INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Systementwicklungsprojekt

IPSec Integration in einer Linux Umgebung

Bearbeiter:	Dietmar Fackelmann
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Helmut Reiser

INSTITUT FÜR INFORMATIK

DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Systementwicklungsprojekt

IPSec Integration in einer Linux Umgebung

Bearbeiter:	Dietmar Fackelmann
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Helmut Reiser
Abgabetermin:	30. März 2000

Hiermit versichere ich, daß ich die vorliegende Arbeit über das Systementwicklungsprojekt selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 23. Maerz 2000

....
(Unterschrift des Kandidaten)

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Das Internet bietet eine Reihe von Funktionen, die beruflich als auch privat genutzt werden können. Durch die breite Akzeptanz und der steigenden Anzahl von Anwendungen wurde auch schnell ersichtlich, daß zusätzliche Sicherheitsvorkehrungen zum Schutz des Datenverkehrs getroffen werden mußten. IPSec ist ein Sicherheitsprotokoll, das sich einigen Sicherheitsproblemen annimmt. Der Entwicklung des IPSec-Protokolls kommt dabei zugute, daß schon reichlich Erfahrungen im Gebiet der Sicherheit gesammelt wurden und zudem schon einige Konzepte vorhanden waren, die es nun galt zu verbessern.

1.1 Aufgabenstellung

Im Rahmen dieser Arbeit geht es um die Integration von IPSec (Internet Protocol Security) in einer geeigneten Testumgebung. Die Testumgebung besteht dabei aus vier Rechnern, auf denen jeweils das Betriebssystem Linux mit IPSec-Unterstützung zu installieren ist. Welche Software dazu zu verwenden ist wurde freigestellt. Die verwendete Software ist auf ihre Einsetzbarkeit hin zu testen (hierzu ist ein Beispielszenario zu erstellen). Die schriftliche Ausarbeitung soll theoretische als auch praktische Einblicke in IPSec geben.

1.2 Aufbau der Arbeit

Kapitel 2 gibt eine kurze Einführung in IPSec. Es werden die wichtigsten Punkte vorgestellt, auf denen in den nachfolgenden Kapiteln zum Teil noch näher eingegangen wird.

Kapitel 3 konzentriert sich auf die Vermittlungsschicht. Dabei wird auf das TCP/IP-Referenzmodell, das ISO-OSI-Referenzmodell und weitere Protokolle der Vermittlungsschicht eingegangen.

Kapitel 4 behandelt die einzelnen Bestandteile von IPSec. Dazu gehört der Authentication Header (AH), der Encapsulating Security Payload (ESP), der Security Parameter Index (SPI) und die Security Associations (SAs), als auch die Tunnel- und des Transport-Modi. Das Ende von Kapitel 4 widmet sich dem Thema der IPSec-Schlüsselverwaltung.

Kapitel 5 befaßt sich mit der Problematik der Netzwerksicherheit. Dazu wird der Begriff "Sicherheit" strukturiert. Themenschwerpunkt sind verschiedene Angriffsmethoden und inwieweit IPSec hier Schutz bieten kann. Schwachstellen von IPSec werden hier ebenso aufgezeigt. Richtlinien, die den Einsatz von IPSec noch sicherer gestalten sollen beenden Kapitel 5.

Der Schwerpunkt von **Kapitel 6** ist FreeS/WAN, eine Software, die ermöglicht IPSec auf Linux-Rechnern zu installieren. Beschrieben wird die Integration von FreeS/WAN in eine Testumgebung. Als Beispielszenario wird ein Virtuelles Privates Netzwerk (VPN) aufgebaut. Es wird auf die Installation, Konfiguration und Funktionsweise von FreeS/WAN eingegangen. Das Schlüsselmanagement, die Entwicklungsgeschichte und die Ziele des FreeS/WAN Projektes sind weiter Themen von Kapitel 6.

Kapitel 7 beschreibt den IPSec-Einsatz anhand von Einsatzzielen und Einsatzbeispielen. Zudem werden Interoperabilitäts- und Performance-Gesichtspunkte betrachtet.

Kapitel 8 beschäftigt sich mit der Entwicklungsgeschichte von IPSec. Vorgestellt werden die Internet Engineering Task Force (IETF) [IETF], die zu IPSec gehörenden Drafts und RFCs und vorhandene IPSec-Implementierungen.

Im **Anhang** werden weitere Sicherheitsprotokolle und IPSec beim Internet Protokoll Version 6 (IPv6) vorgestellt.

2 IPSec Internet Protocol Security

Ziel von IPSec ist eine sichere Internet-Kommunikation auf der Vermittlungsschicht. IPSec soll Schutz vor Verfälschung, unberechtigten Einblicken und Attacken durch externe Angreifer bei der TCP/IP-Kommunikation bieten. Mit IPSec können verschiedene Standorte ihren Datenverkehr sichern, wenn die Kommunikation über das öffentliche Internet stattfindet.

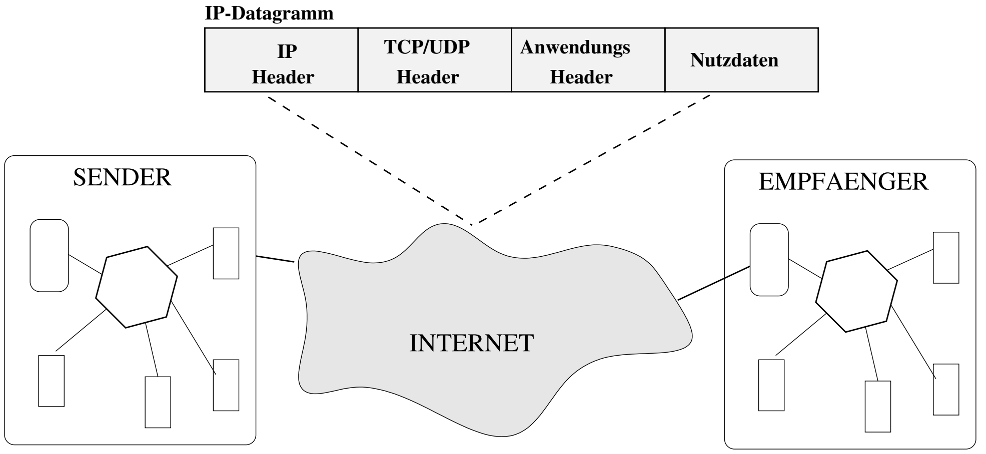

Abbildung 1: Verbindung zweier LANS über das Internet

IPSec erweitert das bestehende IPv4 Internet Protokoll um den Schutz der Vertraulichkeit, der Integrität und der Authentifizierung von Datagrammen. Im IPv6 Internet Protokoll ist IPSec standardmäßig enthalten. (nach [DeHi 98])

IPSec ist z.B. für den Aufbau komplexer VPNs (Virtuell Privat Networks) geeignet. VPNs nutzen keine exklusiven Leitungen sondern offene Netze wie das Internet wobei gewährleistet sein muß, das die Daten vor fremder Einsicht und Manipulation gesichert sind und ein Partner geeignet authentifiziert werden kann.
IPSec eignet sich für den sicheren Datenaustausch zwischen vertrauenswürdigen Rechnern, dabei kann festgestellt werden, ob der Datenverkehr von anerkannten Standorten stammt. Es kann jedoch nicht zuverlässig unterschieden werden, von welchem Benutzer des anerkannten Standortes der Datenverkehr stammt.

Ist IPSec im Einsatz, so werden die Sicherungsmaßnahmen automatisch vom System durchgeführt. Für den Benutzer sind diese Sicherungsmaßnahmen völlig transparent. Die Verantwortung einer sicheren Kommunikation liegt nicht beim einzelnen Benutzer. Der Benutzer braucht kein spezielles Wissen und kann auch nichts falsch machen.

IPSec ist ein Protokoll auf der Netzwerkschicht (Schicht 2) des TCP/IP Referenzmodells. Ein Sicherheitsprotokoll der Netzwerkschicht bietet Schutz für alle Daten von Applikationen in den darüberliegenden Schichten, ohne daß Applikationen modifiziert werden müssen. Somit profitieren sozusagen alle darüberliegenden Schichten von der mit IPSec gegebenen Sicherheit. IPSec schützt alle Protokolle die auf dem Internet Protokoll aufsetzen (z.B. HTTP, FTP, SNMP, TELNET, ...) [Fran 99].

Sicherheitsvorkehrungen auf Basis von Anwendungsprogrammen können das Ausspionieren von Kommunikationsbeziehungen und Angriffe durch gefälschte Adressen, also das vortäuschen falscher Absenderadressen bzw. Adressenverfälschung zum Umleiten von Daten auf z.B. unsichere

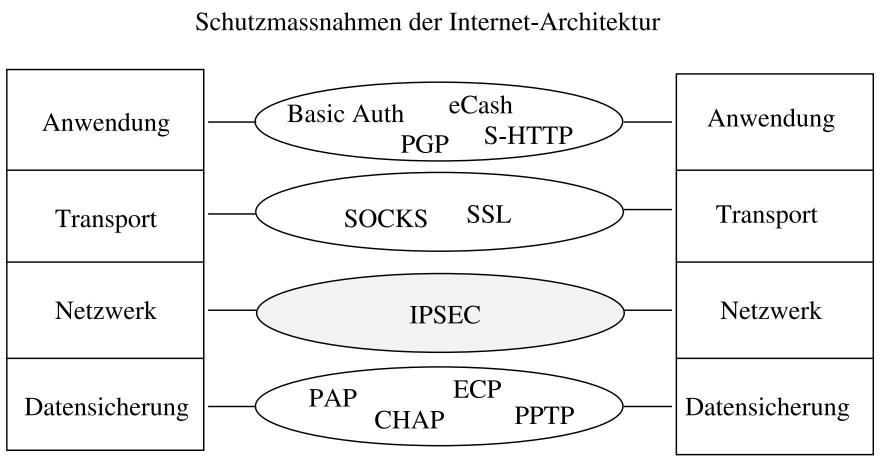

Abbildung 2: Einordung von IPSec [Raep 98]

Rechner, nicht verhindern. Hier hilft nur eine Sicherheitsvorkehrung im Transportprotokoll, also auf der Vermittlungsschicht. Bei Verfahren, die direkt auf der Kommuniaktionsleitung verschlüsseln, muß vor jedem Router dekodiert werden. Dadurch enstehen Perfomance-Probleme bei Hochgeschwindigkeitsleitungen. Zudem fehlt unter Umständen das Vertrauen in jeden Router.

Die neuen Sicherheitsfunktionen werden durch den IP Authentication Header AH [KeAt 98a] und des Encapsulating Security Payload ESP [KeAt 98b] umgesetzt. Dabei gewährleistet der Authentication Header mittels einer Prüfsumme die Datenintegrität und die Authentifizierung eines jeden Paketes. Der Encapsulating Security Payload sorgt auf Wunsch für die Vertraulichkeit des Datenverkehrs.

Die IPSec Working Group von der Internet Engineering Task Force (IETF) standardisierte das IPSec Framework (Rahmenwerk) um eine sichere Übertragung des Internet Protokolls IP auf der Vermittlungsschicht bereitzustellen. Das IPSec Framework stellt einen Rahmen dar, der festlegt, mit welchen Ergänzungen das IP Protokoll die Integrität der Datenpakete und die Vertraulichkeit der Inhalte sichert und welche kryptographische Algorithmen eingebunden werden. Diese Standardisierung ist eine gute Vorraussetzung um IPSec weit verbreitet antreffen zu können und sich auf dem Markt durchsetzen zu können. Hält man sich an diese Normen so können Implementierungen verschiedenster Hersteller in einer Einsatzumgebung eingesetzt werden.

3 IPSec ein Protokoll der Vermittlungsschicht

Im folgenden soll ein Einblick in die Vermittlungsschicht und deren Lage im OSI- und im TCP/IP-Referenzmodell gegeben werden (vgl. [Tane 96]). Dazu werden die einzelnen Modelle vorgestellt und es wird noch kurz auf die Funktionsweise der Vermittlungsschicht eingegangen. Zuletzt werden zwei Vermittlungsschicht-Protokolle, das Internet Protocol IP und das Internet Control Message Protocol ICMP vorgestellt.

3.1 Was ist eine Schicht / ein Protokoll?

Schicht
Die Aufgabe einer Schicht ist die Bereitstellung von Diensten für eine darüberliegende Schicht. Ein Dienst ist dabei eine Gruppe von Operationen, die zur Verfügung gestellt werden.

Protokoll
Ein Protokoll ist das Regelgefüge, das das Format und die Bedeutung der von den Partnereinheiten innerhalb einer Schicht ausgetauschten Rahmen, Pakete oder Nachrichten festlegt. Mit Protokollen führen die Einheiten ihre definierten Dienste aus. Sie können ihre Protokolle nach Belieben verändern, solange sie nicht ihre für den Dienstnutzer sichtbaren Dienste ändern.

3.2 Einordnung des IPSec-Protokolls in die Referenzmodelle

Das OSI-Modell und das TCP/IP-Referenzmodell haben viele Gemeinsamkeiten. Beide Modelle basieren auf dem Konzept eines Stapels unabhängiger Protokolle und beide besitzen Schichten mit z.T. ähnlichen Funktionen.

3.2.1 Das ISO-OSI-Referenzmodell

Das OSI-Modell (Open Systems Interconnection) basiert auf einen Vorschlag von der International Standard Organization ISO und beschäftigt sich damit, offene Systeme miteinander zu verbinden.

Das OSI-Modell ist in sieben Schichten eingeteilt:

7. Verarbeitungsschicht
6. Darstellungsschicht
5. Sitzungsschicht
4. Transportschicht
3. Vermittlungsschicht
2. Sicherungsschicht
1. Bitübertragungsschicht

Einordnung des IPSec Protokolls:
Das IPSec-Protokoll ist im OSI-Modell auf der Schicht 3, also der Vermittlungsschicht anzusiedeln. Die Vermittlungsschicht (Network Layer) hat die Aufgabe, Pakete vom Sender zum Empfänger zu bringen. Dazu gehört auch, daß Pakete evtl. über zwischengeschaltete Router hinweg transportiert werden müssen. Sie ist die unterste Schicht, die mit Ende-zu-Ende-Übertragung zu tun hat. (Die Sicherungsschicht muß im Vergleich lediglich Rahmen von einem Ende der Leitung zum anderen befördern).
Um Pakete über Router hinweg vermitteln zu können muß die Vermittlungsschicht über den Aufbau des Kommunikationsteilnetzes (d.h. der Router) Bescheid wissen und geeignete Pfade wählen. Es muß auch eine geeignete Auswahl der Router getroffen werden um Router effektiv einzusetzen, also nicht zu überlasten und eine geeignete Lastenverteilung zu garantieren.

3.2.2 Das TCP/IP-Referenzmodell

Das TCP/IP-Referenzmodell wird in dem weltweiten Internet angewendet. Es handelt sich dabei um ein paketvermittelndes Netz auf der Grundlage einer verbindungslosen Vernetzungsschicht.

Das TCP/IP-Referenzmodell besitzt 4 Schichten:

4. Verarbeitung
3. Transport
2. Internet
1. Host-an-Netz

Einordnung des IPSec Protokolls:
Das IPSec-Protokoll ist auf der Internet-Schicht anzusiedeln. Die sogenannte Internet-Schicht hat die Aufgabe den Hosts zu ermöglichen, Pakete in jedes beliebige Netz einzuschießen und unabhängig an das Ziel zu befördern. Die Internet-Schicht definiert ein offizielles Paketformat und Protokoll namens Internet Protokoll IP, dabei hat sie die Aufgabe, IP-Pakete richtig zuzustellen (Paket-Routing) und für einen Lastenausgleich zu sorgen. Die Internet-Schicht TCP/IP ist der OSI-Vermittlungsschicht sehr ähnlich.

3.3 Weiter Protokolle der Vermittlungsschicht

Zusätzlich zu IP, das für den Datentransfer benutzt wird, werden im Internet verschiedene Steuerprotokolle auf der Vermittlungsschicht benutzt, darunter ICMP, ARP, RARP und BOOTP. Nachfolgend werden die Protokolle IP und ICMP vorgestellt.

3.3.1 IP - Internet Protokol

Einzelne Schichten fügen jeweils einen Protokollspezifischen Header an den Anwendungsdaten an. So besteht ein komplettes IP-Datagramm aus den Steuerinformationen aller übergeordneten Protokollen und den Anwendungdaten.

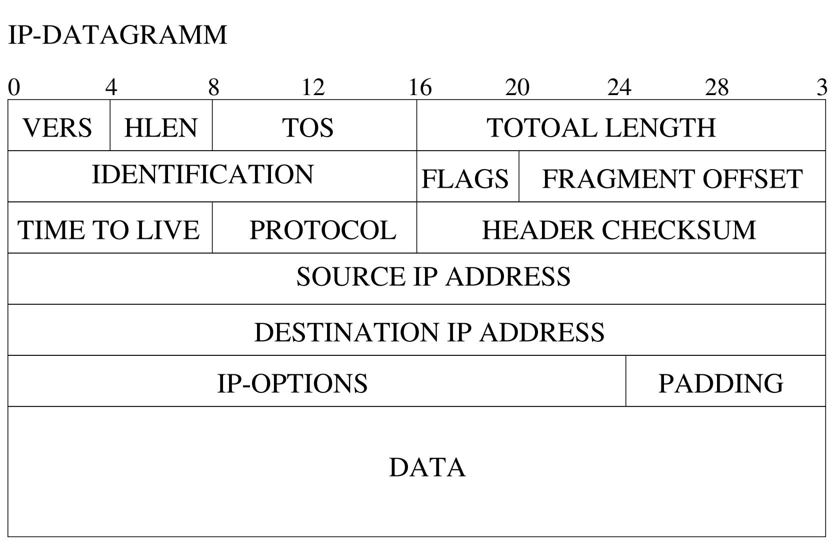

Abbildung 3: IP-Datagramm

HLEN (Header LENgth field) gibt die Länge des IP-Kopfsatzes in 32-Bit-Worten an. Meistens hat HLEN den Wert 5 (also 20 Bytes), da die Header oft kein IP-OPTIONS- und PADDING-Feld enthalten.

TOS (Type Of Service) - Feld hat 8 Bit und kann die Dienstgüte über das Datagramm festlegen. Dieses Feld kann wenn, dann nur von Routern interpretiert werden und es kann von einer höheren Schicht aus z.B. eine höhere Zuverlässigkeit beantragt werden. Jedoch können diese Angaben nur als Hinweis für die IP-Routingalgorithmen verwendet werden.

TOTAL LENGTH (16 Bit Feld) ist die Länge des gesamten IP-Datagramms in Bytes. Berechnung der Nutzdatengröße durch HLEN - TOTAL LENGTH. Dabei ist die Grenze von 2 hoch 16 also 65.535 Bytes vorgegeben.

TTL (Time To Live) legt die Lebensdauer fest. Dabei wird bei jedem Router der Wert um eins dezimiert. Die Lebensdauer wird also auf die Anzahl der passierten Router (Hops) zurückgeführt. Hat TTL den Wert 0 so wird das Datagramm verworfen und eine ICMP-Fehlermeldung an den Sender geschickt.

