Open Source Library

Ralf Spenneberg

2. Auflage

VPN mit Linux

Grundlagen und Anwendung virtueller privater Netzwerke mit Open-Source-Tools

ADDISON-WESLEY

CD

3. Schutz durch ein VPN

Ein VPN bietet Authentifizierung, Vertraulichkeit und Schutz der Integrität der übertragenen Informationen. Diese Punkte werden wir in diesem Kapitel genauer betrachten.

3.1 Authentifizierung

Die Authentifizierung ist ein sehr wichtiger Bestandteil beim Aufbau eines virtuellen privaten Netzwerks. Eine erfolgreiche Authentifizierung ist die Voraussetzung für den Aufbau einer anschließenden verschlüsselten Verbindung. Wird die Authentifizierung übersprungen oder nicht korrekt durchgeführt, besteht die Gefahr eines sogenannten Man-in-the-Middle-Angriffes (s.u.). Für eine Authentifizierung können drei unterschiedliche Faktoren einzeln oder in Kombination genutzt werden:

- *Wissen*: zum Beispiel ein Kennwort. Die Authentifizierung kann erfolgen, da der Benutzer etwas weiß.
- *Besitz*: zum Beispiel eine Smartcard. Die Authentifizierung kann erfolgen, da der Benutzer eine Smartcard besitzt. Dieser Besitz zeichnet ihn als korrekten Benutzer aus.
- *Person*: zum Beispiel ein Fingerabdruck. Die Authentifizierung erfolgt biometrisch und testet die Person direkt. Die Identität der Person wird so eindeutig erkannt.

Häufig werden diese Verfahren in Kombination eingesetzt. So sind Smartcards meist zusätzlich mit einem Kennwort geschützt. Die biometrischen Verfahren haben leider keine Reife erlangt, die ihren Einsatz im Consumerbereich rechtfertigen würde.

Im Folgenden werde ich nun die Wichtigkeit der Authentifizierung eines Kommunikationspartners an zwei Beispielen verdeutlichen.

Stellen Sie sich vor, Sie möchten ein gebrauchtes Auto privat für 5000 Euro erwerben. Dann werden Sie sicherlich nicht mit dem Verkäufer lediglich die Schlüssel gegen den Geldbetrag tauschen. Sie werden zusätzlich eine Authentifizierung verlangen, dass das Fahrzeug auch tatsächlich dem Verkäufer gehört. Diese Authentifizierung erfolgt zum Beispiel, indem der Verkäufer Ihnen sowohl den Fahrzeugbrief als auch seinen Personalausweis vorzeigt. Erst dann können Sie sicher sein, dass er das Recht hat, das Fahrzeug zu verkaufen, denn er besitzt den Brief. Und Sie wissen, dass er tatsächlich derjenige ist, der er vorgibt zu sein, wenn Sie sein Gesicht mit dem Foto im Personalausweis vergleichen.

Stellen Sie sich nun vor, dass der Verkäufer Ihnen einen italienischen Fahrzeugbrief zeigt und selbst über einen spanischen Personalausweis verfügt. Sie werden sicherlich nicht leicht bereit sein, ihm das Fahrzeug abzukaufen, da Sie zunächst nicht in der Lage sind, die Validität

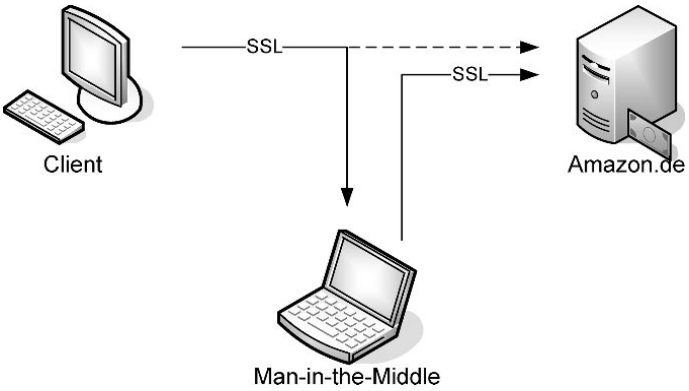

Abbildung 3.1: **Ein Man-in-the-Middle-Angriff**

seiner Dokumente zu überprüfen. Im Falle der deutschen Dokumente stellt das jedoch kein Problem dar, da das Layout eines Personalausweises und eines Fahrzeugbriefes grundsätzlich bekannt sind.

Ein ähnliches Problem tritt auf, wenn Sie fünf Tage vor Weihnachten feststellen, dass Ihnen noch ein Geschenk fehlt. Sie haben leider keine Zeit mehr, um lange durch Geschäfte zu streifen und nach einem Geschenk zu suchen. Sie erinnern sich, dass Online-Shops wie zum Beispiel Amazon.de den Versand bis Weihnachten noch garantieren, und suchen dort entsprechende Geschenke aus. Nachdem Sie sämtliche Geschenke in Ihrem virtuellen Einkaufskorb gesammelt haben, gehen Sie zur virtuellen Kasse. Hier stellt Amazon.de fest, dass Sie bisher noch nicht Kunde sind, und bittet Sie um die Eingabe Ihrer Konto- oder Kreditkarteninformationen.

