesen von anderen wie WLAN-Nutzern unterscheiden zu können.

❑ *Client wlanc*

```
C=DE, O=Testfirma, OU=WLANC, CN=user2@testfirma.de
```

Wie bei einem Internet-Client, nur jetzt zur Unterscheidung mit OU=WLANC im Namen.

Auf dem Gateway selbst sind Zertifikate und der private Schlüssel des Gateways in den dafür vorgesehenen Verzeichnissen zu speichern. Der private Schlüssel sollte nur von root lesbar sein.

Was	Verzeichnis
Zertifikat der CA	/etc/ipsec.d/cacerts/
Zertifikat des Gateways	/etc/ipsec.d/certs/
Privater Schlüssel des Gateways	/etc/ipsec.d/private/

Das Zertifikat des Clients muß unter Windows im Format PKCS 12 importiert werden. Die Beschreibung folgt in Abschnitt 3.

2.2 FreeS/WAN-Konfiguration

Zunächst der besseren Übersicht wegen das vollständige Konfigurationsfile `/etc/ipsec.conf` für das VPN-Gateway. Die Erklärung erfolgt dann anschließend abschnittsweise.

```
# vpngw:/etc/ipsec.conf
# --> Verwendung von X.509-Zertifikaten

version 2.0

config setup
    interfaces="%defaultroute ipsecl=eth1"

conn %default
    left=%defaultroute
    leftid="C=DE, O=Testfirma, CN=gateway.testfirma.de"
    leftcert=gateway-cert.pem
    rightrsasigkey=%cert
    auto=add
    authby=rsasig
    rekey=no
    keyingtries=5

conn rw-vpnc
    right=%any
    rightid="C=DE, O=Testfirma, OU=IVPNC, CN="
    leftsubnet=192.168.0.129/32

conn rw-wlan
    left=192.168.9.1
    right=%any
    rightsubnetwithin=192.168.9.0/24
    rightid="C=DE, O=Testfirma, OU=WLANC, CN="
    leftsubnet=192.168.0.128/28

conn private
    auto=ignore
conn clear
    auto=ignore
conn clear-or-private
    auto=ignore
conn private-or-clear
    auto=ignore
conn packetdefault
    auto=ignore
conn block
```

```
auto=ignore
```

Das File beginnt mit allgemeinen Parametern wie die verwendete FreeS/WAN-Version. Die Versionsangabe ist für alle FreeS/WAN-Versionen $\geq 2.x$ erforderlich, sonst nimmt FreeS/WAN an, das es sich um ein Konfigurationsfile für die Version 1.x handelt:

```
version 2.0

config setup
    interfaces="%defaultroute ipsec1=eth1"
```

Im Abschnitt `config setup` werden die Interfaces gesetzt. Hier geht es um zwei Interfaces. Aus der Angabe `%defaultroute` ermittelt FreeS/WAN das Interface, auf den die Defaultroute gesetzt wurde (hier das DSL-Interface). Das virtuelle IPsec-Interface ist dabei implizit `ipsec0`. Für das zweite Interface ist explizit das Interface anzugeben.

```
conn %default
    left=%defaultroute
    leftid="C=DE, O=Testfirma, CN=gateway.testfirma.de"
    leftcert=gateway-cert.pem
    rightrsasigkey=%cert
    auto=add
    authby=rsasig
    rekey=no
    keyingtries=5
```

Hier werden allgemeine Voreinstellungen gesetzt, die für alle Verbindungen gelten. Findet sich der selbe Parameter in einem verbindungspezifischem Abschnitt wieder, hat dieser Vorrang. Der Vorteil einer solchen Default-Section: man muß Angaben, die für alle gelten, wie etwa das Zertifikat oder die Angabe, daß sich die andere Seite ebenfalls per Zertifikat authentifizieren soll (`rightrsasigkey=%cert`), nur ein einziges Mal angeben.

Mit `left=%defaultroute` bestimmt FreeS/WAN die eigene IP-Adresse, die im Falle der Internetverbindung dynamisch ist und erst bei Einwahl gesetzt wird. Leider verfügt FreeS/WAN noch nicht über die Möglichkeit, diese IP-Adresse im laufenden Betrieb zu wechseln. Mit jeder neuen IP-Adresse, also bei jeder neuen Einwahl in das Internet müssen die zu FreeS/WAN gehörenden Dienste über das Skript `/etc/init.d/ipsec` neu

gestartet werden. Das erledigt man am besten über die `ip-up`-Mechanismen des PPP (siehe unten).

`leftcert` ist dabei das eigene Zertifikat, `leftid` der DN zu diesem Zertifikat. Der Parameter `auto=add` besagt, daß das Gateway die Verbindung zwar registrieren, aber nicht automatisch starten soll. Da die Clients nicht permanent online sind und im Falle des Internets auch über eine dynamische IP-Adresse verfügen, nimmt nur der Client die Verbindung zum Gateway auf, das Gateway selbst verhält sich passiv und wartet auf eingehende Anfragen.

Der Parameter `rekey=no` sorgt nur dafür, das FreeS/WAN nicht von sich aus eine Erneuerung des Verbindungs-Schlüssel versucht. Das Gateway nimmt trotzdem anfragen vom Client entgegen. Wenn der Client offline geht oder die IP-Adresse wechselt, verhindert diese Einstellung, das FreeS/WAN immer und immer wieder versucht, mit dem Client neue Schlüssel auszutauschen.

```
conn rw-ivpnc
    right=%any
    rightid="C=DE, O=Testfirma, OU=IVPNC, CN=* "
    leftsubnet=192.168.0.129/32
```

Der Abschnitt `rw-ivpnc` (`rw` steht für Road Warrior) definiert die Verbindungsparameter für die Internet-Clients `ivpnc`. Die IP-Adresse des Clients ist beliebig: `right=%any`. Identifiziert wird der Internet-Client über den Abschnitt `OU=IVPNC` im DN. Der Namensteil im CN ist dabei beliebig, eine Wildcard-Funktion des X.509-Patches zu FreeS/WAN. `leftsubnet` schränkt hier die im lokalen Netzwerk erreichbaren IP-Adressen ein. Im Beispiel ist nur ein einziger Host mit der IP-Adresse `192.168.0.129` erreichbar, da die Subnetzangabe „32“ entspricht 32 gesetzten Bits in der Maske, also eine Subnetmask von `255.255.255.255`.

```
conn rw-wlanc
    left=192.168.9.1
    right=%any
    rightsubnetwithin=192.168.9.0/24
    rightid="C=DE, O=Testfirma, OU=WLANC, CN=* "
    leftsubnet=192.168.0.128/28
```

Abweichend von der bisherigen Konfiguration wird hier explizit die IP-Adresse des VPN-Gateways mit `left` angegeben, sie ist ja in diesem Falle

bekannt (und vom Interface der Default-Route verschieden). Die Kombination aus `right` und `rightsubnetwithin` sorgt dafür, daß die rechte IP-Adresse aus dem Bereich von 192.168.9.1 - 192.168.9.254 kommen muß (0 ist die Netzwerkadresse, 255 die Broadcast-Adresse).

Die Identifikation erfolgt wieder über einen Abschnitt im DN, hier `OU=WLANC`.

```
conn private
    auto=ignore
conn clear
    auto=ignore
conn clear-or-private
    auto=ignore
conn private-or-clear
    auto=ignore
conn packetdefault
    auto=ignore
conn block
    auto=ignore
```

Der Rest des Konfigurationsfiles dient nur dazu, implizite Methoden für „Opportunistic Encryption“ abzuschalten, da das hier nicht gewünscht ist. „Opportunistic Encryption“ soll dazu führen, daß sich Router gegenseitig als IPsec-Router erkennen und Daten automatisch verschlüsselt austauschen, auch wenn sie vorher noch nie etwas voneinander gehört haben. Da es hier nicht verwendet werden soll, werden die impliziten Verbindungen einfach abgeschaltet.

2.3 Setup bei Verbindungsauf-/abbau des PPPD

Eine dynamische IP-Adresse steht erst fest, wenn die Verbindung in das Internet steht. Durch eine bereits genannte Einschränkung von FreeS/WAN kann FreeS/WAN erst gestartet werden, nachdem die Internet-Verbindung aufgebaut wurde. Das geschieht am besten über das Verzeichnis `/etc/ppp/ip-up.d`. Alle ausführbaren (!) Skripte, die sich dort befinden, werden gestartet. Jedes Skript bekommt dabei einige Variablen übergeben, hier wird aber nur die Variable `PPP_IFACE` benötigt, die das Interface bestimmt. Nur, wenn das PPP-Interface hier `ppp0` ist, wird FreeS/WAN gestartet:

```
#!/bin/sh
#
# /etc/ppp/ip-up.d/000-pppd
```

```

#
case "$PPP_IFACE" in
    ppp0)
        /etc/init.d/ipsec start
        ;;
esac

```

Ein VPN-Gateway sollte immer erreichbar sein. Allerdings erfolgt bei den meisten DSL-Tarifen nach 24 Stunden eine Zwangstrennung. Man muß daher dafür sorgen, das unmittelbar nach der Zwangstrennung ein erneuter Verbindungsaufbau durchgeführt wird.

```

#!/bin/sh
#
# /etc/ppp/ip-down.d/99-pppd
#
case "$PPP_IFACE" in
    ppp0)
        /etc/init.d/ipsec stop
        ( sleep 10 && /bin/ping -c 4 -i 3 <ip> \
          > /dev/null ) &
        ;;
esac

```

Im Verzeichnis `/etc/ppp/ip-down.d/` wird dazu ein weiteres Skript platziert, das zunächst FreeS/WAN stoppt, um über einen Ping unmittelbar danach wieder wieder eine erneute Einwahl zu veranlassen. Als IP-Adresse sucht man sich am besten eine heraus, die immer erreichbar ist, etwa der Nameserver des eigenen Providers.