HEADER CHECKSUM stellt die Integrität des Headers sicher. Dabei wird die Prüfsumme nach jedem Gateway neu berechnet, da sich das TTL-Feld und noch andere Felder ändern.
Eine Prüfsummenberechnung auf die kompletten Daten würde die Weitergabe der Datagramme stark verzögern. Zur Berechnung wird der Kopfsatz als Folge von 16-Bit-Werten betrachtet, deren Einer-Komplemente addiert werden. Das Einer-Komplement der Summe liefert das Ergebnis.

SOURCE/DESTINATION-IP-ADRESS geben die 32-Bit-IP-Adresse der Kommunikationspartner wieder.

OPTION Das Feld kann verwendet werden, um bestimmte Routing-Pfade im Datenfeld durch den Absender vor der Versendung festzulegen (Source Routing) oder von den Gateways protokollieren zu lassen. Zudem kann es zur Zeiterfassung dienen, um Aussagen über die Verzögerungen auf den Teilstrecken treffen zu können.

DATA beinhaltet die Nutzdaten.

Fragmentierung Höheren Schichten wird die Möglichkeit bereitgestellt, beliebig große Datenmengen zu versenden. Da im Internet Protokoll IP aber nur Datagramme einer bestimmten Größe versandt werden können, findet eine Fragmentierung statt. Jedes Paketvermittelnde Netzwerk definiert eine Obergrenze für die Größe eines Datagrammes. Jedes Fragment muß den kompletten IP-Header des Datagramms kopieren und füllt die restliche maximale Anzahl von Bytes mit den Nutzdaten.

Die drei Felder IDENTIFICATION, FLAGS und FRAGMENT OFFSET kontrollieren die Fragmentierung und Zusammenfügung der Datagramme.

IDENTIFICATION bekommt vom Sender einen eindeutigen Wert des Datagramms, das unverändert jedem Fragment mit übergeben wird. Somit kann der Empfänger zusammengehörige Fragmente mit der Ziel-IP-Adresse identifizieren.

Mithilfe des FRAGMENT OFFSET kann das ursprüngliche Datagramm wieder zusammengesetzt werden. Auch wenn die einzelne Pakete in einer falschen Reihenfolge eintrafen, da der Abstand der Nutzdaten im originalen Datagramm festgehalten werden. Anhand des niederwertigsten der drei FLAG-Bits wird erkannt, ob das Fragment Daten aus dem Begin, der Mitte oder dem Ende enthält.

3.3.2 ICMP - Internet Control Message Protocol

Hauptaufgabe von ICMP sind Meldungen von Fehlern und Statusinformationen auf der Internetschicht. Diese werden im TYPE-Feld des ICMP-Headers über standardisierte Nummern referenziert (RFC 792).

ICMP-DATAGRAMM

0 4 8 12 16 20 24 28 31

TYPE	CODE	CHECKSUM
IDENTIFIER		SEQUENCE NUMBER
DATA (optional)		

Abbildung 4: ICMP-Datagramm

CODE hier wird der Inhalt der Nachricht eingegrenzt, um dem Empfänger die genaue Fehlerursache mitzuteilen.

Das Programm ping verwendet z.B. ICMP-Nachrichten um eine TCP/IP-Verbindung zu testen. Ping sendet dabei Echo-Request-Nachrichten an einen Rechner der, falls eine Verbindung besteht, mithilfe des in jedem Protokollstack integrierten ping-Servers mit Echo-Reply Nachrichten antwortet.

4 Bestandteile von IPSec

IPSec besteht aus unterschiedlichen Bestandteilen, die je nach Einsatzzweck eingesetzt werden können bzw. fest in IPSec verankert sind [IPSEC]. Dazu gehört der **Authentication Header AH** [KeAt 98a], der **Encapsulating Security Payload ESP** [KeAt 98b], der **Security Parameter Index SPI** , die **Security Association SA** und die **Schlüsselverwaltung**.

4.1 IP Authentication Header (AH)

AH gewährleistet mittels einer Prüfsumme die **Datenintegrität** und die **Authentifizierung** von Daten und statischen IP-Felder. Ein zusätzliches Nummernfeld schützt vor **Replay-Angriffen**.

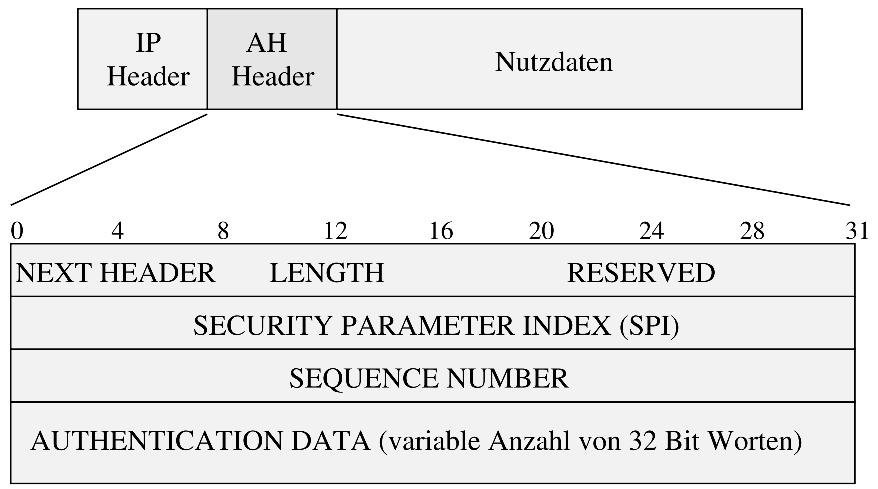

Abbildung 5: AH-Header

Datenintegrität
Checksumme generiert durch einen Message Authentication Code (z.B. MD5).
Datenauthentizität
durch einen in den Daten enthaltenen secret shared key.
Schutz vor Replay Attacken
durch eine Sequenznummer.

AH schützt Inhalte eines IP-Datagrammes bis auf die veränderlichen Felder (mutable fields), die sich während dem Transport verändern (z.B. TimeToLive-Feld). Die mutable fields werden bei der Berechnung als Null-wertige Felder angesehen. Der Wert des integritäts Checks wird im AH Header mitgeführt.

4.1.1 AH Transport Mode

Beim AH Transport Mode bleibt der Original IP-Header erhalten. Danach folgt der AH-Header und die Nutzdaten des Original-Datagrammes. Es können alle Veränderungen der Originaldaten und des AH-Headers festgestellt werden. Der AH Transport Modus bietet keinen Schutz der Vertraulichkeit, da alle Informationen im Datagramm in Klartext sind.

4.1.2 AH Tunnel Mode

Beim AH Tunnel Mode wird ein neuer IP-Header generiert, wobei sich in der Regel die Herkunft- und die Zieladresse im IP-Header von dem Original IP-Header unterscheidet. Auf

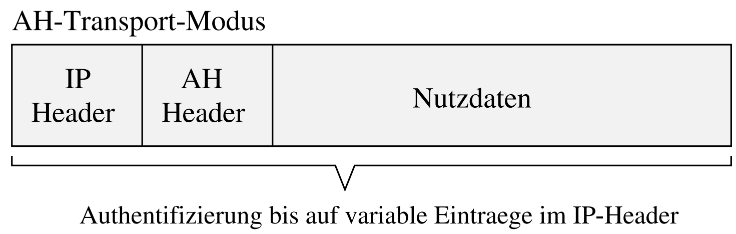

Abbildung 6: AH-Transport-Modus

den neuen IP-Header folgt der AH-Header und dann das original Datagramm (IP-Header mit Nutzdaten). Dabei können am neuen IP-Header, am AH-Header sowie am original IP-Header und den Nutzdaten Änderungen festgestellt werden. Auch hier sind alle Informationen in Klartext!

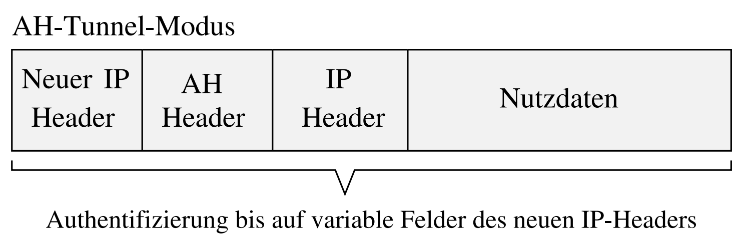

Abbildung 7: AH-Tunnel-Modus

Bemerkung:
AH kann alleine, in Kombination mit ESP (wird nachfolgend vorgestellt), oder verschachtelt eingesetzt werden. Durch diese Kombinationsmöglichkeiten kann z.B. eine Authentifikation zwischen zwei Hosts, zwischen zwei Firewalls bzw. einer Firewall und einem Host realisiert werden.

4.2 Encapsulating Security Payload (ESP)

ESP sorgt mittels Verschlüsselung für die **Vertraulichkeit** des Datenverkehrs. Wie bei AH kann ESP zur **Authentifizierung** dienen und besitzt auch geeignete Maßnahmen gegen **Replay-Angriffe**.

- **Schutz vor unberechtigtem Lesen** durch Verschlüsselung der Daten. Die Daten können nur von Personen gelesen werden, die den Schlüssel besitzen.
- **Originalzustand der Pakete** durch die Verschlüsselung der Pakete, wobei jede kleinste Änderung erkannt wird. Die Pakete können demnach nicht unbemerkt verändert werden.
- **Datenauthentizität** durch den HMAC-Algorithmus (s. AH - Authentication Header). Der Sender kann hier eindeutig Authentifiziert werden.
- **Schutz vor Replay Attacken** durch Sequenznummern.

ESP setzt einen **Symmetric Shared Key** ein. Diesen Schlüssel benützen beide Parteien zum Ver- und Entschlüsseln der Daten.

4.2.1 ESP Transport Mode

Beim ESP Transport Mode werden nur die Nutzdaten des original IP-Datagrammes und des ESP-Trailer verschlüsselt. Der IP-Header ist weder authentifizierbar noch verschlüsselt ! Die ESP

Authentifikationsfunktion schützt nur die Original Nutzdaten aber nicht den Original IP-Header. (AH schützt hier beides).

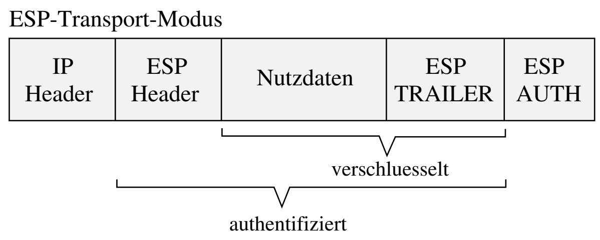

Abbildung 8: ESP-Transport-Modus

4.2.2 ESP Tunnel Mode

Beim ESP Tunnel Mode wird ein neuer IP-Header erzeugt. Das Original-Datagramm (IP-Header und Nutzdaten) und der ESP-Trailer werden verschlüsselt. Somit kann ein Angreifer also keine Informationen aus dem Original IP-Header gewinnen.
Die ESP Authentifikationsfunktion schützt den Original IP-Header und die IP-Nutzdaten, jedoch nicht den neuen IP-Header. (AH schützt hier beides).

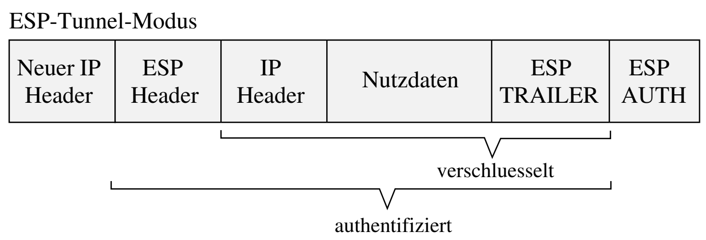

Abbildung 9: ESP-Tunnel-Modus

ESP kann alleine, in Kombination mit AH oder verschachtelt eingesetzt werden. Auch hier kann zum Beispiel eine Authentifikation zwischen zwei Hosts, zwei Firewalls bzw. einer Firewall und einem Host realisiert werden.

4.3 Einsatz von Tunnel- und Transport Mode

Der Einsatz vom Tunnel- bzw. vom Transport-Modus hängt von der jeweiligen Gegebenheit und Aufgabenstellung ab.

4.3.1 Transport Mode

Der Transport Modus wird normalerweise zwischen den Endpunkten einer Verbindung eingesetzt. Dies bedeutet eine sichere Kommunikation über den gesamten Verbindungspfad hinweg.
Die Protokolle authentifizieren das gesamte Paket bis auf veränderliche Einträge des IP-Kopfes. Die Verschlüsselungsmechanismen greifen nur für die höheren Protokollebenen und den Datenbereich des Paketes. Der ursprüngliche IP-Header bleibt erhalten.
Der Transport Modus spart im Vergleich zum Tunnel Modus Rechenzeit und ist für Verbindungen innerhalb eines sicheren Netzwerkes vorgesehen. (Beim Transport Mode muß kein neuer IP-Header generiert werde, es müssen nur die Nutzdaten verschlüsselt werden, dafür wird bietet der Transport Mode jedoch weniger Schutz).

4.3.2 Tunnel Mode

Der Tunnel Modus wird eingesetzt, falls zumindest ein Rechner keinen Endpunkt in der Verbindung darstellt. (Sichere Kommunikation zwischen Firewalls, oder entferntes Einwählen in ein Local Area Network (LAN) um eine sichere Verbindung bis zum Eingangs-Gateway zu erhalten).
Der Tunnel Modus bietet eine höhere Sicherheit, da das gesamte IP-Paket kodiert ist. Das Paket erhält einen neuen IP-Header gefolgt von einem AH- bzw. eine ESP-Header, die sowohl die Daten als auch die ursprünglichen Protokollköpfe sichern.

4.3.3 Nesting oder Bundling

Übersetzt bedeutet Nesting Verschachtelung bzw. Einschachtelung. Speziell beim Einsatz von IP-Sec spricht man von Nesting bzw. Bundling, wenn zwei Rechner (Rechner "A" und Rechner "B") unter einem bestimmten Modus miteinander kommunizieren und zwischen diesen zwei Rechnern zwei weitere Rechner (Rechner "1" und Rechner "2") einen weiteren Modus einsetzen. Dabei wird das Original IP-Datagramm in das Nutzdatenfeld eines neuen IP-Datagrammes eingebettet.

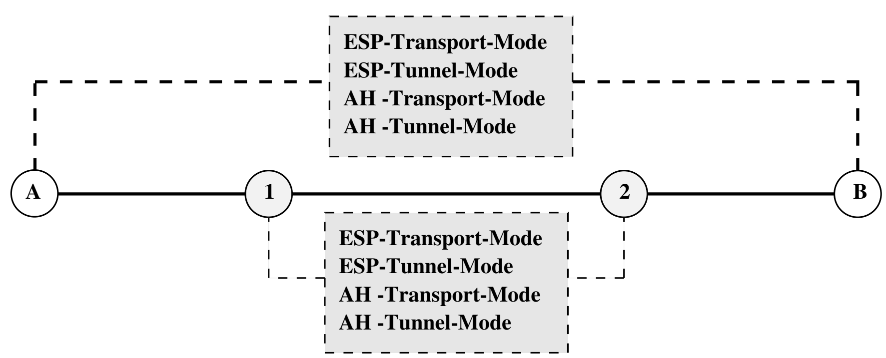

Abbildung 10: Nesting Möglichkeiten

Insgesamt existieren (bei einer Verschachtelungstiefe von zwei) sechzehn Kombinationsmöglichkeiten.

Ein Beispielszenario wäre z.B. der verschachtelte Einsatz vom ESP-Transport-Mode und dem AH-Tunnel-Mode. Rechner A und Rechner B nutzen den ESP-Transport-Mode, während zwischen der Firewall 1 und der Firewall 2 der AH-Tunnel-Mode zum Einsatz kommt.

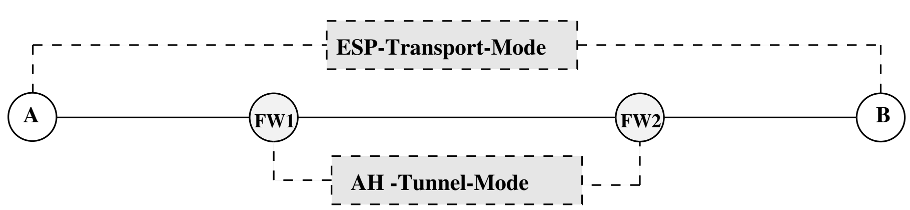

Abbildung 11: Nesting Beispiel

Zwischen den Firewalls sind somit beide Modi aktiv (ESP-Transport-Modus und der AH-Tunnel-Modus).

Ablauf der Kommunikation:

1. Rechner A setzt den ESP-Transport-Modus ein

2. Firewall 1 setzt den AH-Tunnel-Modus ein

3. Firewall 2 empfängt das AH-Tunnel-Datagramm, authentifiziert es und sendet das ausgepackte Paket an den Rechner B

4. Rechner B empfängt das Paket und kann es mit dem richtigen Schlüssel entschlüsseln.

4.4 Keyed- und HMAC-Hashes

Kryptografische Prüfsummen oder Hash-Funktionen bieten Schutz vor Manipulation von Daten. Dabei wird nach der Vorschrift **H=h(M)** aus einer Nachricht M ein Hashwert H mit fester Länge berechnet. (Einzelne Bitveränderungen in einer Nachricht ergeben einen ganz anderen Hash-Wert H, außerdem ist es nahezu unmöglich eine Nachricht M' zu erstellen, die denselben Hashwert H erzeugt.)

In IPSec Implementierungen kommen spezielle Varianten der Funktion **Message Digest 5 (MD5)** oder des **Secure Hash Algorithm (SHA)** zum Einsatz. (z.B. Keyed MD5 [MeSi 95], Keyed SHA-1, HMAC-MD5-96 [MaGl 98a], HMAC-SHA-1-96 [MaGl 98b])

Hashed Message Authentication Code (HMAC nach [KBC 97]) beschreibt ein Verfahren zur Berechnung einer kryptographischen Prüfsumme, bei der eine beliebige Hash-Funktion zum Einsatz kommen kann. HMAC gilt gegenüber den Keyed-Hashes (s.u.) als das sichere Verfahren.

4.4.1 Erzeugung von Keyed- und HMAC-Prüfsummen

Ein auf Sender und Empfänger gleichlautender Schlüssel (shared key) und das Datagramm sind in beiden Fällen die Eingabedaten für die Hash-Funktion.
Während bei Keyed MD5 und Keyed SHA-1 schon das Endergebnis in Form eines 128-(MD5) beziehungsweise 160-Bit-(SHA-1)-Hash-Wert vorliegt, wird der Vorgang bei HMAC-Implementierungen wiederholt. Diesmal jedoch mit dem Ergebnis aus der ersten Operation und dem geheimen Schlüssel. Aus dem Resultat der zweiten Berechnung finden die 96 höchstwertigen Bits als Authentication Data im AH- beziehungsweise im ESP-Header Verwendung. Sofern der geheime Schlüssel zuvor sicher ausgetauscht wurde, kann der Empfänger davon ausgehen, daß die Daten vom echten Absender kommen.