Nun stehen Sie vor einem Problem. Zum einen möchten Sie Ihre Informationen verschlüsselt übertragen. Hierzu müssen Sie einen verschlüsselten Tunnel aufbauen. Dies ist, seit Diffie und Hellman den nach ihnen benannten Schlüsselaustausch erfunden haben (siehe Abschnitt 7.11), sehr einfach mit dem *Secure Socket Layer* (SSL) des *Hypertext Transport Protocol* (HTTP) möglich. Ihnen fehlt jedoch zuvor eine Authentifizierung von Amazon.de. Nur weil die Website überzeugend aussieht, bedeutet das ja noch lange nicht, dass Sie tatsächlich auf der Website von Amazon gelandet sind. Es könnte ja sein, dass ein Angreifer unsere Anfrage an Amazon.de abgefangen und auf seinen Rechner umgeleitet hat (siehe unten den Exkurs zum DNS-Spoofing). Bei dem Autokauf konnten Sie sich den Personalausweis des Verkäufers zeigen lassen. Ganz so einfach ist das in diesem Fall nicht. Ein Man-in-the-Middle-Angriff ist möglich (siehe Abbildung 3.1).

EXKURS

DNS-Spoofing. *Computer im Internet kommunizieren über ihre IP-Adresse miteinander. Dies ist eine Nummer, die aus vier Bytes besteht. Zur einfachen Darstellung werden die Bytes üblicherweise durch Punkte voneinander getrennt, zum Beispiel 10.5.171.253. Da es sehr schwer ist, sich diese IP-Adressen zu merken, erhalten Computer meistens zusätzlich einen Namen. Für die Auflösung des Namens in die entsprechende IP-Adresse und umgekehrt haben sich im Laufe der Zeit verschiedene Systeme etabliert, von denen das* Domain Name System *(DNS) das heute am*

meisten verwendete System ist. Dieses System ist verantwortlich dafür, dass ein Rechner einen DNS-Namen in die entsprechende IP-Adresse auflösen und für eine IP-Adresse auch den entsprechenden Namen ermitteln kann. Wenn Sie in Ihrem Browser die Adresse `http://www.os-t.de` *eingeben, so wird dieser Browser zunächst eine DNS-Anfrage stellen, um die IP-Adresse des entsprechenden Rechners in Erfahrung zu bringen. Erhält er hierbei eine falsche IP-Adresse, so spricht man von DNS-Spoofing. So besteht zum Beispiel die Möglichkeit, dass ein Angreifer einen Benutzer auf eine andere Website umlenkt und ihm falsche Informationen unterschiebt.*

Es existieren grundsätzlich zwei Methoden, mit denen das DNS-Spoofing erfolgen kann:

1. *DNS-Server kennen nicht alle DNS-Namen des Internets. Daher müssen Sie häufig bei anderen DNS-Servern nachfragen, um die Auflösung eines DNS-Namens in eine IP-Adresse zu gewährleisten. Um nicht für denselben DNS-Namen nach kurzer Zeit eine neue Anfrage zu starten, cachen die DNS-Server diese von anderen DNS-Servern gelieferten Ergebnisse. Die Dauer der Zwischenspeicherung bestimmt der liefernde DNS-Server.*

   Gelingt es dem Angreifer, hierbei falsche Informationen zu senden, die dann im Cache gespeichert werden, spricht man vom DNS Cache Poisoning*. So erlaubt es das DNS-Protokoll dem antwortenden DNS-Server, zusätzliche Informationen, die nicht ursprünglich angefragt wurden, mitzuliefern. Diese werden von dem fragenden DNS-Server dann häufig auch gecacht (siehe Abbildung 3.2). Moderne DNS-Server bieten üblicherweise Funktionen, um dies zu unterbinden.*

   Solch ein Angriff wurde in dem New Yorker Wahlkampf 1999 auf die Website von Hillary Clinton angewendet, um Zugriffe auf Ihre Website `http://www.hillary2000.org` *auf die Website* `http://www.hillaryno.org` *umzulenken.*

   Dan Kaminsky hat im Sommer 2008 auf der Blackhat-Konferenz eine neue Methode vorgestellt, die ein DNS-Cache-Poisoning bei fast jedem DNS-Server erlaubte.

2. *Bei der zweiten Variante werden direkt die Anfragen des Browsers an den DNS-Server oder des DNS-Servers an weitere DNS-Server aufgefangen und durch ein Programm des Angreifers direkt beantwortet. Da dieses Programm wahrscheinlich wesentlich schneller die Anfrage beantworten kann als ein DNS-Server, der zunächst in seiner Datenbank suchen muss, wird diese Antwort als korrekte Antwort akzeptiert. So kann ein Angreifer also warten, bis er eine entsprechende Anfrage im Netz erkennt, und dann sein Opfer gezielt auf die falsche IP-Adresse lenken.*

Um dies im Zusammenhang mit einer SSL-verschlüsselten Verbindung ausnutzen zu können, wird noch eine Anwendung benötigt, die auf dem Rechner des Angreifers läuft, den verschlüsselten Tunnel aufbaut und dem Opfer den Eindruck vermittelt, dies sei der korrekte Rechner. Dug Song hat derartige Werkzeuge bereits 1998 öffentlich vorgestellt. Hierbei handelt es sich um die Werkzeuge `webmitm` *und* `dnsspoof` *seines Programmpaketes* `dsniff`[1]*.*

Public-Key-Kryptografie bietet hier Hilfe. Eine genauere Betrachtung dieser Methode erfolgt in den späteren Kapiteln. Bei der Public-Key-Kryptografie erzeugt ein Benutzer für sich zwei

