Technologische Methoden und Mechanismen zur Erhöhung der Vertraulichkeit von VoIP-Verbindungen

Magisterarbeit

zur Erlangung des akademischen Grades
Diplomingenieur (Dipl.-Ing.)

ausgeführt am
Institut für Rechnergestützte Automation
Forschungsgruppe Industrial Software
der Technischen Universität Wien

unter der Anleitung von
Univ.-Prof. DI Dr. Thomas Grechenig

durch
Johannes Kuch
Werfelgasse 26/10
7400 Oberwart
johannes.kuch@ieee.org

Oberwart, 5. Mai 2008

Zusammenfassung

Das Telefon begleitet die Menschheit seit seiner Erfindung vor nun mehr als 150 Jahren. Es gab ständig Weiterentwicklungen und Neuerungen, die den anfänglichen Apparat zur Kommunikation via Sprache mittlerweile zu einem multifunktionalen Gerät für vielerlei Aufgaben reifen ließen. Der neueste Trend auf diesem Gebiet heißt Voice-over-IP (VoIP) und befindet sich seit einigen Jahren auf einem rasanten Vormarsch. Durch die Verbreitung des Internets und TCP/IP im LAN-Bereich findet sich fast überall eine geeignete Infrastruktur, um darauf aufbauend Telefoniedienste zu betreiben. Durch das Aufkommen von immer einfacheren Produkten, die sich diese Technologie zunutze machen, steigt nun die Verbreitung weiter. Wie jede neue Technologie, die Vorteile bringt, gibt es auch bei VoIP Risiken und Probleme beim Einsatz, welche auch berücksichtigt werden müssen.

Dazu soll aufgezeigt werden, wo Schwachstellen von derzeitigen VoIP-Protokollen sind, was man dagegen tun kann und inwieweit der Einsatz von Sicherungsmaßnahmen Besserung bringt bzw. was für Nachteile durch den Einsatz dieser auftreten können. Die vorliegende Arbeit soll hierbei einen aktuellen Überblick über verwendete und mögliche Technologien zur Sicherung von VoIP und der zugrunde liegenden Infrastruktur geben. Aufgrund der rasant wachsenden Technologien bieten andere Werke meist eine überalterte und/oder nicht vollständige Sicht auf die aktuelle Entwicklung, sodass diese zwar als Quellen herangezogen werden können, aber nicht ausreichen. Auch steht bei dieser Arbeit die ganzheitliche Betrachtung im Vordergrund, sodass sowohl VoIP als auch die zugrunde liegende Infrastruktur als Gesamtheit analysiert wird. Im Rahmen der vorliegenden Arbeit wird klar aufgezeigt, dass es für die Absicherung von VoIP-Verbindungen eine Vielzahl von Möglichkeiten gibt, deren Einsatz stark von den Anforderungen an die Sicherheit und den technischen Möglichkeiten der VoIP-Geräte abhängt.

Inhaltsverzeichnis

1. Einleitung 1

2. Grundbegriffe der IT-Sicherheit 3
- 2.1 Geschichtliche Entwicklung 4
- 2.2 Schutzziele 5
  - 2.2.1 Allgemeine Beschreibung 6
  - 2.2.2 Schutzziele im Kontext von VoIP 7
- 2.3 Ausgewählte Typen von Angriffen 10
  - 2.3.1 Eindringen in fremde IT-Systeme 10
  - 2.3.2 Denial-of-Service (DoS) 10
  - 2.3.3 Man-in-the-Middle-Attacken (MitM) 14
  - 2.3.4 Sniffing 15
  - 2.3.5 Replay 16

3. Innovation in der Telefonie 17
- 3.1 Einführung in die Telefonie-Welt 18
  - 3.1.1 Geschichte der Sprachtelefonie 18
  - 3.1.2 PSTN - (analoge) Telefonnetze 20
  - 3.1.3 ISDN - Integrated Services Digital Network 23
- 3.2 Sicherheitsaspekte der klassischen Sprachtelefonie 27
- 3.3 Einführung in Voice over IP 29
- 3.4 Technische Grundlagen von VoIP 32
  - 3.4.1 RTP & RTCP 32
  - 3.4.2 Einführung in H.323 36
  - 3.4.3 Einführung in SIP 41
  - 3.4.4 Media Gateway Protokolle 46
- 3.5 Beschreibung der Basis-Infrastruktur für VoIP 49

3.5.1 Ethernet 49
3.5.2 Internet Protocol - IP 53
3.5.3 User Datagram Protocol - UDP 55
3.5.4 Transmission Control Protocol - TCP 56

4. Sicherheitsprobleme von VoIP-Systemen und deren Basis-Infrastruktur . . 60
4.1 Angriffe auf die Basis-Infrastruktur für VoIP 60
4.1.1 Man-in-the-Middle-Attacken (MitM) 60
4.2 Angriffe und Schwachstellen von SIP-basierten VoIP-Systemen 63
4.2.1 Denial-of-Service-Attacken (DoS) 63
4.2.2 Session-Hijacking: Unbemerktes Umleiten von Anrufen 65
4.2.3 Identitätsdiebstahl durch Fälschen der SIP-Registrierungsinformationen 66

5. Abwehr von Angriffen und Erhöhung der Sicherheit von VoIP-Anlagen . . 67
5.1 Sicherung der Basis-Infrastruktur 68
5.1.1 Physische Sicherung 68
5.1.2 logische Netzwerktrennung 69
5.1.3 Firewalls 70
5.1.4 Instrusion-Prevention-Systeme (IPS) 75
5.1.5 Honeypots/-nets 77
5.2 Sicherung von VoIP-Verbindungen 80
5.2.1 IP Security (IPSec) 81
5.2.2 Transport Layer Security (TLS) 89
5.2.3 HTTP Authentifizierung 92
5.2.4 S/MIME 96
5.2.5 Secure RTP (SRTP) 96
5.2.6 VoIP Application Gateways 97
5.3 Konsequenzen von verschärften Sicherheitsmaßnahmen 98
5.3.1 Rechenaufwand 98
5.3.2 Vergrößerung der Datenmengen und des Bandbreitenbedarfs 99
5.3.3 Steigender Administrationsaufwand 99
5.3.4 Fehlende Kompatibilität von einzelnen Komponenten 99

6. Zusammenfassung 101
6.1 Fazit 101
6.2 Ausblick 102

Verzeichnisse 103
Abkürzungsverzeichnis 103
Abbildungsverzeichnis 106
Literaturverzeichnis 108

1. Einleitung

1871 veröffentlichte Antonio Guiseppe Meucci erstmals den Entwurf eines Telefons. Seitdem ist viel Zeit vergangen und Telefone haben sich zum wahrscheinlich am öftesten genutzten Kommunikationsmedium der Zeit etabliert. Die in den letzten Jahrzehnten entwickelte und immer stärker werdende Technologie von Voice-over-IP (VoIP) versucht nun die "herkömmliche Telefonie", wie sie bekannt und beherrscht ist, mit der Internet-Technologien zu vereinen und so eine neue Anwendung zu schaffen. Gerade, weil Telefonie etwas Bekanntes und Allgegenwärtiges ist, haben Benutzer eine bestimmte Erwartungshaltung, die es zu erfüllen gilt.

Hierbei handelt sich sowohl um Faktoren, wie Sprachqualität, Verzögerungsfreiheit, aber auch Verfügbarkeit und Sicherheit werden erwartet. Diese technischen Aspekte der Sicherheit von VoIP sind daher das Thema der vorliegenden Arbeit.

Um einen möglichst umfassenden Überblick in die Thematik von VoIP und dessen Sicherheit zu geben besteht die Arbeit aus zwei Teilen, die nicht explizit als solche ausgewiesen sind, aber thematisch zusammengehören. Die ersten zwei Kapitel, welche den ersten Teil bilden, umfassen hierbei Grundlagen-Informationen über IT-Sicherheit, Telefonie und Voice over IP. Sie bilden also die Wissensbasis für die folgenden Kapitel, welches sich dann eingehend mit geeigneten Sicherungsmaßnahmen beschäftigt. Dies kann als zweiter Teil angesehen werden.

Das erste Kapitel der Arbeit gibt eine Beschreibung der Grundbegriffe der IT-Sicherheit. Hierbei wird vor allem auf den Begriff der "Schutzziele" und deren Definition Wert gelegt. Von dem Begriff der Schutzziele wird im weiteren Verlauf immer wieder Gebrauch gemacht, weshalb an dieser Stelle eine detaillierte Einführung erfolgt. Des weiteren gibt es eine Einführung in die Begriffsdefinitionen für Angriffe auf IT-Systeme.

Nach der kurzen Einführung in die Begriffswelt der IT-Sicherheit folgen technische Grundlagen. Die drei großen Themengebiete hierbei sind die klassische Telefonie, Voice over IP und die IT-Infrastruktur, auf der ein VoIP-System aufsetzt. Für jedes dieser Gebiete wird zuerst eine allgemeine Einführung gegeben, um danach die jeweiligen Sicherheitsaspekte

bzw. Angriffe zu erläutern.

Nach dieser umfassenden theoretischen Einführung in die Materie beginnt der Hauptteil dieser Arbeit. Hierbei geht es im Wesentlichen um die Sicherung von VoIP. Weil aber die Sicherheit eines VoIP-System nicht nur von der "Eigensicherheit" des selbigen abhängt, sondern auch von der Sicherheit der zugrunde liegenden Infrastruktur, werden Sicherungs-Maßnahmen sowohl für VoIP, als auch für die Infrastruktur betrachtet. Da der Gewinn an Sicherheit, welchen diese Maßnahmen bringen, nicht ganz ohne "Kosten" bzw. Nachteile entsteht, werden diese abschließend gesondert beleuchtet.

Den Abschluss bildet eine Zusammenfassung der Thematik in Form eines Fazits über den derzeitigen Stand der Technik und einen Ausblick auf künftige Technologien.

2. Grundbegriffe der IT-Sicherheit

Bevor man sich mit Sicherheit bzw. im Speziellen mit IT-Sicherheit auseinander setzt, ist ein Befassen mit den wesentlichen Grundbegriffen und der Entstehung derselben notwendig.

Dazu beginnt dieses Kapitel mit einem kurzen Rückblick in die Anfänge der IT-Sicherheit und zeigt dessen geschichtliche Entwicklung auf.

Darauf folgend wird ein wesentlicher Begriff der IT-Sicherheit: die Schutzziele, erklärt. Schutzziele sind Definitionen, die beschreiben, was an einem IT-System überhaupt schützenswert ist. Nach der allgemeinen Betrachtung des Begriffs der Schutzziele folgt zusätzlich noch eine Betrachtung der Schutzziele speziell im Kontext von VoIP.

Den dritten und letzten Punkt dieses Kapitels bildet eine Erläuterung der Angriffstypen gegen IT-Systeme. Hierzu werden potentielle Angriffe nach ihrer Vorgangsweise kategorisiert beschrieben.

2.1 Geschichtliche Entwicklung

Sicherheit war schon immer ein wichtiger Aspekt im Zuge der Verwendung von Informationstechnologie, weshalb die Geschichte derselben nicht erst in der jüngsten Vergangenheit, sondern weitaus früher beginnt. Im Folgenden werden die jeweils wichtigsten Ereignisse bzw. Gegebenheiten in der Entwicklung der IT-Sicherheit aufgezeigt, wie man sie z.B. auch unter [SS05] nachlesen kann.

Mitte der 70er Jahre gewann die IT-Sicherheit in den Unternehmen an Bedeutung. Nicht nur die Sicherung der Daten, sondern auch die der Technik wurde immer wichtiger. Durch die Einführung von Teilnehmer-Betriebssystemen wurde es möglich online auf Daten zuzugreifen und diese auch zu bearbeiten oder zu löschen.

In den 80er Jahren wurde die Verwaltung mehrerer Benutzer und deren Zugriffsrechte durch die schnelle Verbreitung von PCs in den Unternehmen unerlässlich. PCs wurden somit zu einer bedeutenden Konkurrenz für zentrale Großrechner, da die Netzwerkbetriebssysteme über eine leistungsfähige Benutzerverwaltung verfügten und IT-Sicherheit in ähnlich großem Umfang möglich war.

In den späten 80er Jahren wurde das Internet für zivile Zwecke freigegeben. Ursprünglich sollte das Internet für nicht-kommerzielle Zwecke verwendet werden, doch schon 1994 war die Zahl der kommerziellen Nutzer größer, als die der wissenschaftlichen Teilnehmer.

Wenngleich das weltumspannende Internet einen wesentlichen Beitrag für die schnellere und bessere Entwicklung der meisten Bereiche des Lebens (z.B. Wirtschaft, Forschung und andere) getan hat, eröffnet es aber auch Möglichkeiten für kriminelle Elemente Schaden anzurichten oder sich selbst zu bereichern. Das Hauptproblem ist hierbei sicherlich das rasante Wachstum. Die Allgegenwärtigkeit und Unverzichtbarkeit des Internets hat niemand in dieser Form vorausgesehen. Dadurch wurde auch nicht auf spezielle Schutzbedürfnisse massiv vernetzter Rechner oder deren Benutzer eingegangen.

2.2 Schutzziele

Unter dem Namen "Schutzziele" versteht man im Kontext der IT-Sicherheit jene Schlagwörter, welche die zentralen Sicherheitsthemen bilden bzw. die Teilaspekte der IT-Sicherheit. Jedes "Schutzziel" / jeder Teilaspekt beschäftigt sich mit eigenen Fragen und beinhaltet eine Gruppe von Problemen für die es auch wieder eine Vielzahl von Lösungen gibt. Oftmals werden mit bestimmten Lösungen mehrere Schutzziele gleichzeitig erfüllt, was durchaus erwünscht ist. Ein umfassender Schutz in der IT kann nur gewährleistet werden, wenn für jede Komponente eines IT-Systems alle relevanten Schutzziele abgedeckt sind und dementsprechend gewährleistet werden. Je nach Anwendung bzw. System treten bestimmte Schutzziele in den Vordergrund oder werden manchmal auch bewusst ignoriert oder als weniger wichtig erachtet.

Über die genaue Auflistung und Benennung der Schutzziele findet man teilweise durchaus (wenn auch meistens nur leicht) unterschiedliche Definitionen (siehe hierzu z.B. [BSI05a] und [JP06]), was darauf zurückzuführen ist, dass einzelne Schutzziele in mehrere Teile aufgespalten oder mehrere Schutzziele zusammengefasst werden. Eine weit verbreitete Einteilung der IT-Schutzziele wäre wie folgt: (die englischen Begriffe sind in Klammer hinzugefügt)

- Verfügbarkeit (*Availability*)
- Integrität (*Integrity*)
- Vertraulichkeit (*Confidentiality*)
- Verbindlichkeit / Unabstreitbarkeit (*Non-Repudiation*)
- Authentizität (*Authenticity*)

Nachfolgend in diesem Kapitel wird auf die einzelnen Schutzziele eingegangen und diese näher erläutert. Dies geschieht zuerst allgemein und dann speziell für den Anwendungsfall "Voice over IP".

2.2.1 Allgemeine Beschreibung

Die Beschreibung eines Schutzziels ist in der Praxis oftmals mit konkreten Systemen verbunden. Eine derartige Erklärung der Schutzziele findet sich auch in dieser Arbeit an späterer Stelle in Kapitel 2.2.2 auf der nächsten Seite. Davor werden jedoch die Schutzziele in allgemeiner Art und Weise beschrieben, da dies wichtig für das Grundverständnis der Thematik ist. Die folgenden Beschreibungen stützen sich hierbei auf die Definitionen, wie sie in [BSI05a] verfügbar sind.

2.2.1.1 Verfügbarkeit

Das erste Schutzziel, welches hier näher erläutert wird, ist die Verfügbarkeit. Gleichzeitig bildet es die Basis für alle weiteren Schutzziele bzw. für ein funktionierendes IT-System. Ein verfügbares IT-System ist in der Lage alle benötigten Daten und Anfragen entsprechend der Vorgaben zu liefern bzw. zu erfüllen, welches gleichzeitig als eine Grundvoraussetzung für jede Form von Informationsverarbeitung und deren Sicherheitsüberlegungen darstellt, denn ein nicht-reagierendes/-arbeitendes IT-System braucht auch nicht (mehr) beschützt zu werden. Für spezielle IT-Systeme gelten auch besondere Kriterien - man spricht hierbei von "Hoch-Verfügbarkeit" (*High-Availability*). Dies ist vor allem dort zu finden, wo ein Ausfall eine unmittelbare Bedrohung von Menschen(leben) und/oder dem (wirtschaftlichen) Fortbestehen von Firmen darstellen würde.

2.2.1.2 Integrität

Um einen reibungslosen Betrieb eines IT-Systems zu gewährleisten, muss sichergestellt werden, dass sämtliche Daten, Befehle, usw. sich stets in einem definierten "Soll-Zustand" befinden. Durch das Verletzen der Integrität kann es zu einem unvorhergesehenen System-Verhalten kommen, der einen Daten-Verlust oder ähnlich schlimme Fehlverhalten hervorrufen kann. Integrität wird unter anderem gewährleistet indem man sämtliche Daten, Befehle, etc. vor der Verarbeitung auf eine gültige Syntax bzw. Vollständigkeit überprüft und den Zugriff auf gespeicherte Daten nur über genau definierte Schnittstellen zulässt.

2.2.1.3 Vertraulichkeit

Unter Vertraulichkeit versteht man die Sicherstellung, dass Daten auch nur den entsprechend autorisierten Personen zur Verfügung stehen. Des weiteren geht es um die erfolgreiche Verhinderung von nichtberechtigten Zugriffen. Vertraulichkeit hat vor allem im Bereich

von sensiblen (z.B. medizinischen, persönlichen, juristischen) Daten eine hohe Bedeutung, da durch eine falsche Offenlegung Schaden für Leib und Leben oder eine Verletzung der Privatsphäre die Folge wären.

2.2.1.4 Verbindlichkeit / Unabstreitbarkeit

Die Verbindlichkeit/Unabstreitbarkeit gewährleistet, dass der Urheber und Empfänger einer Nachricht bzw. allgemein einer Datenübertragung eindeutig festgelegt sind. Dieses Schutzziel ist eng mit der Authentizität (siehe 2.2.1.5) verbunden, da nur durch die Echtheit von Angaben eine Verbindlichkeit/Unabstreitbarkeit erreicht werden kann.