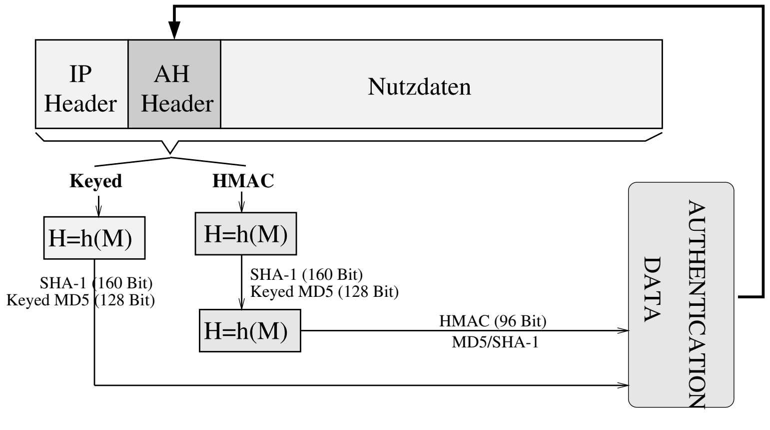

Abbildung 12: Erzeugung von Keyed- und HMAC-Prüfsummen [Raep 99]

Keyed- und HMAC-Hashes sorgen daher sowohl für eine Integritätskontrolle als auch für eine Authentifizierung des Kommunikationspartners.

Wird als Algorithmus der MD5 benützt so spricht man von einem HMAC-MD5. Das AH-Protokoll unterstützt auch HMAC-SHA. Bei Keyed MD5 und Keyed SHA-1 ist das Endergebnis ein 128 MD5 bzw. ein 160 SHA-1 Bit Hash Wert.

4.5 SPI - Security Parameter Index

IPSec bietet für IP-Pakete Geheimhaltung, Fälschungssicherheit oder beides. Für jede dieser Funktionen gibt es einen eigenen optionalen Paket-Header. Dieser Paket-Header enthält einen numerischen Wert und zwar den Security Parameter Index. Anhand des SPI stellt der Rechner fest, welche Chiffrierschlüssel und Verfahren er verwenden muß.

4.6 SA - Security Association

IPSec muß die beteiligten Rechner mit verschiedenen Schlüsseln und kryptographischen Verfahren assoziieren und deren Einsatz mit den IPSec-Headern koordinieren. Dazu müssen jeweils zwei Rechner eine Sicherheitsassoziation untereinander einrichten. Mithilfe dieser Beziehung wird dann das Verschlüsselungs- als auch das Authentifizierungsverfahren bestimmt.

Eine Sicherheitsbeziehung enthält unter anderem folgende Werte:

- Zieladresse
- Verfahren zur Verschlüsselung oder Authentisierung
- den jeweiligen Modus (Transport- oder Tunnel-Modus)
- aktueller geheimer Schlüssel für diese Beziehung
- weitere Parameter speziell für dieses Verfahren
- Zeitangabe der Gültigkeit eines Schlüssels

Erreicht ein Paket mit einem IPSec-Header einen IPSec unterstützenden Rechner, so wird anhand des Security Parameter Index (SPI) und der IP-Adresse bestimmt, welche Security Association (SA) auf diesen IPSec-Header anzuwenden ist. Sicherheitsbeziehungen (SAs) zwischen zwei Rechnern werden mithilfe der eindeutigen IP-Adressen formuliert. Es wird eine Security Association (SA) pro Kommunikationsrichtung benötigt.

4.7 IPSec-Schlüsselverwaltung

Für Sicherheitsprotokolle sollte die Beziehung zwischen einzelnen berechtigten Einheiten sowie deren Chiffrierschlüsseln und Identifizierungscodes inerhalb der Nachrichten definierbar sein. Bei IPSec wird anhand des Security Parameter Indexes (SPI) die jeweils geltende Sicherheitsassoziation (SA) für den jeweiligen IPSec-Header ausgewählt. Benötigt wird noch eine Methode, um diese SAs einzurichten und ihnen SPIs zuzuordnen.

Die Verwaltung und Verteilung der Schlüssel wird nicht innerhalb von IPSec gelöst. Man geht davon aus, daß es ausreichend sichere Verfahren gibt. Es gibt vier Standardkonzepte zum IPSec-Schlüsselaustausch. Der Unterschied einzelner Konzepte besteht dabei in der Produktverfügbarkeit und Benutzerfreundlichkeit und weniger hinsichtlich Sicherheitsaspekte. Die manuelle Schlüsselverwaltung ist standardmäßig in jeder IPSec-kompatiblen Implementierung verfügbar.

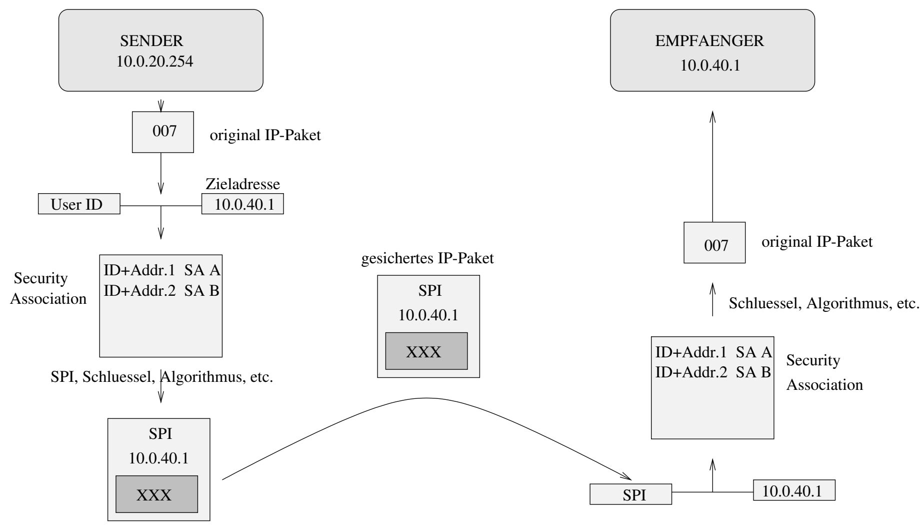

Abbildung 13: Kommunikationsablauf

4.7.1 Manuelle Schlüsseleinrichtung

Sicherheitsassoziationen werden hier manuell konfiguriert. Hier werden die SPIs, kryptographische Verfahren und die Schlüssel sowie die Angabe eines Rechners, der diese Angaben verwendet definiert. Die Sicherheitsassoziationen werden in einer Textdatei mit einheitlichem Format definiert. Alle Zahlen werden dezimal dargestellt. Bei längeren Zahlen werden die einzelnen Bytes durch Punkte getrennt.

```
src-host:          10.0.0.1
dst-host:          10.0.40.1
dst-spi:           1.2.3.4
RFC1828-Key-128:   1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16
RFC1829-Key-56:    1.2.3.4.5.6.7.8

src-host:          10.0.40.1
dst-host:          10.0.0.1
dst-spi:           5.6.7.8
RFC1828-Key-128:   9.10.11.12.13.14.15.16.8.7.6.5.4.3.2.1
RFC1829-Key-56:    8.7.6.5.4.3.2.1
```

4.7.2 Automatischer Schlüsselaustausch

Der automatische Schlüsselaustausch erfolgt nach dem Internet-Key-Exchange Protokoll (s. 4.7.4).

4.7.3 SKIP - Simple Key Interchange Protocol

SKIP tauscht Schlüssel zwischen IPSec-Rechnern aus [MoGu 98]. Es wird ein spezieller Header vor den IPSec-Headern in die IP-Pakete eingetragen. Dabei beruht der Schlüsselaustausch auf einem gemeinsamen Geheimnis, oder es wird ein authentifizierter Schlüsselaustausch mit dem Diffie-Hellmann-Algorithmus (ein Public-Key-Algorithmus) durchgeführt. Dabei tauschen die Kommunikationspartner ihre öffentlichen Schlüssel aus und können in Verbindung mit ihren privaten Schlüsseln ohne eine zusätzliche Verbindungsaufnahme einen gemeinsamen geheimen Wert erzeugen. SKIP verursacht dabei einen Overhead von 20 bis 30 Byte in jedem Paket. SKIP wurde von

Sun Microsystems entwickelt und im Produkt SunScreen zur Aushandlung von Schlüsseln in VPNs verwendet.

4.7.4 IKE/ISAKMP

IKE ist ein Protokoll, das der Verwaltung von Security Associations innerhalb IPSec dient. Mit IKE ist der Protokollrahmen ISAKMP der IETF umgesetzt worden. IKE wird gebraucht, da IPSec die zur Verschlüsselung notwendigen Informationen (Algorithmus, Schlüssel, Gültigkeitsdauer etc.) nicht selbst überträgt, sondern sie aus einer lokalen SA bezieht.

Ineternet Key Exchange (IKE) ist ein in RFC 2410 beschriebenes Schlüsselaustauschverfahren, das speziell auf die Benutzung mit ISAKMP abgestimmt ist. In dem RFC befinden sich genaue Definitionen für das gesamte Verfahren des Schlüsselaustauschs.

Die dazu benötigten Schlüssel werden mittels des asymmetrischen Diffie-Hellman-Verfahrens ausgetauscht. Die wiederum dazu notwendigen öffentlichen Schlüssel müssen über den Verzeichnisdienst einer CA (PKI) bereitgestellt werden.

Das Internet Security Association and Key Management Protocol ist ein Protokoll zur Verwaltung von Sicherheitsassoziationen als auch zum Schlüsselaustausch [MSST 98]. Es bietet Funktionen für die Aushandlung, Einrichtung, Modifikation und Zurücknahme von SAs. ISAKMP gibt letztendlich nur den formalen und organisatorischen Rahmen für die Verwaltung von SAs vor. Die eigentliche Schlüsselgenerierung liegt in der Verantwortung anderer Protokolle (z.B. OAKLEY [Orma 98]). ISAKMP ist ein Protokoll der Anwendungsschicht und deshalb z.T. schlechter zu integrieren, belastet den Netzverkehr jedoch weniger, da keine Schlüsselverwaltungsdaten in jedem Paket übertragen werden. ISAKMP wird zur Automatisierung der Generierung und der Erneuerung kryptografischer Schlüssel eingesetzt.

Automatisches setzen der SA (Security Associations) Automatisches Key Management ist vor allem bei großen Installationen anzuraten. Es muß darauf geachtet werden, daß die unterschiedlichen Sicherheitrichtlinien (Security Policies) der miteinander kooperierenden Systeme angeglichen werden. Das System kann nur so sicher sein, wie sein schwächstes Glied.

Durch eine dynamische Adressvergabe der IP-Adressen lassen sich keine SAs auf Basis der IP-Adresse bilden. Dieses Problem kann in ISAKMP durch vorher ausgetauschte Schlüssel oder einer auf individuelle Benutzerzertifikate basierenden Authentifizierung gelöst werden. Man könnte so z.B. den Namen bzw. die E-mail Adresse verwenden [Raep 99]. In Verbindung mit einem gültigen, von einer vertrauenswürdigen Instanz unterschriebenem X.509-Zertifikat [PKIX] ist die Identität des mobilen Mitarbeiters durch seine digitale Signatur sicher überprüfbar.

ISAKMP automatisiert die Generierung und die Erneuerung der kryptografischen Schlüssel, dabei soll die manuelle Konfiguration soweit wie möglich ausgeschlossen werden. Der sichere Austausch der Schlüssel ist die kritischste Phase in Bezug auf die Sicherheit der Verbindung. Der ISAKMP Informationsaustausch muß verschlüsselt und mit einer geeigneten Authentifizierung ablaufen, damit niemand die Schlüsselinformationen lesen kann und der Austausch nur zwischen authentifizierten Partnern stattfindet.
Das ISAKMP-Protokoll beinhaltet die komplexesten und Prozessor-intensivsten Operationen im IPSec Protokoll.

Folgende Punkte wurden beim Design beachtet:

- **Denial of Service:** Die Nachrichten enthalten eindeutige “cookies” zur schnellen Identifizierung ohne intensive Berechnungen durchführen zu müssen.

- **Man-in-the-Middle:** Schutz vor dem Löschen und Verändern, dem Umlenken und dem Reflektieren der Nachrichten sowie dem nochmaligen versenden alter Nachrichten.
- **Perfect Forward Secrecy:** Unabhängige Schlüsselauswahl

Die 2 Phasen von ISAKMP

- **Phase 1:** Hier kommt das "master secret" zum Einsatz, das bei der gesamten Kommunikation zum Schutz verwendet wird. Diese Phase dient auschließlich zum Schutz der ISAKMP Nachrichten. Die Phase 1 wird einmal (am Tag oder in der Woche) ausgehandelt, woraufhin die Phase 2 dann mehrmals (in einigen Minuten) abgehandelt werden kann [Raep 99].
- **Phase 2:** Hier werden die SA's und die Schlüssel ausgehandelt, welche die Nutzdaten schützen sollen. Die nötigen ISAKMP-Nachrichten werden durch die in der Phase 1 generierten SAs geschützt [Raep 99].

4.7.5 Photuris

Photuris konkurriert mit ISAKMP [MSST 98] Es dient zur Einrichtung von Sicherheitsassoziationen sowie zum Schlüsselaustausch und verwendet den Diffie-Hellman-Algorithmus. Dabei ist das Protokoll nicht so komplex aber auch nicht so flexibel wie ISAKMP. Es enthält keine Komponenten, mit denen ein einfacher und direkter Austausch von Sitzungsschlüsseln auf Basis geheimer Schlüssel möglich ist.

5 Netzwerksicherheit durch IPSec

Sicherheit ist ein komplexes Thema, es müssen verschiedenste Aspekte betrachtet werden. Dieses Kapitel soll dabei zur allgemeinen Strukturierung dienen um IPSec besser anhand dieser Struktur einordnen zu können.

Um die potentielle Gefährdung zu verdeutlichen, werden Programme und Methoden beschrieben, mit denen versucht wird unerlaubten Zugriff auf ein System zu erhalten oder unberechtigt Einblicke in Daten zu gewinnen, als auch Daten zu manipulieren. Dabei wird auch die Frage beantwortet, ob in dem entsprechendem Fall der IPSec-Einsatz den genannten Angriffen standhalten kann. Es sollte klar werden, für welche Aufgaben sich der IPSec-Einsatz eignet und für welche nicht. Um den IPSec Einsatz so sicher wie möglich zu machen, sollten die am Ende aufgezählten Punkte beachtet werden.

5.1 Strukturierung des Begriffes Sicherheit

Sicherheitsanforderungen, Sicherheitsbedrohungen, Kommunikationspartner und das Schichtenmodell mit den einzelnen Schichten und deren Funktionen sind eng miteinander verstrickt. Diskutiert man über Sicherheit so ergeben die vier oben genannten Punkte eine wichtige Diskussionsgrundlage [Reis 97]

5.1.1 Sicherheitsanforderungen

Es existieren folgende Sicherheitsanforderungen

1. **Authentisierung (Authentication)**
   Authentifikation des Kommunikationspartners.

2. **Zugriffskontrolle (Access Control)**
   Jeder Benutzer bekommt jeweils die für ihn zugeordneten Rechte.

3. **Vertraulichkeit (Confidentiality)**
   Unberechtigte Personen können Daten nicht einsehen.

4. **Integrität (Integrity)**
   Die gesendete Nachricht ist unverändert.

5. **Verbindlichkeit**
   Unterschriftenersatz.

6. **Dienstgüte (Denial of Service)**
   Verfügbarkeit eines Dienstes.

Je nach Bedarf, müssen bei einer Kommunikationsverbindung zum Teil auch alle oben genannten Sicherheitsanforderungen erfüllt werden.

5.1.2 Sicherheitsbedrohungen

Sicherheitsbedrohungen können in Angriffe auf Rechen- und Kommunikationssysteme und der Kryptoanalyse unterteilt werden.

Angriffe auf Rechen- und Kommunikationssysteme Hier kann noch feiner zwischen aktiven und passiven Angriffen unterschieden werden:

- **passive Angriffe** (Daten bleiben unverändert)
  - Abhören des Netzverkehrs (Wiretapping)
  - Lesen von Daten (z.B. lesen von Backupdaten)
  - Verkehrsflußanalyse (Traffic Analysis)
- **aktive Angriffe** (Daten werden verändert, entfernt, erzeugt oder die Verfügbarkeit von Ressourcen wird gestört)
  - Verfügbarkeit von Diensten wird verschlechtert oder verhindert (Denial-of-Service Attack)
  - Vortäuschen einer anderen Identität (Masquerade, Man in the Middle Attack)
  - Nochmaliges Senden einer vorher abgehörten Nachricht (Replay Attack)
  - Täuschung von berechtigten Nutzern (Social Engineering Attack)

Kryptoanalyse Es existieren folgende Angriffsmethoden:

- **Kryptoanalyse**

  Unter Kryptologie versteht man das Entwerfen von Chiffren (Kryptographie) und das Aufbrechen derselben (Kryptoanalyse).
  Dabei existieren drei Varianten der Kryptoanalyse:
  - nur Chiffretext (Ciphertext-only Attack)
  - bekannter Klartext (Known-Plaintext Attack)
    besteht aus Chiffretext und dem passenden Klartext
  - gewählter Klartext (Chosen-Plaintext Attack)
    Teile eines selbst gewählten Klartextes verschlüsselt.

- **Protokollangriffe**

  Kryptographische Protokolle werden zum Beispiel zum sicheren Schlüsselaustausch eingesetzt. Hier gibt es auch wieder die Unterteilung in passive und aktive Angriffe.

5.1.3 Kommunikationspartner

Um geeignete Maßnahmen bzgl. der Sicherheit treffen zu können, müssen die einzelnen Kommunikationspartner betrachtet werden. Es besteht ein erheblicher Unterschied, ob z.B. ein Endbenutzer mit einem Prozeß auf einem Transitsystem oder mit einem anderen Endbenutzer kommuniziert. Dabei gibt es noch den Unterschied, daß der Endbenutzer selbst Sicherheitseinstellungen vornehmen kann bzw. daß der Endbenutzer gar nichts von den Sicherheitsvorkehrungen mitbekommt (transparente Sicherheitsvorkehrungen).

5.1.4 Maßnahmen in einer Schicht

Anhand von dem Schichten-Modell kann man nun Überlegungen anstellen, in welcher Schicht am effizientesten eine Maßnahme getroffen werden kann, um den gestellten Anforderungen zu genügen. Dabei spielen unter anderem Kosten, Nutzen, Aufwand als auch die Einsetzbarkeit und die Handhabung eine Rolle.

Für ein und dasselbe Problem kann es mehrere Lösungen in z.T. auch unterschiedlichen Schichten geben.
Auf der **Bitübertragungsschicht** kann man z.B. Kabelkanäle die Argongas unter Hochdruck enthalten einsetzen, die einen Alarm bei dem Versuch den Kabelkanal anzubohren auslösen.
Auf der **Sicherungsschicht** können Pakete in einer Punkt-zu-Punkt-Leitung kodiert und beim Empfänger wieder dekodiert werden. Müssen die Pakete jedoch mehrere Router durchlaufen, so müssen sie bei jedem Router dekodiert werden, wodurch sie wieder Angriffen ausgesetzt sind.
Auf der **Vermittlungsschicht** können z.B. Firewalls eingesetzt werden, um Pakete zu schützen und Unberechtigten Personen den Zugang zu versperren.
Auf der **Transportschicht** können komplette Verbindungen von Prozeß zu Prozeß verschlüsselt werden. Dies bedeutet hohe Geheimhaltungssicherheit, ist jedoch nicht ausreichend für Authentifikation und Nichtanerkennung.

5.1.5 Einordnung des IPSec-Protokolls

Es stellen sich also folgende Fragen:

1. Welche Sicherheitsanforderungen habe ich?
2. Welche Sicherheitsbedrohungen existieren?
3. Welche Kommunikationspartner will ich sichern?
4. In welcher Schicht will ich Maßnahmen zur Sicherung durchführen?