1 `http://www.monkey.org/~dugsong/dsniff/`

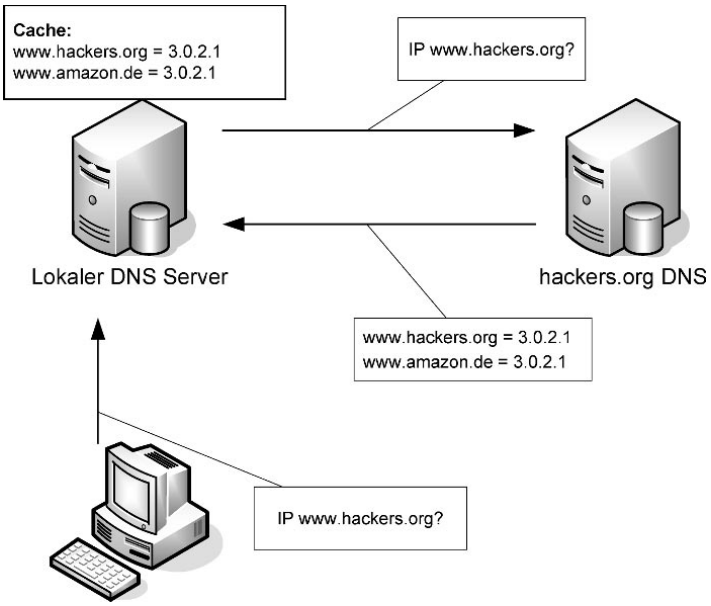

Abbildung 3.2: **DNS Cache Poisoning**

Schlüssel. Der eine Schlüssel wird als privater Schlüssel bezeichnet und ist nur dem Benutzer bekannt und kann daher als Identitätsnachweis verwendet werden. Jeder, der über diesen Schlüssel verfügt, kann sich als der entsprechende Benutzer authentifizieren. Häufig wird dieser Schlüssel zum Schutz noch mit einem Kennwort verschlüsselt und auf einer Smartcard gespeichert. Der zweite Schlüssel wird als öffentlicher Schlüssel (Public Key) bezeichnet. Dieser Schlüssel kann frei abgegeben werden.

Die Besonderheit der Public-Key-Kryptografie besteht nun in der Beziehung der beiden Schlüssel. Eine Nachricht, die mit dem privaten Schlüssel verschlüsselt wurde, kann nur mit dem entsprechenden öffentlichen Schlüssel entschlüsselt werden. Dies gilt dementsprechend auch in die andere Richtung. Eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht kann nur mit dem privaten Schlüssel entschlüsselt werden (siehe Abbildung 3.3).

Dieses Verfahren kann nun zur Authentifizierung eines Webshops genutzt werden. Für den Webserver von wird ein derartiges Schlüsselpaar erzeugt. Der öffentliche Schlüssel wird zum Kunden übertragen. Anschließend kann der Kunde, bevor er vertrauliche Daten an den Webshop überträgt, die Authentifizierung verlangen. Hierzu kann er eine große zufällige Zahl an den Webshop übermitteln und diesen auffordern, diese Zahl mit seinem privaten Schlüssel zu verschlüsseln. Der Webshop sendet diese verschlüsselte Herausforderung (Challenge) an den Kunden zurück, der sie mit dem öffentlichen Schlüssel entschlüsseln und mit der Original-Zahl vergleichen kann.

Kommen wir zurück zum Problem: Sie wollen wenige Tage vor Weihnachten noch die Geschenke einkaufen. Wie erhalten Sie den öffentlichen Schlüssel des Webshops? Ganz einfach. Der Webshop sendet Ihnen diesen Schlüssel über das Internet. Dies erfolgt noch nicht ver-

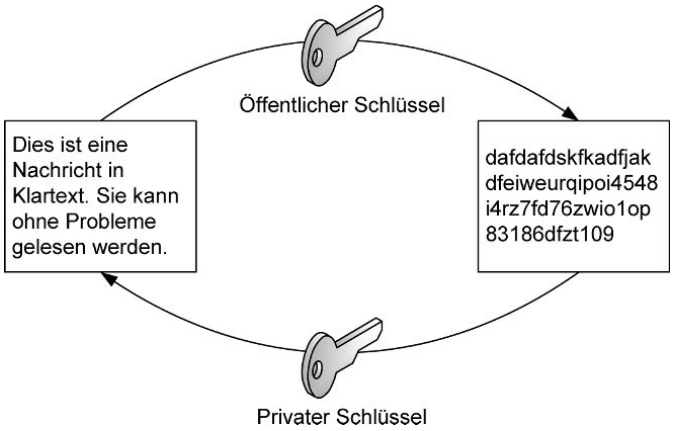

Abbildung 3.3: **Ver- und Entschlüsselung mit dem Public-Key-Verfahren (vereinfacht)**

schlüsselt. Da es sich um den öffentlichen Schlüssel handelt, ist das auch nicht erforderlich. Woher wissen Sie nun, dass der Schlüssel tatsächlich von dem Webshop ist und nicht von einem Man-in-the-Middle gesendet wurde? Dies stellt nun das zentrale Problem dar.

Im Falle des Fahrzeugkaufs war es einfach, wenn der Verkäufer einen deutschen Personalausweis besaß. Dieser Personalausweis zertifizierte ihn als deutschen Staatsbürger und konnte sehr einfach überprüft werden, da das Layout und die Stempel allgemein bekannt sind. Im Grunde vertraut der Käufer der Stelle, die den Personalausweis ausgegeben hat, dahingehend, dass sie die Identität der Person geprüft hat. Es wird also hier eine dritte Zertifizierungstelle (Certificate Authority, CA) genutzt.

Das Verfahren wurde auf das Internet übertragen. Hierzu hat der Webshop seinen öffentlichen Schlüssel mit einer „Kopie des Personalausweises" an eine Zertifzierungsstelle gesendet. Diese bestätigt die Echtheit des Schlüssels, indem sie ihn mit ihrem eigenen privaten Schlüssel signiert. Diese Signatur kann nun mit dem öffentlichen Schlüssel der Zertifizierungsstelle (CA) validiert werden. Ist die Signatur echt, dann ist auch der öffentliche Schlüssel des Webshops echt, und der Browser des Kunden kann den Challenge an den Webshop senden. Wenn der Webshop den Challenge richtig verschlüsselt, handelt es sich tatsächlich um den richtigen Webserver.