Ebenfalls beim Verbindungsaufbau ist die IP-Adresse bei einem DynDNS-Provider zu registrieren, damit der Client das Gateway auffinden kann. Die notwendigen Schritte sind in Abschnitt C zusammengefaßt.

Kapitel 3

Konfiguration des Windows-Clients (IPSEC.EXE)

Windows 2000 und Windows XP bringen bereits die Fähigkeit zu IPsec mit. Für Win2k ist mindestens Service Pack 2 erforderlich, besser ist es aber immer, den aktuellsten Servicepack einzuspielen oder ein Update über <http://www.winupdates.com> durchzuführen. Am besten informiert man sich regelmäßig über die Microsoft-Seiten oder einschlägige Nachrichtenkanäle wie

<http://www.heise.de/security/>

um neu entdeckte Sicherheitslücken rechtzeitig schließen zu können. Im Abschnitt 1.3 wurden bereits wichtige Sicherheitshinweise dazu gegeben.

Die Konfiguration ist eigentlich recht unspektakulär, sofern man beim Zertifikatimport (siehe Anhang B) die Anweisungen exakt befolgt. An dieser Stelle sei noch einmal die englische Dokumentation des Autors des IPSEC.EXE Tools, Marcus Müller, erwähnt:

<http://vpn.ebootis.de>

3.1 Installation der Hilfstools

Vor der eigentlichen Konfiguration benötigt man zwei Dinge: zum einen das Tool von Marcus Müller, daß man unter <http://vpn.ebootis.de> erhält. Das Tool ist gleichermaßen für Win2k und WinXP einsetzbar. Am besten kopiert man den Inhalt des Pakets `package.zip` in das Verzeichnis `C:\Programme\VPN` am Client. Den Inhalt des Paketes zeigt Abbildung 3.1.

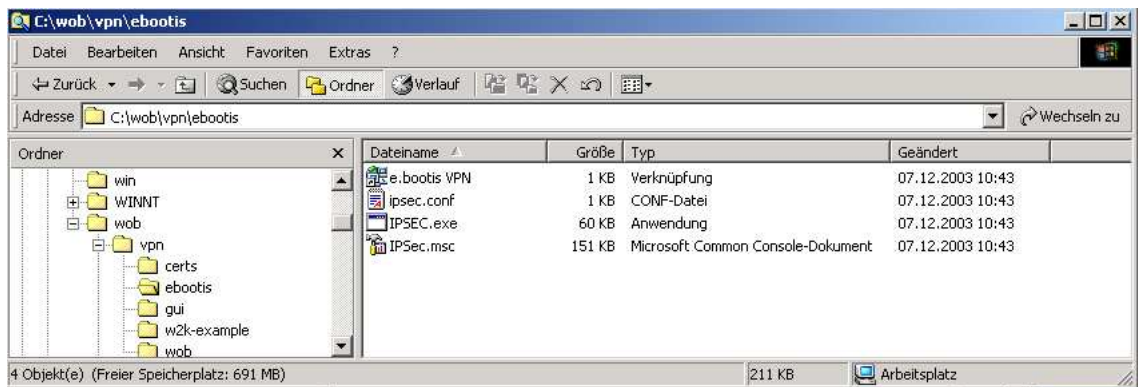


Abbildung 3.1: Inhalt von <http://vpn.ebootis.de/package.zip>

IPSEC.EXE

ist das eigentliche Werkzeug für die Einrichtung des VPN-Tunnels. Es wird **nach** jedem Verbindungsaufbau aufgerufen (sofern es sich um eine dynamische IP-Adresse, etwa bei einer Internet-Anbindung handelt).

ipsec.conf

das Konfigurationsfile für IPSEC.EXE

IPsec.msc

ein Konfigurationsfile für die Microsoft Management Console, die für den Import der Zertifikate benötigt wird. Marcus Müller hat hier freundlicherweise gleich eine Vorkonfiguration mitgeliefert, die ein umständliches Laden der benötigten Plugins erspart.

e.bootis VPN

Ein Link auf die ausführbare Datei IPSEC.EXE, den man sich auf den Desktop ziehen kann.

IPSEC.EXE ruft dann betriebssystemabhängig ein Programm von Microsoft auf:

Windows 2000

ipsecpol aus dem Windows 2000 Resource Kit. Man kann sich das Tool einzeln von Microsoft herunterladen, den entsprechenden Link findet man wieder auf <http://vpn.ebootis.de>.

Windows XP

ipseccmd aus den Windows XP Support Tools. Diese befinden sich auf der XP-CD im Verzeichnis \SUPPORT\TOOLS. Dort einfach setup aufrufen.

Wichtig: IPSEC.EXE muß das jeweilige Programm finden. Entweder installiert man das Programm in das gleiche Verzeichnis C:\Programme\VPN (empfehlenswert bei Win2k), oder man sorgt dafür, daß das Programm im Pfad gefunden wird (empfehlenswert bei XP, die Support Tools installiert man besser mit der Defaulteinstellung).

3.2 Import des Zertifikates

Der Import des Zertifikates unter Windows wird im Anhang B beschrieben.

3.3 Lokale Konfiguration: ipsec.conf

Die Konfigurationsdatei ipsec.conf muß sich im selben Verzeichnis wie das Programm IPSEC.EXE befinden. Der Aufbau lehnt sich deutlich an die FreeS/WAN-Konfiguration an, auch wenn nicht alle Parameter unterstützt werden:

```
conn %default
    dial=CallByCall

conn ivpnc
    left=vpngw.dyndns.org
    leftsubnet=192.168.0.0/255.255.255.0
    leftca="C=DE, O=Testfirma, CN=Testfirma CA"
    right=%any
    network=ras
    auto=start
    pfs=yes
```

Der Abschnitt conn %default soll dazu dienen (nicht getestet), daß beim Start von IPSEC.EXE eine Einwahlverbindung gleich mit gestartet wird. Dabei geht ein Fenster auf, das eine Einwahl erfolgen soll, das mit OK bestätigt werden muß. Der Name CallByCall ist durch den Namen der RAS-Verbindung zu ersetzen.

In der hier einzigen Verbindung `conn ivpnc` wird der DynDNS-Name des VPN-Gateways angesprochen. Der Client muß das Gateway ja irgendwie im Internet auffinden, und bei dynamischer IP-Adressenvergabe und regelmäßiger Zwangstrennung der üblichen DSL-Tarife ändert sich die IP-Adresse ständig. Die Konfiguration von DynDNS wird im Anhang C beschrieben.

`leftsubnet` ist hier das interne Netzwerk, die Angabe sollte identisch zu der Konfiguration am VPN-Gateway sein, in unserem Beispiel wäre das:

```
leftsubnet=192.168.0.129/32      # ivpnc
leftsubnet=192.168.0.128/28     # wlanc
```

Als `leftca` **muß** unbedingt der DN der ausstellenden CA verwendet werden, nicht der DN des User-Zertifikates!

Der Parameter `network=ras` deutet an, das es sich um eine Wählverbindung per RAS handelt. Andere Werte sind `lan` (etwa für das WLAN oder lokale Netzwerk) oder `auto` (`IPSEC.EXE` versucht automatisch zu bestimmen, welcher Verbindungstyp erforderlich ist).

Die anderen beiden Parameter bestimmen, ob die Verbindung automatisch bei Bedarf gestartet werden soll: `auto=start`, und das sogenannte „Perfect Secret Forwarding“: `pfs=yes`.

3.4 Verbindung starten

Für das Setup des IPsec-Tunnels benötigt `IPSEC.EXE` eine bestehende Verbindung, aus der die lokale IP-Adresse abgeleitet werden kann. Für RAS-Verbindungen heißt das, daß man zuerst die Einwahl in das Internet starten muß, bevor `IPSEC.EXE` aufgerufen wird. Am besten legt man sich zwei Icons auf den Desktop, die man der Reihe nach aufruft.

Beim Starten von `IPSEC.EXE` dauert es ca. 10 sec., bis sich das Programm wieder beendet. Diesen Timeout kann man umgehen, wenn man das Programm mit der Option `-nosleep` aufruft. Nach dem Start gibt `IPSEC.EXE` eine Reihe von Parametern aus:

```
C:\Programme\VPN>ipsec
IPSec Version 2.2.0 (c) 2001-2003 Marcus Mueller
Getting running Config ...
Microsoft's Windows 2000 identified
```

```

Setting up IPsec ...

Deactivating old policy...
Removing old policy...

Connection rrvpn:
MyTunnel      : 212.144.62.75
MyNet         : 212.144.62.75/255.255.255.255
PartnerTunnel: vpngw.dyndns.org
PartnerNet    : 192.168.0.129/255.255.255.255
CA (ID)       : C=DE, O=Testfirma, CN=Testfirma CA..
PFS           : y
Auto          : start
Auth.Mode     : MD5
Rekeying      : 3600S/50000K
Activating policy...

C:\Programme\VPN>

```

Mit IPSEC . EXE wird nur der Tunnel initialisiert, aber noch nicht tatsächlich gestartet. Um den Tunnel dann vollständig aufzubauen, müssen erst Pakete übertragen werden. Zum Test kann man einen Ping auf den erlaubten, internen Bereich durchführen. Im Beispiel:

```

C:\Programme\VPN>ping 192.168.0.129

Ping wird ausgeführt für 192.168.0.129 mit 32 Bytes:

Zeit<FC>berschreitung der Anforderung.
Zeit<FC>berschreitung der Anforderung.
Antwort von 192.168.0.129: Bytes=32 Zeit=111ms TTL=63
Antwort von 192.168.0.129: Bytes=32 Zeit=110ms TTL=63

Ping-Statistik für 192.168.0.129:
    Pakete: Gesendet=4, Empfangen=2, Verloren=2 (50%),
    Ca. Zeitangaben in Millisek.:
        Minimum=110ms, Maximum=111ms, Mittelwert=55ms

C:\Programme\VPN>

```

Manchmal erfolgt auch einige Male die Ausgabe von „IP-Sicherheit wird verhandelt“. Anschließend sollte aber der Ping normal antworten (sofern die Firewallregeln am VPN-Gateway eingehende Pings über den VPN-Tunnel erlauben).