Der IPSec-Ansatz liefert dabei folgende Antworten:

1. **Sicherheitsanforderungen**
   Sicherung auf der IP-Ebene durch Authentisierung, Vertraulichkeit, Integrität und Zugriffskontrolle
2. **Sicherheitsbedrohungen**
   Kryptoanalyse, Angriffe auf Rechen- und Kommunikationssysteme (Wiretapping, Masquerade, Man in the Middle Attack, ...)
3. **Kommunikationspartner**
   Endsysteme bis hin zu Teilnetzen (VPN)
4. **Betreffende Schicht**
   Vermittlungsschicht und dadurch transparent für die Benutzer

5.2 Angriffsmethoden und IPSec

Es werden nun einige Methoden bzw. Programme vorgestellt, die eine Sicherheitsbedrohung darstellen.
Es wird versucht, Sicherheitvorkehrungen zu umgehen. Dies wird z.T. erreicht durch Verletzung der

- Vertraulichkeit (Sniffing, Man-in-the-middle Attack, Verkehrsflußanalyse)
- Integrität (Veränderungen, Session Hijacking, Replay Angriffen) und
- Authentisierung (IP-Spoofing)

5.2.1 Wiretapping

Unter Wiretapping versteht man das Abhören des Netzverkehrs. Dabei gibt es nützliche Tools, die als Sniffer-Programme bezeichnet werden. Sniffer sind Programme, die an ihrer Netzwerkschnittstelle Netzwerk-Pakete aufzeichnen und auswerten. Sniffer können als Analyse-Werkzeuge eingesetzt werden, um Fehler zu lokalisieren. Daher werden sie häufig von Systemadministratoren zu Diagnosezwecken eingesetzt. Normalerweise können alle Rechner desselben Netzwerks den gesamten Datenverkehr mithören, jedoch reagieren sie normalerweise nur auf die an sich gesendeten Pakete, da nur diese für den einzelnen Rechner von Bedeutung sind.

Ein Sniffer ist nun in der Lage Pakete aufzuzeichnen, die nicht an ihn adressiert worden sind. Diese Pakete können ausgewertet werden und sicherheitskritische Inhalte (z.B. Paßwörter) beinhalten.
Schutz bietet eine Verschlüsselung der Daten, dabei können die Daten z.B. auf der Vermittlungsschicht aber auch in einer darüberliegenden Schicht verschlüsselt werden.
Schutz durch Erkennen und zu Beseitigen eines Sniffers ist mit sehr viel Aufwand verbunden.

Die ESP-Verschlüsselung von IPSec schützt den gesamten Netzverkehr einschließlich wiederverwendbarer, geheimer Paßwörter und macht somit z.B. das Abhören von Paßwörtern (password sniffing) unmöglich.

5.2.2 Spoofing

Als Spoofing (dt. Vortäuschen) wird ein Angriff bezeichnet, bei dem der Angreifer versucht unberechtigt Zugriff auf ein System zu bekommen, indem er eine falsche Identität vortäuscht. (Von der Datensicherungsschicht bis hin zur Anwendungsschicht möglich).

IP-Spoofing
Vortäuschen einer IP-Adresse (Vermittlungsschicht). Somit greifen z.B. die Paketfilter nicht mehr, da hier die IP-Adresse als Identifikator für vertrauenswürdige Rechner benützt wird.
Der Angreifer benötigt:
- Kenntnis des zeitlichen Ablaufs des Protokolls
- Adresse eines internen Systems

ARP-Spoofing
ARP-Nachrichten verwenden keine eindeutige Kennung. Der Angreifer kann also gefälschte ARP-Antwortpakete an andere Rechner schicken, die eine andere IP-Adresse der MAC-Adresse des Angreifers zuordnen. Der jeweilige Rechener akzeptiert die Antwortpakete und trägt das neue Adressenpaar in seinen Cache ein. (Lösung: Statische ARP-Einträge)

UDP-Spoofing
Erfolgt die Identitätsprüfung nur auf IP-Adressebene und Portnummer, können UDP-Pakete problemlos eingeschleust werden, da UDP-Pakete zustandslos sind. Gefälschte Pakete können nicht erkannt werden.

Der IPSec-Einsatz schützt vor Spoofing-Angriffen z.B. durch Verwendung von Authentication Headers (AHs). Der angreifende Rechner kann keine zulässigen AHs erzeugen, da er den Chiffrierschlüssel nicht kennt.

5.2.3 Denial-of-Service-Attacken

Hier sind auch alle Schichten betroffen. SYN-Flooding nutzt z.B. eine Schwachstelle beim TCP-Verbindungsaufbau aus und überlastet den Server durch eine Menge an Anfragen nach einer neuen Verbindung. Die Pakete enthalten verschiedene Absenderportnummern, als auch IP-Adressen anderer Maschinen. Der Server muß alle ankommende Anfragen abarbeiten wobei für jedes Segment Empfangspuffer und Speicherplatz für Kontrollinformationen bereitgestellt wird. Da Ressourcen erst wieder nach einer bestimmten Zeit, falls z.B. keine Bestätigungs-Pakete mehr eintreffen, freigegeben werden, können womöglich andere Benutzer diese Ressourcen nicht mehr nutzen.

IPSec kann SYN-Flooding-Angriffe abwehren, da die SYN-Nachrichten einen gültigen AH mit korrekter kryptographischer Prüfsumme besitzen müssen. Werden korrekte Anfragen mitgeschnitten und dann für ein SYN-Flooding-Angriff verwendet, so kann man dem durch einen Schlüsselaustausch entgegnen.

5.2.4 Session-Hijacking

Übernahme von bestehenden Verbindungen. Hier werden gezielt TCP-Segmente mit anderen Daten und gefälschten IP-Adressen und den zugehörigen Initial Sequence Numbers (ISNs) eingeschleust. Die Verbindung zwischen dem eigentlichen Client und dem Server wird dann aufgrund der falschen ISN unterbrochen.

Verbindungen die durch Authentication Headers (AHs) geschützt sind, können nicht übernommen werden, da der Angreifer nicht in der Lage ist gültige AHs zu erzeugen. Für die Generierung gültiger AHs benötigt er den geheimen Schlüssel.

5.2.5 ICMP-Angriffe

Manipulation von Vermittlungswegen, Neukonfiguration von Rechnern und Flusskontrollmechanismen für Datagramme können für Angriffe genutzt werden.

Destination Unreachable Meldungen können an einen Rechner gesendet werden, wodurch alle Verbindungen zu der in der Nachricht angegebenen IP-Adresse unterbrochen werden.

Source Quench Nachrichten im ICMP verringern die Sendeleistung von Datagrammen, um eine Überlastung bei dem Empfänger zu vermeiden. Falsch erzeugte Nachrichten können hier die Übertragungsraten drücken.

Redirect Nachrichten können z.B. dadurch mißbraucht werden, um den Datenverkehr über einen bestimmten unsicheren Rechner umzuleiten.

ICMP Echo Requests führen (führten) bei einer Überdimensionierung bei einigen BS zum Systemstillstand, aufgrund eines Programmierfehlers in der Protokollsoftware. Diese Attacke bezeichnet man als Ping-of-Death (ping -t -1 65510 IP-Adresse)

ICMP-Tunneling
Die Schwachstelle befindet sich hier bei den Firewalls bzw. Paketfiltern, die bestimmen, ob ein ICMP-Paket aufgrund der im Paket enthaltenen IP-Adresse und der Nummer des Nachrichtentyps weitergeleitet werden darf oder nicht. So können Systemkommandos innerhalb eines zugelassenen ICMP-Requests die Firewall passieren. Befindet sich am Zielrechner ein modifizierter ICMP-Server können die Nachrichten ausgelesen und ausgeführt werden.

Bei IPSec spielt es keine Rolle, über welche Router der Verkehr läuft. Letztendlich geht man von einem nicht vertrauenswürdigen Rechner aus. Hier greift die Verschlüsselung und die Authentifikation um etwaige Manipulationen bzw. das Ausspionieren zu verhindern.

5.2.6 TCP-Sequenznummern-Attacke

TCP ordnet einzelne Pakete einer Verbindung aufgrund der Sequenznummern zu. So können Daten in einer bestehenden Verbindung manipuliert werden, indem derselbe Kopfsatz jedoch mit

geänderten Nutzdaten gesendet wird, indem der Angreifer ein ACK ISN mit der benötigten ISN sendet.
Die initiale Sequenznummer (Initial Sequence Number ISN) wird z.T. durch ein einfaches bzw. bekanntes Prinzip ermittelt, so daß sie vorhersagbar ist. Nach RFC 793 wird z.B. ein interner Zähler alle 4ms um eins erhöht. Bei BSD Unix wird alle 0.5s und bei jeder neuen Verbindung zu der bei Systemstart auf den Wert 1 gesetzten ISN die Zahl 64000 addiert.

IPSec unterbindet diesen Angriff ebenfalls durch den Einsatz von AHs. Kein Angreifer kann gültige AHs erzeugen ohne den geheimen Schlüssel zu kennen.

5.2.7 Abstreiten die sendende Person zu sein

Eine Person streitet ab, bestimmte Nachrichten gesendet zu haben. Wird über das Internet z.B. ein Auftrag an eine Firma erteilt 100 PC im Wert von je 2000 DM an Person X zu liefern, so kann Person X beim Anliefern der PCs diesen Auftrag abstreiten.

IPSec kann zur Authentifikation eingesetzt werden. Jedoch kann IPSec nur einen Rechner im Internet authentifiziern. Man kann sich also sicher sein, das diese Nachricht von einem vertrauenswürdigen Rechner gesendet worden ist. Man weiß jedoch nicht, welche Person die Nachricht letztendlich gesendet hat.

5.3 Was kann IPSec nicht?

Es werden nun einige Punkte aufgezählt, die nicht in den Sicherungsbereich von IPSec fallen.

- IPSec ist eingeschränkt auf den Transport von Internet-Protokollen.
- IPSec bietet keinen Schutz vor Viren oder anderen Programmen die z.B. einzelne Tastendrucke aufzeichnen. (Kein Schutz vor Trojansichen Pferden, ...)
- IPSec bietet keinen Schutz, falls jemand Zugang zum geheimen Schlüssel bekommen hat. Die Sicherheit die IPSec bietet hängt immer von dem Schlüssel ab. Liegt der Schlüssel z.B. auf einem Rechner der von einem Hacker geknackt worden ist, so kann IPSec keinen Schutz mehr bieten.
  Nicht außer Acht gelassen werden sollte, daß auch verschlüsselte Netzwerk-Pakete aufgezeichnet werden können. Dies hat zur Folge, daß die Schlüssel bis hin zu ihrer Vernichtung mit besonderer Sorgfalt verwaltet werden sollten. Falls der Schlüssel auch erst Wochen oder Monate später in falsche Hände gelangt, könnten die aufgezeichneten Netzwerk-Pakete entschlüsselt werden.
  Was Aufzeichnungen betrifft, kann man auch davon ausgehen, daß evtl. die Gefahr besteht eine heute noch gut verschlüsselte Nachricht in der Zukunft entschlüsseln zu können. Wichtig in diesem Zusammenhang ist die Schlüssellänge, die je nach Risikoeinschätzung ausreichend lang sein sollte.
- IPSec bietet z.T. keinen umfassenden Schutz vor Schnüfflern.
  Bei der Nutzung von IPSec zur Sicherung einer Kommunikation sind für Außenstehende mehr Komponenten des Netzverkehrs sichtbar als z.B. bei der verbindungsorientierten Verschlüsselung. Je nachdem, wie IPSec konfiguriert wurde, können Außenstehende feststellen, welche Rechner an Ihren Standorten miteinander kommunizieren. Zumindest kann man den Netzverkehr zwischen den Standorten ablesen. Ungeschützt sind Informationen über Paketgröße, Anzahl der Verbindungen und Daten in Protokollfeldern, die zu Netzwerkverkehrsanalysen ausgewertet werden.
- Cut-and-Paste und Rewrite Angriffe
  IPSec kann je nach Konfiguration nicht vor Cut-and-Paste Angriffen schützen. Wird IPSec eingesetzt um ein VPN einzurichten, ohne den AH-Header zu verwenden, so kann der

Netzverkehr aufgezeichnet werden und an eine dritte Person (Insider) innerhalb des vertrauenswürdigen Standortes gesendet werden. Diese dritte Person kann somit diese Nachrichten in Klartext empfangen, da die Nachricht vorher durch den IPSec-Proxy entschlüsselt wurde. Wird die Zieladresse durch den ESP-Tunnelmodus verborgen, so können immer noch Rewrite-Angriffe gestartet werden. Wenn ein Angreifer ein im ESP-Tunnelmodus verschlüsseltes Paket zwischen Person X und Person Y abfängt, so kann er das abgefangene Paket an ein gesendetes Paket von Person X an den Insider anhängen, um ein neues Paket zu erzeugen. Dieses Verfahren enthüllt dabei einige Daten, selbst wenn es bei DES CBC angewendet wird. Schutz bietet hier jeweils die Verwendung vom Authentication Header (AH).

- Replay Attacken
  Sequenznummern werden zum Schutz von Replay Attacken eingesetzt. Jedoch kann man hier nicht von einem kompletten Schutz von Replay-Attacken sprechen, es handelt sich hierbei vielmehr um eine Erschwerung des Replay Angriffs. Ein kompletter Schutz ist nicht möglich aufgrund der Notwendigkeit, mit Pakete die verlorengegangen oder dupliziert worden sind, umgehen zu müssen.

- IPSec schützt nicht gegen Angriffe, die auf Fehler in diversen Server-Implementationen zurückzuführen sind.
  IPSec kann nicht die Aufgabe eines Firewalls übernehmen. Durch IPSec erhält ein Angreifer unter Umständen eine abgesicherte Verbindung zu einem unsicheren Server. Erst durch den Einsatz eines Firewalls wird es möglich, Zugriffe nur auf Server im LAN zuzulassen, bei denen eine sichere Implementation installiert ist.

5.4 Richtlinien zur Verbesserung der IPSec-Sicherheit

IPSec kann für verschieden Zwecke konfiguriert werden. Um den bestmöglichen Schutz zu haben sollten folgende Punkte beachtet werden.

1. Getrennte Netzverbindungen für Klar- und Chiffretext.
   Die sicherste Methode ist alle Nachrichten ohne IPSec-Header abzuweisen.

2. Länge des Chiffrierschlüssels.
   Es muß ein Schlüssel mit ausreichender Schlüssellänge benützt werden.

3. Kryptographische Algorithmen.
   Für Anwendungen von IPSec mit mittlerem Risiko sind z.B. der DES-Algorithmus im CBC-Modus und der MD5-Hashalgorithmus geeignet.

4. Anwendung des AH auf den ESP.
   Nur mit beiden Headern können die vorher aufgezeigten Angriffe abgewehrt werden.

5. Schutz des Verschlüsselungsrouter und Schutz gegen unnötige Preisgabe von Schlüsseln.
   Sicherheitssysteme sollten sich stets in verschlossenen Räumen befinden und nur für authorisiertes Personal zugänglich sein.

6. Kompatibilitätstests.
   Kommt ein Router zum Einsatz, so sollte dieser einem Kompatibilitätstest unterzogen werden. (z.B. durch die S/WAN-Initiative oder dem SKIP Forum).

7. Automatische Einrichtung neuer Schlüssel.
   Chiffrierschlüssel sollten am besten in regelmäßigen Abständen oder bei jedem Verbindungsaufbau automatisch geändert werden.

8. Protokollierung von Ereignissen und Erkennung unerwünschter Vorfälle.
   Damit können gegebenenfalls Angriffe erkannt werden.

9. Unterstützung des IPSec-Tunnelmodus.
   Durch den Einsatz des Tunnelmodus wird der größte Teil an Adreßinformationen verborgen. Dieses Verfahren ist besonders vorteilhaft, wenn die Standorte mit nicht-registrierten Internet-Rechneradressen arbeiten.

6 IPSec mit Linux FreeS/WAN

Ausgewählt wurde die Software IPSec FreeS/WAN. IPSec FreeS/WAN hält sich an die vorgegebenen Standards, interoperiert somit mit den verschiedensten System und wird gut gepflegt. Die Software wird ständig weiterentwickelt und gut dokumentiert.

6.1 Free Secure Wide Area Network

Linux FreeS/WAN ist eine Implementierung von IPSec und IKE für Linux. Implementiert werden die von der IPSEC Working Group entwickelten Standards der IETF. Linux FreeS/WAN setzt RSA Data Security's S/WAN ein.

Zur Realisierung einer IPSec unterstützten Umgebung ist unter anderem Linux FreeS/WAN geeignet. FreeS/WAN ist Freeware und unterliegt keinen Exportbeschränkungen der US-Regierung. Zudem ist die Konfiguration und Installation gut beschrieben und man erhält innerhalb der Mailing Liste zügig Antworten. Der Quellcode sowie weiter Informationen findet man auf der Homepage unter folgender URL:

`http://www.xs4all.nl/~freeswan/`

Das aktuelle Entwicklungs- und Zielsystem ist RedHat5.2 mit einem 2.0.36 Kernel. Die aktuelle Distribution benötigt ca. 5MB und in compilierter Form nocheinmal 5MB. Bei anderen Distributionen werden i.a. keine Anfragen beantwortet, jedoch ist ein Erfahrungsaustausch auch anderer Distributionen erwünscht. (Der Entwicklungsschwerpunkt liegt hier auf der vollständigen und fehlerfreien Version einer Distribution und weniger auf Kompatibilität mehrerer Distributionen.)

In der Testumgebung wurde die SUSE Distribution (http://www.suse.de) verwendet . Besonderheiten bei der Installation der SUSE Distribution werden speziell angegeben, um so auch einen Eindruck zu gewinnen, inwieweit FreeS/WAN IPSec auch bei anderen Distributionen einsetzbar ist. Besonderheiten bzgl. weiteren Distributionen kann man auch der Homepage bzw. der Installationsanleitung entnehmen.

6.2 Installation von IPSec

Zuerst benötigt man die neueste Version (Latest Release) von FreeS/WAN, die man z.B. auf der Homepage finden kann. Hier erhält man ein tar-File das die Sourcen, Dokumentation, Patches usw. für Linux (momentan für die 2.0.3X Kernels) enthält. Es gibt auch noch die Möglichkeit, sich den neuesten Snapshot herunter zu laden, jedoch ist dieser nur für Experimentelle Zwecke geeignet. Diese Snapshots werden für die momentan aktuelle Entwicklung eingesetzt und können sich auch innerhalb Stunden ändern. Will man einen solche Version testen, was natürlich erwünscht ist um etwaige Fehler zu erkennen, ist es sinnvoll sich in der Mailingliste anzumelden.

Momentan aktuelle Version: **freeswan-1.00.tar.gz**

Installation von IPSec FreeS/WAN:

1. Linux mit Kernel 2.0.X installieren (Momentan noch keine 2.2.X Unterstützung)

2. Auspacken des tar-Archives.

   ```
   > tar xvfz freeswan-1.00.tar.gz
   ```

3. README Dateien lesen (um sich eine Überblick zu verschaffen)
   Index ../doc/index.html
   Wo ist Was ../doc/roadmap.html (What's Where in Linux FreeS/WAN)
   Setup Prozedur ../doc/setup.html (Installationsschritte)

4. Aufbau einer funktionsfähigen Umgebung ohne IPSec.
   (Als Beispielnetzwerk soll später eine Verbindung mit einem geschützten Tunnel aufgebaut werden).