Wie erhält nun der Kunde den öffentlichen Schlüssel der Zertifizierungsstelle, um deren Signatur zu prüfen? Hier besteht ja dasselbe Problem wie zuvor mit dem öffentlichen Schlüssel des Webshops. Der Trick liegt in der Tatsache, dass die verwendeten Browser bereits sämtliche öffentlichen Schlüssel der anerkannten Zertifizierungsstellen enthalten. Die Browserhersteller haben diese bereits in ihren Browsern hinterlegt[2]. So können die Browser Zertifikate, die von diesen CAs unterzeichnet wurden, validieren.

Dieses Verfahren der Authentifizierung von Kommunikationspartnern mit Zertifikaten wird in späteren Kapiteln noch genauer erläutert. Im Grunde arbeiten die meisten guten Authentifi-

2 Eine interessante Frage ist: Woher wissen Sie, dass hierbei keine Fehler unterlaufen sind?

zierungssysteme auf diese oder ähnliche Weise. Jedoch sollte der Stellenwert der Authentifizierung deutlich geworden sein. Ohne eine vorherige Authentifizierung der Kommunikationspartner kann keine Datensicherheit garantiert werden. Die Authentifizierung garantiert den Ursprung der Daten und stellt damit sicher, dass die Daten von dem gewünschten Kommunikationspartner stammen.

3.2 Vertraulichkeit

Die Garantie der Vertraulichkeit der übertragenen Daten ist ein weiterer wichtiger Aspekt eines virtuellen privaten Netzwerks. Diese Vertraulichkeit kann technisch durch einen Provider in Form eines ATM-Netzwerks gewährleistet oder durch eine sichere Verschlüsselung der Daten während des Transports garantiert werden. Die Realisierung durch einen Provider in Form eines ATM-Netzwerks ist nicht Thema dieses Buches und soll daher hier vernachlässigt werden. Wenn heute von einem Software-VPN-Produkt gesprochen wird, so garantiert dieses die Vertraulichkeit durch eine Verschlüsselung (meist mit IPsec) der übertragenen Informationen. Bei den heute eingesetzten Verschlüsselungsverfahren werden symmetrische und asymmetrische Verfahren unterschieden. In beiden Fällen sind die mathematischen Verfahren bekannt und werden dauernd auf Herz und Nieren geprüft.

Bei den *symmetrischen Verfahren* wird für die Ver- und Entschlüsselung der identische Schlüssel eingesetzt. Bei den *asymmetrischen Verfahren* handelt es sich um Public-Key-Algorithmen, die mit zwei Schlüsseln arbeiten. Dabei wird die Nachricht mit einem Schlüssel verschlüsselt und kann nur mit dem entsprechenden Pendant entschlüsselt werden (vereinfacht, siehe Kapitel 7).

Die heute im Einsatz befindlichen symmetrischen Verfahren wie DES, 3DES, AES, Blowfish, Twofish, CAST und RC5 weisen bei richtiger Anwendung keine wesentlichen Sicherheitslücken auf, die es ermöglichen würden, aus einem verschlüsselten Text auf den Klartext oder den verwendeten Schlüssel zu schließen. Leider hat der Anwender keinen Einfluß auf die Umsetzung durch den Programmierer. Daher sollten Sie Open-Source-Programmen den Vorzug geben, bei denen Sie oder auch die Gemeinschaft den Code kontrollieren können. Ein Angriff ist lediglich durch einen sogenannten Brute-Force-Angriff möglich. Hierbei muss der Angreifer sämtliche möglichen Schlüssel ausprobieren. Dies dauert in Abhängigkeit vom verwendeten Algorithmus, der verwendeten Schlüssellänge und der zur Verfügung stehenden Hardware unterschiedlich lange. So errechnete das Projekt distributed.net (`http://www.distributed.net`), dass sie bei einer dauerhaften Rechenleistung von 45,998 2-GHz-AMD-Athlon-XP-Rechnern 790 Tage benötigt hätten, um sämtliche möglichen RC5-64-Schlüssel auf einen verschlüsselten Text anzuwenden. Diese für das Knacken aufzuwendende Zeit lässt sich sehr einfach durch einen längeren Schlüssel exponentiell verlängern. So erfordert ein 1 Bit längerer Schlüssel den doppelten und ein 2 Bit längerer Schlüssel bereits den vierfachen Aufwand. Heutzutage übliche Längen eines symmetrischen Schlüssels sind 40, 56, 64, 128, 168 und 256 Bit. Schlüssellängen kleiner als 128 Bit werden jedoch als nicht sicher eingestuft. Inzwischen existieren jedoch spezialisierte Hardwaresysteme, die diese Analyse schneller verrichten können. So haben die Universitäten Bochum und Kiel im März 2007 mit

der Hardware COPACOBANA[3] gezeigt, dass mit marktüblichen FPGA-Prozessoren ein DES-Schlüssel in 6,4 Tagen im Brute-Force-Verfahren ermittelt werden kann. Die Kosten für die Hardware beliefen sich damals auf 10.000 Dollar. Aktuell werden vor allem aktuelle Grafikkartenprozessoren und Spielekonsolen für diese Anwendungen missbraucht. Um ausreichend Rechenleistung zur Verfügung zu haben, werden die Systeme in einem Cluster eingesetzt.