2.2.1.5 Authentizität

Bei der Authentizität handelt es sich um das Sicherstellen der Echtheit eines Subjekts. Bei Einhaltung der Authentizität ist gewährleistet, dass Angaben z.B. über Kommunikationsteilnehmer einer Datenübertragung tatsächlich richtig sind. Dies einerseits als Grundlage für die Verbindlichkeit bzw. Unabstreitbarkeit (siehe 2.2.1.4) wirkt sich aber auch auf andere Schutzziele, wie die Vertraulichkeit (siehe 2.2.1.3 auf der vorherigen Seite aus).

2.2.2 Schutzziele im Kontext von VoIP

Während die in Kapitel 2.2.1 auf der vorherigen Seite genannten Schutzziele allgemein für IT-Systeme bzw. für die IT-Sicherheit gelten, folgt nun eine nähere Betrachtung oben genannter Schutzziele für den Einsatz von VoIP-Systemen. Da Telefonie ein integraler Bestandteil der modernen Kommunikation ist und sich auch schon jahrelang etabliert hat, haben sich gewisse Erwartungen aufgebaut, die von VoIP nun auch erfüllt werden müssen, um als Ersatz für die derzeitigen Telefonie-Systeme dienen zu können.

2.2.2.1 Verfügbarkeit bei VoIP

Wie schon für die allgemeine Verfügbarkeit von IT-Systemen (siehe 2.2.1.1 auf der vorherigen Seite) gilt für VoIP-Systeme, dass sämtliche Komponenten eine gewisse Verfügbarkeit aufweisen müssen, um einen reibungslosen Betrieb zu gewährleisten. Ohne die jeweiligen Endgeräte ("VoIP-Telefone") und die Zwischenkomponenten ("VoIP-Server" und andere)

ist ein Betrieb nicht möglich. Entsprechende Vorkehrungen zur Gewährleistung der Verfügbarkeit sind daher zu treffen. Öffentliche Telefonnetze weisen meist eine sehr gute Verfügbarkeit auf, da die Kontrolle bei einzelnen, großen Telefon-Providern liegt, wohingegen bei VoIP Systeme zum Einsatz kommen, die außerhalb der Kontrolle von einzelnen/wenigen liegen (wie z.B. das Internet) und es daher ungleich schwieriger ist Garantien abzugeben oder einen reibungslosen Betrieb zu gewährleisten. Zusätzlich gehört zum Thema der Verfügbarkeit auch eine gewisse Mindestleistung der beteiligten Systeme (inklusive des Übertragungsmediums), da bei einer zu großen Verzögerung keine sinnvolle Sprachtelefonie durchgeführt werden kann.

2.2.2.2 Integrität bei VoIP

Ein herkömmliches Telefongespräch zwischen zwei Teilnehmern wird üblicherweise als "Punkt-zu-Punkt-Verbindung" angesehen. Während dies für das leitungsvermittelte PSTN tatsächlich gilt, wird bei VoIP ein paketvermittelter Dienst in Anspruch genommen. Durch diesen Umstand werden die Sprachinformationen über gemeinsam genutzte Verbindungen übertragen - im Falle des Internets nehmen die Pakete Wege, die sie an verschiedensten Knotenpunkten vorbeiführen. Die Sicherstellung der Integrität der Pakete ist hierbei wichtig, um mögliche Veränderungen der übertragenen Daten feststellen zu können.

2.2.2.3 Vertraulichkeit bei VoIP

Gerade bei Kommunikationsmitteln legt jeder Benutzer Wert darauf, dass nur jene Personen die versandte Nachricht erreichen, die vom Absender ursprünglich ausgewählt wurden. Bei einem Telefongespräch wird erwartet, dass nur die unmittelbar daran beteiligten Parteien das Gesprochene hören. Diese Erwartung wird natürlich auch in VoIP gesetzt und da bei einer Übertragung über das Internet der genaue Weg der IP-Pakete nicht vorab definiert ist, müssen entsprechende Vorkehrungen getroffen werden, um zu gewährleisten, dass der Sprach-Datenstrom nicht mitgeschnitten oder verändert (siehe 2.2.2.2) werden kann.

2.2.2.4 Verbindlichkeit / Unabstreitbarkeit bei VoIP

Betrachtet man das Schutzziel der Verbindlichkeit bzw. Unabstreitbarkeit im Zusammenhang mit bzw. bei der Verwendung von VoIP, so erkennt man zwei primäre Anforderungen, die bei konsequenter Einhaltung des Schutzziels gewährleistet sein müssen:

- Niemand kann abstreiten einen bestimmten Anruf nicht getätigt zu haben.

- Niemand kann abstreiten einen bestimmten Anruf nicht erhalten zu haben.

Um dies zu gewährleisten ist einerseits eine Protokollierung notwendig und andererseits eine funktionierende Arbeit der Authentizität (siehe 2.2.2.5 und 2.2.1.5 auf Seite 7) zur Sicherstellung der Identitäten der einzelnen Kommunikationsteilnehmer.

Ohne Authentizität hätte eine Aufzeichnung weniger Sinn, da dann nicht gewährleistet wäre, dass die Identität eines Teilnehmers bzw. Anschlusses echt ist.

2.2.2.5 Authentizität bei VoIP

Die Gewährleistung der Authentizität bei der Sprachtelefonie beschäftigt sich mit dem Problem, dass unter einer angerufenen oder anrufenden Teilnehmer-Kennung (beispielsweise eine Telefon-Nummer im PSTN) sich auch tatsächlich der erwartete Teilnehmer meldet. Die Authentizität ist die Basis für andere Schutzziele (wie z.B. der Verbindlichkeit, siehe 2.2.2.4 auf der vorherigen Seite).

2.3 Ausgewählte Typen von Angriffen

Angriffe auf IT-Systeme gibt es seitdem es die IT gibt. Je nach Angriffsmuster und dem Ziel der Attacke kann man diese in unterschiedliche Gruppen einteilen. Die Vorstellung und Einteilung, der für VoIP relevantesten, möglichen Angriffe, sowie die wesentlichen Kriterien, wodurch eine Attacke einer bestimmten Kategorie zugeordnet wird, ist Teil dieses Abschnitts.

2.3.1 Eindringen in fremde IT-Systeme

Die erste Gruppe von Angriffen auf ein IT-System ist der allgemeine Begriff des "Einbruchs" bzw. "Eindringens". Ein Einbruch in ein IT-System ist stark vergleichbar mit dem Eindringen einer Person in ein Haus. Es stellt das widerrechtliche Betreten bzw. Benützen eines IT-Systems durch eine unbefugte Person dar. Im Gegensatz zu den folgenden Angriffs-Arten gibt es beim Einbruch kein generelles Muster oder eine Konstellation, sodass man den "Einbruch" als "allgemeinen Typ" eines IT-Angriffs werten kann.

Genau so, wie die Methodik eines Eindringens sich mit dem angegriffenen System selbst immer verändert, ist auch das Ziel immer ein anderes. Ein wichtiger Grund ist sicherlich das Erlangen von Informationen. Was in der realen Welt Wertgegenstände oder Geld direkt ist, sind in der Informationstechnologie, wie es der Name schon vermuten lässt, Informationen. Diese lassen sich, direkt verkaufen oder verhelfen einer Person/Firma, die sie besitzt, zu einem wirtschaftlichen Vorteil.

2.3.2 Denial-of-Service (DoS)

Unter einer "Denial-of-Service" (DoS) - Attacke versteht man einen Angriff mit dem Ziel ein bestimmtes Gerät (in diesem Fall ein VoIP-Endgerät oder einen Server) außer Betrieb zu setzen. Dies kann durch eine bewusste Manipulation oder durch Überlastung geschehen, wobei das Ziel ist, dass das Gerät seine Dienste nicht mehr im vollen Funktionsumfang anbieten kann oder ganz ausfällt.

DoS-Attacken werden meist automatisiert durchgeführt, da sie oftmals durch Überlastung des Ziels realisiert werden und die entsprechenden Vorgänge nicht in der benötigten Menge manuell durchführbar wären (wie z.B. eine hohe Anzahl von Zugriffen auf eine Ressource binnen kürzester Zeit). Derartige Anfragen können entweder darauf abzielen die Internet-Anbindung des Rechners zu überlasten, sodass kein weiterer Verkehr mehr

zu dem Rechner durchdringen kann, oder durch falsche bzw. unvollständige Anfragen die Ressourcen des Servers zu überlasten (z.B. indem Ressourcen verbraucht werden, um halbfertige Anfragen zwischenzuspeichern). Im zweiteren Fall können es durchaus kleinere Anfrage-Pakete sein, sodass ein Senden einer großen Anzahl keine entsprechend gute Anbindung erfordert. Ein Angreifer, der aber durch viele Pakete die Anbindung zum Rechner selbst überlasten will muss über zumindest die selbe Bandbreite verfügen, wie das Angriffs-Ziel, um erfolgreich zu sein. Da jedoch derartige Anbindungen normalerweise teuer bzw. nicht einfach zu bekommen sind, hat sich ein Spezialfall für DoS-Attacken gebildet: DDoS-Attacken.

DDoS

DDoS steht für "Distributed Denial-of-Service" und beschreibt eine Technik, bei der mehrere Rechner in eine Attacke involviert sind. Dazu werden meist Viren bzw. Trojaner in Umlauf gebracht, die eine Hintertür bei dem infizierten Rechner öffnen, sodass ein potentieller Angreifer diese Rechner nutzen kann, um einen Angriff durchzuführen oder aber ab Aktivierung (also Infektion des Rechners) kontinuierlich ein fest vorgegebenes Ziel angreifen. Dadurch ist es möglich mit einer großen Anzahl an Rechnern ein viel höheres Datenaufkommen zu erzeugen, als es mit einem einzelnen angreifenden Rechner möglich wäre und zusätzlich ist das Angriffsziel nicht mehr direkt ausmachbar, da meist "unbeteiligte" Privat-PCs benutzt werden, die keine Rückschlüsse auf den Ursprung der Attacke zulassen.

DRDoS

In der Literatur (wie z.B. [Ver01])findet man manchmal auch eine Sonderbezeichnung namens "DRDoS" (Distributed Reflected Denial of Service). Bei dieser Art von Angriff sendet ein Angreifer Anfragen (z.B. Ping-Requests) an verschiedene Internet-Rechner. Der Trick hierbei ist, die Absenderadresse zu fälschen ("spoofen"), sodass die angesprochenen Geräte ihre Antworten an das Angriffs-Ziel senden. Sendet man nun sehr viele Anfragen aus, so bekommt das Ziel entsprechend viele Antworten und wird durch diese lahm gelegt - im Gegensatz zum "klassischen" (D)DoS, bei dem die Anfragen direkt an das Opfer gehen. Mit so einer DRDoS-Attacke kann man auch theoretisch mehrere Ziele gleichzeitig durch Überlastung angreifen: zum einen das Opfer, welches mit (unverlangten) Antworten überhäuft

und zum anderen die Ziele, die mit Anfragen überhäuft werden. Durch den "Umweg" über andere Rechner kann außerdem das Opfer des Angriffs nicht mehr direkt die Absender der Attacke herausfinden, da ja nur Antworten von anderen Rechnern ankommen - die Suche nach dem Schuldigen wird also schwieriger gestaltet.

Beschreibung von DoS-Angriffen

Im folgenden Kapitel wird näher auf die tatsächlichen Möglichkeiten eines DoS-Angriffs eingegangen, welche hier in zwei Gruppen unterteilt sind:

1. DoS durch Überlastung
   Angriffe, die durch pure Überlastung einer Verbindung oder eines Systems diese/s am normalen Betrieb hindern

2. DoS durch Ausnutzung von Software-Fehlern
   Angriffe, die durch spezifische, auf das Ziel-System angepasste, Vorgehensweisen dieses außer Betrieb setzen

DoS durch Überlastung

DoS-Angriffe basieren meist auf der Überlastung von IT-Systemen. Im Falle einer IT-Infrastruktur können dies entweder Netzwerkgeräte (wie z.B. Router und Switches), einzelne Rechner (z.B. Server) oder Übertragungsverbindungen sein.

Da die Methoden in Abhängigkeit des attackierten Ziels verschieden sind, werden im Anschluss sowohl Möglichkeiten der Überlastung von Verbindungen, als auch zur Überlastung von Servern (bzw. allgemein von Rechnern) dargestellt.

Überlastung von Verbindungen

Im Falle von Internet-Anbindungen werden durch Überlastung sehr oft die Internet-Anbindungen selbst angegriffen. Bewerkstelligt wird dies durch Methoden wie DDoS, die bereits früher erläutert wurden. Einen wirklich zuverlässigen Schutz hierfür gibt es höchstens von Seiten des Internet-Providers, der entsprechenden Verkehr nicht weiterleiten darf, wenn er eine DoS-Attacke erkennt.

DoS-Attacken, die eine Überlastung von Verbindungen zum Ziel haben, werden sehr oft mittels ICMP-Echo-Requests gehandhabt, da diese entsprechend klein sind, eine Antwort generieren, welche auch wieder Bandbreite benötigt und auf jeder Plattform verfügbar sind.

Überlastung von Servern

Im Unterschied zur Überlastung von Leitungen, werden für DoS-Angriffe auf Server bzw. allgemein Rechnern meist andere Möglichkeiten verwendet, da die oben beschriebenen ICMP-Echo-Requests oftmals standardmäßig ignoriert werden, um automatisierten Scan-Programmen nicht als potentielles Ziel aufzufallen. Auch ist der ICMP-Echo-Mechanismus nicht für den ordnungsgemäßen Betrieb notwendig und wir daher auch weggelassen, um das Maximum an Rechnerkapazität für die jeweilige eigentliche Aufgabe frei zu haben.

Ein Angriff ist daher meist nur durch Verwenden bzw. Ausnützen von Protokollen möglich, die auch für den normalen Betrieb notwendig sind und dort verwendet werden. Ein beliebtes Ziel hierbei ist TCP bzw. dessen Handshake-Mechanismus (siehe hierzu Kapitel 3.5.4 auf Seite 56).

Eine Attacke besteht nun aus einer raschen Abfolge von unzähligen TCP-Verbindungs-Anfragen (also dem Eröffnen von TCP-Handshakes), wodurch die Kapazitäten des Ziels aufgebraucht werden sollen, da jede Verbindung eine gewisse Menge System-Ressourcen benötigt. Diese so genannten halb-offenen TCP-Verbindungen werden erst nach Ablauf einer gewissen Timeout-Zeit vom System aufgegeben und belegen bis dorthin Ressourcen und können so legitime Benutzer des Servers aussperren, da ein Rechner, in Abhängigkeit von seinen System-Ressourcen, nur eine gewissen Anzahl an TCP-Verbindungen verwalten kann.

DoS durch Ausnutzung von Software-Fehlern

Im Gegensatz zu den im Kapitel 2.3.2 (Seite 12ff.) beschriebenen Methoden, welche alle unabhängig von den tatsächlichen Geräten sind, gibt es noch eine große Zahl von Software-Fehlern (Bugs) in diversen Geräten und Programmen, die sich für den Einsatz als DoS-Attacke eignen.

Da diese Schwachstellen und Fehler von der jeweils eingesetzten Software und dort im Speziellen von deren Version, ist eine umfassende Beschreibung nicht annähernd möglich, weshalb hier darauf verzichtet wird.

Die häufigsten Probleme ergeben sich bei der Behandlung von Zeichenketten, die nicht auf ihre Länge überprüft werden. Dadurch kann es bei der Speicherung im Hauptspeicher dazu kommen, dass eine Zeichenkette mehr Speicherplatz benötigt, als Speicher für sie reserviert ist. Durch dieses Phänomen wird dann an Stellen des Speichers geschrieben,

wo eigentlich Informationen von anderen Variablen (oder noch schlimmer) internen Befehlen abgelegt ist. Durch geschickte Wahl von Zeichenketten lassen sich nun so gezielt Programm-Anweisungen in den Speicher schreiben, welche dann auch ausgeführt werden. Wird die Zeichenkette nicht speziell präpariert, sodass bestimmte Anweisungen in den Speicher geschrieben werden, so zerstört der Eingriff auf jeden Fall die korrekte Ausführung des angegriffenen Programms. Allgemein bekannt ist dieses Verfahren unter dem Namen "Puffer-Überlauf" (engl. *Buffer Overflow*).

2.3.3 Man-in-the-Middle-Attacken (MitM)

Bei einer "Man-in-the-Middle"-Attacke wird eine Verbindung durch einen Dritten, der sich logisch gesehen zwischen den zwei Teilnehmern befindet, angegriffen sodass der Angreifer Zugriff auf den gesamten Datenstrom dieser Verbindung hat.

MitM-Angriffe werden entweder durch tatsächliches physisches "Anklemmen" (engl. "wiretap") an eine tatsächliche physische Verbindung realisiert oder durch Umleiten des Netzwerkverkehrs. Hat ein Angreifer beispielsweise Zugriff auf das selbe lokale Netzwerk wie ein Teilnehmer der anzugreifenden Verbindung, kann durch Manipulation der Switches der Datenverkehr entsprechend umgeleitet werden, welches auch das "Mithören" ermöglicht. Bei IP-Verbindungen müssen dann entsprechend Router mit falschen Routing-Informationen gefüttert oder gehackt werden, um den Datenstrom umzulenken.

Eine schematische Darstellung eines derartigen Angriffs zeigt Abbildung 2.1 auf der nächsten Seite. Beide Teilnehmer gehen hierbei von einer direkten Verbindung aus, jedoch werden jetzt sämtliche Übertragungen am Man-in-the-Middle vorbei geleitet. Bei Bedarf kann nun dieser die Verbindung einseitig lösen, z.B. Teilnehmer "B" ein Ende der Verbindung signalisieren, und danach "in dessen Rolle schlüpfen". Man spricht in diesem Zusammenhang auch oftmals von "Identitäts-Diebstahl" oder "Session-Hijacking".

Beim "Session-Hijacking" ist das Ziel eines Angreifers nicht die Information selbst, sondern die Verbindung bzw. Sitzung. Dies geschieht hauptsächlich, um gewisse Vertrauensverhältnisse zwischen den Teilnehmern auszunutzen. Ein Beispiel hierfür wäre eine unbemerkte Umleitung von Anrufern bei der Hotline ihrer Bank. Während vermutlich niemand einem Fremden die genauen Daten seines Online-Konto-Zugangs geben würde, lassen sich etliche Leute dazu bringen, diese mitzuteilen, wenn sie im festen Glauben sind, an der Hotline ihrer Bank zu sein.