Gleichzeitig kann man am Gateway über die Ausgaben im Syslog (bei den meisten Distributionen in der Standardkonfiguration über die Logdatei `/var/log/messages` oder `var/log/auth`) kontrollieren, ob der Verbindungsaufbau klappt.

Probleme macht manchmal eine mangelhafte DNS-Auflösung. Bei manchen Providern kommt es zwischendurch mal zu Problemen mit der Übertragung der Nameserver-IP-Adressen bei der Einwahl.

Zum Test wählt man sich ins Internet ein und versucht, einen bekannten Webserver anzupingen. Unabhängig davon, ob der Ping tatsächlich funktioniert, muß über die DNS-Namensauflösung eine IP-Adresse ermittelt werden, die der Ping anzeigt. Wenn der Ping auf `vpngw.dyndns.org` funktioniert, ist das VPN-Gateway eingewählt, hat sich registriert und lokal klappt alles mit der Namensauflösung.

Kapitel 4

IPsec/L2TP

Microsoft benutzt auf neueren Windowsbetriebssystemen für VPN-Verbindungen eine Kombination aus IPsec und L2TP (Layer 2 Tunneling Protocol). Die Betriebssysteme Win2k/XP liefern den passenden Client gleich mit, für alle älteren Windows-Betriebssysteme ist ein externer Client ebenfalls von Microsoft erhältlich.

Nutzt man den Builtin-Client mit IPsec/L2TP, fällt die Konfiguration auf Clientseite deutlich einfacher aus, dafür steigt der Aufwand am Gateway. Im Abschnitt 1.1 wurden Vor- und Nachteile dieser Lösung gegenüber einem nativen IPsec diskutiert.

Auf die Verwendung von Preshared Keys (PSK) sollte aus Sicherheitsgründen generell verzichtet werden. Die nachstehende Beschreibung verwendet X.509-Zertifikate auf der Clientseite. Am Gateway kommt der `l2tpd` von <http://www.l2tpd.org> zum Einsatz. Andere L2TP-Daemons sind ebenfalls verfügbar, darauf wird hier jedoch nicht eingegangen.

Die Verwendung von IPsec/L2TP im Zusammenhang mit den Microsoft-Clients wird von Jacco de Leeuw ausführlich beschrieben (englisch):

<http://www.jacco2.dds.nl/networking/>

Seine Dokumentation enthält unzählige Hinweise zur Fehlersuche, aber auch Querverweise zu anderen Dokumentationen.

4.1 L2TP-Protokollstack

Das „Layer 2 Tunneling Protocol“ L2TP kann man als weiterentwickelte Variante von PPP betrachten. Während PPP zum Ziel hat, unterschiedliche Protokolle zu transportieren, aber direkt im Layer 2 implementiert ist,

transportiert L2TP eben genau dieses PPP, aber aufgesetzt auf IP. Damit lassen sich unterschiedlichste Protokolle, die sonst direkt auf dem Layer 2 aufsetzen (etwa IPX, aber auch andere Protokolle), über Standard-IP-Strecken transportieren, die eigentlich ausschließlich uniprotokollfähig sind (eben nur IP beherrschen). Für den Transport von TCP/IP – wie es hier im VPN-Tunnel verwendet wird – ist das eigentlich Overkill, auf der anderen Seite bringt das verwendete PPP-Protokoll die Fähigkeit mit, dem Client Einstellungen mitzuteilen, wie Defaultroute, IP-Adressen, DNS-Server und WINS-Server. L2TP wird in RFC 2661, L2TP over IPsec wird in RFC 3193 beschrieben.

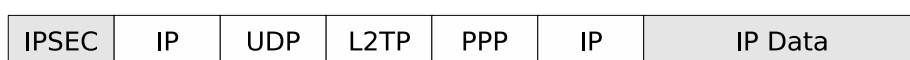


Abbildung 4.1: Protokoll-Overhead bei IPsec/L2TP

Wie IPsec in Kombination mit L2TP eigentlich funktioniert, zeigt Abbildung 4.1. Die Daten werden über IPsec transportiert, das hier nur schematisch dargestellt ist. Aus dem IPsec-Tunnel kommt zunächst einmal wieder IP. Bei nativem IPsec würde danach direkt die Nutzlast kommen, also der Datenteil des IP-Datagramms. Hier aber wird das L2TP über UDP transportiert (Destination-Port 1701, gemäß RFC 2661). L2TP wiederum transportiert die PPP-Frames. PPP wiederum verhandelt über eine zweite Authentifikation hinaus die Parameter für den Transport des IP-Protokolls und transportiert dieses nun endlich zusammen mit der Nutzlast (Datenteil des IP-Datagramms).

Die primäre, sichere Verschlüsselung und Authentifikation findet auf IPsec-Ebene statt. Darüber hinaus ist L2TP ebenfalls in der Lage, eine Authentifikation durchzuführen, darauf kann man aber getrost verzichten. Noch eine weitere Authentifikation findet auf PPP-Ebene statt: Während Zertifikate unter Windows nur an den Host gebunden werden können und sich auf der IPsec-Ebene nur die Workstation gegenüber dem Gateway authentifiziert, findet die Authentifikation auf PPP-Ebene userspezifisch statt. Hier gibt es in der unten vorgestellten Kombination mit dem `l2tpd` die Möglichkeit, einzelnen Usern eine feste IP-Adresse zuzuweisen.

4.2 Windows-Client: private Verbindung über VPN

Vorbereitungen

Generell empfehlenswert ist es insbesondere bei Win2k, alle notwendigen Updates von `www.windowsupdates.com` einzuspielen (um möglichst al-

le vorhandenen Sicherheitslücken zu schließen). Dort gibt es auch den NAT-T-Patch, der außer der NAT-Fähigkeit auch noch ein standard-konformeres Verhalten von L2TP einspielt. Die FreeS/WAN-Konfiguration hier geht von einem eingespielten NAT-Patch aus.

ACHTUNG: Wer bisher das IPsec-Tool von Markus Müller oder andere IPsec-Clients benutzt hat, muß unter Umständen einen Registry-Key löschen:

```
C:\ regedt32
HKEY_LOCAL_MACHINE
  \System
    \CurrentControlSet
      \Services
        \Rasman
          \Parameters
            prohibitipsec
```

Nach dem Löschen unbedingt den Rechner neu starten. Ist der Registry Key gesetzt, wird verhindert, daß die L2TP-Verbindung über IPsec aufgebaut wird. Die L2TP-Verbindung läuft dann ungeschützt und im Klartext über das Internet. Falls doch eine Verbindung zustande kommt, läßt das Gateway möglicherweise eine unverschlüsselte L2TP-Verbindung zu. Das ist ein riesiges Sicherheitsloch. Die Firewallregeln am Gateway müssen dafür sorgen, daß eine Verbindung zum lokalen L2TP-Daemon nur über das IPsec-Interface von FreeS/WAN erlaubt wird.

Für die Authentifikation der IPsec-Verbindung wird ein X.509-Zertifikat benötigt. Die Generierung wird in Anhang A beschrieben, die Installation unter Windows in Anhang B.

Welches Zertifikat für die Verbindung benötigt wird, ermittelt Windows aus dem Stammzertifikat des Gateways. Um bei den ersten Schritten nicht unnötig Stolperstellen einzubauen, sollte man zunächst unter Windows ausschließlich das Zertifikat installieren, das für die IPsec/L2TP-Verbindung benutzt werden soll.

Einrichtung der VPN-Verbindung

Vor dem Erstellen der eigentlichen VPN-Verbindung muß eine Internet-Verbindung vorhanden sein, etwa eine RAS-Verbindung. Falls noch keine Internet-Verbindung existiert, sollte man diese jetzt erstellen.

Die Einrichtung der VPN-Verbindung selbst wird mit „Neue Verbindung erstellen“ aus „Netzwerk und DFÜ-Verbindungen“ gestartet.



Abbildung 4.2: Start des Netzwerk-Verbindungsassistenten



Abbildung 4.3: Auswahl des Netzwerkverbindungstyps

Nach einem Begrüßungsbildschirm (Abbildung 4.2) erfolgt die Auswahl des Netzwerkverbindungstyps: „Verbindung mit einem privaten Netzwerk über das Internet herstellen“ (Abbildung 4.3). Danach kann die Verbindung mit dem Internet (meist eine RAS-Verbindung, muß vorher bereits konfiguriert sein) angegeben werden (Abbildung 4.4). Die dort angegebene Anfangsverbindung ist eine RAS-Verbindung zu einem Internetprovider via ISDN).