Die symmetrischen Verfahren haben jedoch den Nachteil, dass der verwendete Schlüssel beiden Kommunikationspartnern bekannt sein muss. Das bedeutet, dass der symmetrische Schlüssel vor dem Aufbau der Verbindung auf geheimem Weg ausgetauscht werden muss. Erhalten dritte Personen Zugang zu diesem Schlüssel, so sind sie in der Lage, die Verbindung mitzulesen.

Diesen Nachteil weisen asymmetrische Public-Key-Verfahren (RSA, DSA, ElGamal etc.) nicht auf. Hierbei erzeugt jeder Kommunikationspartner ein Schlüsselpaar aus öffentlichem und privatem Schlüssel. Anschließend werden die öffentlichen Schlüssel ausgetauscht und können zur Verschlüsselung von Nachrichten genutzt werden. Da eine Nachricht, die mit einem öffentlichen Schlüssel verschlüsselt wurde, nur mit dem privaten Schlüssel gelesen werden kann, können die so erzeugten Mitteilungen nur von der gewünschten Person gelesen werden.

Damit die asymmetrischen Verfahren jedoch als sicher gelten können, sind wesentlich längere Schlüssel erforderlich. Übliche asymmetrische Schlüssellängen sind 512, 768, 1024, 2048 und 4096 Bit. Schlüssellängen kleiner als 1024 können nicht mehr als sicher eingestuft werden. Eine asymmetrische Verschlüsselung ist daher aus rechentechnischer Sicht aufwendiger als eine symmetrische Verschlüsselung. Daher werden üblicherweise beide Verfahren gemeinsam in einem sogenannten *Hybridverfahren* eingesetzt. Dabei wird die Nachricht mit einem zufälligen symmetrischen Schlüssel verschlüsselt und dieser mit einem öffentlichen Schlüssel verschlüsselt und angehängt. Nur der Besitzer des entsprechenden privaten Schlüssels kann den symmetrischen Schlüssel und damit die ganze Nachricht entschlüsseln.

3.3 Integrität

Schließlich ist ein VPN auch in der Lage, die Integrität der übertragenen Daten zu sichern. Dies ist erforderlich, damit die übertragenen Daten nicht verfälscht oder zusätzliche Daten injiziert werden können.

Hierfür werden üblicherweise kryptografische *Prüfsummen* verwendet. Diese haben eine ähnliche Bedeutung wie zum Beispiel die Quersumme. Wenn zwei Personen eine Zahl austauschen und sicherstellen möchten, dass bei der Übertragung kein Fehler passiert, so ermitteln sie eine Prüfsumme (zum Beispiel in Form der Quersumme) und übertragen diese ebenfalls (siehe Abbildung 3.4).

Eine einfache Prüfsumme, wie eine Quersumme, ein Paritätsbit oder die CRC32-Prüfsumme, genügt üblicherweise, um zufällige Datenübertragungsfehler zu entdecken. Für diese Anwen-

3 http://www.copacobana.org

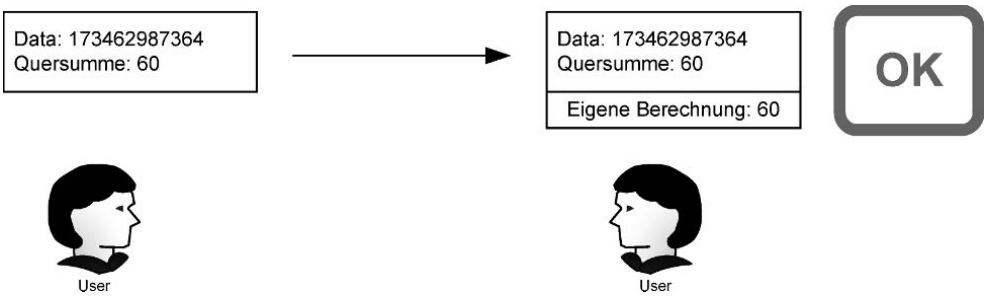

Abbildung 3.4: **Schutz vor Übertragungsfehlern mit einer Quersumme**

dung genügt ihre Komplexität. Wenn jedoch bewusste Veränderungen durch einen Angreifer erkannt werden sollen, reichen diese Prüfsummen nicht mehr. Hier sind kryptografische Prüfsummen, wie MD5, SHA-1 oder SHA-2 erforderlich. Diese Prüfsummen (Hash) verwenden Algorithmen, die eine Veränderung eines Textes unter Beibehaltung der Prüfsumme unmöglich machen sollen. Haben zwei Texte eine identische Prüfsumme, so spricht man von einer *Kollision*. Natürlich gibt es unendlich viele Kollisionen bei den heute eingesetzten Algorithmen. Jedoch sollen die eingesetzten Algorithmen eine bewusste Berechnung dieser Kollisionen in praktikabler Zeit unmöglich machen. Hier hat es jedoch in den letzten Jahren einige besorgniserregende Entwicklungen gegeben, die später genauer angesprochen werden.

Mit diesen Prüfsummen können nun sogenannte Authentifizierungswerte (Hash Message Authentication Codes, HMAC) erzeugt werden. Dazu erzeugt der Absender aus einem vorher ausgetauschten Geheimnis (PreShared Key, PSK[4]) und der Nachricht eine Prüfsumme und hängt diese an. Der Empfänger liest die Nachricht und erzeugt auf identische Weise die Prüfsumme. Stimmen beide Prüfsummen überein, so wurde die Nachricht nicht verfälscht und stammt aus der erwarteten Quelle. Ein Angreifer kann nicht die Nachricht so verändern, dass der Empfänger es nicht merkt, da ihm das PSK zur Erzeugung des HMAC fehlt.