5. Vor der Installation von IPSec sollte der Kernel übersetzt und dann auf seine Funktionsfähigkeit hin überprüft werden.

6. Installation von IPSec. Dabei erfolgt die Installation je nach gewünschte Funktion entweder auf allen beteiligten Rechnern oder nur zum Beispiel auf 2 Gateways, die eine sichere Verbindung aufbauen sollen.

6.2.1 Struktur der Testumgebung

Aufgabe: Zwei Netzwerke sollen sicher über ein drittes Netzwerk, dem nicht getraut werden kann kommunizieren.

Die Testumgebung in der IPSec installiert werden soll besteht aus drei Netzen, wovon ein Subnetz ein nicht vertrauenswürdigen Standort wie z.B. das Internet darstellt. Die anderen zwei Subnetze stellen die kommunizierenden Clients dar, die sicher über das Internet kommunizieren wollen. Es werden vier Rechner vernetzt und geeignet konfiguriert. Zwei Rechner werden als Gateway eingesetzt, wobei jeweils einer stellvertretend für ein Subnetz steht.

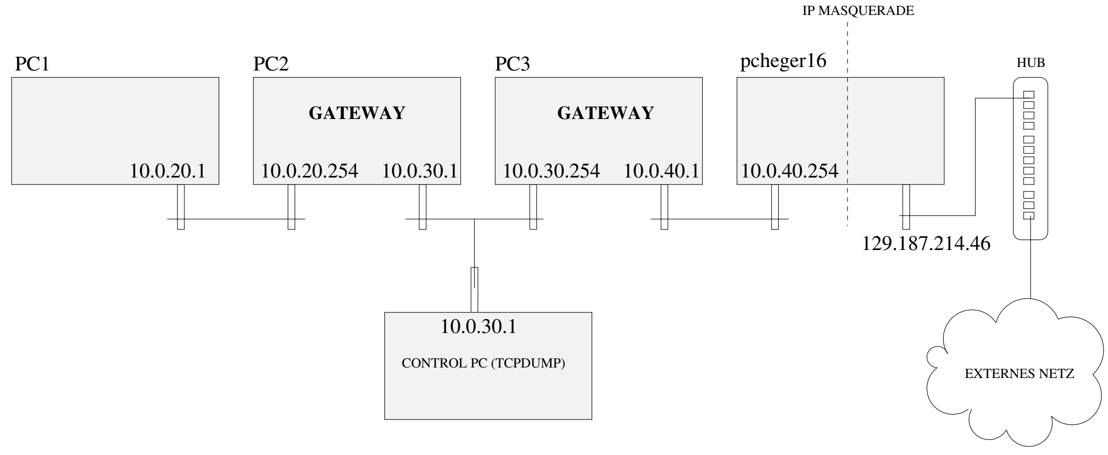

Abbildung 14: Versuchsaufbau

Die Clients besitzen routebare IP-Adressen (lassen sich also jeweils durch ein *ping* ansprechen). Eine andere Möglichkeit wäre der Einsatz von IP Masquerading (hier kann der jeweilige maskierte Client natürlich nicht mehr angepingt werden, sondern nur noch der Gateway).

Im Allgemeinen ist vor der Installation von IPSec die Umgebung geeignet zu testen, damit Fehler aufgrund einer Fehlkonfiguration ausgeschlossen werden können. Insbesondere sollte noch ohne IPSec eine Kernelübersetzung erfolgreich abgeschlossen werden. Die Software, die mit IPSec gesichert werden soll, sollte auch fehlerfrei arbeiten.

Um nun eine sichere Kommunikation über den nicht vertrauenswürdigen Standort zu erreichen, werden zwischen den Standorten Gateways mit IPSec installiert. Nachrichten zwischen den Gateways werden somit verschlüsselt. Das Resultat ist ein VPN.

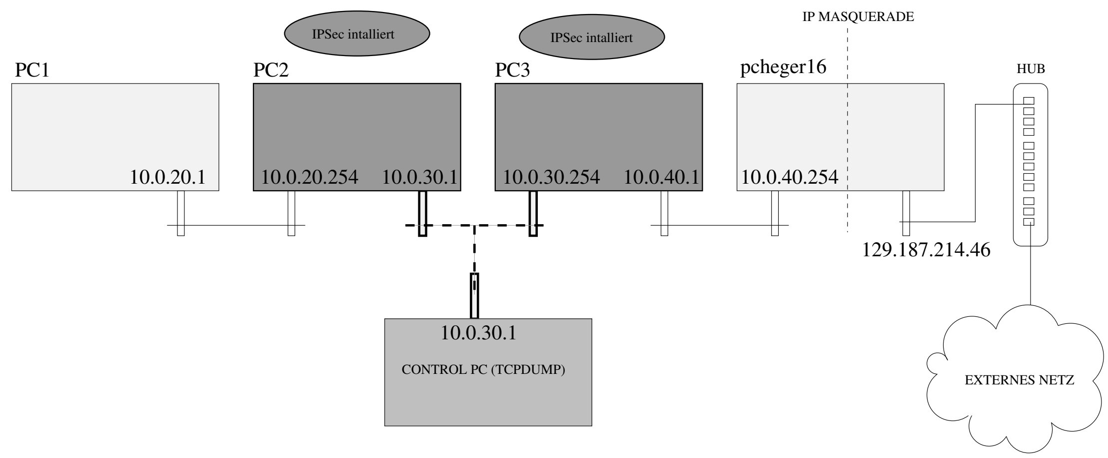

Abbildung 15: Versuchsaufbau

6.2.2 Installation von IPSec auf den Gateways

Um IPSec auf einem Gateway zu installieren, ruft man im Hauptverzeichnis von FreeS/WAN (z.B. unter /usr/src/freeswan-1.00, falls man das tar-Verzeichniss unter /usr/src entpackt hat) **einen** der drei folgenden Befehle für die Kernelkonfiguration auf.

```
make menuconfig (f"ur menuconfig)
make xgo (f"ur xconfig)
make ogo (f"ur config)
```

Die benötigten Optionen (IP-forwarding/gatewaying, IP-tunneling und Kernel/User network link driver) sind in der Regel automatisch aktiviert, und es steht nun eine zusätzliche Option für IPSec zur Verfügung, die ebenfalls schon aktiviert ist.

```
[*]IP: forwarding/gatewaying
[*]IP: tunneling
[*]Kernel/User network link driver

[*]IP Security Protocol (FreeS/WAN IPSEC)
--- IPSec options (FreeS/WAN)
[*]    IPSEC: IP-in-IP encapsulation
[ ]    IPSEC: Enable Insecure algorithms (strongly discouraged)
[*]    IPSEC: Enable ICMP PMTU messages
[*]    IPSEC: Authentication Header
[*]       HMAC-MD5 authentication algorithm
[*]       HMAC-SHA1 authentication algorithm
[*]    IPSEC: Encapsulating Security Payload
[*]       3DES encryption algorithm
[*]    IPSEC Debugging Option
```

Sind alle Einstellungen vorgenommen, und ist die Konfiguration abgeschlossen, wird der Kernel automatisch übersetzt. Dies kann je nach Prozessor Minuten bis mehrere Stunden dauern. Dabei sollte am Ende einer erfolgreichen Übersetzung folgende Zeile zu lesen sein:

```
make[2]: Leaving directory .....
make[1]: Leaving directory .....
utils/errcheck out.kbuild
```

Nach erfolgreicher Übersetzung sollte man das System mit dem neuen Kernel neu starten. (Siehe Beschreibung der jeweiligen Linux Distribution). Bei einem Neustart erkennt man nun anhand neuer Einträge das IPSec installiert worden ist. Es erscheint auch eine Fehlermeldung, die auf noch nicht korrekte Einstellung des IPSec Interfaces beruht. (Das IPSec Interface ist bei einer neuen Installation defaultmäßig auf dev999 konfiguriert). Im folgenden Abschnitt wird auf die nötigen Konfigurationen eingegangen.

Besonderheiten bei der SUSE Distribution:
Bei der SUSE Distribution funktionierte nur der **make ogo** Aufruf korrekt.
Desweiteren muß man die *Warnings* Einträge in der Datei
/usr/src/linux/include/linux/types.h
auskommentieren, da ansonsten die Kernelübersetzung abbricht. (Auskommentieren der Zeilen 3 bis 14).
Die Kerneloptionen sind nochmals zu überprüfen. Es wurden einige Optionen nicht wie versprochen automatisch aktiviert. ("[ ] IPSec" wird z.B. nicht aktiviert).
Nach erfolgreicher Übersetzung befindet sich der neue Kernel wie gewohnt unter /usr/src/linux/arch/i386/boot/zImage.

6.2.3 Konfiguration

Nach erfolgreicher Implementierung sollten die folgenden Dateien existieren und geeignet konfiguriert werden.
Diese Dateien enthalten sicherheitskritische Inhalte, die dementsprechend vorsichtig gehandhabt werden müssen. Kann eine unauthorisierte Person die Inhalte dieser Dateien einsehen, so besteht keinerlei Schutz mehr.

```
/etc/ipsec.conf    Datei zur Konfiguration von IPSec-Verbindungen
/etc/ipsec.secrets beinhaltet ein Geheimnis zur Authentifikation
```

Nach der Installation stehen Manpages zur Verfügung, die eine genaue Beschreibung der Befehle bzw. der Dateien liefert. Unter anderem sind folgende Manpages hilfreich für die Konfiguration:

```
man 5 ipsec.conf
man 8 ipsec-manual
man 8 ipsec-auto
```

Die Konfigurationsdatei /etc/ipsec.conf Innerhalb der Konfigurationsdatei ist eine Beispielkonfiguration enthalten, wobei man sich am besten anhand des Beispiels die einzelnen Bedeutungen klar machen kann.

In der Basiskonfiguration gibt man das IPSec Interface an. Desweiteren kann man noch Einstellungen vornehmen ob z.B. Debugging-Informationen mitgeloggt werden sollen. Dies ist für eine spätere Fehlersuche sinnvoll, jedoch für den regulären Gebrauch nicht anzuraten, da z.B. bei der Option "all" sehr viel Text anfällt. Zusätzlich können hier noch Einstellungen in Bezug auf automatischer Verbindungsaufnahme beim Start und PLUTO-Optionen eingestellt werden (Optionale Parameter sind ohne Anführungszeichen dargestellt).

Mögliche Einträge in der CONFIG SECTIONS:

interfaces
das Interface das IPSec verwenden soll.
forwardcontrol
soll nach dem IPSec-Start setup IP forwarding gestartet und vor dem Ende gestoppt werden? ([default] no, yes)
syslog
syslog-Einstellungen für startup/shutdown log messages. ([default] daemon.error)
klipsdebug
KLIPS Debugging Output. ([default] none, all)
plutodebug
Pluto Debugging Output. ([default] none, all)
dumpdir
in welchem Directory sollen dump cores erlaubt werden?
manualstart
welche Manuelle-Verbindung soll beim Start gestartet werden?
pluto
soll Pluto gestartet werden? (no, yes)
plutoload
welche Verbindung soll beim Start in die Pluto DB geladen werden?
plutostart
welche Verbindung soll beim Start initialisiert werden?
plutowait
soll beim Start auf jede plutostart-Initialisierung gewartet werden? ([default] yes, no)

Beispiel für den config-Abschnitt:

```
# basic configuration
config setup
        # virtuelles und physikalische Interface f"ur IPSEC
        interfaces="ipsec0=eth1"
        # IP forwarding beim Start an und beim Stop ausschalten
        forwardcontrol=no
        # KLIPS debugging text (all f"ur viele Ausgaben)
        klipsdebug=all
        # Pluto debugging text
        plutodebug=all
        # manually-keyed connections to set up at startup
        manualstart=
        # connections to load into Pluto's internal database at startup
        plutoload=
        # connections for Pluto to try to negotiate at startup
        plutostart=
        # should Pluto wait for each negotiation to finish before proceeding?
        plutowait=yes
```

In einem weiteren Abschnitt in der ipsec.conf Datei werden die einzelnen Optionen einer Verbindung festgelegt. Jeder Verbindung muß ein Name zugeordnet werden (z.B. conn pc2-pc3). Dann muß der Typ der Verbindung eingetragen werden.
Die Rechner der Verbindung, die zum nicht vertrauenswürdigen Netz führen, werden durch ihre IP-Adresse eingetragen. Hierbei ist ein Rechner der linke und der andere der rechte (left, right). Jetzt müssen noch die Netze, die erreicht werden sollen, eingetragen werden (rightsubnet, leftsubnet). Zuletzt bestimmt man noch die einzusetzenden Verschlüsselungs- und Authentifikationsalgorithmen.

Relevante Einträge für die automatische und die manuelle Schlüsselverteilung:

type
bezeichnet den Typ der Verbindung (tunnel, transport)
auto
welche Operation automatisch beim Start von IPSec ausgeführt werden soll (add, start, ignore)
left
gibt die IP-Adresse des linken Partners an (aaa.bbb.ccc.ddd)
leftsubnet
hier steht das private Subnetz des linken Partners (network/netmask)
leftnexthop
next-hop Gateway IP-Adresse für den linken Partner ([default] right)
leftupdown
welches updown Skript ausgeführt werden soll, wenn der Status der Verbindung sich ändert ([default] ipsec_updown).
leftfirewall
setzt der linke Partner Forwarding-Firewalling (beinhaltet Masquerading) ein? (yes, no)

Relevante Einträge für die automatische Schlüsselverteilung:

keyexchange
gibt die Methode des Schlüsselaustausches an (ike)
auth
Authentifikation mittels ESP-Verschlüsselung oder seperat mithilfe des AH-Protokolls (esp, ah)
authby
wie sich die Sicherheits-Gateways authentifizieren sollen. Möglich ist hier *shared secrets* oder *rsasig* für eine RSA digital signatures ([default] secret, rsasig)
leftid
wie der linke Partner zur Authentifikation identifiziert werden soll. ([default] left, aaa.bbb.ccc.ddd)
leftrsasigkey
der öffentliche Schlüssel für RSA signature authentication (im RFC 2537 Format)
pfs
ob Perfect Forward Secrecy für die Verbindung erwünscht ist. Mit PFS ist bei einem Verlust des Geheimnises die Sicherheit der Verbindung nicht gefährdet. (yes, no)
keylife
wie lange die SAs und die Schlüssel verwendet werden sollen. ([default] 8.0h, maximum 24h)
rekeymargin
wie lange Pluto versuchen soll, eine (gewechselte) SAs Verbindung aufzubauen, bevor abgebrochen werden soll. ([default] 9m, (s. keylife))
keyingtries
wieviele Versuche gemacht werden sollen, um eine Verbindung aufzubauen. ([default] 3, (ganze Zahl))
ikelifetime
wie lange die Verbindung zum anderen Key-Management Deamon höchstens bestehen soll, bevor sie neu aufgebaut werden soll.

Relevant Einträge für die manuelle Schlüsselverteilung:

spi
SPI Nummer für die Verbindung. (0xhex)
spibas
Basis Nummer für SPIs. (0xhex0)
esp
eingesetzter ESP-Algorithmus. ([default] (not to use), 3des-md5-96)
espenchkey
ESP Encryption Key
espauthkey
ESP Authentication Key
espreplay_window
ESP-Replay-Window-Einstellungen. Nur relevant beim Einsatz von ESP Authentication. (0,..,64)
leftespspi
zu benützende SPI für die ESP SA
ah
AH Authentication Algorithmus. ([default] (not to use), hmac-md5-96)
ahkey
AH Authentication Key.
leftahspi
Security Parameter Index der AH-Security-Assosiation.

Beispiel für den conn-Abschnitt:

```
# connection specifications
# Beispiel f"ur einen  Tunnel (mit manuellen oder automatischen Schl"ussel)
# ESP f"ur Verschl"usselung und Authentifikation

conn pc2-pc3
        type=tunnel
        # left security gateway (public-network address)
        left=10.0.30.254
        # next hop to reach right
        # subnet behind left (omit if left end of the tunnel is just the s.g.)
        leftsubnet=10.0.40.0/24
        # right s.g., subnet behind it, and next hop to reach left
        right=10.0.30.1
        rightsubnet=10.0.20.0/24
        # (manual) base for SPI numbering; must end in 0
        spibase=0x200
        # (manual) encryption/authentication algorithm and parameters to it
        esp=3des-md5-96
        espenckey=0x32c03fb3_0f5fddc7_208f6baa_bee3fde9_ec34b1f7_bc1cb9dc
        espauthkey=0xd0fa0fad_cb3ea576_7879bc69_1b4aad45
        # (auto) key-exchange type
        keyexchange=ike
        # (auto) key lifetime (before automatic rekeying)
        keylife=8h
```

Die jeweils kommunizierenden Rechner müssen ein und dieselbe “ipsec.conf”-Datei besitzen. Der einzige Unterschied der Dateien kann der Interface-Eintrag sein. Um Fehler zu vermeiden, kopiert man am besten die Konfigurationsdatei und überprüft dann noch den Interface-Eintrag. Man kann im allgemeinen natürlich mehrere Verbindungen angeben, wobei eine neue Verbindung nach dem gleichen Prinzip eingetragen werden muß.

Die Datei /etc/ipsec.secrets In der Datei ipsec.secrets befindet sich das für den Diffie-Hellman Schlüsselaustausch benötigte Geheimnis. Dieses Geheimnis dient der Authentifikation der Kommunikationspartner.

Ein Beispieleintrag könnte so aussehen:

```
# This file holds shared secrets which are currently the only inter-Pluto
# authentication mechanism.  See ipsec_pluto(8) manpage.  Each secret is
# (oversimplifying slightly) for one pair of negotiating hosts.

# The shared secrets are arbitrary character strings and should be both
# long and hard to guess.  A long hex number is supplied automatically as
# an example just because it's convenient to generate.

# Note that all secrets must now be enclosed in quotes, even if they have
# no white space inside them.

10.0.30.254 10.0.30.1   "0x69eb2480_2b6b12f1_0de6a1ec_ceb0d226_af73d046_6cdadbad
_c85896b4_75efe380"
```

Details können der Manpage ipsec.secrets(5) entnommen werden.

Im Installationsprozeß wurde ein solches Geheimnis zu Testzwecken generiert. Um jedoch einen neuen Schlüssel zu generieren, kann das ranbits-Utility (Manpage ranbits(8)) verwendet werden.

```
ipsec ranbits 256 > temp
```

Konfiguration des zweiten Gateways Nach erfolgreicher Konfiguration des ersten Gateway muß jetzt noch der zweite Gateway konfiguriert werden. Dazu müssen die folgenden Dateien des ersten Gateways auf den zweiten Gateway kopiert werden.

```
/etc/ipsec.conf
/etc/ipsec.secrets
```

Hier ist wieder darauf zu achten, daß der geheime Schlüssel übertragen wird und diese Übertragung somit sicher sein muß.

Die kopierten Dateien müssen dann nur noch geringfügig geändert werden. So muß der Interface-Eintrag auf Korrektheit überprüft werden. Er zeigt dem KLIPS IPSEC Code im Linux Kernel welches Interface genützt werden soll.

Mit dem ifconfig Kommando können die einzelnen Interfaces abgefragt und kontrolliert werden.

Werden bei der Konfiguration Fehler gemacht, so kann FreeS/WAN nicht ordnungsgemäß gestartet werden, wobei je nach Einstellung mehr oder weniger Fehlermeldungen erstellt werden (siehe Abschnitt Fehlersuche).