Abbildung 4.4: Auswahl der Anfangsverbindung (z.B. RAS)

Es folgt die Angabe des Gateways (entweder eine feste IP-Adresse, oder ein Full Qualified Domain Name, Abbildung 4.5). Die Verbindung soll für alle Benutzer der Workstation verwendbar sein (Abbildung 4.6). Im letzten Bild des Assistenten erhält die Verbindung noch eine eindeutige Bezeichnung (Abbildung 4.7).

Zum Schluß fragt der Assistent ab, ob direkt eine Einwahl erfolgen soll. Das sollte verneint werden, da noch einige Einstellungen zu tätigen sind.

Konfiguration der VPN-Verbindung

Die neue Verbindung findet sich jetzt im Ordner „Netzwerk- und DFÜ-Verbindungen“. Mit der rechten Maustaste kommt man an die „Eigenschaften“

Der Reiter „Allgemein“ zeigt die Einstellungen zum Gateway und zur gewählten Internet-Verbindung (Abbildung 4.8).

Im Reiter „Optionen“ ist insbesondere der Eintrag „Windows-Anmelde-domäne“ zu beachten. Wenn der Eintrag angekreuzt, muß die Angabe des



Abbildung 4.5: Angabe der Gateway-IP-Adresse oder des Full Qualified Domain Name



Abbildung 4.6: Diese Verbindung für alle Benutzer verwenden



Abbildung 4.7: Bezeichnung der neu angelegten Verbindung

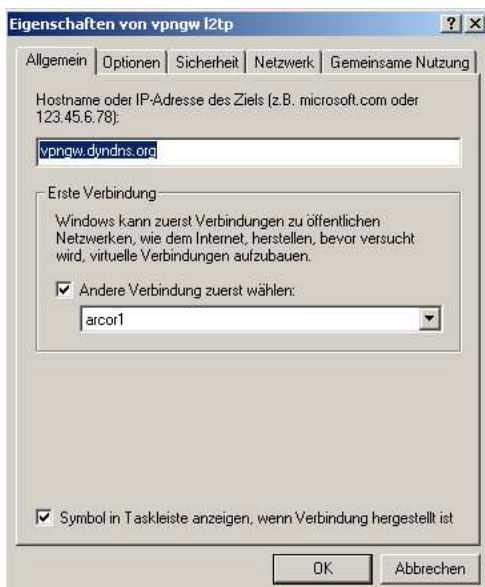


Abbildung 4.8: Eigenschaften/Allgemein

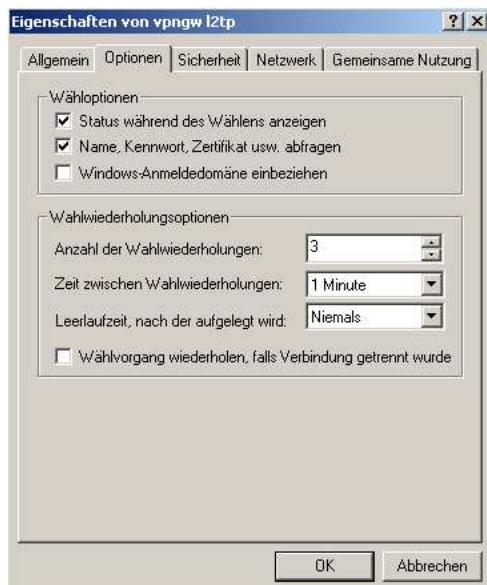


Abbildung 4.9: Eigenschaften/Optionen

Users beim Login später in der Form <Domain>\<User> erfolgen.

Der wichtigste Reiter ist „Sicherheit“ (Abbildung 4.10). Dort wählt man „Erweitert“ und öffnet das Unterfenster „Einstellungen“ (Abbildung 4.11). Bei Datenverschlüsselung wählt man optional, bei den Protokollen wählt man alle CHAP-Protokolle aus. Wichtig: welches CHAP-Protokoll verwendet werden kann, hängt von den Fähigkeiten des PPPD ab. Wer sicher gehen will, weil er nicht weiß, ob auch MS-CHAP und MS-CHAP v2 unterstützt wird, wählt immer zusätzlich das einfache CHAP aus.

Beim Reiter „Netzwerk“ muß schließlich noch L2TP als Typ des anzurufenen VPN-Servers angegeben werden (Abbildung 4.12). Ob man dort über TCP/IP hinaus auch noch den Microsoft-Client und die Datei-/Druckerfreigabe aktiviert, hängt vom Anwendungsfall ab.

4.3 Konfiguration des VPN-Gateways

Am Gateway werden folgende Dienste benötigt:

- ❑ Ein Kernel mit FreeS/WAN, Openswan oder strongSwan, der die X.509-Fähigkeit mitbringt. Die X.509-Patches von Strongsec enthalten Konfigurationsmöglichkeiten wie `leftprotoport`, die benötigt werden.

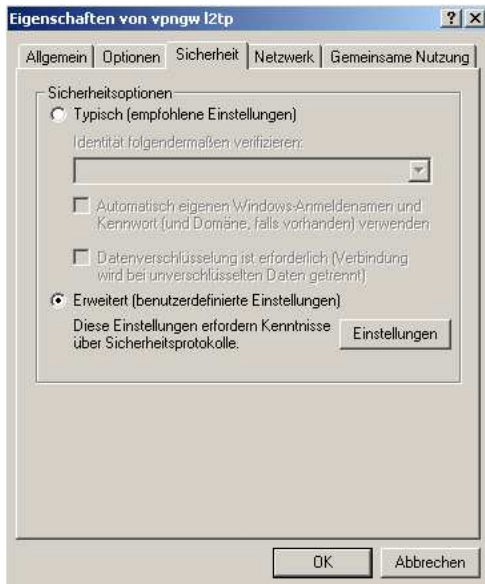


Abbildung 4.10: Eigenschaften/Sicherheit

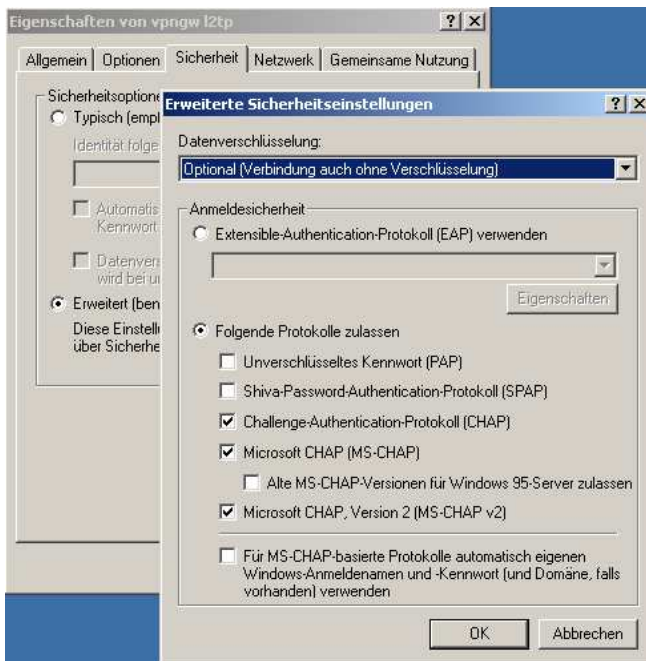


Abbildung 4.11: Eigenschaften/Sicherheit, Unterfenster Einstellungen

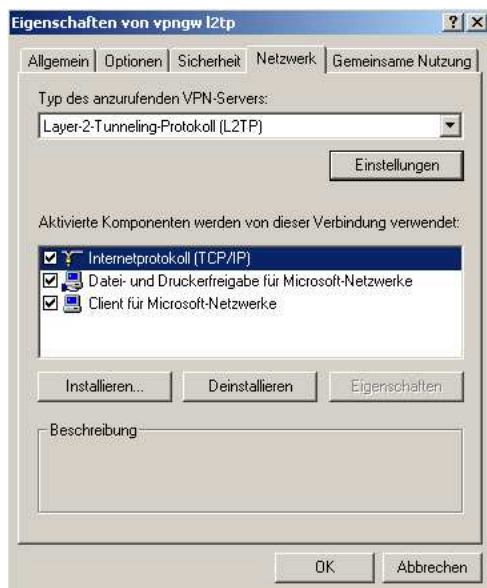


Abbildung 4.12: Eigenschaften/Netzwerk

- ❑ Ein `l2tpd`-Daemon. Die hier vorgestellte Konfiguration erfordert den `l2tpd` von www.l2tpd.org. Wichtig: bis zur Version 0.69 taugt der `l2tpd` nicht wirklich. Solange die Version 0.70 oder höher nicht verfügbar ist, verwendet man entweder ein bereits gepatchtes RPM-Paket von <http://www.jacco2.dds.nl/networking/>, oder man zieht sich die Sourcen von der Homepage aus dem SVN-Verzeichnis (Subversion, der große Bruder von CVS) und übersetzt sich den Daemon selbst (relativ problemlos).
- ❑ Ein aktueller PPP-Daemon (2.4.2 oder höher).