3.4 Vor- und Nachteile eines VPN

Der Einsatz eines virtuellen privaten Netzwerks weist sowohl Vor- als auch Nachteile auf. Zunächst scheint ein VPN nur Vorteile zu bieten. Seine Funktionen umfassen den Schutz der Vertraulichkeit, der Integrität und garantieren die Authentifizierung der Kommunikationspartner. Damit wird die sichere und vertrauliche Übertragung sämtlicher Daten im VPN gewährleistet. Dies gilt für alle transportierten Informationen. Bei einem VPN ist es nicht erforderlich, jedes Applikationsprotokoll einzeln abzusichern.

In der Vergangenheit wurden häufig einzelne Applikationsprotokolle mit zusätzlichen Methoden (zum Beispiel Secure Socket Layer, SSL) gesichert. Diese zusätzliche Ebene garantierte

4 Dies ist nicht der *PreShared Key*, der auch zum Aufbau eines VPNs genutzt wird. Der Begriff *PreShared Key* kann in vielen Umgebungen verwendet werden und bedeutet lediglich, dass zwei Kommunikationspartner vorher einen Schlüssel ausgetauscht haben.

die Vertraulichkeit, Integrität und Authentifizierung der mit dem Applikationsprotokoll übertragenen Daten. Jedoch traf dies nur für die Daten zu, die mit dem entsprechenden Protokoll übertragen wurden. Mit SSL können HTTP, Telnet, POP, IMAP und die meisten weiteren TCP-Applikationsprotokolle gesichert werden. Zusätzlich wurden aber auch komplett neue Anwendungen entwickelt, die die Verschlüsselung bereits enthielten. Die Secure Shell ist ein Beispiel für eine derartige Anwendung. Sie ersetzt die klassischen UNIX-r-Dienste durch entsprechende verschlüsselnde s-Dienste.

Dennoch war für jedes Applikationsprotokoll die eigene Entwicklung einer derartigen Verschlüsselung oder eine Anpassung der Secure Socket Layer oder ihrer Weiterentwicklung, der Transport Layer Security (TLS), erforderlich. Ein VPN ist nicht auf ein Applikationsprotokoll beschränkt. Sämtliche übertragenen Daten werden unabhängig von dem verwendeten IP-Protokoll verschlüsselt übertragen. Hierbei spielt es keine Rolle, ob es sich um eine TCP-Verbindung oder eine UDP-Verbindung handelt. Auch ICMP-Pakete und selbst das Appletalk-DDP-Protokoll können über ein VPN übertragen werden (siehe Abbildung 3.5).

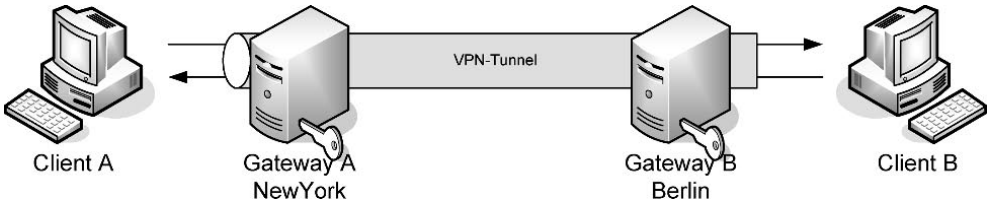

Abbildung 3.5: **Ein typischer VPN-Tunnel**

Der Aufwand für die sichere Übertragung all dieser Protokolle hält sich in Grenzen. Es muss lediglich einmal der verschlüsselte Tunnel aufgebaut werden, anschließend können über diesen Tunnel jegliche Informationen ausgetauscht werden.

Ein derartiges VPN hat jedoch nicht nur Vorteile. Die Verschlüsselung ist nur zwischen denjenigen Maschinen gewährleistet, die die Verschlüsselung durchführen. Der Bereich zwischen dem Client im Netzwerk Berlin und dem Gateway Berlin beziehungsweise zwischen dem Gateway im Netzwerk NewYork und dem Client NewYork in der Abbildung 3.5 ist nicht verschlüsselt. Hier werden die Daten im Klartext übertragen.

Der Endbenutzer ist darüber hinaus nicht in der Lage, die korrekte Verschlüsselung seiner Daten zu überprüfen. Beim Einsatz von zum Beispiel HTTPS hat der Benutzer direkt eine positive Rückmeldung der Verschlüsselung durch den Browser. Dieser kennzeichnet den erfolgreichen Aufbau einer verschlüsselten Verbindung üblicherweise mit einem geschlossenen Vorhängeschloß in der unteren Ecke. Hierbei handelt es sich also um eine Ende-zu-Ende-Verschlüsselung (vom Webserver zum Webbrowser). Bei einem VPN muss der Endbenutzer darauf vertrauen, dass das VPN (Abbildung 3.5) seine Aufgabe korrekt erfüllt.

Möchte der Endbenutzer gar sicherstellen, dass eine E-Mail von keinem außer dem gewünschten Empfänger gelesen werden kann, so kann ein VPN dies nicht leisten. Eine derartige Verschlüsselung kann nur durch Werkzeuge wie *Pretty Good Privacy* (PGP) oder *GNU Privacy Guard* (GnuPG) erreicht werden.

3.5 VPNs und Firewalls

Die größten Probleme bei dem Einsatz eines VPN entstehen jedoch, wenn ein VPN gemeinsam mit einer Firewall eingesetzt werden soll. Dabei ist das zunächst gar nicht zu verstehen. Beide Systeme versuchen, die Sicherheit der Daten zu gewährleisten. Sie erhöhen die Sicherheit des Unternehmens. Bei genauer Betrachtung stellt man jedoch fest, dass eine Firewall und ein VPN vollkommen unterschiedliche Methoden einsetzen, um dieses Ziel zu erreichen, und eigentlich auch zwei verschiedene Ziele verfolgen. Tabelle 3.1 zeigt bereits die wesentlichen Unterschiede auf.