6.2.4 Testen der IPSec Installation

Um FreeS/WAN zu starten müssen beide Gateway-Rechner neu gebootet werden. Danach ist es sinnvoll, die /var/log/messages Datei nach etwaigen Problemen durchzusehen. Bei beiden Gateways sollten folgende Einträge im /proc/net Verzeichnis vorhanden sein:

```
ipsec_eroute
ipsec_spi
ipsec_spigrp
ipsec_spinew
ipsec_tncfg
ipsec_version
```

Desweiteren sollte das IPSec Interface vorhanden sein. Kontrolliert kann dies z.B. mit dem Befehl:

```
cat /proc/net/ipsec_tncfg
```

Hier erkennt man das Interface anhand dem ipsec0 Eintrag. Daten, die durch dieses Pseudo Device, mithilfe des eroute(8) Utilitys geroutet werden, werden vor dem Senden verschlüsselt.

Verbindung können mit den folgenden Befehlen aufgebaut werden:

```
ipsec manual --start name
ipsec auto --start name
```

Dabei steht manual für die manuelle Schlüsselwahl wobei der Schlüssel in der Datei /etc/ipsec.conf enthalten ist. Mit der Option auto kommt der Pluto-Daemon der das IKE-Protokoll implementiert zum Einsatz.
Im regulären Betrieb ist die automatische Wahl sicherer und demnach vorzuziehen.

Test mit einem manuellen Schlüssel Zuerst sollte die Konfiguration mit den manuellen Schlüsseleinträgen getestet werden. Man startet also IPSec mit dem entsprechendem ipsec-Kommando und kann danach mit dem Kommando ipsec look die Konfiguration einsehen. Vor dem Start erfolgt durch den ipsec look -Aufruf folgende Ausgabe:

```
> ipsec look

pc2 Sat Oct  9 21:03:20 MEST 1999
-------------------------
-------------------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.30.0       0.0.0.0         255.255.255.0   U      1500 0          0 eth1
10.0.40.0       10.0.30.254     255.255.255.0   UG     1500 0          0 ipsec0
```

Für eine manuelle Verbindung muß bei beiden Gateways ipsec gestartet werden. Folgend wird nun auf beiden Gateways ipsec gestartet und mithilfe der ipsec look -Option eine Ausgabe der Konfiguration angezeigt.

——— Start von ipsec auf PC2: ———

```
> ipsec manual --up pc2-pc3
> ipsec look

pc2 Sat Oct  9 20:13:37 MEST 1999
-------------------------
10.0.20.0/24 -> 10.0.40.0/24 => tun0x201@10.0.30.254 esp0x203@10.0.30.254
-------------------------
tun0x201@10.0.30.254 IPIP_proto:  dir=out  flags=0x0   10.0.30.1 -> 10.0.30.254
esp0x202@10.0.30.1 ESP_proto_3DES_encrypt_MD5_auth:  dir=in  win=0  seq=0
bit=0x00000000  flags=0x1
esp0x203@10.0.30.254 ESP_proto_3DES_encrypt_MD5_auth:  dir=out  win=0  seq=181
bit=0x00000000  flags=0x0
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.30.0       0.0.0.0         255.255.255.0   U      1500 0          0 eth1
10.0.40.0       10.0.30.254     255.255.255.0   UG     1500 0          0 ipsec0
```

——— Start von ipsec auf PC3: ———

```
> ipsec manual --up pc2-pc3
> ipsec look

pc3 Sat Oct  9 22:29:25 MEST 1999
-------------------------
10.0.40.0/24 -> 10.0.20.0/24 => tun0x200@10.0.30.1 esp0x202@10.0.30.1
-------------------------
tun0x200@10.0.30.1 IPIP_proto:  dir=out  flags=0x0   10.0.30.254 -> 10.0.30.1
esp0x202@10.0.30.1 ESP_proto_3DES_encrypt_MD5_auth:  dir=out  win=0  seq=0
bit=0x00000000  flags=0x0
esp0x203@10.0.30.254 ESP_proto_3DES_encrypt_MD5_auth:  dir=in  win=0  seq=0
bit=0x00000000  flags=0x1
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.30.0       0.0.0.0         255.255.255.0   U      1500 0          0 eth0
10.0.20.0       10.0.30.1       255.255.255.0   UG     1500 0          0 ipsec0
```

Dabei kann man folgendes erkennen:
PC2: tun0x201@10.0.30.254 (ein Tunnel von 10.0.30.1 nach 10.0.30.254)
PC3: tun0x200@10.0.30.1 (ein Tunnel von 10.0.30.254 nach 10.0.30.1)

PC2: esp0x202@10.0.30.1 (die eingehende Verbindung)
PC2: esp0x203@10.0.30.254 (die ausgehende Verbindung)
PC3: esp0x202@10.0.30.1 (die ausgehende Verbindung)
PC3: esp0x203@10.0.30.254 (die eingehende Verbindung)

Beide Verbindungen benützen dabei ESP mit einer 3DES Verschlüsselung und der MD5 Authentifikation.

Routing Informationen: Bei PC2 erkennt man, daß Pakete mit der Netzadresse 10.0.30.0 über das Interface eth1 und Pakete mit der Netzadresse 10.0.40.0 über das Interface ipsec0 versendet werden, wobei das ipsec0-Interface die IPSec-Funktionalität bietet.

Bei PC3 werden analog Pakete mit der Netzadresse 10.0.30.0 über das Interface eth0 und Pakete mit der Netzadresse 10.0.20.0 über das Interface ipsec0 versendet.

Das ipsec-Interface ist ein Pseudo-Interface. An das ipsec-Interface gesendete Pakete werden verschlüsselt und danach über das ethernet-Interface versendet. Bei PC3 werden z.B. Pakete mit der Netzadresse 10.0.20.0 an das ipsec0-Interface gesendet, woraufhin die Pakete verschlüsselt werden und danach zur IP-Adresse 10.0.30.1 gesendet werden.

Es wird also der gesamte Verkehr vom Netzwerk 10.0.20.0 zum Netzwerk 10.0.40.0 durch einen IPSec Tunnel zum Gateway 10.0.30.254 bzw. zum Gateway 10.0.30.1 geroutet.

Wurde der Befehl jeweils korrekt ausgeführt, sollte man die Verbindung mithilfe ping und dann mit den Programmen, die eingesetzt werden sollen, testen. Um jedoch auch sicher zu gehen, daß die Verbindung wirklich verschlüsselt wird, ist es am besten, die Pakete mittels eines Paketschnüfflers (Sniffer) zwischen den Gateways zu untersuchen.

Test mit Tcpdump Um zu kontrollieren, ob die Pakete auch wirklich verschlüsselt werden, also ob IPSec fehlerfrei funktioniert, sollte man mit einem Sniffer (z.B. Tcpdump) den Datenverkehr zwischen den Gateways aufzeichnen und analysieren.

Das nachfolgende Beispiel zeigt jeweils einen Ausschnitt einer aufgezeichneten Telnet-Sitzung zwischen PC1 und PC4.

```
pc1:> telnet pc4
```

Dabei wurden im Subnetz, quasi zwischen den zwei Gateways die IP Paket mithilfe von TCPDUMP(1) aufgezeichnet.

```
schnueffel:> tcpdump -w aufzeichnung.dat
```

Die erste Aufzeichnung zeigt die IP-Pakete ohne IPSec und der zweite Ausschnitt zeigt dann die IP-Paket mit laufendem IPSec im Hexadezimal Format (-x).

```
>tcpdump -x -r aufzeichnung.dat

...

03:04:42.587978 pcheger16.informatik.uni-muenchen.de.telnet > 10.0.20.1.1051: P 117:136(19) ack 94 win 32736 (DF)
                         4500 003b 4014 4000 3f06 aaaa 0a00 28fe
                         0a00 1401 0017 041b f1f1 0848 65fd 5ac5
                         5018 7fe0 c80e 0000 0d0a 7063 6865 6765
                         7231 3620 6c6f
03:04:42.597978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: . ack 136 win 16060 (DF)
                         4500 0028 0274 4000 3f06 e85d 0a00 1401
                         0a00 28fe 041b 0017 65fd 5ac5 f1f1 085b
                         5010 3ebc 60d8 0000 fffd 0100 3338

03:04:43.157978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 94:95(1) ack 136 win 16060 (DF)
                         4500 0029 0275 4000 3f06 e85b 0a00 1401
                         0a00 28fe 041b 0017 65fd 5ac5 f1f1 085b
```

```
                        5018 3ebc face 0000 66fd 0100 3338
                                            ^^----------------------------- f

03:04:43.427978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 95:96(1) ack 137 win 16060 (DF)
                        4500 0029 0277 4000 3f06 e859 0a00 1401
                        0a00 28fe 041b 0017 65fd 5ac6 f1f1 085c
                        5018 3ebc ffcc 0000 61fd 0100 3338
                                            ^^----------------------------- a

03:04:43.597978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 96:97(1) ack 138 win 16060 (DF)
                        4500 0029 0279 4000 3f06 e857 0a00 1401
                        0a00 28fe 041b 0017 65fd 5ac7 f1f1 085d
                        5018 3ebc fdca 0000 63fd 0100 3338
                                            ^^----------------------------- c

03:04:43.747978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 97:98(1) ack 139 win 16060 (DF)
                        4500 0029 027b 4000 3f06 e855 0a00 1401
                        0a00 28fe 041b 0017 65fd 5ac8 f1f1 085e
                        5018 3ebc f5c8 0000 6bfd 0100 3338
                                            ^^----------------------------- c

03:04:43.887978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 98:99(1) ack 140 win 16060 (DF)
                        4500 0029 027d 4000 3f06 e853 0a00 1401
                        0a00 28fe 041b 0017 65fd 5ac9 f1f1 085f
                        5018 3ebc fbc6 0000 65fd 0100 3338
                                            ^^----------------------------- e

03:04:43.997978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 99:100(1) ack 141 win 16060 (DF)
                        4500 0029 027f 4000 3f06 e851 0a00 1401
                        0a00 28fe 041b 0017 65fd 5aca f1f1 0860
                        5018 3ebc f4c4 0000 6cfd 0100 3338
                                            ^^----------------------------- l

03:04:44.077978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 100:101(1) ack 142 win 16060 (DF)
                        4500 0029 0281 4000 3f06 e84f 0a00 1401
                        0a00 28fe 041b 0017 65fd 5acb f1f1 0861
                        5018 3ebc f3c2 0000 6dfd 0100 3338
                                            ^^----------------------------- m


03:04:44.237978 10.0.20.1.1051 > pcheger16.informatik.uni-muenchen.de.telnet: P 101:102(1) ack 143 win 16060 (DF)
                        4500 0029 0283 4000 3f06 e84d 0a00 1401
                        0a00 28fe 041b 0017 65fd 5acc f1f1 0862
                        5018 3ebc ffc0 0000 61fd 0100 3338
                                            ^^----------------------------- a

...
```

Es können alle Eingaben abgelesen werden. In diesem Beispiel kann man das login *fackelma* erkennen. Da Paßwörter bei einer Telnet-Sitzung auch unverschlüsselt über das Netz gesendet werden, ist es mit dieser Methode auch möglich das Paßwort zu lesen.

Verwendet man IPSec ist der gesamte Datenverkehr verschlüsselt:

```
> tcpdump -x -r aufzeichnung-ipsec.dat

05:07:16.717978 pc2.columbien.de > pc3.columbien.de: ip-proto-50 76
                        4500 0060 007d 0000 4032 28f1 0a00 1e01
                        0a00 1efe 0000 0203 0000 000e a948 f85d
                        920a 8c82 0f87 439d bc38 a020 a5e0 51cd
                        bb01 f7cb 7d89
(30)
05:07:16.737978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 0149 0000 4032 2825 0a00 1efe
                        0a00 1e01 0000 0202 0000 000e 00df 78d4
                        36cb 472b 9d9a 846e 058b 83eb 3710 1c33
                        5ab3 06ac 7de6
(31)
05:07:17.727978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 92
                        4500 0070 014a 0000 4032 2814 0a00 1efe
                        0a00 1e01 0000 0202 0000 000f c56d 3fe2
                        52a3 f5d9 4102 b697 bc52 5ca4 3c22 0999
                        02c3 5638 0cdd
(34)
05:07:19.197978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 014b 0000 4032 2823 0a00 1efe
                        0a00 1e01 0000 0202 0000 0010 e1d9 96fe
                        c250 9a19 3067 9b4e 8ef8 cd74 7d13 4ef3
                        dc39 8745 6776

05:07:19.557978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 014c 0000 4032 2822 0a00 1efe
                        0a00 1e01 0000 0202 0000 0011 6ad9 4315
                        4083 524d 5e3b 3b4f 39cb 29e6 f890 b8fa
                        e50d 2ab9 39bd

05:07:19.847978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 014d 0000 4032 2821 0a00 1efe
                        0a00 1e01 0000 0202 0000 0012 473d bc3a
                        b015 d783 1423 3d21 5b7f c4c2 26f1 eb47
                        1a82 39a9 fef6

05:07:20.077978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 014e 0000 4032 2820 0a00 1efe
                        0a00 1e01 0000 0202 0000 0013 759f 1699
                        e8d2 9dca 7548 aa8b 04b9 bdec f4fa 862b
                        664f be82 ee7a

05:07:20.247978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                        4500 0060 014f 0000 4032 281f 0a00 1efe
                        0a00 1e01 0000 0202 0000 0014 0f19 6291
```

```
                         6cda 751d 8496 e0fa f760 b5c0 5131 a5ab
                         936c 1354 1366

05:07:20.547978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                         4500 0060 0150 0000 4032 281e 0a00 1efe
                         0a00 1e01 0000 0202 0000 0015 3741 bddd
                         deb1 403b 71ab 4219 3861 8bac 106e 01b3
                         4db9 84af ec84

05:07:20.847978 pc3.columbien.de > pc2.columbien.de: ip-proto-50 76
                         4500 0060 0151 0000 4032 281d 0a00 1efe
                         0a00 1e01 0000 0202 0000 0016 bedc e19d
                         65eb 4c86 956f fc98 9cf6 ae98 5c10 e961
                         e64f b6ce d1e5
```

Das login ist nicht mehr abzulesen! Um Fehler auszuschließen sollte bei den Tests ein zusätzlicher Rechner die Aufzeichnungen vornehmen.

Test mit einem automatischen Schlüssel Zuerst sollte man sicher sein, daß keine Verbindung aufgebaut ist. Ansonsten sollte man sie beenden. (z.B. ipsec manual –down pc2-pc3 zum beenden der manuellen Verbindung pc2-pc3)

Um zu testen, ob noch Verbindungen existieren, kann der Befehl ipsec look benützt werden.

Nun sollte der folgender Befehl bei beiden Gateways eingegeben werden:

`ipsec auto --add name`

und dann auf einem der beiden Gateways:

`ipsec auto --up name`

Das erste Kommando fügt die Information dieser Verbindung in der Pluto Datenbank hinzu. Das zweite Kommando aktiviert letztlich die Verbindung.

Um eine Verbindung zu beenden dient folgendes Kommando, das auf beiden Gateways ausgeführt werden muß:

`ipsec auto --down name`

6.2.5 Schlüsselgenerierung

Schlüssel können mithilfe des ranbits Utility generiert werden. Näheres erfährt man auch in der Manpage ranbits(8). So generiert z.B. der Befehl

`ipsec ranbits 192 (oder 128) > temp`

einen Schlüssel mit der gewünschten Länge.

192-bit Schlüssel für die 3DES Verschlüsselung
128-bit Schlüssel für die MD5 Authentifikation

Für SHA anstatt MD5 wird ein 160-bit Schüssel benötigt.

Zu beachten ist, daß die ganze Sicherheit, die IPSec bietet, auf diese Schlüssel aufbaut und sie somit besonders sorgfältig geschützt werden müssen. Es muß darauf geachtet werden, daß die erzeugten Dateien nicht gelesen bzw. so früh wie möglich wieder gelöscht werden. Eventuelle Aufzeichnungen des verschlüsselten Internetverkehrs könnten ja auch zu einem späteren Zeitpunkt mit einem geeigneten Schlüssel entschlüsselt werden.

Einzelheiten können auch den jeweiligen Manpages ipsec-ranbits(8) und random(4) entnommen werden.

6.2.6 Fehlersuche

Mit dem ipsec barf -Kommando wird eine sehr ausführliche Datei erstellt wobei hier alle wichtigen Optionen abgefragt und notiert werden. Hier nur ein kleiner Ausschnitt der Datei:

```
> ipsec barf

pc3
Sat Oct  9 22:38:15 MEST 1999
+ _________________________
+ cat /proc/net/ipsec_eroute
+ _________________________
+ cat /proc/net/ipsec_spi
+ _________________________
+ cat /proc/net/ipsec_spigrp
+ _________________________
+ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.30.0       0.0.0.0         255.255.255.0   U      1500 0          0 eth0
10.0.40.0       0.0.0.0         255.255.255.0   U      1500 0          0 eth1
10.0.20.0       10.0.30.1       255.255.255.0   UG     1500 0          0 ipsec0
127.0.0.0       0.0.0.0         255.0.0.0       U      3584 0          0 lo
0.0.0.0         10.0.40.254     0.0.0.0         UG     1500 0          0 eth1
+ _________________________
+ cat /proc/net/ipsec_tncfg
ipsec0 -> eth0
ipsec1 -> NULL
+ _________________________
+ ipsec auto --status
000
+ _________________________
+ ifconfig -a

...

tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:0.0.0.0  Mask:0.0.0.0
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0

tunl1     Link encap:IPIP Tunnel  HWaddr
          inet addr:0.0.0.0  Mask:0.0.0.0
          NOARP  MTU:1480  Metric:1
```

```
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0

ipsec0    Link encap:IPIP Tunnel  HWaddr
          inet addr:10.0.30.254  Bcast:10.0.30.255  Mask:255.255.255.0
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:181 errors:11 dropped:0 overruns:0 frame:0
          TX packets:141 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0

...


eth0      Link encap:Ethernet  HWaddr 00:00:E8:6A:2E:A2
          inet addr:10.0.30.254  Bcast:10.0.30.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2897 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2142 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          Interrupt:11 Base address:0x6100

eth1      Link encap:Ethernet  HWaddr 00:00:E8:54:48:5F
          inet addr:10.0.40.1  Bcast:10.0.40.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:440 errors:0 dropped:0 overruns:0 frame:0
          TX packets:475 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          Interrupt:10 Base address:0x6200

+ _________________________
+ hostname --fqdn
pc3.columbien.de
+ _________________________
+ hostname --ip-address
10.0.30.254
+ _________________________
+ ipsec _keycensor /etc/ipsec.conf
# /etc/ipsec.conf - FreeS/WAN IPSEC configuration file
```

Außerdem ist es bei einem Fehler immer sehr hilfreich in der ipsec.conf Datei möglichst viel Debugging-Informationen mitloggen zu lassen. Danach kann man dann in der Datei /var/log/messages zumindest Anhaltspunkte finden, die zu einem Fehler geführt haben.

6.3 Einzelheiten von FreeS/WAN

Noch die wichtigsten Kommandos und Begriffe von IPSec FreeS/WAN. Dabei wird noch näher auf Pluto, eine Implementierung des IKE-Protokolls eingegangen. Es erfolgt auch eine Aufzählung der noch nicht implementierten Standards und Methoden.