4.3.1 FreeS/WAN-Konfiguration

Die hier angegebene Konfiguration ist für `strongSwan` getestet, sollte aber auch mit `Openswan` und `FreeS/WAN` funktionieren:

```
# vpngw:/etc/ipsec.conf
#
version 2

config setup
    interfaces=%defaultroute

conn %default
```

```

left=%defaultroute
leftid="C=DE, O=Testfirma, CN=gateway.testfirma.de"
leftcert=gateway-cert.pem
auto=add
authby=rsasig
rekey=no
keyingtries=5
rightrsasigkey=%cert

conn vpn-l2tp
right=%any
rightid="C=DE, O=Testfirma, OU=IVPNC, CN=*"
pfs=no
leftprotoport=17/1701
rightprotoport=17/1701

```

Das einzig spannende an dieser Konfiguration sind folgende Angaben:

- ❑ `leftprotoport`: Destination-Port für L2TP
- ❑ `rightprotoport`: Source-Port für L2TP, bei Win2k/XP ohne den NAT-Patch muß hier 17/0 stehen.
- ❑ `pfs=no`: Microsoft verwendet standardmäßig keine „Perfect Forward Secrecy“, bei FreeS/WAN ist das aber die Voreinstellung, deshalb muß hier explizit PFS abgeschaltet werden.

Ebenfalls wichtig: Es handelt sich zwar um einen Tunnel (im Gegensatz zum Transportmodus von IPsec), aber trotzdem um eine Punkt-zu-Punkt-Verbindung. Es darf also weder auf Gateway- noch auf Clientseite ein Subnetz definiert sein, sonst klappt der Verbindungsaufbau nicht.

4.3.2 L2TP-Konfiguration

Die Konfiguration für den `l2tpd` besteht aus einem Konfigurationsfile, Optionen für den PPP und passenden Einträgen in `chap-secrets` für die Userauthentifikation.

`/etc/l2tp/l2tpd.conf`

```

; comment start with ';'
[global]
; listen-addr = 192.168.19.98

```

```
[lns default]
ip range = 192.168.20.128-192.168.20.254
local ip = 192.168.20.99
require chap = yes
refuse pap = yes
require authentication = yes
name = vpngw
ppp debug = yes
pppoptfile = /etc/ppp/options.l2tpd
length bit = yes
```

Kommentare werden hier generell mit `' ; '` gekennzeichnet. Nach dem hier nicht benutzten globalen Abschnitt folgt ein Abschnitt für den Default-Servereintrag.

Mit L2TP wird ein Subnetz definiert. Daraus erhält der Server die IP-Adresse über `local ip`, der `ip range` definiert die verfügbaren IP-Adressen für den Client, die im selben Subnetz wie `local ip` liegen müssen! Das Subnetz insgesamt darf durchaus ein anderes sein, als das Subnetz des lokalen Interfaces `eth0` am Gateway.

Die nächsten drei Zeilen sorgen dafür, daß eine PPP-Authentifikation stattfindet (`require authentication`), allerdings nur über CHAP, nicht über PAP. `name` ist der Name, den das Gateway in der PPP-Authentifikation verwendet, `ppp debug` sorgt für Debugging des PPP-Daemons zur Fehlersuche, `pppoptfile` gibt das Option-File an, das der PPPD für die L2TP-Verbindung verwenden soll.

Weitere Infos finden sich in der Manpage: `man l2tpd.conf`.

/etc/ppp/chap-secrets

```
# <client> <server> <secret> <ip-address>
wob      *          passwd  192.168.20.129
*        wob      passwd  192.168.20.129
```

Die Datei `chap-secrets` hat ein wohlbekanntes Format. Die bei dem User angegebene IP-Adresse wird dem Client beim Verbindungsaufbau zugewiesen. Dort läßt sich auch ein Subnetz angeben, etwa `192.168.20.128/25`. Wichtig ist aber, daß das Subnetz aus dem Bereich von `ip range` stammt und nicht mit anderen IP-Adressen (etwa `local ip`) kollidiert.

/etc/ppp/options.l2tpd

```
ipcp-accept-local
ipcp-accept-remote
ms-dns 192.168.11.17
# ms-wins 192.168.11.2
auth
crtscts
idle 1800
mtu 1400
mru 1400
nodefaultroute
debug
lock
+chap
# proxyarp
connect-delay 5000
```

Die hier angegebenen Optionen sind dem Vorschlag von Jacco de Leeuw entnommen und funktionieren beim Autor bestens. Sinnvollerweise weist man dem Client mit `ms-dns` wenigstens einen funktionierenden DNS-Server zu, damit der Client eine ordnungsgemäße Namensauflösung im internen Netzwerk durchführen kann. Die niedrige MTU ist erforderlich, da das ursprüngliche Datenpaket durch L2TP und PPP-Header vergrößert wird. Zusammen mit einem PPPoE-Header bei DSL darf das Paket am Ende 1500 Bytes nicht überschreiten, sonst gibt es fast immer Probleme.

4.3.3 Firewalling

Der L2TP-Daemon darf nur für die verschlüsselte Verbindung erreichbar sein, andernfalls hat man ein großes Sicherheitsloch. Wenn man von einer Default-Policy ausgeht, die zunächst erst einmal alles verbietet und nur gezielte Dienste freischaltet, kann man folgende Regel einsetzen, wenn man `iptables` mit Stateful Inspection verwendet:

```
% iptables -A INPUT -i ipsec0 -m state --state NEW \
-p UDP --sport 1701 --dport 1701 -j ACCEPT
```

Hier wird erwartet, daß der Rückweg über eine Stateful-Inspection-Regel freigeschaltet wird, zum Beispiel durch:

```
% iptables -A OUTPUT \
-m state --state ESTABLISHED,RELATED -j ACCEPT
```

4.4 Troubleshooting

- ❑ Zuerst muß eine funktionierende IPsec-Verbindung aufgebaut werden. Findet sich im Logfile keine Angaben über einen IPsec-Verbindungsaufbau, sollte man kontrollieren, ob das Gateway überhaupt kontaktiert wird:

```
% tcpdump -i ppp0
```

Ein IPsec-Verbindungsaufbau beginnt mit dem Schlüsselaustausch auf UDP Port 500. Taucht dort aber ein unverschlüsselter L2TP-Kontaktversuch auf, klappt die IPsec-Richtlinie am Client nicht (vorher Marcus Müllers Tool eingesetzt?) Den Registry-Key in Abschnitt 4.2 kontrollieren, falls gesetzt, unbedingt löschen. Falls der Key gelöscht werden muß, auf jeden Fall neu booten und die alte VPN-Verbindung löschen, also die Einrichtung noch einmal von vorne beginnen.

- ❑ Wird versucht, eine IPsec-Verbindung aufzubauen und klappt diese nicht, die Logfiles am Gateway kontrollieren. Dort steht meistens im Klartext, warum es nicht geht, etwa

```
no suitable connection for peer 'C=DE, O=My ...'
```

oder ähnliches (auch eine vorhandene Subnetz-Angabe auf Gateway-Seite führt zu einer Ablehnung der Verbindung mit ähnlicher Fehlermeldung).

- ❑ Die IPsec-Verbindung steht, aber die L2TP-Verbindung kommt nicht zustande? Wichtig sind die richtigen Ports, Win2k/XP ohne NAT-Update reagiert anders als Win2k/XP mit NAT-Update. Tip: immer das Update durchführen, dann hält sich L2TP eher an die Standards. Ebenfalls die Firewallregeln kontrollieren, ob dort nicht der Port 1701 gefiltert wird (sollte schon gefiltert werden, aber nicht aus dem IPsec-Tunnel).
- ❑ Wenn die L2TP-Verbindung zustande kommt, wird der PPPD gestartet. Im Logfile erhält man (bei eingeschaltetem Debugging) eine Reihe von Hinweisen, die auf mögliche Fehlerquellen schließen lassen. Liefert die Windows-Seite eine Fehlernummer, kann man in den Howtos von Jacco de Leeuw nachsehen (www.jacco2.dds.nl/networking/), ob es eine bekannte Ursache für das aufgetretene Problem gibt.

Anhang A

Generierung von X.509-Zertifikaten

Voraussetzung für die Erstellung von X.509-Zertifkation ist eine Installation von OpenSSL. OpenSSL wird als Paket bei allen Linux-Distributionen mitgeliefert, so daß dessen Installation keine Probleme machen sollte.

Normalerweise befindet sich die dazugehörigen Dateien in `/etc/ssl`. Der besseren Übersicht halber sollte man dort Unterverzeichnisse anlegen, um die erstellten Schlüssel, Requests und Zertifikate nicht alle in das Verzeichnis `/etc/ssl` zu kopieren. Wenn man bei der Erstellung der einzelnen Dateitypen der nachfolgenden Konvention folgt, sollte es nicht weiter schwer sein, die unterschiedlichen Arten – Schlüssel, Request und Zertifikat – auseinanderhalten zu können.

Datei	Beschreibung
<code>private/*-key.pem</code>	privater Schlüssel
<code>reqs/*-req.pem</code>	Zertifikats-Anforderung
<code>certs/*-cert.pem</code>	Signiertes Zertifikat
<code>p12/*-cert.pem</code>	Zertifikat im PKCS 12-Format

Für die Vergabe der Namen (oben mit * markiert) kann man folgende Konvention verwenden: CA oder besser `<kürzel>CA`, wenn man gegebenenfalls mit mehreren CAs später arbeiten muß (Beispiel: `meierCA-cert.pem`).

Für das Gateway sollte man einen sprechenden Namen benutzen wie `<kürzel>gw` (Beispiel: `meiergw-cert.pem`).

User-Zertifikate kennzeichnet man am besten mit einem eindeutigen Loginnamen (Beispiel: `richie-cert.pem`).