VPN	FIREWALL
Verschlüsselung erlaubt keinen Einblick.	Untersucht den IP-Header und den Inhalt und protokolliert dies.
Erlaubt üblicherweise über das VPN ungehinderten Zugang.	Schränkt den Zugriff stark ein.
Erweitert das Netz um weitere Rechner und Netze.	Reduziert das zu schützende Netz auf einen Single Point of Defense.

Tabelle 3.1: **Vergleich VPN – Firewall**

Die wesentliche Tätigkeit eines VPNs ist die Verschlüsselung sämtlicher übertragener Informationen. Eine Firewall kann diese verschlüsselten Daten dann nicht mehr analysieren, unterscheiden oder protokollieren. Die Firewall ist sozusagen blind. Eine Firewall kann lediglich die unverschlüsselten Daten filtern.

Ein weiterer wesentlicher Bestandteil eines VPNs ist häufig der ungehinderte Zugang zum Intranet über das VPN. Der Vorstandsvorsitzende eines Unternehmens möchte von zu Hause aus über das VPN genau so arbeiten können, als ob er sich an seinem Arbeitsplatz in der Firma befindet. Hierzu benötigt er ungehinderten Zugang zu allen Systemen und Ressourcen, die die Firma bietet, einschließlich der Datenbanken, Mailserver oder Dokumentenrepositories. Die Aufgabe einer Firewall ist es jedoch, derartige Zugriffe von außen zu unterbinden oder auf ein Mindestmaß zu reduzieren. Auch hier kommt es zu einem Interessenkonflikt zwischen dem Firewall- und dem VPN-Administrator.

Der letzte Punkt stellt jedoch nach meiner Ansicht das größte Problem dar. Sobald eine VPN-Verbindung mit einem anderen Netzwerk oder einem Außendienstmitarbeiter aufgebaut wurde, werden die entsprechenden Rechner Teil des eigenen Netzes. Die eigene Firewall ist plötzlich auch für den Schutz dieser Rechner vor Angriffen von außen verantwortlich. Diese Rechner befinden sich nun logisch hinter der Firewall. Die Sicherheit dieser Rechner definiert plötzlich die Sicherheit des gesamten Rechnernetzes. Wenn die Firewall von dem Netzwerk NewYork in Abbildung 3.5 nicht richtig konfiguriert ist und ein Einbruch in Netzwerk NewYork erfolgte, so kann der Angreifer direkt auf die Rechner in Netzwerk Berlin unter Umgehung der Firewall in Netzwerk Berlin zugreifen.

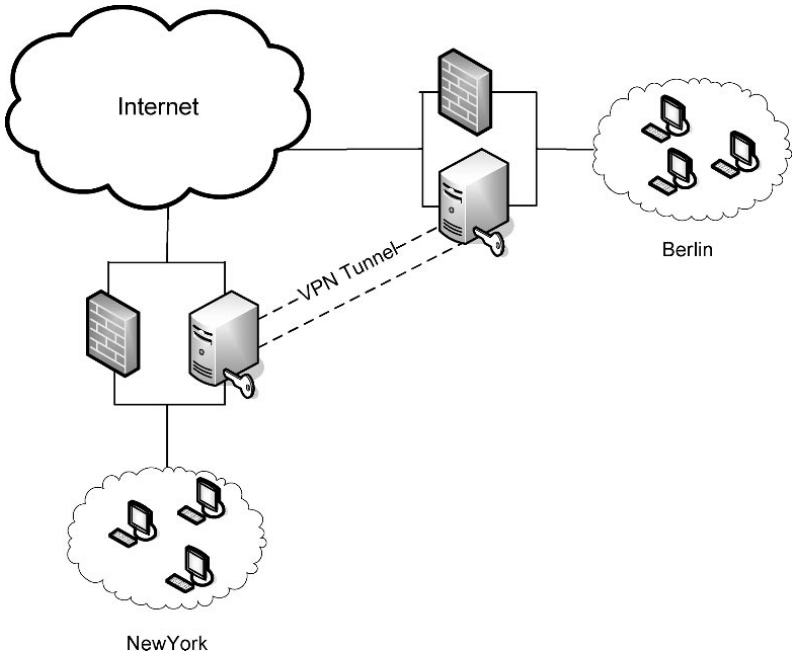

Abbildung 3.6: **Firewall und VPN sind parallel zueinander aufgebaut (schlecht)**