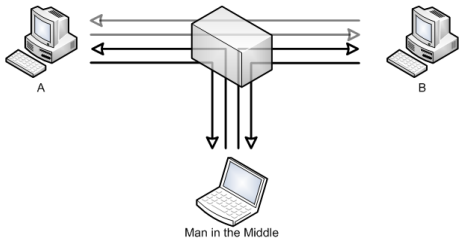

Abb. 2.1: Schema eines Man-in-the-Middle-Angriffs, bei dem der Datenstrom umgelenkt wird

2.3.4 Sniffing

Unter "Sniffing" versteht man das klassische Abhören. Es werden hierbei keinerlei Daten verändert, sondern lediglich mitgeschnitten. Ein derartiger Angriff ist normalerweise vollkommen transparent, da er keinerlei direkte Spuren im Datenstrom hinterlässt. Lediglich über manipulierte Zwischengeräte (die dazu verwendet werden den Datenstrom "weiterzuleiten") kann man einen Missbrauch erkennen. Sniffing ist ähnlich wie Man-in-the-Middle-Angriffe (siehe 2.3.3 auf der vorherigen Seite), jedoch ist Sniffing ohne bestimmtes Ziel einfach zu realisieren, da es nur darauf ankommt, an eine Datenleitung angeklemmt zu sein.

Abbildung 2.2 veranschaulicht den Spezialfall, wo der Datenverkehr zwischen zwei bestimmten Teilnehmern abgehört werden soll. Die beiden Teilnehmer ("A" und "B") gehen von einer direkten Datenübertragung aus (grauer Pfeil), während - aufgrund einer Manipulation o.ä. - der gesamte Datenverkehr abgezweigt und an einen Angreifer weitergeleitet wird (schwarze Pfeile).

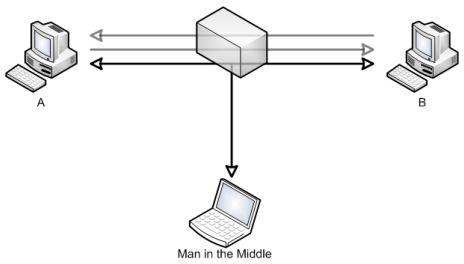

Abb. 2.2: Schema eines Sniffing-Angriffs, bei dem der Datenstrom dupliziert wird

Wesentlich für das Sniffing bzw. den Angreifer ist, dass auch das Vertrauensverhältnis zwi-

schen den beiden Kommunikationsteilnehmern missbraucht wird. Wüsste die Teilnehmer, dass sie abgehört werden, würden sie vermutlich vorsichtiger sein, welche Daten sie austauschen oder bewusst Falsch-Informationen dazu verwenden, um Mithörer zu täuschen. Normalerweise wird Sniffing aber erst im Nachhinein oder nach einer gewissen (längeren) Zeitspanne bemerkt und zu diesem Zeitpunkt sind schon einige Informationen ausgetauscht worden, die dann gestohlen wurden.

2.3.5 Replay

Bei einer so genannten "Replay"-Attacke wiederholt ein Angreifer die gesamten Kommandos (oder einen Teil davon), welche zwischen zwei Teilnehmern gesandt werden. Das kann einerseits zum Zwecke der Dienst-Unterbrechung (Denial-of-Service, siehe auch 2.3.2 auf Seite 10) und andererseits zum Zweck der Integritäts-Verletzung eingesetzt werden. Replay-Attacken haben also einen rein zerstörerischen Aspekt und liefern - im Unterschied zu den vorangegangenen Angriffen - einem Angreifer keine (vertraulichen) Informationen.

Replay-Attacken sind sehr einfach gestrickte Angriffe, die mit relativ geringem Aufwand verhindert werden können. Eine gängige Möglichkeit ist eine Art von Kennzeichnung für Pakete einzuführen (eine laufende Nummer, ein Zeitstempel des Versands, o.ä.), sodass das Duplikat sofort auffällt.

3. Innovation in der Telefonie

Nach der allgemeinen Einführung in die Thematik und in die IT-Sicherheit bietet das folgende Kapitel nun das Grundlagen-Wissen über die verwendeten Technologien. Dies umfasst die "klassische Telefonie", "Internet-Telefonie" (also Voice over IP), sowie die "Basis-Infrastruktur" für VoIP (z.B. die IT-Infrastruktur eines Unternehmens, in welche VoIP-Systeme integriert werden bzw. auf derer VoIP-Systeme verwendet werden).

Zur Erleichterung des Einstiegs und als Grundlage für die Betrachtung von VoIP beginnt das Kapitel mit einer Einführung in die Welt der analogen und digitalen Telefonie (also der so genannten "klassischen Telefonie") und einer kurzen Betrachtung von Sicherheits-Überlegungen in diesem Zusammenhang.

Erst danach folgt ein Einstieg in VoIP, dessen technische Grundlagen und der anschließenden Betrachtung der zwei wichtigsten VoIP-Protokolle H.323 und SIP. Fortgesetzt wird dann mit einer kurze Beschreibung von Media Gateway Protokollen, welche dazu verwendet werden ein VoIP-Netz an ein öffentliches Telefonnetz zu koppeln - also einen nahtlosen Übergang zwischen den beiden "Telefonie-Welten" schaffen.

Um der Zielsetzung der ganzheitlichen Betrachtung der Problematik von Angriffen auf VoIP-Systeme auch wirklich gerecht werden zu können, bedarf es auch der Beschäftigung mit der Infrastruktur, auf die VoIP-Systeme aufbauen. Die Sicherheit von VoIP-Systemen wird auch durch die Sicherheit der Infrastruktur bestimmt, in der ein VoIP-System eingebettet ist. Daher wird an dieser Stelle der Aufbau einer solchen Infrastruktur und im nächsten Kapitel die Möglichkeiten für Angriffe entsprechend erläutert.

3.1 Einführung in die Telefonie-Welt

Beim Herangehen an die Thematik von "Voice over IP" kommt man an einer Betrachtung der "klassischen" Telefonie nicht vorbei, da diese einerseits die technologische Grundlage von VoIP bildet und andererseits moderne VoIP-Systeme de facto immer mit den bestehenden Telefonnetzen gekoppelt werden müssen, da VoIP - im Gegensatz zu den "herkömmlichen" Telefonnetzen - nicht so weit verbreitet und verfügbar ist. Das folgende Kapitel gibt eine kurze Einführung in die Welt der analogen und digitalen Telefonnetze und wie diese intern arbeiten. Weitere Details und ausführlichere Erklärungen finden sich z.B. in dem sehr guten Werk [Bad05] und auch in [JP06].

Der Überblick über die Technologien beginnt mit der geschichtlichen Entwicklung, die von den ersten Labor-Versuchen des Telefons und der Patentierung zur Entwicklung der Technik, wie sie heute in Verwendung ist, führt.

(Analoge) Telefonnetze und ISDN sind auf die Sprachkommunikation ausgelegte Netzwerke, die auf dem Prinzip der Leitungsvermittlung ("Circuit Switching") basieren. Das heißt, dass im Falle einer Verbindung zweier Endgeräte (Telefone) eine direkte Leitung bei allen Geräten dazwischen "durchgeschalten" wird. Es besteht dann eine eigene, dedizierte Leitung zwischen diesen zwei Geräten.

Da die Verbindungen zwischen zwei Teilnehmern entsprechend auf- und abgebaut werden muss, kommen so genannte Signalisierungsprotokolle zur Anwendung. Der Vorgang des Herstellens bzw. des Abbaus einer Verbindung wird dementsprechend auch "Signalisierung" genannt und im weiteren Verlauf dieses Kapitels näher beleuchtet.

3.1.1 Geschichte der Sprachtelefonie

Die Geschichte des Telefons (und damit der Sprachtelefonie) beginnt vor ca. 150 Jahren als Philipp Reis im Jahr 1861 ein Gerät präsentiert, mit dem man Töne (und auch Sprache) mittels elektrischer Impulse übertragen kann. Die Übertragungsleistung und die Sprachqualität waren zwar sehr gering, aber es hat sich im Nachhinein -gezeigt, dass eine Sprachübertragung technisch möglich war. Dennoch gilt Reis nicht als Erfinder des Telefons, da seine Entdeckung nur experimenteller Natur war und er sie nicht bis hin zu einer tatsächlich praktischen Anwendung verbesserte.

Heutzutage wird der US-Italiener Antonio Guiseppe Meucci als Erfinder des Telefons angesehen. Dieser veröffentlichte 1871 in der italienischen Ausgabe eines amerikanischen Jour-

nals seine Entdeckung, die er sich auch patentieren lies. Unglücklicherweise lief dieses 1874 wieder aus, da er die Kosten dafür nicht zahlen konnte. 1876 entwickelte dann der Amerikaner Alexander Graham Bell ein kommerziell brauchbares Telefon, welches er im Februar 1876 zum Patent anmeldete. Nur zwei Stunden später am selben Tag versuchte auch der Amerikaner Elisha Gray einen ähnlichen Apparat zum Patent anzumelden. Das Patent wurde dann 3 Wochen später (am 7. März) an Bell vergeben, der im darauf folgenden Jahr die Firma "Bell Telephone Association" gründete und damit den Bau eines Telefonnetzes in den USA übernahm. 1885 wurde die Firma in den heutigen Namen "American Telephone and Telegraph Company (AT&T)" umbenannt und ist heute der größte Telefonkonzern der Welt. Lange Jahre wurde Bell die Erfindung des Telefons zugerechnet, bis im Jahr 2002 von den USA offiziell beschlossen wurde, dass Meucci die Erfindung des Telefons zuerkannt wird.

In den darauf folgenden Jahren und Jahrzehnten brachten zahlreiche Hersteller verschiedenste Telefon-Modelle auf den Markt (wie z.B. das "Siemens-1900" von Siemens, siehe linkes Telefon in Abbildung 3.1). Die Grundbauweise wurde im Wesentlichen bis in die 1960er Jahre gleich belassen und änderte sich erst durch die Erfindung des Tastentelefons und später durch die Weiterentwicklung der zugrundeliegenden Technologie. In den darauf folgenden Jahrzehnten wurde aus dem analogen Telefonnetz ein digitales und in späterer Folge kam - bedingt durch die rasante Entwicklung des Internets - die Grundsteinlegung für Voice over IP zustande. Dadurch änderte sich natürlich auch das Design der Endgeräte (siehe z.B. ein aktuelles VoIP-Telefon von Cisco, rechtes Telefon in Abbildung 3.1) und das Telefonnetz selbst. Genauere technische Details, wie ein aktuelles Telefon- bzw. VoIP-Netz funktioniert folgen in den kommenden Unterkapiteln.

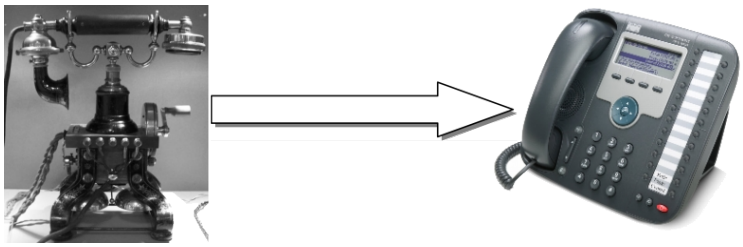

Abb. 3.1: Telefone im Wandel der Zeit [Bad05]

3.1.2 PSTN - (analoge) Telefonnetze

Öffentliche Telefonnetze - abgekürzt meistens als *PSTN* (public switched telephone network) bezeichnet - bilden die Grundlage der Kommunikation der letzten Jahre und Jahrzehnte. Obwohl die meisten Telefonnetze dieser Welt bereits digitalisiert worden sind und im Kern-Bereich bereits mit digitalen Leitungen arbeiten, werden sie dennoch des öfteren als "analoge" Telefonnetze bezeichnet. Dies liegt vor allem daran, dass häufig noch analoge Endgeräte (Telefone) zur Anwendung kommen. In solchen Fällen gibt es dann im Anschlussbereich eine Analog/Digital-Umwandlung, um diese in das übrige Netz zu integrieren.

Vereinfacht lässt sich die Architektur eines Telefonnetzes wie folgt beschreiben:
Im Anschlussbereich stehen "Teilnehmervermittlungsstellen" (TVSt), die mit einer oder mehren "Fernvermittlungsstellen" (FVSt) verbunden sind. Diese Fernvermittlungsstellen sind untereinander wiederum verbunden (siehe dazu Abbildung 3.2). Somit bilden die FVSt den "Kern" des Telefonnetzes, während die TVSt den "Anschlussbereich" darstellen. Die Endgeräte (also die Telefone) sind direkt mit der jeweils (geografisch) nächstliegenden TVSt verbunden. Wie bereits oben beschrieben, sind die FVSt eines Telefonnetzes mit digitalen Leitungen verbunden, während im Anschlussbereich (zwischen TVSt und Telefon des Kunden) auch oftmals analoge Leitungen (und Endgeräte) zur Anwendung kommen.

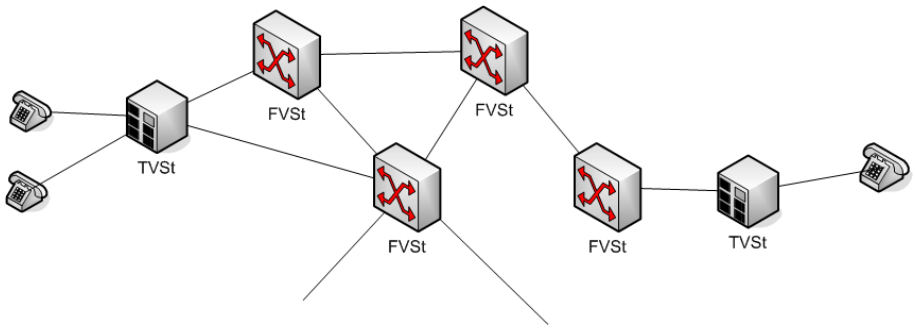

Abb. 3.2: Schematische Darstellung eines Telefonnetzes [Bad05]

Entsprechend den Unterschieden in den Leitungen, mit denen die unterschiedlichen Komponenten zusammengeschlossen sind, unterscheiden sich auch die Signalisierungsprotokolle. Zwischen TVSt und FVSt beziehungsweise zwischen den FVSt untereinander wird das international standardisierte (und auch bei ISDN verwendete) "Signalisierungs-System 7"

(Signaling System 7, SS7) verwendet. Die Signalisierung im Anschlussbereich (zwischen Telefongerät und TVSt) erfolgt wiederum nach einem eigenen Standard.

Während die Signalisierung zwischen Telefonen und TVSt als *Innenband-Signalisierung* erfolgt, d.h. für Signalisierung und Sprachübertragung wird die gleiche Leitung verwendet, erfolgt die Signalisierung zwischen den einzelnen FVSt als *Außenband-Signalisierung*, d.h. es gibt getrennte Leitungen für Signalisierungs-Informationen und Sprachübertragung.

Ablauf der Signalisierung

Die Signalisierung im Telefonnetz läuft (vereinfacht dargestellt) nach folgenden Schritten ab (grafische Darstellung: siehe Abbildung 3.3 auf der nächsten Seite):

1. Teilnehmer 1 hebt den Hörer ab
2. TVSt erhält Signalisierungsinformation *Belegung*
3. Akzeptiert die TVSt die Belegung, bestätigt diese durch Anlegen eines Wähltons
4. Teilnehmer 1 tippt die entsprechenden Wählziffern ein. Nach Erhalt der ersten Ziffer schaltet die TVSt den Wählton ab.
5. Nach Erhalt der letzten Wählziffer signalisiert die TVSt durch die SS7-Nachricht *IAM* (Initial Address Message) über die FVSt an die TVSt des Teilnehmers 2 (dem Angerufenen) den Anruf
6. Sollte der Anschluss des Teilnehmers 2 frei sind, aktiviert dessen TVSt ein Klingeln bei diesem und signalisiert an die wählende TVSt die SS7-Nachricht *ACM* (Address Complete Message), worauf diese ein Freizeichen an Teilnehmer 1 schickt.
7. Nach Abheben von Teilnehmer 2 (welches der TVSt als *Melden* signalisiert wird), wird dies der TVSt von Teilnehmer 1 mittels der SS7-Nachricht *ANS* (Answer Message) signalisiert
8. Beim Eintreffen der ANS-Nachricht bei der TVSt. von Teilnehmer 1 wird dessen Freizeichen abgeschalten und die Verbindung ist aufgebaut. Ab diesem Zeitpunkt werden Gebühren verrechnet und das Telefongespräch kann stattfinden
9. Nach dem Auflegen eines Teilnehmers erhält dessen TVSt die Signalisierungs-Nachricht *Auslösen*, die als SS7-Nachricht *REL* (Release) weitergeleitet. Dadurch

wird auch die Gebührenerfassung beendet und dem zweiten Teilnehmer die *Auslöse-Anzeige* signalisiert

10. Sobald der zweite Teilnehmer aufgelegt hat (was seinerseits als *Auslösen* an die TVSt signalisiert wird), signalisiert dessen TVSt noch die SS7-Nachricht *RLC* (Release Complete). Damit ist das Telefongespräch vollständig abgebaut.

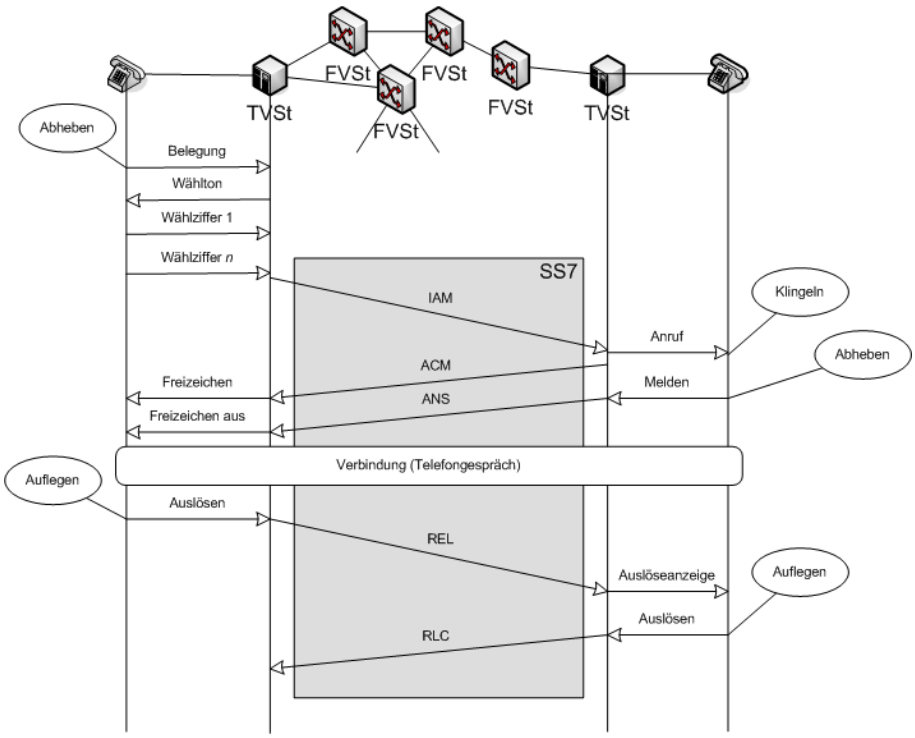

Abb. 3.3: Ablauf der Signalisierung in einem Telefonnetz [Bad05]

Wie man nun erkennen kann, ist ein "simples" Telefongespräch mit einem hohen Signalisierungsaufwand verbunden. Da VoIP-Gespräche, wie bereits eingangs erwähnt, aber nicht nur von IP-Gerät zu IP-Gerät erfolgen sollen, sondern auch in ein öffentliches Telefonnetz, ist die Betrachtung (und das Verständnis) der Signalisierung ein Erfordernis, will man VoIP vollständig verstehen und beherrschen.