A.1 Zertifikation Authority

Das sogenannte Root-Zertifikat, das ist das Zertifikat der Certification Authority CA, ist nichts anderes als ein selbst unterschriebenes Zertifikat. Mit OpenSSL läßt sich das in einem einzigen Durchgang erstellen (ohne zwischendurch einen gesonderten Request aus dem privaten Schlüssel heraus zu erzeugen):

```
openssl req -x509 -days 1500 -newkey rsa:2048 \  
-keyout private/testfirmaCA-key.pem \  
-out testfirmaCA-cert.pem
```

Nach der Beantwortung einiger Fragen – hier muß man aufpassen, daß man nur die Country-Frage (C) mit DE, die Organization (O) mit Testfirma und den Common Name (CN) mit Testfirma CA beantwortet, die anderen Fragen leer läßt. Die Passphrase ist sozusagen das Passwort der Root-CA, man sollte dieses Passwort sicher aufbewahren, da man ohne dieses keine Zertifikate mehr erstellen kann. Sollte das Passwort bekannt werden, könnte jeder mit dem Schlüssel der Root-CA Zertifikate erstellen, in dem Fall kann man also gleich alle Schlüssel und Zertifikate wegwerfen und komplett von vorne beginnen.

Der Aufruf erzeugt einen privaten Schlüssel mittels RSA in der Länge 2048, und ein selbst unterschriebenes X.509-Zertifikat. Das Zertifikat ist 1500 Tage (etwas mehr als 4 Jahre gültig).

Den Inhalt des Zertifikates kann man sich mit folgendem Aufruf ansehen (geht auch bei anderen X.509-Zertifikaten, die kein Root-Zertifikat darstellen):

```
openssl x509 -in testfirmaCA-cert.pem -noout -text
```

Irgendwann läuft das Root-Zertifikat ab, dann muß ein neues erstellt und das alte ersetzt werden. Da Clientzertifikate ebenfalls begrenzt sind, ist es nicht verkehrt, die Lebensdauer eines Clientzertifikates kürzer zu halten und beim Austausch der Clientzertifikate gleich ein neues Root-Zertifikat anzufertigen. Insbesondere beim Importformat für Windows (*.p12) werden ja Root- und Clientzertifikat gleichzeitig importiert, der Client erhält so bei der Erneuerung des Clientzertifikates gleich wieder ein frisches Root-Zertifikat, das wiederum länger gilt als das neue Client-Zertifikat.

Der Aufruf für ein neues Root-Zertifikat ist fast identisch mit dem oben angegebenen, nur daß jetzt eben nicht ein neuer Schlüssel erzeugt wird, sondern der bereits vorhandene verwendet wird:

```
openssl req -new -x509 -days 1500 \  
-key private/testfirmaCA-key.pem \  
-out testfirmaCA-cert.pem
```

Wichtig: in jedem Falle bei der Erneuerung des Zertifikates exakt den gleichen DN verwenden!

A.2 Zertifikat erstellen

Für die Erstellung eines normalen Zertifikates gibt es keinen prinzipiellen Unterschied zwischen Gateway-Zertifikat und User-Zertifikat. Dort, wo im nachfolgenden <user> steht, ist für das Gateway-Zertifikat einfach der Name des Gateways anzugeben.

Zunächst wird der private Schlüssel, und daraus ein Request für die CA erzeugt. Der Request wird anschließend von der CA unterschrieben und dabei das Zertifikat erzeugt.

```
openssl req -newkey rsa:2048 \  
-keyout private/<user>-key.pem \  
-out reqs/<user>-req.pem
```

Hier sind wieder einige Fragen nach C, O und CN zu beantworten, der Schlüssel erhält wieder eine Passphrase (Passwort). Für Userzertifikate sollte man zusätzlich einen Wert bei Organizational Unit (OU) angeben: zum Beispiel IVPNC für den Internet-Client.

Anschließend muß das Zertifikat von der Root-CA unterschrieben werden:

```
openssl ca -days 730 -in reqs/<user>-req.pem \  
-out certs/<user>-cert.pem
```

Abgefragt wird die Passphrase der RootCA, am Ende steht das fertige Zertifikat im Verzeichnis ./certs.

Für eine Erneuerung von Zertifikaten zahlt es sich aus, die Requests (*-req.pem-Files) aufzuheben. Die Zertifizierung erfolgt nach der gleichen Prozedur, und man muß sich nicht um den DN des Users oder des Hosts kümmern, da dieser im Request enthalten ist. Also Requests aufheben!

A.3 Zertifikat nach Windows exportieren

Für die Verwendung am Windows-Client muß das Zertifikat im PKCS 12-Format exportiert werden. Neben dem Zertifikat des Users und dessen Private Key kann ein weiteres Zertifikat im eingepackt werden. Auf diese Art und Weise läßt sich in einem einzigen File gleichzeitig auch das Root-Zertifikat der ausstellenden CA mit übertragen. Das Root-CA benötigt der Client, um die Authentizität des Gateways überprüfen zu können.

```
openssl pkcs12 -export \  
  -in certs/<user>-cert.pem \  
  -inkey private/<user>-key.pem \  
  -certfile testfirmaCA-cert.pem \  
  -name "<user>" -caname "Testfirma CA" \  
  -out p12/<user>.p12
```

Für den Import des privaten Schlüssels wird die Passphrase des User-Schlüssels benötigt. Anschließend wird ein Export-Passwort abgefragt. Das Export-Passwort wird benötigt, wenn später das *.p12-File am Windows-Client wieder importiert werden soll.

Anhang B

Zertifikate unter Windows

Für den Import von Zertifikate müssen diese im PKCS 12-Format vorliegen. Der Import wird über die Management Konsole durchgeführt. Ein Doppelklick auf die Datei `IPsec.msc` öffnet die Konsole mit den notwendigen Plugins.

Mit einem rechten Mausklick auf „Eigene Zertifikate“ öffnet sich ein Menü, dort gelangt man über „Alle Tasks“ an den Menüpunkt „Importieren“ (siehe Abbildung B.1).

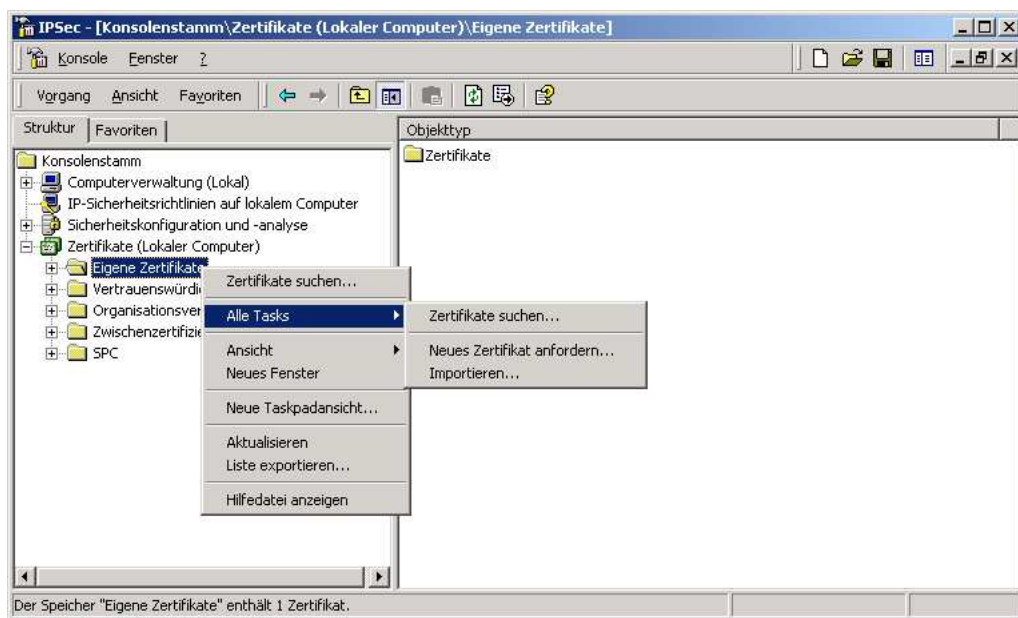


Abbildung B.1: Start des Importvorganges: rechter Mausklick auf „Eigene Zertifikate“

Es öffnet sich der Import-Assistent (Abbildung B.2), über „Weiter“ gelangt man an das Eingabefenster für den Dateinamen (* .p12-Datei, Abbildung B.3).



Abbildung B.2: Start des Import-Assistenten

Das * .p12-Format ist passwortgeschützt, für den Import ist daher die Angabe des Exportpasswortes aus Abschnitt A.3 erforderlich (Abbildung B.4).

Der nächste Schritt ist kritisch: unbedingt „Zertifikatsspeicher automatisch auswählen“ ankreuzen, ansonsten landen die Zertifikate nicht immer dort, wo sie hin sollen, der spätere Verbindungsaufbau wird von Windowsseite abgebrochen (weil das Zertifikat fehlt), am VPN-Gateway erscheint nur eine nichtssagende Fehlermeldung „incomplete ISAKMP SA“ und man sucht, und sucht, und sucht...

Jetzt kommen noch zwei Fenster, daß alles geklappt hat, das zweite ist der unter Windows übliche Glückwunsch zum Gelingen des Vorhabens ;-): „Der Importvorgang war erfolgreich“ (Abbildung B.6 und B.7).

Zur Kontrolle sollte man jetzt den Ordner „Eigene Zertifikate/Zertifikate“ öffnen und nachsehen, ob das gewünschte User-Zertifikat dort gelandet ist (Abbildung B.8). Meistens wird das frisch importierte Zertifikat erst nach „Aktualisieren“ angezeigt (rechte Maustaste verwenden), also nicht gleich verzweifeln.