6.3.1 Das Kommando ipsec

Das ipsec Kommando erlaubt Kontrolle über die IPSec Aktivität.

```
Usage: ipsec command argument ...
where command is one of:
        auto
        barf
        eroute
        klipsdebug
        look
        manual
        pluto
        ranbits
        setup
        spi
        spigrp
        tncfg
        whack
```

6.3.2 KLIPS - KerneL IPSec Support

KLIPS sind die nötigen Modifikationen des Linux Kernels um IPSec zu unterstützen.

6.3.3 Der Pluto-Daemon

Pluto ist ein Daemon der das IKE-Protokoll implementiert. Er läuft als Daemon auf einem Netzwerk Knoten. Dieser Netzwerkknoten muß ein LINUX-System mit der IPSec-KLIPS Implementation sein. Pluto implementiert IKE noch nicht vollständig jedoch genügt dies, um mit anderen Instanzen von Pluto, und vielen anderen IKE-Implementationen zusammenzuarbeiten. Die Policy für die Sicherheitsbeziehungen wird noch hart in den Code von Pluto implementiert (s. spdb.c). Eventuell wird dies später in eine Security Policy Database verschoben.
ISAKMP SAs können Oakley groups MODP768 und MODP1024, 3DES, SHA1-96 und MD5-96 verwenden. Die IPSec SAs können 3DES, MD5-96 oder SHA1-96 verwenden.

Pluto erstellt automatisch untereinander geteilte Sicherheitsbeziehungen (SAs). Dies ist eine Arbeitserleichterung, da die aufwendige, manuelle verteilung entfällt.

Eine Security Association (SA) ist eine übereinkunft zweier Netzwerk Knoten, wie sie miteinander kommunizieren wollen (verschlüsselt, authentifiziert, eingekapselt, komprimiert).

Läuft auf einem Knoten IKE, kann dieser SAs mit anderen Netzwerkrechnern einrichten. Dies funktioniert aber nur, falls die anderen Knoten ebenfalls IKE implementiert haben. IKE muss also auf jedem Knoten, der ein Endpunkt einer IKE-Initiierten SA ist, laufen.

Eine IKE-Instanz kommuniziert mit einer anderen IKE-Instanz mit UDP IP Packeten, deshalb muß auch eine Route zwischen den jeweiligen Rechnern existieren.

Bei der Aushandlung einer SA können unterschiedliche Einstellungen vorgenommen werden. Diese Einstellungen entsprechen oft einem Kompromiß zwischen Sicherheit, Vertraulichkeit und

Effizienz. Diese sogenannten Policys Issues der IKE-Instanz werden vom System Administrator spezifiziert.

IKE arbeitet dabei in zwei Phasen. Die erste Phase baut eine ISAKMP SA auf. ISAKMP SAs werden von IKEs zur sicheren Kommunikation untereinander benötigt. Danach werden von den IKEs die IPSec SAs, die dann von anderen Programmen benutzt werden können, aufgebaut. Ein wichtiger Punkt ist, daß die IKE-Instanzen sich untereinander authentifizieren können.

Pluto verwendet zur Authentification Shared Secrets oder RSA-Signaturen. In Zukunft sollen noch andere Techniken unterstützt werden. Pluto implementiert auch ISAKMP SAs selbst. Beim beenden von Pluto werden alls Security Asscociations beendet.

Pluto benötigt eine Datenbank mit Preshared Secrets und RSA private keys.

Ein Pluto-Daemon und ein anderer IKE-Daemon müssen sich authentifizieren, bevor eine Verbindung erfolgreich zustande kommen kann. Die Authentifikation geschieht durch ein Geheimnis (ipsec.secrets), daß manuell oder mithilfe der RSA Signatur ausgetauscht wurde. Es existieren auch andere Techniken, die jedoch nicht in Pluto implementiert worden sind. Gespeichert werden die preshared secret keys und die RSA private keys in der Datei /etc/ipsec.secrets.

Die default IKE-Port Nummer ist 500. Pluto muß dabei unter superuser laufen, um diesen Port nutzen zu können. Ist Pluto gestartet, wartet er auf Anfragen von whack. Whack dient zur Kommandoübergabe an Pluto. Whack verwendet dazu ein UNIX domain socket. Bei jeder Anfrage führt Pluto normalerweise ein fork aus. Dies ist eine gängige Technik, um mehren Anfragen antworten zu können.

Pluto verifiziert Identitäten, sucht Sicherheits-Policies aus und initiiert Schlüssel für die KLIPS Schicht.

6.3.4 Verschlüsselung bei FreeS/WAN

Die MD5 Implementation ist Eigentum von RSADSI (RSA Data Security Inc.). Hierbei handelt es sich nicht um GPL. Details sind unter */klips/net/ipsec/ipsec_md5c.c* zu finden. Ebenso die LIBDES library (libdes/COPYRIGHT). Der SHA-1 Code ist Puplic Domain

6.3.5 Key Management

Hier gibt es verschieden Möglichkeiten. Es sind allerdings nicht alle Möglichkeiten in FreeS/WAN implementiert. Momentan gibt es folgende Methoden:

Manual keying IPSec erlaubt das manuelle setzen von Schlüsseln. Dabei sind die Schlüssel mit der Verbindungsdefinition in /etc/ipsec.conf gespeichert. Manual keying ist sinnvoll um den KLIPS-Code (Kernel-IPSec) zu testen. Generell ist das automatic keying zu bevorzugen, da es sicherer ist.
Manuelle Schlüssel können nur manuell erneuert werden, dazu muß die IPSec Verbindung unterbrochen werden und auf der jeweiligen Seite muß der Administrator die notwendigen Einträge vornehmen. Dabei muß der Schlüssel auf einen sicheren Weg ausgetauscht werden. (z.B. mit PGP oder SSH)
(Sequenznummer: In Linux FreeS/WAN wird bei einer manuellen Schlüssel Verbindungen die Sequenznummer ignoriert. Bei automatischer Schlüsselwahl wird die Sequenznummer einmal überprüft.)

Automatic keying Hier bestimmt der **Pluto Daemon** den Schlüssel mithilfe des IPSec Key Exchange Protokolls (IKE).

Verbindungen werden automatisch periodisch mit neuen Schlüsseln versehen (rekeyed). Automatic keying ist sicherer, da der Schlüssel periodisch also jede Stunde bzw. nach einigen Minuten ausgetauscht werden kann, ohne die Verbindung zu unterbrechen. Hat ein Angreifer root Rechte erlangt, so reicht es beim manual keying aus die Datei /etc/ipsec.conf zu lesen um alle verschlüsselten Nachrichten entschlüsseln zu können. Beim automatic keying enthält die Datei /etc/ipsec.secrets keine Schlüssel sondern nur ein Geheimnis, um den authentifizierten Schlüsselaustausch zu bewerkstelligen. Diese Eigenschaft vom automatischen Schlüsselaustausch ist auch unter dem Namen Perfect Forward Secrecy (PFS) bekannt. Besitzt eine unberechtigte Person jedoch das Geheimnis, kann ein aktiver Angriff (man-in-the-middle Attack) gestartet werden. Auch dieses Geheimnis sollte somit geschützt sein.

Noch nicht implementierte Methoden in FreeS/WAN

- Unauthenticated key exchange
- internet default shared secret
- Key exchange using DNS
- Key exchange using PKI (Puplic Key Infrastructure)
- Photuris (alternative zu IKE und ISAKMP)
- SKIP (von Sun)

6.4 Entwicklungsgeschichte

Je eine Referenzimplementierung stammen von Cisco und dem National Institute of Standards and Technology (NIST). Aufgrund den US-Exportgesetzten sind diese jedoch nur innerhalb Nordamerikas erhältlich. Für andere Benutzer entstand das frei verfügbare Programm **Pluto** von Angelos D. Kermoytis, das im Rahmen des **FreeS/WAN-Pojekt** weiterentwickelt wird.
Einige VPN-Funktionen basieren bereits auf IPSec, wobei die Schlüssel allerdings noch vorwiegend manuell konfiguriert oder über Suns SKIP (Simpley Key Management over IP) bereitgestellt werden.
Desweiteren herrschen auch Einschränkungen aufgrund der US-Exportgesetze.

Implementierungen ausserhalb Nordamerikas:

1. FreeS/WAN-Pojekt ist Hauptprojekt für JIs IPSec von John Ioannidis und dessen Weiterentwicklung im FreeS/WAN-Projekt sowie die Entwicklung von einem IKE-Daemon Pluto (Erste Implementierung von IKE ausserhalb Nordamerikas).

2. Eine an die OpenBSD-Implementierung angelehnte Version von Petr Novak (verwendet IKE-Vorgänger Photuris für das Schlüsselmanagement). Anfang Mai 98 erschien eine erste FreeS/WAN Version 0.8 mit Verbesserungen und Überarbeitungen. Der Name wurde in KLIPS (Kernel Level IP Security geändert).

Da alle Entwicklungen auf dem IETF-standardisierten Verfahren basieren, erreicht man eine Kompatibilität zu allen, auch kommerziellen Produkten und somit auch eine Herstellerunabhängige Auswahl von Komponenten die zum Einsatz kommen.

6.5 Ziel von FreeS/WAN IPSec

Wunschvorstellung ist, daß jeder IPSec einsetzt und so eine sichere, verschlüsselte Kommunikation über das Internet stattfindet. Also ein reales privates Netzwerk.
Längerfristig ist der Einsatz von Secure DNS zur Authentifikation der Seiten geplant, so das ein Schlüssel mithilfe des Diffie-Hellman Protokoll erstellt werden kann.

7 Der IPSec-Einsatz

IPSec bietet Sicherheit im Internet, Anwahl des ISP, Firewalls und letztendlich auch im eigenen Netz. Dabei unterstützt IPSec Schutz bei End-zu-End Verbindungen als auch bei Segment-zu-Segment Verbindungen. Dies erlaubt dem Netzwerkverantwortlichen in jedem Pfad den jeweils nötigen Schutz anbieten zu können.
In diesem Kapitel werden einige Einsatzbeispiele vorgestellt. Wird IPSec eingesetzt so sollte man sich immer im klaren sein inwieweit IPSec schützen kann bzw. was man beachten sollte um diesen Schutz nicht zu gefährden. Oft spielt aber auch die Performance für den Einsatz von IPSec eine große Rolle. Interoperablilität von IPSec in einer heterogenen Umgebung ist noch ein weiterer Aspekt, der erläutert wird.

7.1 Einsatzziele

Grob gibt es die Einteilung von Segment-zu-Segment, End-zu-End, bzw. End-zu-Segment Verbindungen. Dabei kann dann noch jeweils zwischen dem Einsatz von AH, ESP oder beidem im Tunnel- oder im Transport-Mode entschieden werden. Der Einsatz von AH, ESP und den möglichen Mischformen erfüllt dabei verschiedene Anforderungen. Der AH gewährleistet die Authentifikation und Integrität von Datagrammen, der ESP sorgt für die Integrität und Verschlüsselung. Grundsätzlich gilt, daß durch eine Verschlüsselung das Mitlesen einer Nachricht verhindert werden soll. Zudem wird durch eine komplette Verschlüsselung auch sichergestellt, daß die Nachricht nicht verändert werden konnte und nur von einem bestimmten Absender stammen kann. Eine Authentisierung einer Nachricht durch eine Prüfsumme über die Daten, die mit einem nur dem Absender bekannten Wert verschlüsselt wird, stellt ein sicheres Verfahren zur Identifikation des Absenders dar, da nur er den geheimen Schlüssel kennen sollte. Das letzte Verfahren verzichtet dabei auf die völlige Verschlüsselung und ist deshalb auch wesentlich schneller.

7.1.1 Einsatzziele von AH,ESP,AH/ESP

Je nach Einsatz von AH, ESP, AH und ESP jeweils im Tunnel oder im Transport Modus werden unterschiedliche Ziele verfolgt.

- **AH-Transport Mode**
  - Zugangskontrolle und Verkehrsfilterung.
  - Nutzdaten dürfen von allen gelesen werden, jedoch soll die Nachricht im Originalzustand sein.
  - IP-Header bis auf variable Einträge und Nutzdaten dürfen nicht verändert werden. Der Absender soll identifiziert werden (shared key).
- **AH-Tunnel Mode**
  - Änderungen am IP-Header und an den Nutzdaten sollen erkannt werden.
- **AH-Tunnel Mode / ESP-Transport Mode**
  - Nutzung von privaten IP-Adressen.
  - Anbindung von externen Zulieferern an das Firmennetz:
    Zugriffskontrolle auf eindeutig abgegrenzte Bereiche im Netz des Herstellers. (Host A will nicht, daß Host B bzw. ein unauthorisierter Mitarbeiter der Firma X an sensitiven Daten gelangt. Umgekehrt möchte sich der Hersteller gegen Lauschangriffe aus dem Intranet seiner Partner absichern.
    Eine Lösung ist eine End-zu-End Verbindung, die man unter der Verwendung von AH im Tunnel-Mode und eingeschlossener ESP im Transport-Mode erreicht. Nur authentifizierte Pakete werden verarbeitet und (an höheren Schichten) weitergeleitet werden,

(dadurch verhindert man auch Denial-of-Service-Attacken). Diese Filterfunktion können durch eine End-zu-End Verschlüsselung und Authentifizierung des ESP im Intranet ergänzt werden. Die sensitive Daten werden dadurch vom Client im Intranet des Zulieferers bis zum Server beim Hersteller verschlüsselt übertragen. Unberechtigte Zugriffe durch vortäuschen von zulässigen Absenderadressen (Spoofing) kann die Authentifizierungsfunktion des ESPs abwehren. (Es können nur berechtigte und im Besitz des geheimen Schlüssels befindliche Partner mit den entsprechenden Servern Kontakt aufnehmen). Hierzu müssen allerdings alle Clients und Server IPSec unterstützen. Auch die Firewalls müssen mit den neuen Next Header-Kennzahlen in den IP-Paketen umgehen können, die nun auf die AH bzw. ESP verweisen.

- **ESP-Transport Mode**
  - Nutzdaten müssen verschlüsselt werden.
  - Vertraulichkeit auch innerhalb "sicherer" Standorte.
  - Verschlüsseln der Nutzdaten und des ESP-Trailers.
  - Erkennen von veränderten Nutzdaten.
- **ESP-Tunnel Mode**
  - Vertraulichkeit zwischen Standorten.
  - Geheimhaltung der internen Topologie der Standorte.
  - IP-Daten und Nutzdaten sollen verschlüsselt werden.
  - Authentfizierung des originalen IP-Header und der Nutzdaten.
  - VPN-Gateways an den Übergängen zum Internet sollen nur Pakete annehmen, die tatsächlich von einem im Extranet befindlichen Rechner stammen.

    ESP im Tunnel-Modus mit aktivierter Authentifizierung erfüllt die Anforderung ideal. Hier werden die Daten und die intern verwendeten IP-Adressen verschlüsselt über das Internet transportiert.
- **ESP-Tunnel Mode / AH-Transport Mode**
  - Anwendungen mit sehr hohen Sicherheitsanforderungen, die auch den äußeren IP-Header authentifiziert wissen wollen. (Biete sich als Alternative AH und ESP im Tunnel-Modus an. ESP-Authentifizierung alleine würde nur die unveränderlichen Daten im Original-Datagramm zuzüglich ESP-Header und Trailer absichern).
- **ESP-Transport Mode / AH-Transport Mode**
  - Intranet A soll z.B. nur Verkehr von Intranet B zulassen und dabei eine sichere End-zu-End Verbindung von einem Host aus dem Intranet A mit einem Host vom Intranet B aufbauen. Das Intranet soll vor unberechtigte Zugriffen aus dem Internet geschützt werden und außerdem noch innerhalb des Intranets eine gesicherte Kommunikation bieten.

    Durch Verwendung des AH wird Verkehr unerlaubter Hosts innerhalb des Intranets verhindert (secret shared key) und durch den ESP Transport Modus wird eine sicher End-zu-End Verbindung realisiert.

7.2 Einsatzbeispiele

Hier gibt es die Möglichkeit einer Segment-zu-Segment, einer End-zu-Segment oder einer End-zu-End Verbindung.

Segment-zu-Segment Verbindung IPSec kann eingesetzt werden, um den Datenverkehr zwischen zwei Standorten zu schützen. Dabei verhält sich diese Konfiguration aus Sicht der Netzbenutzer wie eine verbindungsorientierte Verschlüsselung. Internet Nachrichten werden dabei innerhalb des vertrauenswürdigen Standortes im Klartext übertragen, oder können mit anderen Verfahren geschützt sein.

Beim Einsatz von IPSec im Proxy-Modus (IPSec auf einem Proxy-Host (Firewall oder Router)) wird der Netzverkehr mehrerer Rechner durch den Proxy-Host geschleust, bevor er an nicht vertrauenswürdige Rechner weitergeleitet wird. Der Zielrechner verfügt selbst über IPSec-Unterstützung, oder die Nachrichten werden ebenfalls vorher durch einen IPSec-unterstützten Proxy-Host geleitet.
Hierfür muß nur bei wenigen Routern (min. 2) IPSec installiert und konfiguriert werden. Der Vorteil liegt an der Ersparnis von Konfigurationsaufwand. Die Komplexität der Schlüsselverwaltung darf i.a. nicht unterschätzt werden. In diesem Einsatzbeispiel ist sie im Vergleich sehr gering, da die Schlüssel nur den Proxy-Hosts zugeordnet werden müssen. Für IPSec-Proxies kommt der Transport-Mode und der Tunnel-Mode in Frage. Im Transport-Mode enthält jedes IP-Paket die Adresse des eigentlichen Ursprungs- und Zielrechners. Dazu muß das Proxy-Gerät die richtige Sicherheitsassoziation in einem Platzhaltermodus auf die Nachrichten anwenden, da eine Vielzahl von Endpunktrechnern über eine einzige Sicherheitsassoziation angesprochen werden. Im Tunnel-Mode werden die eigentlichen Rechneradressen im ESP-Header übertragen und die Sicherheitsassoziation wird eindeutig zwischen den einzelnen Proxy-Rechnern eingerichtet.

Ein weiteres Beispiel ist ein Standortzugang für VPN und öffentliche Dienste. Das Ziel besteht darin, sowohl VPN-Verbindungen als auch ungeschützte Verbindungen zu öffentlichen Internet-Standorten zu unterstützen. Um eine sichere Kommunikation zu ermöglichen werden Firewall-Produkte mit IPSec-Verschlüsselung eingesetzt. Dabei bieten IPSec-Firewalls Kontrollmechanismen durch Paketfilter, Circuit-Filter oder Anwendungs-Proxies.

End-zu-Segment Verbindung Es werden IPSec-Clients verwendet, um den Datenverkehr zwischen einem zentralen Standort und entfernten Rechnern zu schützen. Dies entspricht einem Standortzugang vom Client zum Server.

Ein IPSec-Client zum Beispiel ist ein Softwarepaket, das IPSec-Sicherheitsmaßnahmen auf Workstations oder Notebooks bietet. Die Sicherheitsmaßnahmen werden entsprechend der vom Workstation-Operator eingerichteten Sicherheitsassoziationen angewendet. Die Sicherheitsassoziationen mit dem zentralen Standort muß beim mobilen Einsatz in der Lage sein, mit verschiedenen IP-Adressen umzugehen, wobei diese nicht im voraus bekannt sind.