3.1.3 ISDN - Integrated Services Digital Network

ISDN stellt eine Weiterentwicklung von herkömmlichen Telefonnetzen dar und arbeitet vollkommen digital. Die wesentlichen Grundprinzipien bleiben zwar die selben, aber dennoch wurde die Komplexität (und auch die Mächtigkeit) erhöht.

ISDN arbeitet ebenfalls nach dem Prinzip der Leitungsvermittlung und benötigt für den Auf- und Abbau von Verbindungen auch eine Form von Signalisierung. Im Unterschied zu herkömmlichen Telefonnetzen erfolgt die Signalisierung bei ISDN immer *out-of-band*, d.h. die Signalisierungs-Informationen werden immer über einen eigenen Kanal übertragen.

Jeder Teilnehmer am ISDN benötigt einen so genannten "ISDN-Anschluss", bei dem es im Gegensatz zu herkömmlichen Telefonnetzen zwei grundlegende Varianten gibt:

- den Basisanschluss (im Englischen *BRI - Basic Rate Interface* genannt)
- den Primärratenanschluss (im Englischen *PRI - Primary Rate Interface* genannt)

Der Basisanschluss besteht aus zwei so genannten *B-Kanälen*, über welche die eigentliche Datenübertragung (bzw das Telefongespräch) erfolgen kann und einem so genannten *D-Kanal*, welcher für die Übertragung der Signalisierungs-Informationen verantwortlich ist. Die Datenübertragungsleistung eines B-Kanals beträgt dabei 64 kBit/s und die eines D-Kanals 16 kBit/s, wobei der D-Kanal nicht für die Datenübertragung durch einen Benutzer verwendet werden kann. Bei ISDN ist zur Erhöhung der Datendurchsatzrate auch die Möglichkeit zur "Kanal-Bündelung" vorgesehen und möglich. In diesem Fall werden beide B-Kanäle verwendet (quasi "zusammengeschalten") und es steht die doppelte Bandbreite (d.h. 128 kBit/s) zur Verfügung, wobei aber auch die doppelten Leitungskosten anfallen, da zwei Leitungen vom Telefonnetz bereitgestellt (und durchgeschalten) werden müssen. Die Kanalbündelung ist ein "Feature", welches beide Gegenstellen unterstützen müssen, um funktionieren zu können.

Ein Primärratenanschluss besteht im Gegensatz dazu aus 30 B-Kanälen zu je 64 kBit/s und einem D-Kanal zu 64 kBit/s für die Steuerungsinformationen. Die gesamte Bandbreite eines Primärratenanschlusses beträgt damit ca. 2 MBit/s. Ein derartiger Anschluss wurde häufig für Standleitungen zur Datenübertragung herangezogen. Der in Europa gebräuchliche Ausdruck für eine eine derartige Standleitung lautet *E1*.[1]

[1] Am Rande sei noch erwähnt, dass die ISDN-Implementierung, wie sie in den USA Anwendung findet, sich leicht von der europäischen unterscheidet. Der Hauptunterschied ist, dass ein PRI in den USA nur 23 B-Kanäle besitzt und somit eine Bandbreite von rund 1,5 MBit/s aufweisen kann. In den USA wird diese Form der Standleitung *T1* genannt.

Außer dem Unterschied der zwei verschiedenen Anschluss-Typen ändert sich an der grundlegenden Struktur des Netzes von ISDN im Gegensatz zum PSTN nichts. Jeder ISDN-Anschluss ist mit einer TVSt verbunden, die dann weitere Verbindungen zu einer oder mehreren FVSt besitzt (siehe dazu Abbildung 3.2 auf Seite 20).

Die Signalisierung bei ISDN wird, wie bereits beschrieben, über den so genannten D-Kanal abgewickelt. Die Übertragung der Signalisierungsinformationen geschieht dabei nach dem so genannten *D-Kanal-Protokoll*, welches nach dem OSI-Referenzmodell aufgebaut ist und durch die ITU-T standardisiert wurde:

- Die unterste Schicht ist die "Bitübertragungsschicht" (Schicht 1), welche durch die ITU-T-Standards I.430 und I.431 beschrieben wird
- Darüber liegt die "Sicherungsschicht" (Schicht 2), die für den Transport der Sicherungsinformationen (Schicht 3) verantwortlich ist. Realisiert wird das durch *LAPD* (Link Access Procedure on D-Channel), welches eine HDLC-Variante darstellt. Die ITU-T hat dazu die Standards Q.920 und Q.921 veröffentlicht
- Die eigentlichen Signalisierungsinformationen werden dann in der Schicht 3 ("Vermittlungsschicht") durch den Austausch von Nachrichten, die nach dem ITU-T-Standard Q.931 festgelegt sind. Zusätzlich beschreibt noch der Standard Q.930 die allgemeinen Aspekte des D-Kanal-Protokolls.

Ablauf der Signalisierung

Die eigentliche Signalisierung eines Telefon-Gesprächs im ISDN erfolgt dann nach folgenden Schritten (grafische Darstellung: siehe Abbildung 3.4 auf Seite 26):

1. Teilnehmer 1 hebt den Hörer ab
2. TVSt erhält Signalisierungsinformation *SETUP*, die bei Verfügbarkeit mit der Bestätigung *SETUP ACK* antwortet
3. Teilnehmer 1 tippt die entsprechenden Wählziffern ein. Für jede Wählziffer wird die Nachricht *INFO* generiert und an die TVSt gesandt
4. Nach Erhalt der letzten Wählziffer bestätigt die TVSt mit der Nachricht *CALL PROC*. Gleichzeitig wird die SS7-Nachricht *IAM* über die FVSt an die TVSt des Teilnehmers 2 (dem Angerufenen) gesandt

5. Sollte der Anschluss des Teilnehmers 2 frei sein, sendet dessen TVSt die Nachricht *SETUP* an diesen, welches zu einem Klingeln beim Teilnehmer führt. Dieser bestätigt den Erhalt der SETUP-Nachricht und das Klingeln mit der Nachricht *ALERT*

6. Den Erhalt der *ALERT*-Nachricht signalisiert nun die TVSt des Angerufenen an die wählende TVSt mittels der SS7-Nachricht *ACM*, die daraufhin ebenfalls eine *ALERT*-Nachricht an den Anrufer schickt

7. Nach Annehmen des Telefonats durch Teilnehmer 2 (welches der TVSt als *CONN* signalisiert wird), wird dies der TVSt von Teilnehmer 1 mittels der SS7-Nachricht *ANS* (Answer Message) signalisiert. Gleichzeitig erhält der Teilnehmer 2 von seiner TVSt die Bestätigung *CONN ACK*.

8. Beim Eintreffen der ANS-Nachricht bei der TVSt von Teilnehmer 1 wird dieser mit einer *CONN*-Nachricht benachrichtigt, die mittels *CONN ACK* bestätigt wird. Danach ist die ISDN-Verbindung aufgebaut und ab diesem Zeitpunkt werden Gebühren verrechnet und das Telefongespräch (oder die Datenübertragung) kann stattfinden.

9. Nach dem Beenden des Gesprächs durch einen der Teilnehmer (in diesem Beispiel Teilnehmer 1), erhält dessen TVSt die Signalisierungs-Nachricht *DISC*. Diese bestätigt mit der Signalisierung von *REL* und sendet gleichzeitig die SS7-Nachricht *REL* an die TVSt des Teilnehmers 2.

10. Auf die REL-Nachricht der TVSt von Teilnehmer 1 erfolgt nun noch eine Bestätigung durch das Endgerät von Teilnehmer 1 mittels der *REL COM*-Nachricht. Dadurch wird auch die Gebührenerfassung beendet und dem zweiten Teilnehmer die *Auslöse-Anzeige* signalisiert

11. Nach dem Empfang der REL-Nachricht durch die TVSt von Teilnehmer 2 sendet diese die Signalisierungs-Nachricht *DISC*, die - sobald der Benutzer tatsächlich aufgelegt hat - vom Endgerät des Teilnehmers durch die Nachricht *REL* bestätigt wird.

12. Nachdem Teilnehmer 2 aufgelegt hat und die TVSt die REL-Nachricht erhalten hat, bestätigt sie diese durch *REL COM* und signalisiert der TVSt von Teilnehmer 1 die SS7-Nachricht *RLC* (Release Complete). Damit ist die Verbindung vollständig abgebaut.

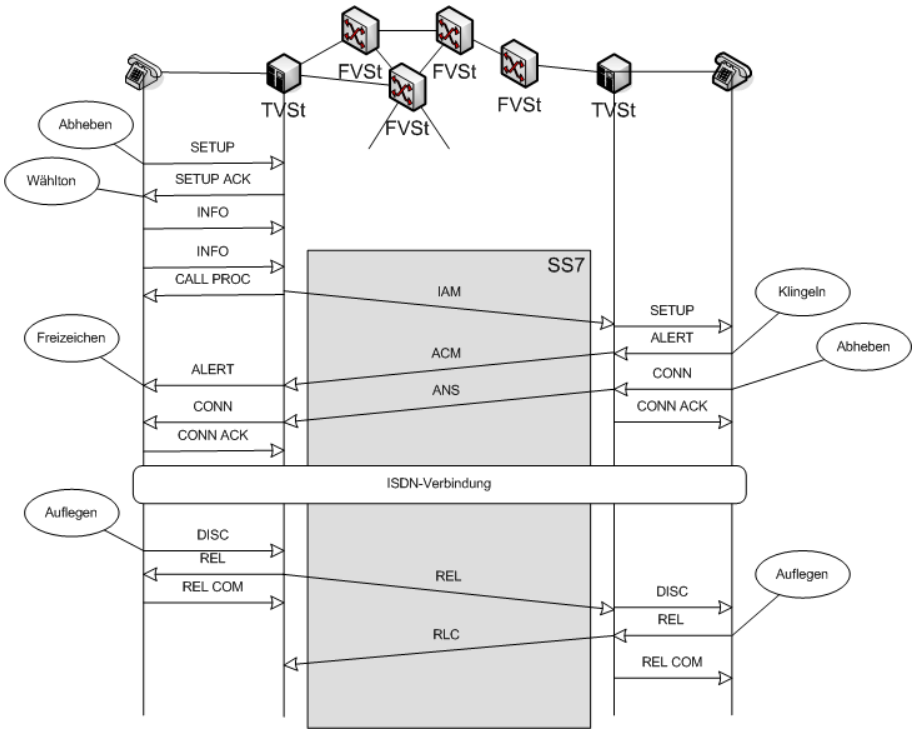

Abb. 3.4: Ablauf der Signalisierung bei ISDN [Bad05]

3.2 Sicherheitsaspekte der klassischen Sprachtelefonie

Mit der großflächigen Verbreitung von Telefonnetzen Anfang des 20. Jahrhunderts begannen die Überlegungen und Bedenken bezüglich der Sicherheit. Das Telefon ist in seiner ursprünglichen Konzeption als Medium für die Ende-zu-Ende-Kommunikation entwickelt worden. Das heißt, es gibt genau zwei Teilnehmer und niemanden sonst, der an der Kommunikation teilnimmt (weder aktiv noch passiv). Das ist auch gleichzeitig die älteste und wichtigste Forderung eines sicherheitskritischen Teilnehmers an ein Telefonnetz/-system: Gespräche sollen nur von den am Gespräch "eingeladenen" Personen mitverfolgt werden können (siehe hierzu auch 2.2.1.3 auf Seite 6 beziehungsweise 2.2.2.3 auf Seite 8). Weiters sollen Informationen über die Telefonverbindungen ("wer ruft wen an?", etc.) nur gewissen Stellen zur Verfügung stehen (dem Telefonnetz-Betreiber für die Abrechnung und den Gesprächsteilnehmern).

Ein System zur Sprachtelefonie soll jederzeit, wenn es benötigt wird, seinen Teilnehmern zur Verfügung stehen (man spricht in diesem Zusammenfassung von der Forderung nach Verfügbarkeit). Gerade in der Anfangszeit der Telefonnetze war Ausfallssicherheit eines der Hauptprobleme, da die Technologien noch nicht ausgereift waren und das Hauptaugenmerk lag auf der Bereitstellung der Technologie und nicht auf der Absicherung gegen eventuelle Ausfälle. Mit Voranschreiten der technologischen Entwicklung traten Ausfalls-Probleme immer weiter in den Hintergrund, sodass das Telefon als dauernd verfügbar angesehen wurde und wird.

Sicherheitsüberlegungen in PSTN- und ISDN-Netzen führten hauptsächlich zu einer physikalischen Abschirmung der einzelnen Komponenten. So wurden Vermittlungsstellen in eigenen (für Außenstehende nicht zugänglichen) Gebäuden untergebracht und kleinere Verteiler, welche in kleinen Gehäusen, die am Straßenrand stehen, untergebracht sind, durch entsprechende Schlösser gesichert. Dies bot einen ausreichenden Schutz gegen unbefugtes Mithören von Telefongesprächen oder die widerrechtliche Verwendung des Telefonnetzes (z.B. auf Kosten eines legitimen Anschlusses ohne dessen Zustimmung)

Durch das Fehlen von offengelegten Standards und dergleichen verstanden nur sehr wenige, die genaue Arbeitsweise der Komponenten und zusätzlich wurden die technischen Spezifikationen, welche anfangs noch stark proprietär waren, sehr gut behütet. Diese so genannte "Security by obscurity" erwies sich eine sehr lange Zeit als geeignet genug, was aber auch darauf zurückzuführen ist, dass generell ein hohes Maß an technischem Wissen erforderlich war und viele Menschen dieses schlicht nicht besaßen.

Dennoch gab es Hacker, die das Telefonnetz erfolgreich angegriffen haben. Am berühmtesten wurde der Fehler in AT&Ts Telefonnetz, welches Innenband-Signalisierung nutzte, der dazu führte, dass tausende Menschen kostenlos telefonieren konnten. Dazu wurde ein Ton mit 2600 Hz eingespeist, welcher veranlasste, die Vermittlungsstelle zu glauben, dass die Leitung frei ist und kein Gespräch stattfindet, welches aber normal geführt werden konnte - nun jedoch ohne Gebührenverrechnung. Die Entdeckung dieses Umstands und die Ausnutzung mittels so genannter Blue Boxes und anderer Geräte prägten dann eine ganze Ära. Auch ein Hacker-Magazin wählte seinen Namen nach der Frequenz dieses schicksalträchtigen Tons: "*2600*". In diesem Zusammenhang wurde für das Verwenden eines Telefonnetzes für Gespräche (oder Datenübertragungen), ohne dafür zahlen zu müssen, der Begriff "Phreaking" geprägt.

3.3 Einführung in Voice over IP

Nach der allgemeinen Einführung in die Telefonie in Form von PSTN und ISDN folgt nun eine Beschreibung von "Voice over IP", welches auch "Internet-Telefonie" genannt wird. Unter "Voice over IP" im engeren Sinn meint man die Übertragung von Sprache bzw. die Telefonie auf Basis der TCP/IP-Protokoll-Suite.

Anwendungsszenarien für Voice over IP

Man unterschiedet hierbei meist zwischen mehreren Anwendungsszenarien (siehe hierzu auch [Kuc05]):

- PC-to-PC (2 PCs, welche über ein IP-Netzwerk verbunden sind, werden für die Telefon-Verbindung benutzt)
- PC-to-Phone (1 PC baut eine Telefon-Verbindung zu einem herkömmlichen Telefon auf)
- Phone-to-Phone (2 herkömmliche Telefongeräte werden verwendet und lediglich im Kern-Netz wird VoIP verwendet)

Die Gemeinsamkeit dieser drei Szenarien besteht in der Verwendung von VoIP-Technologie beziehungsweise deren Komponenten. Bei der nachfolgenden näheren Betrachtung dieser drei Anwendungen zeigt sich dann, wo die VoIP-Technologie eingesetzt wird und für wen und wann der jeweilige Anwendungsfall Relevanz hat.

PC-to-PC

2 PCs (oder VoIP-fähige Endgeräte, z.B. IP-Phones), welche über ein IP-Netzwerk verbunden sind, werden für die Telefon-Verbindung benutzt - Abbildung 3.5 auf der nächsten Seite

Hierbei handelt es sich quasi um die erste (und einfachste Form) von VoIP. Durch fehlende Kopplung an das PSTN/ISDN ist der Teilnehmerkreis zwar sehr eingeschränkt, aber für bestimmte Anwendungen bietet diese Anwendungsart Vorteile. Während Internet-Verbindungen meist sehr günstig zu bekommen waren und sind, fallen für ein Langstrecken-Telefongespräch (z.B. ins Ausland) oft sehr hohe Minuten-Gebühren an, was aber durch den Einsatz von VoIP entfallen kann.

Eventuell wird für dieses Szenario noch ein Vermittlungsrechner (in der Abbildung als "Gatekeeper" bezeichnet) verwendet, der für den Verbindungsaufbau zwischen den zwei Teilnehmern vermittelt. Die Existenz und genaue Funktionsweise eines derartigen Geräts hängt vom verwendeten VoIP-Protokoll ab.