Das Zertifikat der ausstellenden CA wird ebenfalls mit importiert. Das Zertifikat ist nötig, da ja auch der Client prüfen soll, ob das Zertifikat des

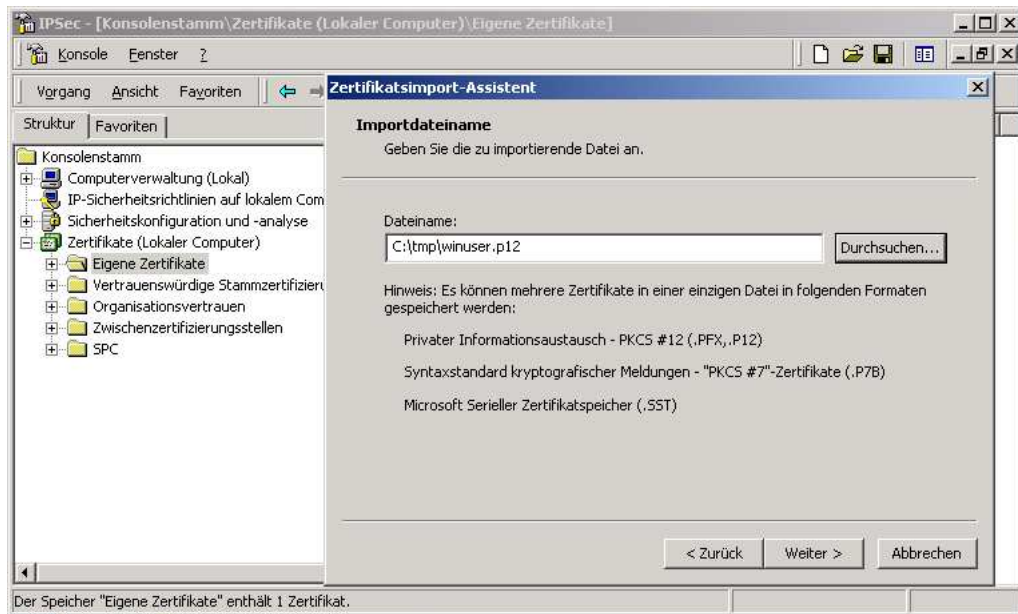


Abbildung B.3: Auswahl des Zertifikates (* .p12-Datei)

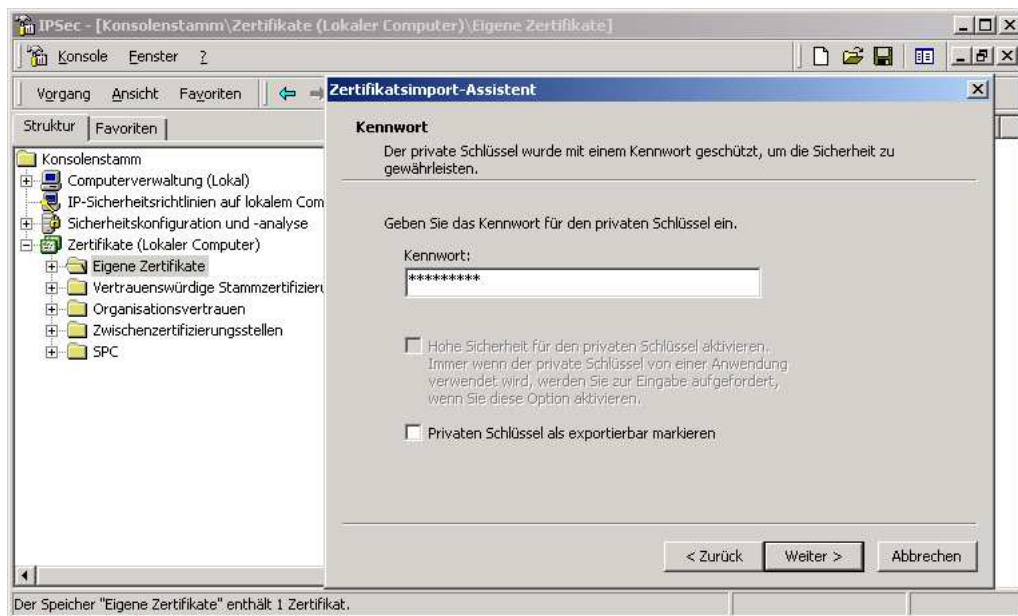


Abbildung B.4: Abfrage des Passwortes zur * .p12-Datei

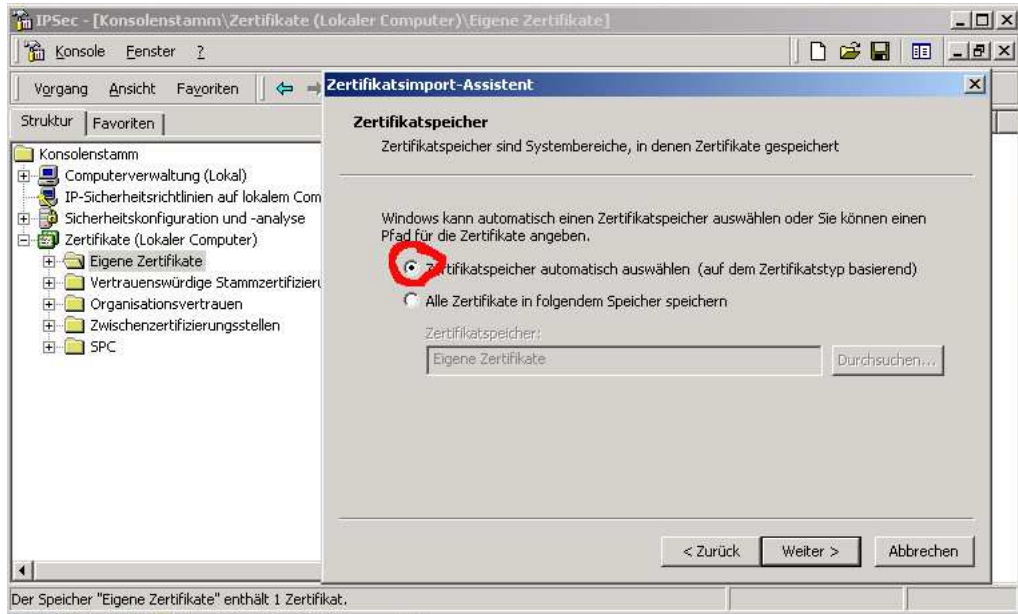


Abbildung B.5: Hier unbedingt „Zertifikatsspeicher automatisch auswählen“ ankreuzen!



Abbildung B.6: Fertigstellung des Zertifikates



Abbildung B.7: Herzlichen Glückwunsch, der Kandidat bekommt 99 Gummipunkte

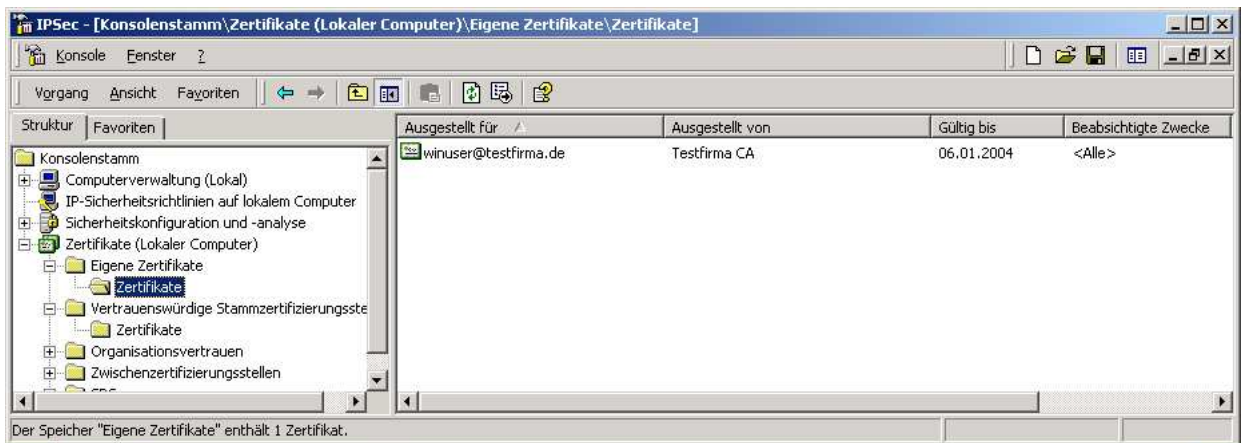


Abbildung B.8: Lokalisierung des User-Zertifikates

VPN-Gateways korrekt ist. Das CA-Zertifikat findet sich unter „Vertrauenswürdige Stammzertifizierungsstellen“ (Abbildung B.9). Hier ebenfalls den Ordner „Zertifikate“ erst mit der rechten Maustaste „Aktualisieren“ bearbeiten, und am besten die Zertifikate noch einmal explizit alphabetisch sortieren, sonst sucht man beim ersten Mal etwas länger ;-)

Wenn man schon beim CA-Zertifikat ist, notiert man sich auch gleich noch einmal zur Sicherheit den Distinguished Name DN der ausstellenden CA („Aussteller“). Dazu öffnet man das Zertifikat wie in Abbildung B.10 und klickt „Aussteller“ an, im unteren Fenster erscheinen dann die Einzelbestandteile des DN. Dieser DN ist wichtig für die Konfigurationsdatei `ipsec.conf` unter Windows (anders als bei Linux/FreeSWAN, dort wird der DN des User/Gateway-Zertifikates benötigt).