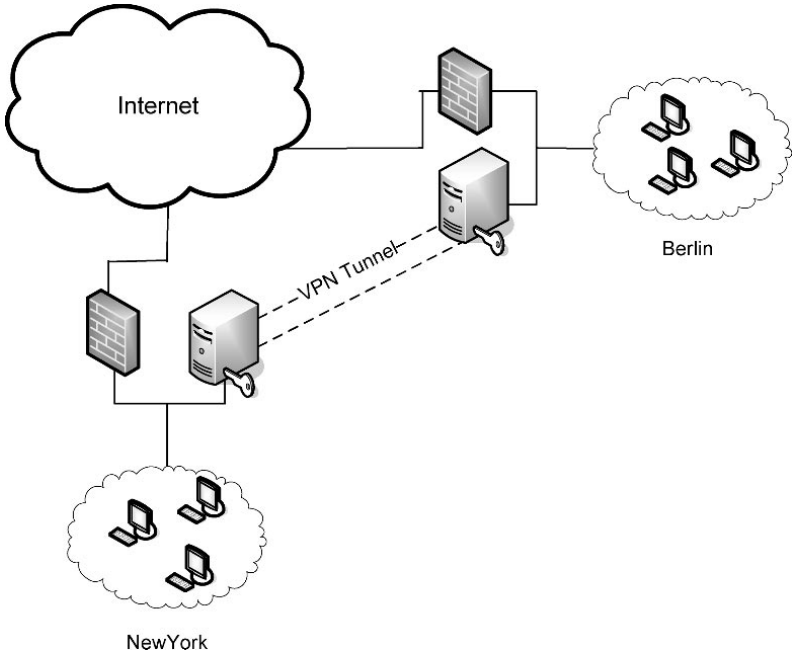

Abbildung 3.7: **Firewall und VPN sind nacheinander geschaltet**

INFO

Bisher gibt es leider kaum Open-Source-Software, die die Integrität und Sicherheit des Clients bei dem Aufbau eines VPNs überprüft. Eine Ausnahme ist die OpenVPN Management-*Software von Thorsten Robers (siehe Abschnitt 55.1). Diese erlaubt nur dann den Aufbau des VPNs, wenn der Rechner über einen aktuellen Virenscanner und eine Firewall verfügt.*

Dieser Zugriff des Angreifers ist natürlich nur möglich, wenn das VPN eine Umgehung der Firewall erlaubt. Leider wird in vielen Fällen das VPN derart konfiguriert, dass es einen Zugang parallel zu Firewall erlaubt (Abbildung 3.6).

Daher sollte sich immer zwischen dem VPN-Gerät und dem internen Netzwerk noch eine Firewall befinden, die den Zugriff auf das interne Netzwerk über das VPN kontrollieren und beschränken kann. Sinnvollerweise befindet sich auch vor dem VPN-Gerät eine Firewall, die das VPN-Gerät schützen kann (Abbildung 3.7).

So sind die über das VPN transportierten Daten und das dahinter liegende Netz optimal geschützt. Dieser logische Aufbau einer VPN/Firewall-Struktur wird auch von vielen kommerziellen Anbietern geschätzt. Diese bieten häufig ein gebündeltes Produkt an, das beide Funktionen (VPN und Firewall) bietet. Wird dieses Produkt auf einem physikalischen Gerät

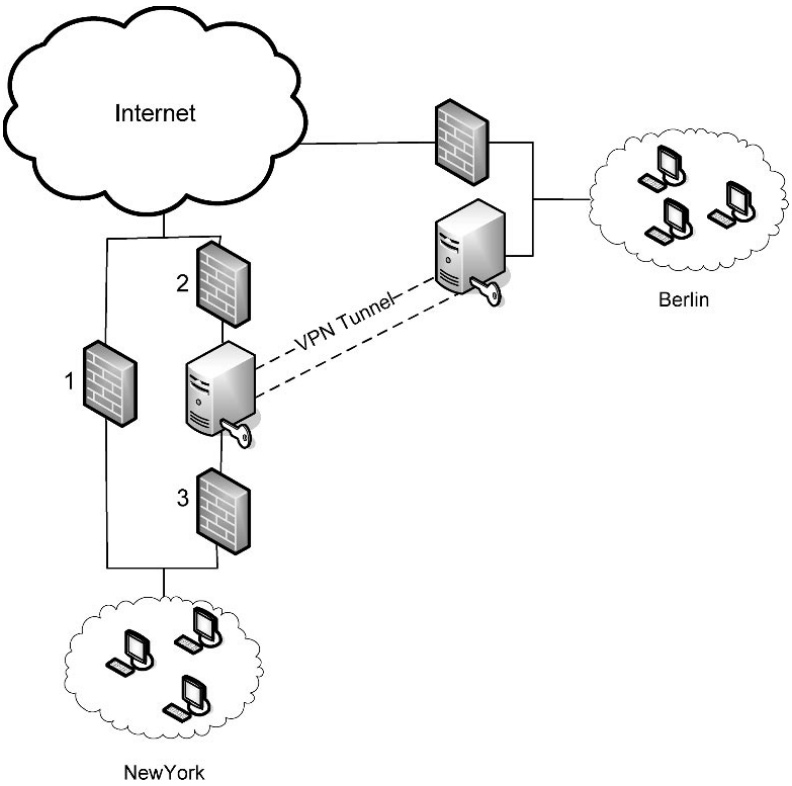

Abbildung 3.8: **Ideale VPN/Firewall-Struktur**

installiert, so kann von der logischen Funktion die in Abbildung 3.8 dargestellte Struktur realisiert werden.

Hierbei wird der normale Verkehr durch die Firewall 1 gefiltert. Parallel hierzu existiert ein VPN-Gateway, das durch zwei weitere Firewalls (2 und 3) geschützt wird. Hierbei filtert die Firewall 2 den verschlüsselten Verkehr von und nach außen und schützt die VPN-Gateway-Software vor Angriffen. Die Firewall 3 filtert den entschlüsselten Verkehr, der über das VPN-Gateway in das interne Netz gelangt.

Eine derartige Struktur kann mit Linux ebenfalls aufgebaut werden. Bei der Besprechung der entsprechenden Szenarien und Implementierungen werden Beispiel-Firewallregeln für Linux vorgestellt und erklärt. Dabei kann ich leider aus Platzgründen nicht sehr in die Tiefe gehen. Wenn Sie weitere Hintergrundinformationen zum Thema Firewall und Linux benötigen, können Sie diese in meinem Buch „Linux Firewalls mit Iptables & Co.“ (ISBN 9783827321367) nachlesen.