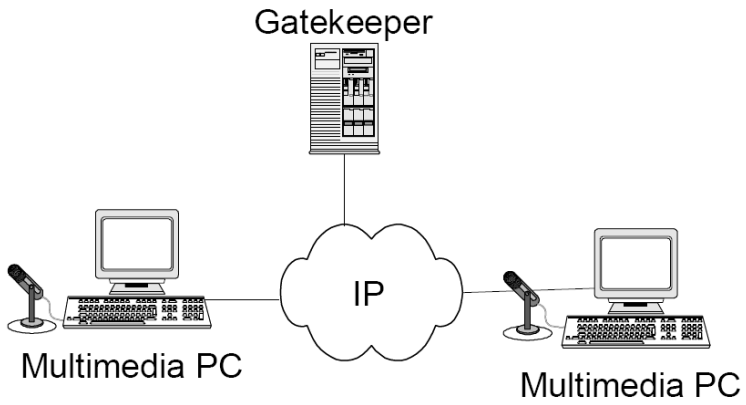

Abb. 3.5: VoIP-Anwendungsszenario: PC-to-PC [Kuc05]

PC-to-Phone

1 PC (oder VoIP-fähiges Endgerät, z.B. IP-Phone) baut eine Telefon-Verbindung zu einem herkömmlichen Telefon (PSTN bzw. ISDN) auf - Abbildung 3.6

Dies stellt eine Mischform in der Anwendung von VoIP dar. Während der eine Gesprächsteilnehmer schon VoIP benutzt, verwendet der zweite Teilnehmer noch PSTN/ISDN. In diesem Fall sorgt ein Gateway-Rechner für den nahtlosen Übergang in das PSTN/VoIP. Diese Variante ist üblicherweise auch bidirektional möglich, d.h. es kann sowohl das PSTN von der VoIP-Seite aus, wie auch das VoIP von der PSTN-Seite aus, angerufen werden.

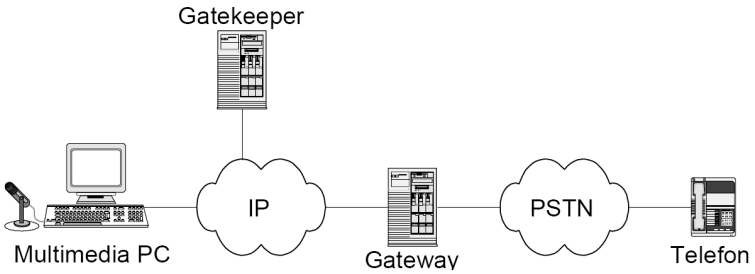

Abb. 3.6: VoIP-Anwendungsszenario: PC-to-Phone [Kuc05]

Phone-to-Phone

Verwendung von 2 herkömmlichen Telefongeräten (PSTN oder ISDN), wobei im Kern-Netz VoIP verwendet wird - Abbildung 3.7

Bei dieser Form der Anwendung benötigt man eigentlich keine VoIP-Technologie, jedoch findet sie in der Praxis immer wieder ihre Anwendung. Vor allem für Telefonnetz-Betreiber, die oftmals auch Internet-Service-Provider sind, ist diese Variante sehr attraktiv, da hierbei nur ein IP-Backbone-Netz betrieben werden muss, ohne radikale Veränderungen für den Endkunden durchführen zu müssen.

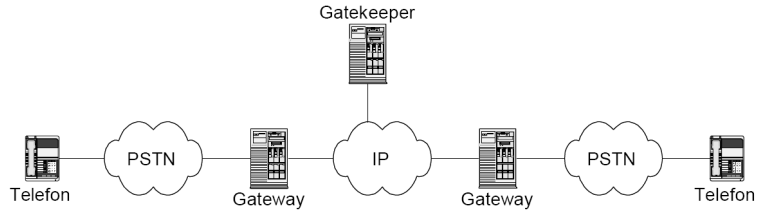

Abb. 3.7: VoIP-Anwendungsszenario: Phone-to-Phone [Kuc05]

Nach dieser Betrachtung der generellen Anwendungsgebiete und -möglichkeiten folgt nun eine genauere Vorstellung der technischen Details und verwendeten Protokolle beim Einsatz von VoIP. Weiterführende Literatur zum Thema VoIP ist sehr zahlreich, wie z.B. [Köh02], [Bad05], [Dur03], [LM00], [TW05] und [Joh04], [MU01].

3.4 Technische Grundlagen von VoIP

Das Thema "Voice over IP" kennt mehrere Schwerpunkte bzw. Kernfragen, die für ein funktionierendes VoIP-System gelöst werden müssen:

1. Transport der Sprach-Daten
2. Signalisierung
3. Sicherung der Signalisierungs- und Sprach-Informationen bzw. Daten-Ströme
4. Medienübergang (z.B. in ein öffentliches Telefonnetz)

Während bei 2. bereits sehr gute Ansätze aus der "Telefoniewelt" (siehe weiter oben) vorhanden waren, mussten die Entwickler von VoIP-Systemen bzw. -Standards für 1. vorhandene Konzepte (die Protokolle der TCP/IP-Protokollfamilie) überarbeiten, um den hohen Anforderungen der Sprach-Telefonie gerecht werden zu können. In diesem Zusammenhang wird auch oft das Problem der Sprachcodierung betrachtet, da in Abhängigkeit von der verwendeten Codierung der Bandbreitenbedarf drastisch verschieden ist - auf die Sprachcodierung und deren besondere Probleme und Bedeutung wird im vorliegenden Dokument aber nicht näher eingegangen. Ein sehr gutes Werk, welche sich mit Sprachcodierungen bei VoIP beschäftigt, ist z.B. [Bad05].

3.4.1 RTP & RTCP

Das *Realtime Transport Protocol* wurde von der IETF entwickelt und erstmals im Jahr 1996 als RFC 1889[GSC+96] spezifiziert. Die ursprüngliche Version wurde dann überarbeitet und durch das RFC 3550[SCFJ03] im Jahr 2003 ersetzt.

Beim RTP handelt es sich um ein verbindungsloses Protokoll, welches auf die Übertragung von Sprach- und Video-Informationen, bei denen es auf eine möglichst verzögerungsfreie (quasi Echtzeit-) Übertragung ankommt, optimiert ist. Um dies zu erreichen wurde mittels konsequentem Minimalismus sämtliche Steuerungsinformationen oder ähnliches aus dem RTP entfernt. Dieses Manko wird durch das *RTP Control Protocol*(RTCP) ausgeglichen, welches parallel zu RTP entwickelt wurde und in den gleichen RFCs[GSC+96, SCFJ03] spezifiziert wird. RTCP dient also der Steuerung und zur Übermittlung von Status-Informationen von RTP-Verbindungen/-Übertragungen.

3.4.1.1 RTP

Der Auf- bzw. Abbau einer RTP-Verbindung selbst findet unter Zuhilfenahme eines Signalisierungs-Protokolls (wie beim PSTN oder ISDN) statt. Signalisierungs-Protokolle können dabei entweder *H.323-SIG* (Signalisierung nach dem H.323-Standard) oder SIP sein. Beide Varianten werden in den folgenden Kapiteln näher erläutert.

RTP besitzt eine Reihe von Besonderheiten, die speziell für den Einsatz als Transportmedium von Echtzeitdaten (wie Sprache, Video und Audio) konzipiert wurden. Im Detail sind das:

- Garantie der Reihenfolge von RTP-Paketen
  RTP-Pakete sind nummeriert, sodass ihre Original-Reihenfolge am Ziel wiederhergestellt werden kann
- Garantie der Isochronität
  RTP-Pakete haben alle einen Zeit-Stempel, mit dem sichergestellt werden kann, dass die gleichen Zeitabstände am Ziel wiederhergestellt werden können
- Transport unterschiedlicher Formate von Echtzeitmedien (z.B. Sprache, Video, ...)
  RTP kennt so genannte Profile für unterschiedliche Typen von Medien, die transportiert werden. Diese Profile sind im RFC 3551[SC03] spezifiziert
- Einsatz von *Translator* und *Mixer* möglich
  RTP kennt so genannte Translator und Mixer. Ein Translator "übersetzt" RTP-Pakete in ein anderes Medienformat und ein Mixer kann mehrere Medienströme zu einem gemeinsamen Medienstrom zusammenfassen

Aufbau eines RTP-Pakets

Medienströme, die via RTP versandt werden, bekommen einen vorangestellten "RTP-Header". Das daraus resultierende RTP-Paket wird dann selbst als Nutzlast an ein UDP-Paket gehängt, welches die Nutzlast eines IP-Pakets bildet. Dies ist das direkte Resultat der Anwendung des TCP/IP-Schichtenmodells. Abbildung 3.8 auf der nächsten Seite stellt dies auch grafisch dar.

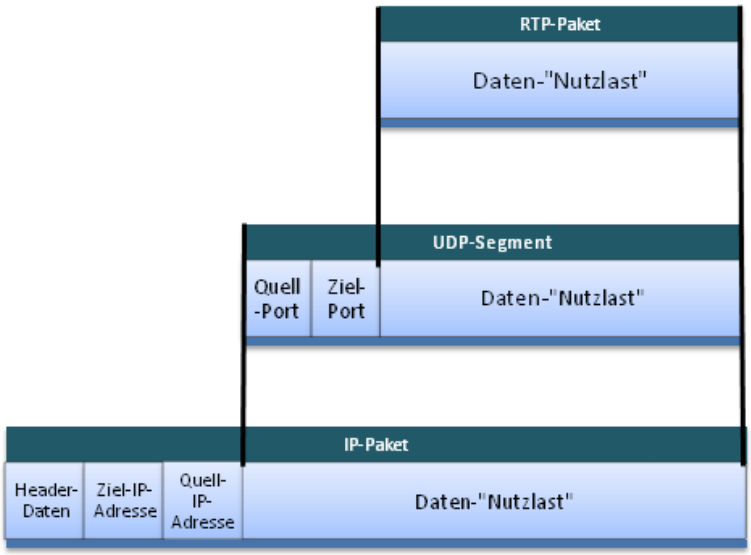

Abb. 3.8: Anwendung des TCP/IP-Schichtenmodells für RTP-Pakete

In Abbildung 3.9 auf der nächsten Seite ist der Aufbau des Headers eines RTP-Pakets zu sehen. Die Elemente sind demnach:

- V - *Version*
- P - *Padding - dient nur zum Auffüllen*
- X - *gibt an, ob hinter dem eigentlichen Header noch (optionale) Erweiterungsfelder (Header Extensions) kommen - dies lässt Spielraum für zukünftigen Versionen*
- CC - *CSRC Counter - gibt an, wieviel CSRC-Einträge im Header vorhanden sind - siehe dazu CSRC weiter unten*
- M - *Marker - die Bedeutung wird durch das verwendete Profil (also die Payload/Nutzlast) bestimmt*
- PT - *Payload Type (Nutzlast-Typ) - dieser Wert gibt an, um welches Format von Daten es sich handelt - gültige Werte sind im RFCF 3551 festgelegten "Profiltypen"*
- Sequence Number - *eine fortlaufende Nummerierung der RTP-Pakete, zur Gewährleistung, dass der Datenstrom beim Empfänger in der richtigen Reihenfolge zusammengesetzt werden kann*

- SSRC - *Synchronization Source Identifier - dient zur Identifikation der Quelle des Medienstroms*
- CSRC - *Contributing Source Identifiers - ist optional und gibt die möglichen "Original"-Quellen der Medienströme an. Wird eingesetzt, wenn der Sender nicht direkt der Ersteller der Daten - wird z.B. beim Einsatz eines Mixers verwendet. Ein Mixer meint hierbei ein Gerät, welches die Datenströme mehrerer Quellen zu einem Datenstrom zusammenfügt. Dabei trägt sich der Mixer selbst als Absender ein und definiert im Feld CSRC die eigentlichen Quellen der Datenströme*

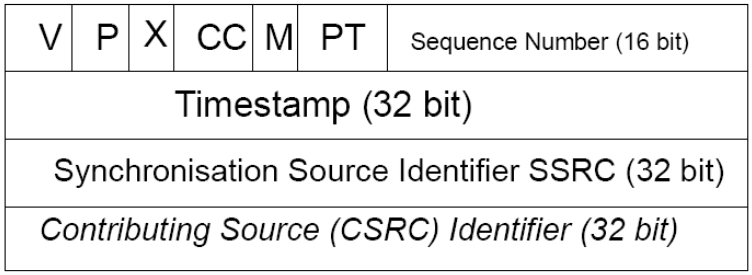

Abb. 3.9: Aufbau des RTP-Headers [Bad05]

Ein RTP-Header besitzt, wenn man alle optionalen Felder weglässt, somit eine Größe von mindestens 12 Bytes. Durch das Kapseln in UDP (8 Bytes) und IP (mindestens 20 Bytes) ist alleine der Header eines vollständigen Pakets mindestens 40 Bytes groß (siehe dazu Abbildung 3.10 auf der nächsten Seite). Aus diesem Grund gibt es für RTP eine "Header-Komprimierung", welche hier aber nicht im Detail besprochen wird (detaillierte Informationen können z.B. unter [Bad05] gefunden werden).

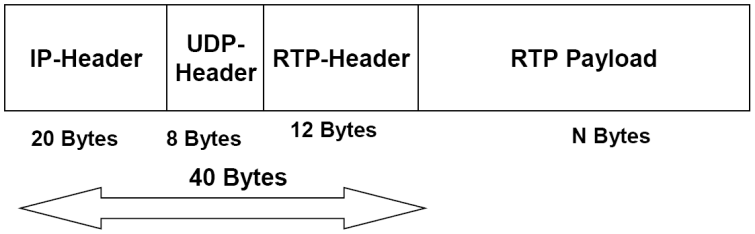

Abb. 3.10: Ein vollständiges RTP-Paket [Bad05]

3.4.1.2 RTCP

RTCP dient, wie bereits erwähnt, der Überwachung von RTP-Verbindungen. Die Hauptaufgaben und -funktionen, lassen sich in folgender Liste zusammenfassen:

- Überwachung der Übertragungsqualität
  Durch den periodischen Austausch der RTCP-Pakete Sender Reports und Receiver Reports wird die Qualität der Übertragung überwacht und gegebenenfalls eine Anpassung (z.B. Reduktion der Datenrate) vorgenommen.

- Unterstützung von Mehrpunkt-Kommunikation ("Konferenz-Schaltungen")
  Durch das periodische Austauschen von Status-Informationen werden Teilnehmer an einer "Konferenz-Schaltung" darüber informiert, wenn Teilnehmer hinzukommen oder ausscheiden. Dies kann bei öffentlichen "Konferenzen" von Nutzen sein.

- Identifikation der Quelle
  Während in einem RTP-Paket die Quelle nur durch den SSRC-Eintrag festgelegt ist, welcher aber durch einen Mixer geändert werden kann, überträgt RTCP einen CNAME (kanonischen Namen), der immer gleich bleibt. Die Übertragung des Namens erfolgt mit einem SDES-Paket (Source Description)

Neben den bereits erwähnten "Sender Reports" (SR) und "Receiver Reports" (RR) wurde im RFC 3611[FCC03] der so genannte *Extended Report* (XR) eingeführt, der mehrere "Report Blocks" enthalten kann, die verschiedene Parameter, die z.B. für VoIP relevant sind, übermitteln kann.

3.4.2 Einführung in H.323

Hinter dem Kürzel "H.323" verbirgt sich ein Rahmenstandard der ITU-T, der mehrere Unter-Standards in sich vereint. Aus diesem Grund wird H.323 auch im Englischen als

"Umbrella-Standard" oder "Framework" bezeichnet.

Der eigentliche H.323-Standard[Int00] beschreibt vor allem das Zusammenspiel der Standards H.225.0[Int96] und H.245[Int98], die auch unter dem Synonym *H.323-SIG* bezeichnet werden, da sie für die Signalisierung verantwortlich sind. Des weiteren wird noch das Zusammenspiel mit RTP bzw RTCP (siehe 3.4.1 auf Seite 32) beschrieben, da RTP als Transport-Protokoll Anwendung findet.

3.4.2.1 Grundbegriffe von H.323

Im H.323-Standard gibt es für spezielle Komponenten eines "H.323-Systems" auch - entsprechend der Aufgaben - spezielle Bezeichnungen. Die im Standard genannten Komponenten sind nachfolgend aufgezählt und zur besseren Veranschaulichung stellt Abbildung 3.11 ein Netzwerk mit den H.323-Komponenten dar, die dort entsprechend platziert sind.

- Terminals - *ein "Terminal" ist die Bezeichnung für H.323-Endgerät, also z.B. ein VoIP-fähiges Telefon nach H.323-Standard*
- Gatekeeper - *der "Gatekeeper" ist eine zentrale Komponente eines H.323-Systems und dient der Kontrolle einer Gruppe von Terminals sowie der Unterstützung von QoS (Quality of Service)*
- Zone - *eine "Zone" ist eine Menge an Terminals, die von einem Gatekeeper verwaltet wird. Man kann eine Zone auch als "H.323-Subnetz" sehen*
- Gateway - *ein "Gateway" stellt ein Art Brücke in ein anderes Medium (z.B. PSTN, ISDN) dar*
- MCU - *eine "Multipoint Control Unit" (MCU) dient zur Unterstützung von Konferenzschaltungen.*

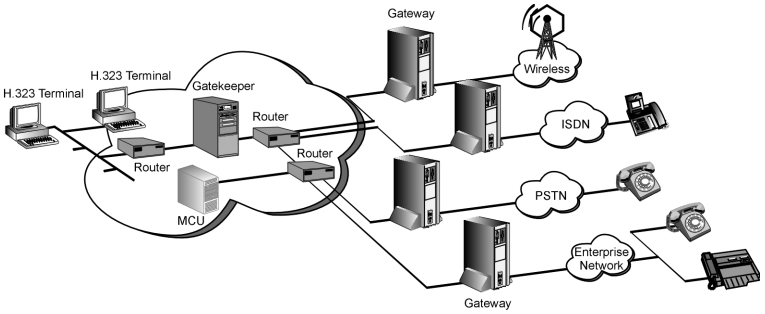

Abb. 3.11: Netzwerk mit H.323-Komponenten [rad98]

3.4.2.2 Aufgaben und Betriebsmöglichkeiten des H.323-Gatekeeper

Der H.323-Gatekeeper stellt die zentrale Verwaltungseinheit einer H.323-Zone dar. Zu seinen Aufgaben zählen:

- Verwaltung der Adressen der Terminals - *ein Gatekeeper verwaltet eine Tabelle, in welcher die Zuordnungen von Telefonnummern der Terminals zu den tatsächlichen IP-Adressen*
- Bandbreiten-Management - *da die Bandbreite eines Netzwerks limitiert ist, überwacht ein Gatekeeper den Bandbreitenverbrauch. Konkret geschieht dies dadurch, dass Terminals, die eine Verbindung aufbauen wollen vom Gatekeeper eine "Amtsleitung" (also die Erlaubnis eine Verbindung aufzubauen) holen müssen*
- Autorisierung - *zusätzlich zur Verwaltung der Bandbreite kann ein Gatekeeper noch gezielt Anrufe erlauben bzw ablehnen*

Wie aus dieser Liste ersichtlich ist, ist ein Gatekeeper die zentralste Komponente bei H.323. Aus diesem Grund werden diese in großen Netzwerken redundant ausgelegt. In diesem Fall bilden mehrere physikalische Gatekeeper einen logischen Gatekeeper, der die H.323-Zone verwaltet. Man spricht dann in so einem Fall von einer "Gatekeeper-Wolke".
Ganz im Gegensatz dazu kann in kleineren Netzwerken auch ein Gatekeeper entfallen. In diesem Fall muss jedes H.323-Terminal eine eigene Kopie der Tabelle der Zuordnungen von seinen Kommunikationspartnern zu IP-Adressen besitzen.