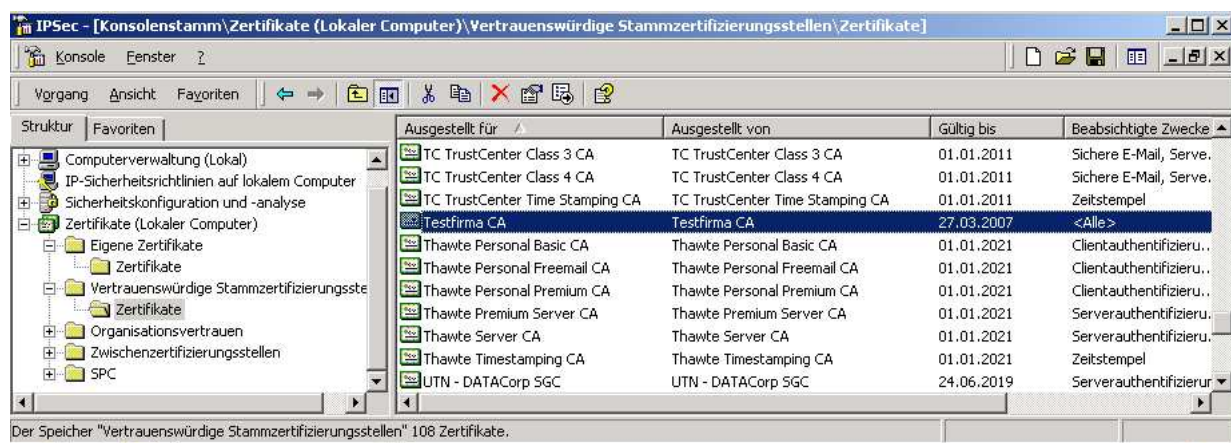


Abbildung B.9: Zertifikat der ausstellenden CA

Noch ein Wort zum Schluß: wer mit unterschiedlichen Zertifikaten herum experimentiert und dabei auch mehrmals die Root-CA neu aufsetzt und den DN der CA ändert, braucht sich nicht zu wundern, wenn Windows ein bisschen durcheinander kommt. Bei solchen Experimenten empfiehlt es sich, vor dem Import neuer Zertifikate die alten zu löschen, sonst kann es zu unerwünschten Nebenwirkungen und Risiken kommen (für die der Apotheker diesmal nichts kann ;-).

Wenn man mit mehreren Zertifikaten arbeitet, die den gleichen User-DN aufweisen, landet das später geladene Zertifikat selten dort, wo es hin soll. Über Details kann man sich die Gültigkeitsdauer (insbesondere ab wann) ansehen, das hilft, alte von neuen Zertifikaten zu unterscheiden.

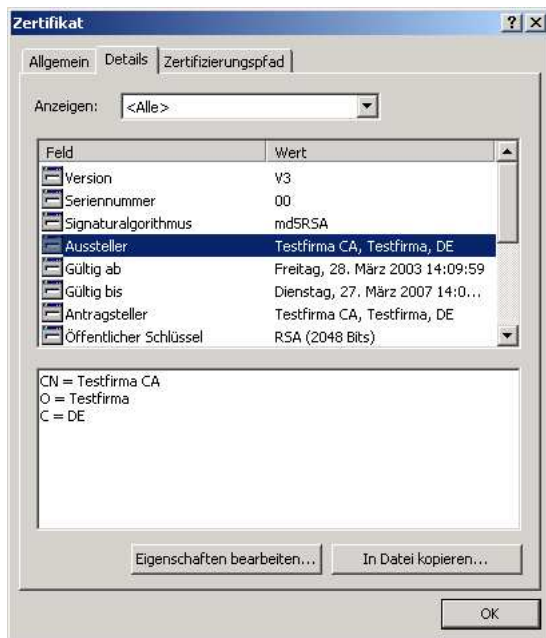


Abbildung B.10: Details zum Aussteller-DN notieren!

Anhang C

Dynamisches DNS für das VPN-Gateway

Einfachen DSL-Tarife haben zwei Nachteile: man erhält keine feste IP-Adresse, und nach 24 Stunden erfolgt eine Zwangstrennung. Mit anderen Worten: spätestens nach 24 Stunden erhält das VPN-Gateway eine neue IP-Adresse. Wie soll nun der Client das Gateway finden, wenn sich dauernd die IP-Adresse ändert?

Die Lösung heißt dynamisches DNS. Man kann sich bei verschiedenen Anbietern registrieren lassen. Mit einem Login und einem Passwort kann man bei jeder Einwahl über ein kleines Hilfsprogramm seine aktuelle IP-Adresse setzen und ist so jederzeit für den Client wieder auffindbar.

Im folgenden wird eine Konfiguration anhand des Programms `ddclient` und dem Provider `DynDNS.org` aufgezeigt. Dabei ist das keine Bedingung. Es gibt andere Clients (siehe www.dyndns.org, dort gibt es eine endlose Liste an verfügbaren Clients), und es gibt andere Provider als `DynDNS.org`.

Zunächst meldet man sich beim Provider an, läßt sich registrieren und erhält neben seinem verwendeten Login-Namen auch ein Passwort, das für den jeweils zu ändernden Namen benötigt wird.

Man kann sich meistens eine Domain herausuchen und läßt sich einen Namen für das Gateway reservieren, hier im Beispiel soll das `vpngw.dyndns.org` sein. Danach muß das Hilfsprogramm, das die Registrierung vornimmt, parametrisiert werden. Für das Programm `ddclient` geschieht dies in `/etc/ddclient.conf`:

```
#  
# /etc/ddclient.conf
```

```
#
pid=/var/run/ddclient.pid
protocol=dyndns2
use=if, if=ppp0
server=members.dyndns.org
login=mylogin
password=mypassword
vpngw.dyndns.org
```

mylogin ist durch den eigenen Login-Namen zu ersetzen, ebenso mypassword. In der letzten Zeile steht der volle Hostname, den man hat registrieren lassen.

Wichtig: für die Bestimmung des Registrierungsservers (members.dyndns.org im Beispiel) ist eine DNS-Auflösung erforderlich, man muß also DNS lokal am VPN-Gateway konfigurieren und die Firewall-Regeln so gestalten, das ein lokaler Verbindungsaufbau für DNS zum verwendeten Nameserver möglich ist. Die Registrierung erfolgt über TCP/Port 80, das VPN-Gateway muß daher auch eine lokale Verbindung zum Server members.dyndns.org aufbauen dürfen, sonst klappt die Registrierung nicht.

Für DNS trägt man in /etc/resolv.conf die IP-Adresse des Nameservers des eigenen Providers ein. Beispiel:

```
nameserver 194.25.2.132
nameserver 194.25.2.131
```

Die hier angegebenen Nameserver sind gut für T-Online-Nutzer, andere Provider verwenden andere Nameserver, bitte in der jeweiligen Dokumentation nachsehen und geeignete Nameserver des DSL-Providers eintragen.

Vorsicht beim Testen: die meisten DynDNS-Provider verhindern eine wiederholte Registrierung in zu kurzen Zeitabständen, um Denial-of-Service-Attacken zu verhindern. Also nicht einfach drauflos konfigurieren und testen, sondern immer die Rückmeldung beobachten und berücksichtigen.

Damit bei jedem Verbindungsaufbau die IP-Adresse registriert wird, liefern die meisten Distributionen direkt ein Skript mit. Hier ein Auszug aus dem Debian-Skript, das um alle Prüfungen gekürzt und auf das wesentliche reduziert wurde:

```
# /etc/ppp/ip-up.d/ddclient
# very short version
/usr/sbin/ddclient -syslog -ip $PPP_LOCAL
```

Die Variable `$PPP_LOCAL` wird vom `pppd` an die `ip-up`-Skripte übergeben. Die Angabe `-syslog` sorgt dafür, dass der Syslog eine Meldung erhält, ob die Registrierung erfolgreich war.

Anhang D

Nützliche Werkzeuge für die Windows-Clients

- ❑ *VPNDialer*

<http://sourceforge.net/projects/vpndialer/>

Nettes GUI für die automatische Einwahl, unterstützt RAS-Verbindungen, benötigt allerdings eine feste IP-Adresse des Gateways (leider in Kombination mit DynDNS derzeit nicht einsetzbar).

Lobenswert ist ein ziemlich ausführliches Handbuch (PDF).

- ❑ *iVPN*

<http://sourceforge.net/projects/ivpn/>

Ein GUI zu IPSECCMD.EXE (Windows XP). Kein Dialer, der die Einwahl unterstützt, sondern eben ein graphisches Frontend, um die Sicherheitsrichtlinien zu manipulieren.

Anhang E

Links

- ❑ *Das FreeS/WAN Projekt*
<http://www.freeswan.org/>
- ❑ *X.509-Erweiterung zu FreeS/WAN*
<http://www.strongsec.com/freeswan/>
- ❑ *strongSwan*
<http://www.strongswan.org/>
- ❑ *Openswan*
<http://www.openswan.org/>
- ❑ *Kernelquellen*
<http://www.kernel.org/>
- ❑ *Windows-Setup für IPsec von Marcus Müller*
<http://vpn.ebootis.de/>
- ❑ *Microsoft ipsecpol.exe für Win2K*
<http://agent.microsoft.com/windows2000/techinfo/reskit/tools/existing/ipsecpol-o.asp>