End-zu-End Verbindung Es ist auch möglich, wenn auch nicht gebräuchlich, auf jedem Rechner IPSec zu installieren und somit auch den ganzen Netzverkehr auf dem Local Area Network (LAN) zu schützen. Jedoch ist der Aufwand der Schlüsselverwaltung erheblich.

IPSec nur zur Authentifikationszwecken Wird IPSec nur zu Authentifikation eingesetzt, so ist der Overhead geringer als mit der zusätzlichen Verschlüsselung. IPSec nur mit dem Authentication Header ist sinnvoll:

- beim Einsatz zur Zugriffkontrolle
- für öffentliche Daten die jedoch im Originalzustand sein müssen
- in vertrauenswürdiger Umgebung, bei der Verschlüsselung nur einen unnötigen Overhead bedeutet
- wenn eine starke Verschlüsselung unterhalb dem IP-Protokoll vorhanden ist
- wenn eine starke Verschlüsselung oberhalb dem IP-Protokoll vorhanden ist

7.3 Performance

IPSec-Lösungen wie FreeS/WAN sind i.d.R. schlechter in der Performance gegenüber proprietären IPSec-Systemen eines Herstellers. Aus Performancegründen sind deshalb z.T. nur kommerzielle Produkte für bestimmte Aufgaben geeignet.

IPSec kann speziell konfiguriert werden. So kann man durch explizites Routing je nach Bedarf Verbindungen mit IPSec bzw. auch Verbindungen ohne IPSec aufbauen. Dadurch verhindert man unnötigen Overhead bei Verbindungen, die nicht explizit geschützt werden müssen. Nach Möglichkeit sollte man aber davon absehen, nur Sicherheitsrelevante Daten zu sichern, da ein potentieller Angreifer hier schon evtl. wichtige von unwichtigen Daten selektieren kann.

Die FreeS/WAN-IPSec-Implementation für Linux wurde von Robert J Gautier auf ihre Performance hin untersucht [Gaut 99]. Dabei wurden u.a. jeweils die Zeitdauer von einem ping-Kommando und einem ftp-Einsatz gemessen. Es wurde ein Durchsatz von 14 Prozent bis zu 85 Prozent im Vergleich zu nicht verschlüsselten Netzverkehr beobachtet.

```
14%: 486DX/50 mit einer 10Mbit/s Anbindung
85%: 486DX/50 mit einer 64kbit/s Anbindung
```

Der ping-Test im Einsatz mit einem 10Mbit/s twisted pair Ethernet, einem 486DX2/50, einem AMD K6/233 und einem Cyrix MII/333+ Prozessor ergab:

```
To/From           Mode   56   1000   2000   5000 Bytes
MII/333+, K6/233  Clear   0,4   2,3   4,0   9,2
                  IPSec   1,2   6,6  10,8  18,6
                          33%   35%   37%   49%

To/From             Mode   56   1000   2000   5000 Bytes
MII/333+, 486DX2/50 Clear  1,5   4,2   6,9   13,0
                    IPSec  5,3  22,6  41,6   93,5
                           28%  18%   17%    14%
```

Ein zusätzlicher Schutz bedeutet immer zusätzlichen Aufwand, den es einzuschätzen gilt. Ein End-zu-End Ansatz hat dabei einen sehr großen Anstieg des Aufwandes zur Folge. Allgemein benötigt eine Änderung in der Vermittlungsschicht einen längeren Zeitraum. Es muß eine Anpassung aller Endsysteme erfolgen und zudem müssen die einzelnen Verbindungen sorgfältig verwaltet werden.

7.4 Interoperabilität

Die Interoperabliltätsaspekte für den IPSec-Einsatz sind dann von Bedeutung, wenn verschiedene IPSec-Installationen verschiedener Hersteller untereinander eingesetzt werden sollen.
Durch die Standardisierung wird IPSec auch in heterogenem Umfeld einsetzbar, falls sich die Hersteller jeweils an diesen Standard gehalten haben. Setzt man Systeme eines bestimmten Herstellers ein, der sich nicht an den Standard hält, so können nur Systeme dieses Herstellers verwendet werden [HoRi 00].
FreeS/WAN wurde einigen Interoperabilitätstests unterzogen, wobei FreeS/WAN mit Triple DES IPSec Produkten von OpenBSD, PGP, Cisco, Raptor, Xedia und Secure Shell zusammen eingesetzt werden konnte [Harr 99] [JJB 00]. Desweiteren gibt es bei der International Computer Security Association ICSA in Reston interoperabilitäts Tests, wobei das Produkt jeweils Zertifiziert werden kann [Asso 00].

7.5 Kostenvorteile beim Einsatz von IPSec

Nach einer Studie von Sun Microsystems wurden die finanziellen Auswirkungen analysiert, wobei statt privater Netze auf IPSec basierende VPNs verwendet werden. Dabei wurden zwei Situationen untersucht: Ein firmeneigener Frame-Relay-Backbone und ein Netz aus Standleitungen zwischen den Filialen. Beim Frame Relay ergab sich durch Einsatz des VPNs eine Kosteneinsparung von 50 Prozent und bei privaten Standleitungen eine Einsparung von 23 Prozent.

8 Entwicklung von IPSec

Die IPSec Arbeitsgruppe der Internet Engineering Task Force (IETF) hat zunächst die eigentlichen Sicherheitsfunktionen für IP spezifiziert und im Anschluß daran ein Key-Management-Protokoll entwickelt.
Dabei hat die IP Security Arbeitsgruppe einen Standard mit der Bezeichnung Security Architecture for the Internet Protocol veröffentlicht (http://www.ietf.cnri.reston.va.us/html.charters/ipsec-charter.html). Dieser Internet-Draft ist sozusagen das Zentrum einer Reihe weiterer Dokumente welche die Steering Group der IETF als Proposed Standard anerkannt hat. Das Protokoll ist weder an bestimmten Produkte noch an Algorithmen gebunden. Die Grundidee dabei ist den Datenaustausch auch unterschiedlicher Hersteller aufgrund eines Standards zu ermöglichen.

Jeder Standard der IETF durchläuft der Reihe nach die Phasen Proposed Standard, Draft Standard, Standard oder Historic Standard. Ein Internet-Draft ist ein Arbeitspapier der IETF mit einer Gültigkeitsdauer von sechs Monaten. Nachfolgend werden die RFCs und Drafts vorgestellt [Kule 98][IBM 99].

8.1 RFCs und Drafts

- **Security Architecture for the IP**
  RFC 1825 (Proposed Standard) Aufnahme von Sicherheitsmechanismen in das IP (August 95).
  Internet-Draft konkretisiert RFC 1825. Anerkennung des Papiers als Proposed Standard von der Steering Group des IETF (August 98).
  Aktuell: RFC 2401

- **IP Authentication Header AH**
  RFC 1826 mit dem Status Proposed Standard beschreibt einen ergänzenden Abschnitt des IPv6, der die Echtheit der übrigen Header und Daten garantiert (August 95).
  Internet-Draft überarbeitet RFC 1826. Der Entwurf ist ein Proposed Standard (Juli 98).
  RFC2402 (November 98)

- **Encapsulating Security Payload ESP**
  RFC 1827 führt verschieden Internet Sicherheitsmechanismen in die Internet Protokolle IPv4 und IPv6 ein. Der ESP Abschnitt legt zum Beispiel den Verschlüsselungsalgorithmus für den Datenaustausch fest (August 95)
  Internet-Draft ersetzt RFC 1827. Das Dokument ist als Proposed Standard anerkannt worden (Juli 98).
  RFC 2406

- **Implementation of VPNs with IPSec**
  Internet-Draft, erörtert verschiedene VPN-Architekturen auf der Grundlage von IPSec (März 97).

- **ESP Triple DES Transform**
  Internet-Draft definiert das Interface, welches den Aufruf von Triple DES Transformationen durch das IP ermöglicht (Juli 97).

- **Internet Security Association and Key Management Protocol ISAKMP**
  Internet-Draft begründet das Konzept der Security Association (SA) und regelt den Umgang mit öffentlichen und privaten Schlüsseln (Juli 98).

- **PKI Requirements for IPSec**
  Internet-Draft nennt Anforderungen, die Public-Key- Infrastruktur-Protokolle erfüllen müssen, wenn sie für IPSec arbeiten (September 98).

- **LDAP Schema for Configuration and Administration of IPSec based VPNs**
  Internet-Draft, begründet die Struktur eines LDAP-Verzeichnisses, mit dem Administratoren VPNs verwalten können (Oktober 98).

- **weitere RFCs**
  RFC 1828 IP Authentication using Keyed MD5
  RFC 1829 ESP DES-CBC Transform
  RFC 2104 HMAC: Keyed-Hashing for Message Authentication
  RFC 2085 HMAC-MD5 IP Authentication with Replay Prevention
  RFC 2410 The NULL Encryption Algorithm and Its Use With IPsec
  RFC 2411 IP Security Document Roadmap
  RFC 2412 The OAKLEY Key Determination Protocol
  RFC 2451 The ESP CBC-Mode Cipher Algorithms
  RFC 2403 The Use of HMAC-MD5-96 within ESP and AH
  RFC 2404 The Use of HMAC-SHA-1-96 within ESP and AH
  RFC 2405 The ESP DES-CBC Cipher Algorithm With Explicit IV
  RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP
  RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP)
  RFC 2409 The Internet Key Exchange (IKE)

Einen aktuellen Überblick gibt es auf der Homepage der IETF .

8.2 IPSec-Implementierungen

IPSec wird von unterschiedlichen Herstellern unterstützt [Mart 98] [Tso 98] [Asso 00] .

IPSec wird unterstützt unter

- Windows 95/98/NT 4 (Dritthersteller)
- Windows NT 2000 angekündigt
- HP-UX: 10.20 [fraglich], 11 [ja]
- AIX 4.3.2
- OS/400 angekündigt von IBM
- OS/390 nur mit manueller Schlüsselverteilung
- Linux
- Solaris 7 im ISP-Server
- CISCO ab IOS 11.2

IBM Produkte in Router, Client- und Server-OS, AIX-Firewall

Radguard-Produkte NetCryptor (VPN-Encryption Device), PyroWall (Firewall), CryptoSystem (Firewall und VPN-Support), clPro (Encryption, Access Control)

Security Project at the TCM Laboratory Beispielimplementierung von IPSec der Helsinki University of Technology [TCM 99].

A Weitere Sicherheitsprotokolle

Die nachfolgenden Sicherheitsprotokolle wurden ebenfalls entwickelt um den Internet-Datenverkehr zu schützen.

A.1 IPSO - IP Security Option

Die IPSO besteht aus einer Reihe von Protokollen, die IP-Paket mit sensitivity labels versieht. Diese Labels werden mit speziellen vertrauenswürdigen Routern oder anderen Netzkomponenten verwendet. Dabei kann ausgeschlossen werden, daß vertrauenwürdige Daten zu unsicheren Standorten gelangen. Zu den IPSO-Protokollen zählen Commercial IPSO (CIPSO), Revised IPSO (RIPSO) und DNSIX, das in militärischen Anwendungen zum Einsatz kommt.

A.2 SDNS - Secure Data Network System

Das SDNS besteht aus einer Protokollfamilie, die zum kryptographischen Schutz des Netzverkehrs in verschiedenen Protokollschichten entwickelt wurde. So zum Beispiel das Message Security Protocol (MSP) auf der Anwendungsschicht, daß schließlich zu PEM wurde. SDNS enthält zudem Protokolle in der Vermittlungs- und Transportschicht. Das SP3 für Schicht 3 und das SP4 für Schicht 4. Dabei bietet das SP3 Protokoll Sicherheitsvorkehrungen, die denen der IPSec-Protokolle ähneln.

A.3 NLSP - Network Layer Security Protocol

Die ISO hat diese Variante des SP3-Protokolls in ihre OSI-Protokollsuite aufgenommen. Da NLSP nicht auf eine klar definierte Menge erforderlicher Dienste festgelegt ist, war es für Hersteller nicht gerade ermutigend, kompatible Produkte zu entwickeln.

A.4 PPTP - Point-to-Point Tunneling Protocol

Dieses Protokoll ist eine Erweiterung des Poin-to-Point Protocol (PPP). Es wird für Einwählverbindungen von seriellen Modems zu ISPs benützt. PPTP setzt auf eine Vielzahl von Protokollen auf. Zum Schutz des Netzverkehrs zwischen den Standorten kann eine Kombination aus Tunneling-Protokoll und kryptographischen Diensten verwendet werden.

A.5 SSL - Secure Sockets Layer

Das SSL-Protokoll bietet Verschlüsselung in der Transportschicht. SSL wendet dabei die Sicherheitsvorkehrungen über eine Variante der TCP-Socket-Schnittstelle an. SSL wurde von Netscape Communications entwickelt. Der Netscape Commerce Server setzt SSL für sichere Web-Dienste ein, als Client fungiert der Netscape Navigator.

A.6 PGP - Pretty Good Privacy

PGP ist ein Protokoll der Anwendungsschicht, es dient zur verschlüsselung und zur Authentifizierung von Mails.

A.7 SSH - Secure SHell

Ist ebenfalls ein Protokoll der Anwendungsschicht und dient zur authentifizierung von Remote Logins und verschlüsselt zudem die Sitzung.

B IPSec beim Internet Protokoll IPv6

Bei der Entwicklung des neuen IP-Standards wollte man das Gebiet der Sicherheit nicht mehr vernachlässigen. Allgemein sollte bei jedem neuen Standard auf Sicherheitsfragen eingegangen werden. Zunächst wurden bei IPnG noch eigene Protokolle zur Sicherung der Übertragung definiert, die Arbeiten wurden später jedoch in die Arbeitsgruppe IPSEC der IETF verlagert. Dort entwickelte man nun Standards für IPv6, die auch noch unter IPv4 Verwendung finden sollten.
Im Gegensatz zu IPv4 ist der Header von IPv6 auf ein Minimum gekürzt worden. Um auch die bisher benutzten sowie neue Funktionen unterzubringen ist eine Erweiterung des Headers notwendig. Bei IPv4 wurde dies durch ein variables langes Optionsfeld realisiert. Bei IPv6 wird dies durch eine Verkettung beliebig vieler Header realisiert. Dabei hat jeder Header eine bestimmte Funktion.

Im Feld Protocol wird anhand des Protokoll-Identifikators der Inhalt der Nutzdaten angegeben.
0 Hop-by-Hop Option
60 Ziel-Option
43 Header für Routing
44 Header für Fragmentierung
51 Header für Authentisierung
50 Header für Verschlüsselung
59 Keine weitere Header

Zur sicheren Übertragung werden die in der Arbeitsgruppe IPSEC definierten Verfahren herangezogen. Für eine detaillierte Beschreibung sei auf [DeHi 98] verwiesen.

Literaturverzeichnis

[Asso 00] ASSOCIATION, INTERNATIONAL COMPUTER SECURITY: *IPSec Certified Product listings.* Technischer Bericht ICSA, 2000.

[DeHi 98] DEERING, S. und R. HINDEN: *RFC 2460: Internet Protocol, Version 6 (IPv6) Specification.* RFC, IETF, Dezember 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2460.html`.

[Fran 99] FRANKEN, MALTE: *Sicherheit auf der Netzwerkschicht (IPsec).* http://einstein.informatik.uni-oldenburg.de/lehre/Seminar/SemSichWS99/07/, 1999.

[Gaut 99] GAUTIER, ROBERT J: *Performance of the FreeS/WAN IPSEC Implementation.* Technischer Bericht TSC Technical Support Centre, 1999.

[Harr 99] HARRISON, ANN: *Free software fights Net snoopers.* Technischer Bericht CNNin, 1999.

[HoRi 00] HOFFMAN, PAUL und MICHAEL RICHARDSON: *Steps for IPsec Interoperability Testing.* Internet Draft, IETF, 02 2000, `http://www.normos.org/ietf/draft/draft-hoffman-ipsec-testing-00.txt`.

[IBM 99] IBM: *Technologies: Virtual Private Networks and IPSEC.* http://www-3.ibm.com/security/technologies/techvpn.shtml, 1999.

[IETF] *The Internet Engineering Task Force.* WWW, `http://www.ietf.org/`.

[IPSEC] *IP Security Protocol (ipsec),* `http://www.ietf.org/html.charters/ipsec-charter.html`. IETF Working Group Charter, http://www.ietf.org/html.charters/ipsec-charter.html.

[JJB 00] JEAN-JACQUES BERNARD, GHISLAINE LABOURET, HERVÉ SCHAUER: *Results of some IPsec interoperability tests conducted by HSC.* Technischer Bericht HSC Hervé Schauer Consultants, 2000.

[KBC 97] KRAWCZYK, H., M. BELLARE und R. CANETTI: *RFC 2104: HMAC: Keyed-Hashing for Message Authentication.* RFC, IETF, Februar 1997, `http://sunsite.auc.dk/RFC/rfc/rfc2104.html`.

[KeAt 98a] KENT, S. und R. ATKINSON: *RFC 2402: IP Authentication Header.* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2402.html`.

[KeAt 98b] KENT, S. und R. ATKINSON: *RFC 2406: IP Encapsulating Security Payload (ESP).* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2406.html`.

[Kule 98] KULESSA, ANNE: *Keine Angst vor Schnuefflern.* Gateway, 11/1998.

[MaGl 98a] MADSON, C. und R. GLENN: *RFC 2403: The Use of HMAC-MD5-96 within ESP and AH.* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2403.html`.

[MaGl 98b] MADSON, C. und R. GLENN: *RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH.* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2404.html`.

[Mart 98] MARTIUS, KAI: *IPSec und ISAKMP in der Praxis.* iX, 8/1998.

[MeSi 95] METZGER, P. und W. SIMPSON: *RFC 1828: IP Authentication using Keyed MD5.* RFC, IETF, August 1995, `http://sunsite.auc.dk/RFC/rfc/rfc1828.html`.

[MoGu 98] MONTENEGRO, G. und V. GUPTA: *RFC 2356: Sun's SKIP Firewall Traversal for Mobile IP.* RFC, IETF, Juni 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2356.html`.

[MSST 98] Maughan, D., M. Schertler, M. Schneider und J. Turner: *RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP).* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2408.html`.

[Orma 98] Orman, H.: *RFC 2412: The OAKLEY Key Determination Protocol.* RFC, IETF, November 1998, `http://sunsite.auc.dk/RFC/rfc/rfc2412.html`.

[PKIX] *Public-Key Infrastructure (X.509) (pkix)*, `http://www.ietf.org/html.charters/pkix-charter.html`. IETF Working Group Charter, http://www.ietf.org/html.charters/pkix-charter.html.

[Raep 98] Raepple, Martin: *Sicherheitskonzepte für das Internet.* dpunkt–Verlag, Heidelberg, 1998.

[Raep 99] Raepple, Martin: *Transportsicherung.* iX, 1/1999.

[Reis 97] Reiser, H.: *Sichere TCP/IP-basierte Kommunikation bei der BMW AG.* Diplomarbeit, Technische Universität München, August 1997.

[Tane 96] Tanenbaum, Andrew S.: *Computer Networks.* Prentice Hall International Editions, Third Auflage, 1996.

[TCM 99] TCM: *Security Project at the TCM Laboratory.* http://www.tcm.hut.fi/Tutkimus/IPSEC/ipsec.html, 1999.

[Tso 98] Tso, Theodore: *Companies which are Implementing (or Planning to Implement) IPSEC/ISAKMP.* Technischer Bericht MIT, 1998.