3.4.2.3 H.323-Domänen

Wenn man sehr große Netzwerke betrachtet, so findet man in diesen typischerweise mehrere H.323-Zonen (z.B. eine Zone pro Standort). Da nun mehrere H.323-Zones existieren würden und diese separat zu administrieren wären, schließt man diese zu einer einzigen administrativen Einheit, einer so genannten *H.323-Domäne*, zusammen.

Dieses Konzept ist angelehnt am Konzept von IP-Subnetzen und "autonomen Systemen" (*autonomous system* - AS) bei Routing-Protokollen, um eine Gleichschaltung zu erleichtern. Während beim Routing zwischen verschiedenen AS so genannte *Border-Gateway*-Protokolle (abgekürzt *BGP*) zum Einsatz kommen, verwendet H.323 *Border Elements* (BEs). Ein BE ist üblicherweise eine Funktion, die in einen Gatekeeper integriert ist und ihn um Methoden zum Telefonnummern-Austausch mit anderen Domains erweitert. Dies ist wiederum eine Analogie zu den BGP, die beim IP-Routing verwendet.

3.4.2.4 Aufgaben von H.323 und Einordnung in den TCP/IP-Stack

Nachfolgend soll ein kurzer Überblick über H.323 und dessen Aufgaben, die durch Subprotokolle gelöst sind, gegeben werden. Ein weiterer wissenswerter Punkt ist die Einordnung von H.323 bzw dessen Subprotokolle in den TCP/IP-Stack. Einen guten Überblick in dem Zusammenhang liefert Abbildung 3.12.

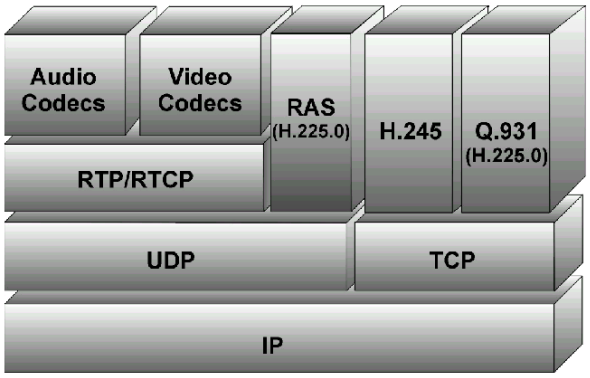

Abb. 3.12: Einordnung der H.323-Subprotokolle in den TCP/IP-Stack [rad98]

Der Aufgabenbereich von H.323 ist weitläufig und umfasst unter anderem:

- RAS-Control - *RAS (Registration, Admission and Status) umfasst die Kommunikation zwischen Gatekeeper und Terminal. Konkret geht es hierbei um die Registrierung ("Registration") von Terminals am Gatekeeper (ein Terminal kann nur eine Verbindung aufbauen, wenn es sich vorher am Gatekeeper registriert hat), Bewilligung("Admission") von VoIP-Verbindungen von VoIP-Verbindungen der Terminals und die Überwachung des Status von Verbindungen. Hierzu wird das Protokoll H.225.0 verwendet*

- Call Control - *bei der Call Control(Anruf-Signalisierung) handelt es sich um die Prinzipien mit denen ein H.245.0-Steuerungskanal zwischen zwei Terminals aufgebaut wird. Hierzu werden H.225.0-Nachrichten (welche vom Standard Q.931 übernommen werden) via UDP ausgetauscht. Der H.245.0-Steuerungskanal wird auf Basis von TCP realisiert und entspricht weitestgehend dem D-Kanal beim ISDN*

- H.245 Control - *Hierbei handelt es sich um die Steuerung des Auf- und Abbaus von logischen Datenkanälen (ähnlich den B-Kanälen des ISDN) zur Datenübertragung. Die logischen Datenkanäle werden dann durch RTP-Verbindungen realisiert*

- Übermittlung von Echtzeitdaten (z.B. Sprache) - *mittels RTP-Verbindungen werden die eigentlichen Sprachdaten transportiert.*

3.4.2.5 Verbindungsaufbau/Signalisierung in H.323

Wie auch beim PSTN oder ISDN muss bei H.323 für das Zustandekommen einer Sprachverbindung vorher ein entsprechender Signalisierungsaufwand betrieben werden. Dies geschieht (vereinfacht dargestellt) nach folgendem Schema (siehe dazu auch Abbildung 3.13 auf der nächsten Seite):

1. es wird zwischen den beiden Teilnehmern eine TCP-Verbindung aufgebaut
2. über diese TCP-Verbindung wird dann mittels H.225.0 die Q.931-Nachricht *SETUP* gesandt
3. das angerufene Terminal antwortet dann mit der Nachricht *ALERTING*. Gleichzeitig wird dem Benutzer (z.B. durch Klingeln) ein eingehender Anruf mitgeteilt
4. nachdem der Anruf angenommen wurde, bestätigt das angerufene Terminal dies durch die H.225.0-Nachricht *Connect*. Damit ist nun ein entsprechender H.245-Steuerungskanal aufgebaut.
5. nach dem Aufbau des H.245-Steuerungskanals wird mittels RTCP die RTP-Verbindung (der so genannten *Medienkanal*) aufgebaut und die Sprachübertragung kann beginnen.

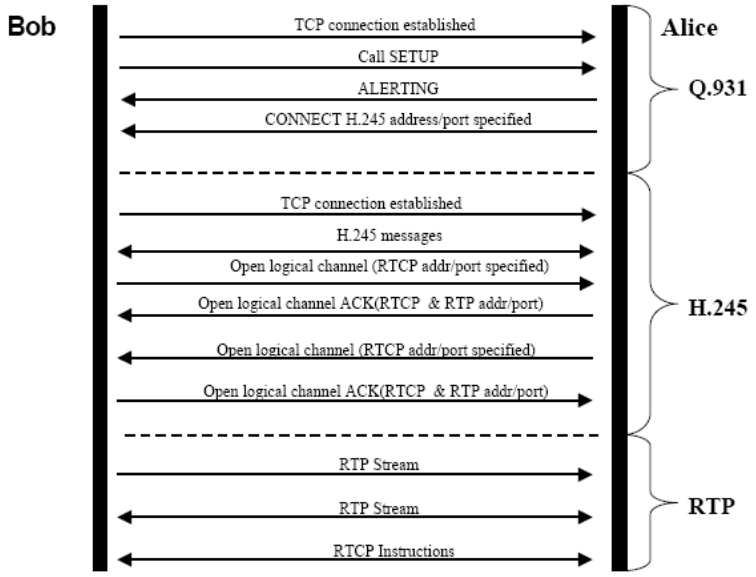

Abb. 3.13: Verbindungsaufbau/Signalisierung bei H.323 [KWF05]

3.4.3 Einführung in SIP

Die Abkürzung SIP steht für das "Session Initiation Protocol" ist ein von der IETF definierter Standard und dient - im Gegensatz zu H.323 - rein der Signalisierung bei VoIP. Die Ursprungsversion von SIP wurde im RFC 2543[HSSR99] im Jahr 1999 vorstellt. Mittlerweile wurde SIP laufend erweitert und verbessert und die aktuelle Version aus dem Jahr 2002 findet sich in den RFCs 3261 bis 3265 [RSC+02, RS02b, RS02c, RS02a, Roa02]. Zu SIP gibt es gute Literatur, wie z.B. [Joh04], [TW05] und [Kan05].

Die Hauptaufgabe von SIP ist der Aufbau von *Sitzungen*, die als logischer Kanal für RTP-Verbindungen dienen. SIP dient damit rein der Signalisierung (ähnlich wie das D-Kanal-Protokoll von ISDN). Während H.323 Binär-Nachrichten verwendet hat die IETF bei der Entwicklung von SIP die Nachrichten als Text-Nachrichten, ähnlich denen von HTTP[BLFF96, FGM+97] und SMTP[Pos82, Kle01], konzipiert. Des Weiteren ähneln SIP-Adressen sehr stark E-Mail-Adressen. Durch diese Maßnahme war die Akzeptanz von SIP anfangs viel höher, da es einfacher zu verstehen bzw. die Abläufe nachzuverfolgen waren.

Der zweite wesentliche Punkt, der bei der Entwicklung von SIP berücksichtigt wurde, ist die Erweiterbarkeit bzw. die Offenheit für zukünftige Erweiterungen. Dadurch wurde mit SIP ein Standard geschaffen, der weithin akzeptiert wird und "zukunftssicher" ist. Eine zukünftige Anwendung von SIP könnte die Verwendung bei UMTS bzw allgemein in der

mobilen Kommunikation sein.

Als Transportprotokoll für die eigentlich Daten (Sprache, Video, usw) dient - wie auch schon bei H.323 - das RTP. Die Aushandlungen, welche Medien übertragen werden sollen bzw können, deren Codierung, usw wird durch das so genannte *Session Description Protocol*(SDP), welches ein Teil von SIP ist, gelöst.

Weitere Merkmale von SIP sind:

- Request/Response-Prinzip - *Bei der SIP-Kommunikation zwischen 2 Endgeräten/Rechner sendet der Initiator einen so genannten (SIP-)Request auf den der Empfänger mit einer so genannten (SIP-)Response antwortet. Auf diesem Prinzip baut jede SIP-Kommunikation auf*
- Adressierung via URLs - *Ähnlich wie bei Web-Anwendungen werden bei SIP die Teilnehmer über einen URL(Uniform Resource Locator) adressiert. Dies ermöglicht/erleichtert die Integration von SIP in bestehende Internet-Anwendungen*
- Syntax der Nachrichten wie bei HTTP - *SIP-Nachrichten sind nach der gleichen Syntax wie HTTP/1.1 aufgebaut*
- Lokalisierung/Namens-Auflösung via DNS - *Die Umwandlung eines Namens bzw eines URLs zu einer IP-Adresse erfolgt mittels des weitverbreiteten DNS(Domain Name System)*
- Authentifizierung - *Maßnahmen zur gegenseitigen Authentifizierung von kommunizierende Endgeräten (Rechner, IP-Telefonen) sind direkt in den SIP-Standard eingebaut*

Wie man aus obiger Aufzählung erkennen kann, wurde bei der Entwicklung von SIP sehr stark auf die Wiederverwendung von bereits etablierten und gebräuchlichen Diensten (z.B. DNS) und Protokollen (z.B. HTTP) zurückgegriffen, um die Integration zu erleichtern bzw die Akzeptanz vorab zu erhöhen.

Verbindungsaufbau mittels SIP

Ein einfacher Verbindungsaufbau mittels SIP geschieht durch den Austausch von wenigen Nachrichten. Als Beispiel wird von zwei IP-Telefonen ausgegangen, die mit den Adressen `user1@earth.space` (für den Teilnehmer 1) und `user2@moon.space` (für den Teilnehmer 2) benannt sind und direkt über ein IP-Netz (z.B. das Internet) verbunden. Die folgende Vorgangsweise bzw Nachrichtenfolge tritt dabei auf (siehe dazu Abbildung 3.14 auf der nächsten Seite):

1. Das IP-Telefon von Teilnehmer 1 schickt einen SIP-Request mit der Nachricht *INVITE* an die Ziel-Adresse von Teilnehmer 2. Dieser Request enthält zusätzlich noch Informationen über die gewünschte Sitzungsparameter (z.B. Codierung), die mittels dem SDP-Protokoll beschrieben sind

2. Sollte das IP-Telefon von Teilnehmer 2 in der Lage eine Verbindung herstellen zu können (d.h. die Einstellungen, die per SDP mitübertragen wurden, sind kompatibel), so startet dieses ein Klingeln und antwortet dem Gerät von Teilnehmer 1 mit einer SIP-Response *180 Ringing*

3. Sobald der Teilnehmer 2 den Hörer abgehoben hat, wird die SIP-Response *ACK* an Teilnehmer 1 gesandt und die RTP-Sitzung ist aufrecht. Ab jetzt können Sprachdaten ausgetauscht werden (d.h. es kann telefoniert werden)

4. Sobald einer der Teilnehmer seinen Hörer auflegt sendet sein IP-Telefon den SIP-Request *BYE* an den Kommunikationspartner

5. Dieser bestätigt mit der SIP-Response ACK und die Verbindung ist vollständig beendet

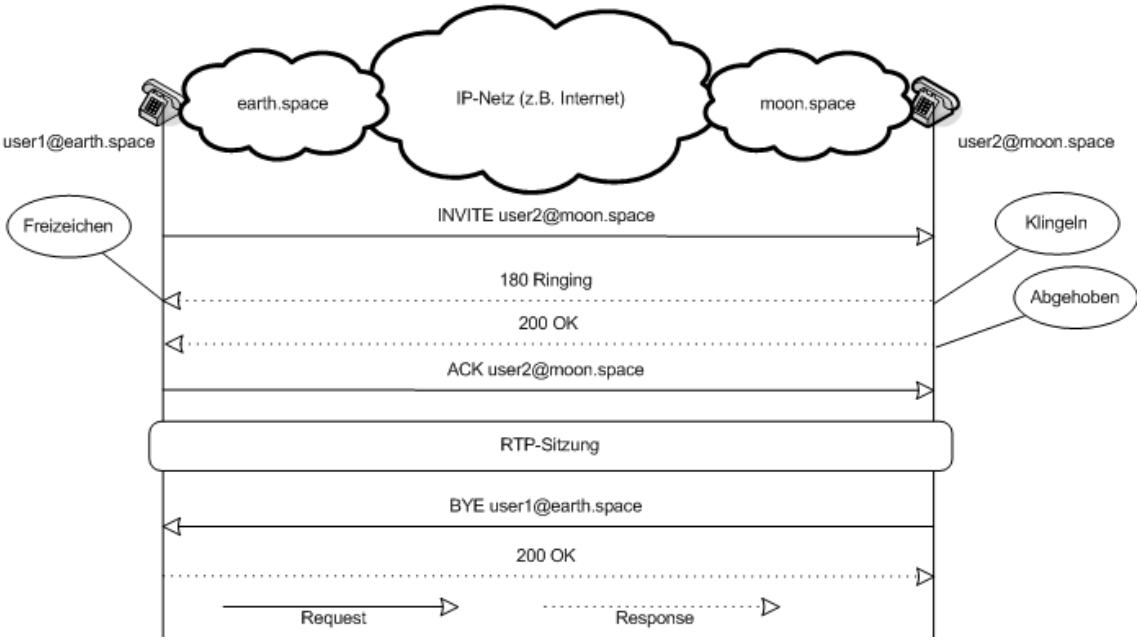

Abb. 3.14: einfacher Verbindungsaufbau bei SIP [Bad05]

SIP-Server

Da bei SIP nicht immer von einer direkten Verbindung der beiden Kommunikationspartner ausgegangen werden kann und für erweiterte Aufgaben, sind im SIP-Standard mehrere Server-Typen spezifiziert:

- *Registrar*, oder auch *Location-Server* genannt
- *Proxy-Server*
- *Redirect-Server*

Einsatz eines Registrars/Location-Servers

Die Funktionsweise eines Location-Servers ist im Wesentlichen die simple Zuordnung einer SIP-Adresse, welche einer E-Mail-Adresse ähnelt, zu der IP-Adresse, unter welcher der Teilnehmer gerade erreichbar ist. Nach dem Einschalten/Starten eines SIP-Geräts/-Clients registriert sich dieser beim Location-Server/Registrar und gibt diesem seine aktuelle IP-Adresse bekannt. Dadurch ist es möglich, dass die gleiche SIP-Adresse an verschiedenen Standorten und von verschiedenen Geräten genutzt wird und somit eine durchgängige Erreichbarkeit gewährleistet ist.

Einsatz eines Proxy-Servers

Da von einer E-Mail-Adresse nicht direkt auf eine IP-Adresse geschlossen werden kann, übernimmt ein Proxy-Server diese "Vermittler-Funktion". Anstatt die INVITE-Nachricht nun direkt an den gewünschten Gesprächspartner zu schicken, wird diese an den Proxy-Server geschickt, der für die jeweilige E-Mail-Domain verantwortlich ist. Der Proxy-Server kontaktiert nun seinerseits einen Location-Server, der die genaue IP-Adresse, des gerade verwendeten IP-Telefons des Benutzers kennt. Danach signalisiert der Proxy dem IP-Telefon den Anruf-Wunsch. Wenn das Ziel-IP-Telefon dem Proxy signalisiert, dass der Anruf entgegengenommen werden kann, dann signalisiert der Proxy dem anrufenden IP-Telefon die Bereitschaft des Ziels und die eigentliche VoIP-Kommunikation kann beginnen. Zu beachten ist hier, dass die Kommunikation auch weiterhin über den Proxy-Server stattfindet.

Einsatz eines Redirect-Servers

Ähnlich wie bei vorher beschriebener Situation mit einem Proxy-Server kann ein Redirect-Server anstatt eines Proxy-Servers eingeschalten werden. Im Unterschied zum Proxy-Server ist ein Redirect-Server nur dafür da die aktuell verwendete IP-Adresse eines Benutzers von einem Location-Server abzufragen und danach an das anrufende IP-Telefon weiterzuleiten. Die weitere Kommunikation findet dann zwischen den beiden IP-Telefonen statt.

Weitere Einsatzmöglichkeiten der SIP-Server

Durch die Kombination von mehreren Proxy- und/oder Redirect-Servern können mehrere verschiedene Szenarien entstehen bzw Situationen abgedeckt werden. Ein paar Beispiele hierfür sind:

- Wenn der gerufene Teilnehmer sich (permanent oder zeitweilig) in einer anderen Domain aufhält, als dem Anrufer bekannt ist, dann kann ein Proxy-Server die weitere Kommunikation auch über einen zweiten Proxy-Server oder Redirect-Server laufen lassen - je nach Konfiguration der Domain, in der sich der gerufene Benutzer aufhält.
- Durch den Einsatz eines Proxy-Server ist auch eine Anrufverzweigung möglich, sodass mehrere Telefone des gleichen Benutzers gleichzeitig läuten (z.B. PDA mit WLAN und IP-Telefon am Arbeitsplatz) - das Gerät, bei dem der Benutzer zuerst abhebt, wird dann für die weitere VoIP-Verbindung herangezogen

- Ein weiterer Anwendungsfall ist der Einsatz eines Voice-Mail-Servers. Der Anrufer kontaktiert wieder einen Proxy, der seinerseits versucht das IP-Telefon des Angerufenen zu erreichen. Erhält der Proxy-Server binnen einer vordefinierten Frist keine Antwort, so leitet er das Gespräch an einen Voice-Mail-Server weiter

Wie man sehen kann, bringt ein Proxy-Server viel Flexibilität in ein System, da der Anrufer eine definierte "Gegenstelle" (nämlich den Proxy-Server) hat und so Zusatzdienste bzw Möglichkeiten zur Mobilität eröffnet werden.

3.4.4 Media Gateway Protokolle

Um einem Benutzer eines VoIP-Telefons bzw -Systems die Möglichkeit zu geben einen Teilnehmer eines öffentlichen Telefonnetzes anzurufen bzw umgekehrt, ist eine Form von "Medienübergang" notwendig. Dieser Übergang erfolgt über so genannte "VoIP-Gateways" (oder auch "Media Gateways" genannt).

Bei den Media Gateways gibt es mehrere Formen der Ausprägung (je nach Verwendungszweck) und dementsprechend werden diese dann genannt:

- Trunking Gateways - *dienen zur Anbindung von IP-Netzwerken an öffentliche Telefonnetze (PSTN, ISDN)*
- Residential Gateways - *an diese werden "herkömmliche" Telefone direkt angeschlossen und somit in ein VoIP-Netz integriert*
- Access Gateways - *bieten die Möglichkeit "klassische" Telekommunikationsanlagen an VoIP-Netze anzuschließen*

Die Kontrolle der Kommunikation eines Media Gateways erfolgt dabei durch einen so genannten *MGC*(Media Gateway Controller), welcher über ein spezielles Protokoll mit dem MG kommuniziert und Signalisierungs-Informationen austauscht. Diese speziellen Protokolle, welche "Media Gateway Protokolle" genannt werden, werden nachfolgend kurz vorgestellt.

Ein MG-Protokoll definiert im Wesentlichen Nachrichten und Regeln, um einen Austausch von Sprache mittels RTP via eines Media Gateways zu ermöglichen. Die Steuerung geht hierbei vom MGC aus. Als Signalisierungs-Protokoll für die RTP-Verbindungen kann in

diesem Zusammenhang SIP und H.323 genommen - die Wahl wird durch das verwendete VoIP-Protokoll des VoIP-Netzes, an welches ein MG angeschlossen ist, bestimmt.

Wie auch schon bei den anderen VoIP-Protokollen gibt es auch hier konkurrierende Standards, die im Nachfolgenden genauer erläutert werden sollen.

3.4.4.1 MGCP

Das *Media Gateway Control Protocol*(MGCP) wurde von der IETF entwickelt und ursprünglich im RFC 2705[ADE+99] veröffentlicht. Eine erweiterte Version wurde später als RFC 3435[AF03] publiziert.

Im Zusammenhang mit MGCP wird ein MGC als *Call Agent*(CA) bezeichnet. Zur Kommunikation mit einem CA verwendet MGCP das verbindungslose UDP-Protokoll auf den Ports 2427 (Kommunikation vom CA zum MG) und 2727 (Kommunikation vom MG zum CA). Die Nachrichten sind text-basiert und werden zeilenweise dargestellt. Die Kommunikation zwischen den Media Gateways erfolgt mittels RTP, wobei SDP zur Beschreibung der RTP-Verbindungen verwendet werden.

MGCP kennt eine Reihe von speziellen Begriffen, die in nachstehender Liste dargestellt sind:

- Endpoint - *Ein Endpoint ist ein (physikalischer) Anschluss an einen MG und Quelle und Senke von Sprache*
- Call - *Eine logische Verbindung zwischen zwei Endpoints. Ein Call kann aus mehrere tatsächlichen Connections(Verbindungen) bestehen*
- Package - *Enthält eine Liste von Ereignissen (Events), die an einem Endpoint-Typ auftreten können - diese Informationen werden für die Signalisierung benötigt*

Wenn nun eine Verbindung über ein MG hergestellt werden soll, dann bekommt der MGC mittels des Packages, welches dem Endpoint des MG zugeordnet ist, die Informationen, welche Ereignisse auftreten können, in welcher Reihenfolge diese Ereignisse auftreten können und über welche Ereignisse das MG informiert werden möchte.

3.4.4.2 Megaco

Alternativ zu MGCP gibt es noch das von der ITU-T entwickelte *Megaco*, welches eine größere Akzeptanz und Verbreitung als MGCP genießt. Megaco wurde von der ITU-T als *H.248* und von der IETF als RFC 3015[CGR+00] als Standard publiziert.

Megaco ist - genau wie MGCP - ein Protokoll, welches die Kommunikation zwischen MG und MGC regelt. Die Nachrichten werden dabei nach einem *Command/Reply*-Schema ausgetauscht. *Commands* werden üblicherweise von einem MGC zu einem MG geschickt, worauf eine *Reply*-Nachricht als Antwort zurück kommt. Die Syntax der Nachrichten wird mit dabei der *ASN-1*-Notation ("Abstract Syntax Notation One") spezifiziert.

Megaco definiert für diesen Nachrichtenaustausch kein spezifisches Protokoll und ist völlig unabhängig vom verwendeten Transport-Protokoll. Dadurch, dass Megaco für VoIP konzipiert und entwickelt wurde, ergibt sich im praktischen Einsatz eine Auswahlmöglichkeit zwischen TCP und UDP.

Megaco definiert eine Reihe von spezifischen Begriffen, die in folgender Liste zusammengefasst sind:

- Termination - *Quelle oder Senke eines Bitstroms. Terminations an einem MG dynamisch mit Hilfe von Megaco-Kommandos erzeugt. Zwischen ihnen besteht dann eine logische Verbindung, über welche in weitere Folge die Daten ausgetauscht werden*
- Package - *Beschreibt eine Termination durch Angabe von Eigenschaften (Properties), mögliche Ereignisse (Events) und mögliche Signale (Signals). Einige Terminations können derart konfiguriert werden, sodass sie mehrere Packages unterstützen*
- Context - *Sind 2 oder mehrere Terminations miteinander verbunden, wird diese Assoziation Context genannt. Ein Context beschreibt die Topologie zwischen den Terminations, als auch die übermittelten Medien-Formate, sowie Switching-Informationen, falls mehr als 2 Terminations verbunden sind. Jeder Context besitzt eine eindeutige Identifikations-Nummer (ID)*

Die Kommunikation erfolgt - wie bereits beschrieben - normalerweise von MGC zu MG und ist der des MGCP sehr ähnlich (da Megaco auch als Weiterentwicklung von MGCP gesehen werden kann).

3.5 Beschreibung der Basis-Infrastruktur für VoIP

Nach der ausführlichen Erklärung über den technischen Aufbau von Voice over IP bzw. dessen Komponenten folgt nun die Beschreibung jener Infrastruktur, auf die VoIP aufsetzt.

Betrachtet man die in den vorangegangenen Kapiteln vorgestellten VoIP-Protokolle entsprechend ihrer Einordnung in den TCP/IP-Stack (siehe Abbildung 3.15) so wird deutlich, dass VoIP nur die oberen zwei Schichten des TCP/IP-Modells abdeckt. Darunter befindet sich die so genannte "Basis-Infrastruktur", welche Bestandteil dieses Kapitels ist. Obwohl die Themen hier in sehr vielen Werken (teilweise auch in den eingangs genannten) ausführlichst behandelt wird, sei hier [Dem04] explizit als Referenz-Werk erwähnt.

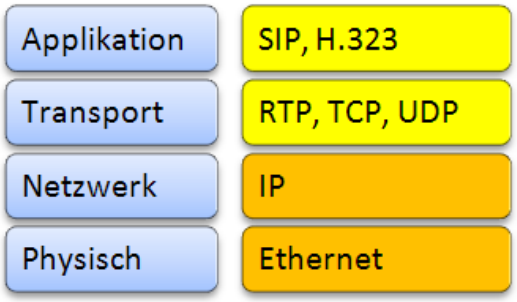

Abb. 3.15: Einordnung der VoIP-Protokolle in den TCP/IP-Stack

3.5.1 Ethernet

Nahezu jedes lokales Netzwerk basiert heutzutage auf der "Ethernet"-Technologie, welche ungefähr in den 1970er Jahren entwickelt und später 1980 von der IEEE unter dem Standard *802.3* spezifiziert wurde.

Im Wesentlichen basiert ein Ethernet-Netzwerk auf einem *shared medium*, einem gemeinsam genutzten Übertragungsmedium, an welches alle teilnehmenden Geräte angeschlossen werden, wie es in Abbildung 3.16 auf der nächsten Seite dargestellt wird. Da ein derartiges Übertragungsmedium nur von einem Gerät gleichzeitig zum Senden genutzt werden kann kommt der so genannte *CSMA/CD*-Algorithmus zum Einsatz, um einen reibungslosen Betrieb zu gewährleisten. Dieser wird in Abschnitt 3.5.1.2 auf Seite 51 noch detaillierter vorgestellt.

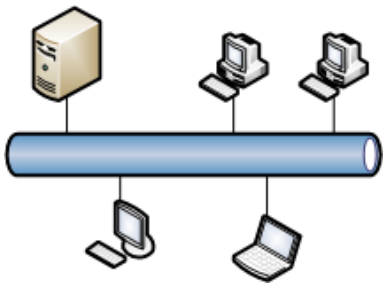

Abb. 3.16: Schematische Darstellung eines Ethernet-Netzwerks

3.5.1.1 Adressierung im Ethernet

Bei der Übertragung im Ethernet wird jedem Gerät eine eindeutige 48 Bit-Adresse zugewiesen. Diese so genannte "MAC-Adresse" (abgeleitet von MAC, dem "Media Access Control"-Teil des Ethernet-Standards) dient zur eindeutigen Identifizierung jedes Geräts, weshalb keine MAC-Adresse doppelt vergeben sein darf. Zu diesem Zweck besteht eine MAC-Adresse aus 2 Teilen (siehe dazu auch Abbildung 3.17):

- einem Organisations-Code (*Organisation Unique Identifier* - OUI)
- einem Wert pro Netzwerk-Interface-Karte (*Network Interface Controller* - NIC)

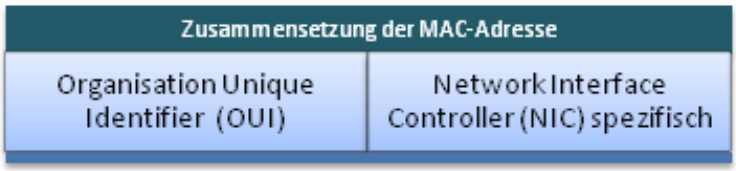

Abb. 3.17: Aufbau einer Ethernet-MAC-Adresse

Jeder Hersteller von Ethernet-Netzwerkkarten erhält nun einen (oder manchmal mehrere) weltweit eindeutige OUI und muss dafür Sorge tragen, dass jede produzierte Netzwerkkarte eine - für diesen Hersteller eindeutigen - Nummer bekommt. Auf diese Weise ist

die MAC-Adresse jeder Netzwerkkarte theoretisch global eindeutig. Dargestellt werden MAC-Adressen nicht mittels Binärzahlen, sondern als Gruppen von Hexadezimalzahlen (6 Gruppen mit jeweils 2 Hex-Zahlen), wie z.B. *00:1A:E9:8F:E8:0A*.

Während in den Anfangszeiten des Ethernets durch das Einbrennen der MAC-Adressen in ein ROM-Modul auf den Netzwerkkarten eine Fälschungssicherheit gegeben war (eine Fälschung hat einen ROM-Tausch nach sich gezogen, der nicht ohne weiters zu bewerkstelligen war), ist es heutzutage sehr simpel die MAC-Adresse unter der sich eine, in einen Rechner eingebaute, Netzwerkkarte meldet bzw. angesprochen fühlt, zu ändern. Nahezu jeder moderne Hersteller hat diese Funktion softwareseitig in die Treiber integriert.

Zusätzlich zur direkten Übertragung an ein Gerät, sieht Ethernet noch den Mechanismus des "Broadcasts" vor. Unter einem Broadcast versteht man einen Ethernet-Frame, der gleichzeitig an alle angeschlossenen Geräte gerichtet ist und bei dem zusätzlich auch keine Kenntnis aller Empfänger-Adressen notwendig ist. Durchgeführt wird ein solcher Broadcast durch Senden eines Frames an eine spezielle Adresse, bei welcher alle Bits auf "1" gesetzt sind (dargestellt als *FF:FF:FF:FF:FF:FF*).

3.5.1.2 CSMA/CD

Wie bereits eingangs erwähnt, kommunizieren angeschlossenen Geräte über ein gemeinsames Medium miteinander. Diese "shared medium" hat aber die Eigenschaft nur eine Übertragung gleichzeitig zu ermöglichen. Um nun einen reibungslosen Betrieb zu gewährleisten wird ein Verfahren mit dem Namen "carrier sense multiple access with collision detection", oder abgekürzt CSMA/CD, verwendet.

Bei diesem Verfahren handelt jedes Gerät nach der gleichen folgenden Vorgangsweise, wenn es Daten übertragen möchte:

1. Medium abhören: "ist das Medium frei?" und dies solange durchführen, bis das Medium frei ist
2. sobald keine Übertragung mehr stattfindet, die eigene Übertragung starten
3. wird eine Kollision erkannt, ein "JAM"-Signal aussenden
4. im Falle einer Kollision (entweder wird diese selbst erkannt, siehe 3, oder ein "JAM"-Signal empfangen): Übertragung abbrechen und eine zufällige Backoff-Zeit warten

5. bei 1 fortsetzen, bis eine erfolgreiche Übertragung durchgeführt werden konnte oder die maximale Anzahl an Versuchen durchgeführt wurde

Diese im Grunde sehr einfache Vorgehensweise garantiert einen reibungslosen Betrieb. Die angesprochenen Kollisionen, die vor allem entstehen, wenn Geräte, die räumlich weit voneinander entfernt stehen, gleichzeitig versuchen Daten zu senden, lassen sich niemals ganz verhindern, aber durch den so genannten Backoff-Algorithmus wird normalerweise eine unmittelbar darauffolgende Kollision wirksam verhindert.

Bei diesem Algorithmus handelt es sich im Wesentlichen um die Vorgabe, dass jedes Gerät, welches eine Kollision erkannt hat, während es gerade beim Übertragen von Daten war, mittels eines Zufallsgenerators eine Zahl ermittelt und die daraus entstehende Zeit wartet, bevor es wieder mit dem Datenübertragungs-Prozess beim Abhören des Mediums beginnt.

3.5.1.3 Ethernet-Frame

Die Übertragungseinheit des Ethernet ist ein so genannter *Frame*). Der Aufbau ist entsprechend des Protokolls sehr simpel und besteht nur aus folgenden 4 Feldern (siehe auch Abbildung 3.18):

1. Ziel-MAC-Adresse
2. Quell-MAC-Adresse
3. Daten-"Nutzlast"
4. Prüfsumme

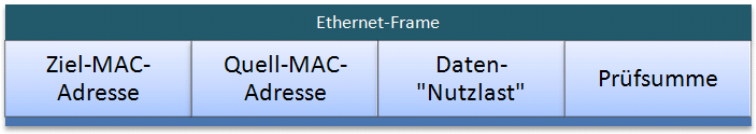

Abb. 3.18: Aufbau eines Ethernet-Frames

Neben der Ziel- beziehungsweise Quell-Adresse des, an der Übertragung beteiligten, Geräts gibt es in einem Ethernet-Frame nur mehr die eigentliche Daten, die übertragen werden sollen und eine so genannte Prüfsumme. Dabei handelt es sich um einen so genannten

CRC-Wert (CRC steht für *Cyclic Redundancy Check*), der aus dem gesamten Ethernet-Frame errechnet wird. Ein Empfänger (oder Gerät, welches den Frame weiter überträgt) kann nun zur Kontrolle diesen CRC-Wert aus dem empfangenen Ethernet-Rahmen bilden. Sollte dieser nicht mit dem, im Frame gespeicherten, mitübertragenen CRC-Wert übereinstimmen, so ist offensichtlich während der Übertragung ein Problem aufgetreten und der Rahmen wird verworfen.

Das Ethernet-Protokoll selbst sieht keinerlei Methoden zur Fehlerkorrektur oder -benachrichtigung vor, d.h. ein fehlerhafter Rahmen wird verworfen, der Sender jedoch nicht informiert. Es ist also Aufgabe der darüber liegenden Schichten (z.B. entweder TCP oder des Anwendungsprotokolls) für eine Fehlerdiagnose und eventuelle Neuübertragung zu sorgen.

3.5.2 Internet Protocol - IP

Direkt über dem Ethernet ist im Schichtenmodell das "Internet Protocol" (kurz IP) angesiedelt. Es ist eines der namengebenden Protokolle für die TCP/IP-Protokollfamilie und aufgrund der Verbreitung des Internets und der damit verbundenen Verbreitung von IP, kann davon ausgegangen werden, dass es heutzutage das am meisten benutzte Übertragungsprotokoll der Welt ist.

Trotzdem oder gerade weil IP relativ einfach konzipiert ist, aber auf der anderen Seite doch mächtig genug, um allen Anforderungen der heutigen Zeit gerecht zu werden, erfreut es sich dieser großen Beliebtheit.

Aktuell am verbreitetsten ist die, in RFC 791[Pos81a] spezifizierte, Version 4 (man spricht hierbei von "IPv4"). Der Nachfolger IPv6 (also die Version 6), ist zwar schon in RFC 2460[DH98] fertig spezifiziert, die Verbreitung geht aber nur sehr schleppend voran. Die Gründe hierfür sind vielfältig, ein sehr wichtiger Faktor ist, dass IPv4 durch den Einsatz von Erweiterungen den momentanen Anforderungen noch genügt und IPv6 zwar sehr interessante Verbesserungen liefern würde, aber diese noch nicht unbedingt notwendig sind oder ihre Wirkung nur in einem IPv6-Netz entfalten können, welche es momentan noch nicht gibt. Eine zentrale Rolle bei der Verbreitung von IPv6 wird den ISPs zugeschrieben, die dies aber im Moment noch sehr spärlich umsetzen.

3.5.2.1 Aufbau eines IP-Pakets

Die Übertragungseinheit des Internet Protocols, ein so genanntes *IP-Paket*, besteht im Wesentlichen aus der Ziel- und Quell-IP-Adresse und zusätzlichen Header-Daten, wie z.B. einer Versionsangabe, der Angabe des Typs des gekapselten Schicht-4-Protokolls, und einigen weiteren. Die genaue Beschreibung ist für das Verständnis der grundlegenden Funktionsweise von IP nicht relevant, weshalb an dieser Stelle darauf verzichtet wird. Eine schematische Darstellung des gesamten Pakets findet sich in Abbildung 3.19.

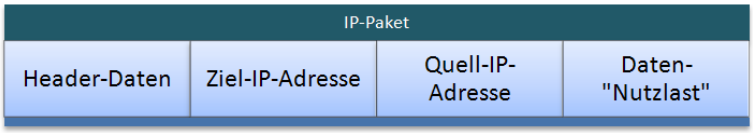

Abb. 3.19: Aufbau eines IP-Pakets

3.5.2.2 Adressierung mittels IP

Zur eindeutigen Identifizierung beim IP-Protokoll dienen 32-Bit-Adressen, welche üblicherweise byteweise, als Dezimal-Zahl durch Punkte voneinander getrennt, angegeben werden. Ein Beispiel hierfür *10.1.2.3*.

Da IP für die Verbindung von Netzwerken konzipiert wurde, hat man bereits in das Adressschema eine Unterscheidung in *Netzwerk-Teil* und *Host-Teil* eingebaut, welche beim Routing (also beim Weiterleiten von IP-Paketen zwischen unterschiedlichen Netzwerken) ihre Anwendung findet.

Um verschieden große Netzwerk-Bereiche im IP-Adressraum festzulegen, hat man bei der Konzeption die so genannten *IP-Adressklassen* eingeführt. Diese geben an wieviele der 4 Bytes für den Host- bzw. Netzwerk-Teil einer IP-Adresse genutzt werden. Daraus resultiert dann gleichzeitig die Anzahl der möglichen Netzwerke pro Adressklasse bzw. die Anzahl der möglichen Hosts in einem bestimmten Netzwerk.

Unterschieden werden die Adressklassen durch jeweils ersten Bits des ersten Bytes einer IP-Adresse bzw. dem daraus resultierende Bereich der entsprechenden Dezimal-Darstellung. Eine übersichtliche Darstellung der IP-Adressklassen, inklusive der Anzahl der verwendeten Bytes für Netzwerk- und Host-Teil findet sich in Abbildung 3.20 auf der nächsten Seite.

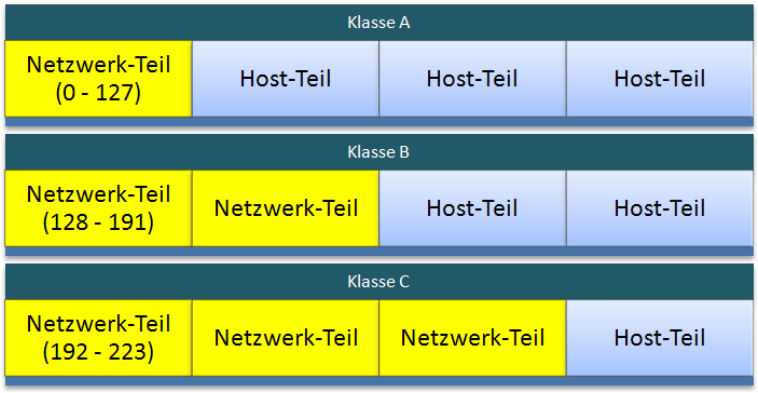

Abb. 3.20: Klassen von IP-Adressen

Zusätzlich zu den abgebildeten 3 Klassen, gibt es noch weitere Klassen für spezielle Typen von Adressen, die hier nicht weiter behandelt werden.

3.5.3 User Datagram Protocol - UDP

Dem Schichtenmodell folgend stößt man nach IP auf das, in RFC 768[Pos80] spezifizierte, *User Datagram Protocol* (abgekürzt UDP), welches die verbindungslose Übertragung von Daten mittels IP ermöglicht. Während IP für den Transport von Paketen zwischen zwei Rechnern gedacht ist, kann UDP, mit dem (später erklärten) Konzept der Ports, zwischen den verschiedenen Anwendungen eines Rechners unterscheiden. Darüber hinaus bietet UDP aber keinerlei Zusatzfunktionalität, weshalb eine Übertragung, genau wie bei IP, verbindungslos und ohne Übertragungsgarantie ist.

3.5.3.1 Aufbau eines UDP-Segments

Da für UDP nur die Angabe der Ports benötigt wird, besteht eine Übertragungseinheit von UDP, ein so genanntes UDP-Segment, wie in Abbildung 3.21 ersichtlich, eben nur aus diesen 2 Feldern und den zu übertragenden Daten.

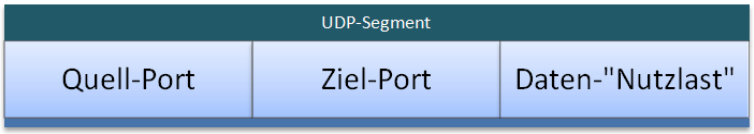

Abb. 3.21: Aufbau eines UDP-Segments

3.5.3.2 Konzept der Ports

Um dies zu gewährleisten wird das Konzept der so genannten "Ports" verwendet. Ein Port ist hierbei eine 2-Byte-Zahl (0 bis 655535), die auf beiden Seiten der Datenübertragung die jeweilige Anwendung des Quell- bzw. Ziel-Rechners beschreibt. Dadurch ist es möglich pro Gerät mehrere voneinander unabhängige Datenübertragungen gleichzeitig durchzuführen zu können.

Die Zuordnung zwischen Port und Anwendung unterliegt keinerlei Kontrolle, d.h. jede beliebige Anwendung kann einen beliebigen Port wählen von dem aus es Daten sendet bzw. empfängt. Es ist dann die Aufgabe des Betriebssystems (bzw. der TCP/IP-Implementierung des Geräts) zu gewährleisten, dass kein Port doppelt vergeben wird und ein in Verwendung stehender Port entsprechend vermerkt wird. Ein und derselbe Port kann niemals von zwei Anwendungen gleichzeitig (wohl aber hintereinander) verwendet werden.

Obwohl diese Freiheit besteht, haben sich im Laufe der Zeit gewisse "well-known-ports" entwickelt, die als Ansprechpunkte für gewisse Applikationen quasi erwartet werden. So reagieren HTTP-Server normalerweise auf Anfragen bei Port 80, SMTP-Server auf Port 25, usw. (eine Liste dieser Ports kann unter anderem bei der IANA nachgelesen werden [IAN08]). Trotz dieser "üblichen Ports" ist es nicht verpflichtend und es kann nicht immer davon ausgegangen werden, dass sich der entsprechende Dienst hinter der zu erwartenden Port-Nummer befindet.

3.5.4 Transmission Control Protocol - TCP

Das, in RFC 793[Pos81b] spezifizierte, Transmission Control Protocol (kurz TCP) ist vom Aufbau dem UDP sehr ähnlich, bietet aber erweiterte Möglichkeiten. Während UDP verbindungslos ist, ermöglicht TCP verbindungsorientierte Übertragungen, bei denen ein virtueller Kanal (oder virtuelle Verbindung) zwischen Sender und Empfänger aufgebaut wird. Zusätzlich bietet TCP noch Methoden zur Flusskontrolle, die eine Überlastung des Empfängers und des Netzwerks verhindern sollen.

3.5.4.1 Aufbau eines TCP-Segments

Um die zusätzlichen Funktionen des Protokolls zu erfüllen, kommen im Vergleich zu einem UDP-Segment, bei einem TCP-Segment mehrere Felder im Protokoll-Header hinzu. Schematisch dargestellt sieht dann ein TCP-Segment wie in Abbildung 3.22 auf der

nächsten Seite aus.

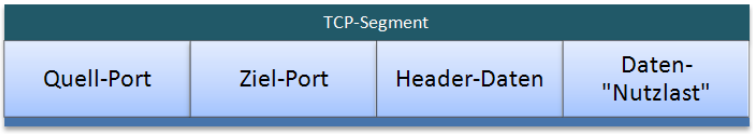

Abb. 3.22: Aufbau eines TCP-Segments

Spezielle Felder des TCP-Headers

Von den im TCP-Header vorhandenen Feldern sind für die folgenden grundlegenden Erläuterungen diese relevant:

- **Sequenz-Nummer** (*sequence number / SEQ*)
  gibt die fortlaufende Nummer des Pakets an
- **Quittierungs-Nummer** (*acknowledgement number / ACK*)
  bestätigt bis zu welchem Paket die Daten gültig empfangen wurden.
- **Flags**:
  - **SYN**, zeigt den Verbindungsaufbau (also den Handshake an)
  - **ACK**, weist auf eine gesetzte Quittierungs-Nummer hin
  - **RST**, dient zur Anzeige, dass die Verbindung zurückgesetzt werden soll
  - **FIN**, zeigt den Wunsch zum Abbau der Verbindung an

3.5.4.2 TCP-Verbindungen

Das wesentliche Merkmal von Übertragungen mittels TCP ist der Einsatz einer virtuellen Verbindung zwischen zwei Gegenstellen. Durch den Einsatz von Sequenz-Nummern gibt es klare Reihenfolge der Pakete und es können Verluste bei der Übertragung erkannt werden. Mit der Quittierungs-Nummer kann dann eine Gegenstelle immer die jeweils höchste Sequenznummer angeben, bis zu welcher die Pakete richtig empfangen wurden (da diese Quittierungs-Nummer nicht immer gesetzt sein muss, weist das ACK-Flag auf ein Vorhandensein dieser hin).

Verbindungsaufbau mittels Drei-Wege-Handshake

Am Beginn jeder Datenübertragung via TCP steht der Aufbau einer virtuellen Verbindung zwischen Sender und Empfänger. Man bezeichnet diesen Vorgang auch als den "Drei-Wege-Handshake" (im Englischen "Three-Way-Handshake"), weil er aus drei Phasen besteht. Rein technisch gesehen dient dieser Handshake zur Übermittlung der Start-Sequenznummern, die in weiterer Folge zur Nummerierung der TCP-Pakete verwendet werden.

Der Ablauf des "Drei-Wege-Handshakes" ist wie folgt: (siehe dazu auch Abbildung 3.23 auf der nächsten Seite)

1. Der Client signalisiert dem Server den Wunsch zum Verbindungsaufbau, indem er diesem ein Paket mit gesetztem SYN-Flag sendet. Dieses Paket erhält bereits eine Sequenz-Nummer.

2. Akzeptiert der Server die Verbindungsanfrage, so antwortet er mit einem Paket, bei dem auch das SYN-Flag gesetzt ist wird. Gleichzeitig wird mittels gesetztem ACK-Flag und der Bestätigungsnummer, welche die nächste zu erwartende Sequenz-Nummer des Clients (also die Sequenz-Nummer des vorigen Pakets um 1 erhöht) trägt, der Empfang des vorigen Pakets quittiert. Dieses Paket hat seinerseits eine Sequenz-Nummer (die aber in keinem Zusammenhang mit der Sequenz-Nummer des Clients steht).

3. Der Client antwortet nun abschließend mit der seinerseitigen Bestätigung des Verbindungsaufbaus (durch Bestätigung des vorigen Pakets mittels ACK-Flag und Quittierungs-Nummer) und beginnt gleichzeitig mit der Datenübertragung. Dieses Paket trägt dann die Sequenz-Nummer, die um 1 höher ist, als die Sequenznummer des vorigen Pakets.

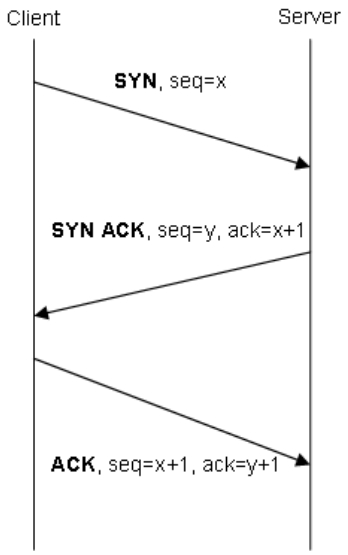

Abb. 3.23: Drei-Wege-Handshake beim Aufbau einer TCP-Verbindung

Abbau einer TCP-Verbindung

Beim Ende einer TCP-Verbindung findet das vorhin erwähnte FIN-Flag Verwendung. Wie aus Abbildung 3.24 ersichtlich ist, funktioniert der Verbindungs-Abbau nach dem Schema des vorhin beschriebenen Drei-Wege-Handshakes, nur mit einem anderen Flag (FIN statt SYN).

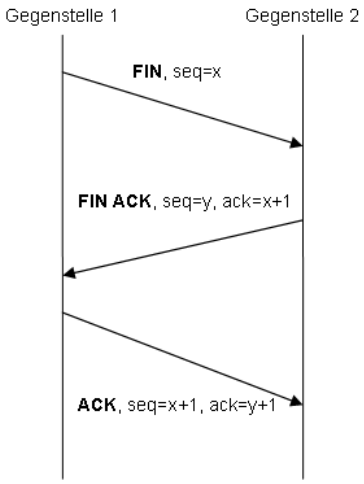

Abb. 3.24: Abbau einer TCP-Verbindung

4. Sicherheitsprobleme von VoIP-Systemen und deren Basis-Infrastruktur

Wie eingangs erwähnt, werden nach den technischen Grundlagen des vorangegangenen Kapitels auch die wichtigsten Angriffsmöglichkeiten gegen VoIP-Systeme behandelt. Dies stellt die Anwendung der allgemeinen Beschreibungen von Kapitel 2.3 auf Seite 10 auf VoIP-Systeme dar. Standard-Literatur ist hierbei spärlich zu finden, aber gerade auf Konferenzen finden sich sehr gute und ausführliche Präsentationen, wie z.B. [Ark02b], [ASRS01] und [Sch05]. Des Weiteren wären noch die Bücher [EC06] und [BSI05b] hervorzuheben.

4.1 Angriffe auf die Basis-Infrastruktur für VoIP

Ein Angriff auf ein VoIP-System beinhaltet neben dem direkten Angriff auch einen Angriff auf die Basis-Infrastruktur, in welche das angegriffene System eingebettet ist. Da IT-Infrastrukturen eine wesentlich längere Entwicklungs- und Einsatzgeschichte als VoIP haben, gibt es hierbei auch etablierte Methoden und Abwehrmaßnahmen (welche in Kapitel 5 ab Seite 67ff. behandelt werden).

Da ein Angriff auf eine bestimmte IT-Infrastruktur meist sehr speziell auf die Anfälligkeiten und Schwachstellen der verwendeten Systeme zugeschnitten ist und von diesen abhängt, würde eine umfassende Beschreibung der Sicherheitsprobleme beziehungsweise Angriffsmöglichkeiten den Rahmen einer einzelnen Arbeit weit sprengen, weshalb im folgenden Kapitel sämtliche Beschreibungen so allgemein wie möglich gehalten sind.

4.1.1 Man-in-the-Middle-Attacken (MitM)

Die so genannten Man-in-the-Middle-Angriffe (abgekürzt *MitM*, siehe dazu auch Kapitel 2.3.3 auf Seite 14) bilden, neben den DoS-Angriffen, die zweite große Gruppe von Angriffstypen. Im Kontext von IT-Infrastrukturen unterscheidet man hierbei zwischen

1. Angriffen innerhalb des internen Netzwerks (LANs)

2. Angriffen auf Datenströme außerhalb des internen Netzwerks (im Internet)

4.1.1.1 MitM im LAN

Wenngleich eine MitM-Attacke in der Theorie sehr einfach wirkt, so ist sie in der Realität nicht ohne Weiteres durchzuführen.

Die erste und wichtigste Voraussetzung für einen Angriff ist der direkte Zugriff auf das anzugreifende LAN. Beim weiteren Vorgehen geht es nun darum, an den Datenstrom, welcher angegriffen werden soll, zu gelangen. In einem LAN wird, außer bei sehr großen Netzwerken, hauptsächlich auf den Einsatz von Switches und, in älteren Netzwerken, Hubs zurückgegriffen. Bei Hubs kann sämtlicher Datenverkehr, welcher diesen passiert, sofort und ohne Weiteres Zutun mitverfolgt werden, während bei Switches dieser angegriffen und manipuliert werden muss.

Ein beliebter Angriffspunkt von Switches ist die Tatsache, dass diese durch Mitverfolgen des Verkehrs "lernen", an welchen Ports welche Geräte angeschlossen sind und so die Informationen zielgerichtet weiterleiten. Überlastet man nun einen Switch, indem man an einem Port tausende Ethernet-Frames mit (gefälschten) Absenderadressen versendet, überlastet man die interne Tabelle der Port-Geräte-Zuordnungen, wodurch ein Switch auf den Notfallbetrieb als Hub umschaltet, in dem sämtlichen Verkehr an sämtliche Ports weitergeleitet wird und ein Mitlesen problemlos möglich ist.

4.1.1.2 MitM im Internet

Wie auch bei einem Man-in-the-Middle-Angriff in einem lokalen Netzwerk erfordert eine MitM-Attacke den Zugriff auf die richtigen Geräte beziehungsweise Netzwerksegmente, um überhaupt an den Datenstrom zu gelangen. Die Schwierigkeit hierbei ist, dass im Gegensatz zu den meisten LANs, wo der Weg der Daten meist (relativ) genau vorhersehbar ist, dies im Internet nicht möglich ist. Pakete der absolut gleichen Übertragung können theoretisch unterschiedliche Wege durch das Internet nehmen.

Wie man nun sehr leicht erkennen kann, ist die beste Möglichkeit an den Datenstrom des Opfers zu gelangen direkt beim Eintritt in das Internet. Diesen Punkt müssen zwangsläufig alle Daten auf dem Weg zu ihrem Ziel im Internet passieren. Was sich nun in der Theorie simpel darstellt, scheitert dann in der Praxis an den Sicherheitsvorkehrungen der Internet-Provider, von denen man ausgehen kann, dass sie höchsten Standards genügen, da die Komponenten zur Kundenanbindung bei einem Ausfall den wirtschaftlichen Ruin

des Providers bedeuten würden.

Zusammenfassend bleibt das Resümee, dass eine MitM-Attacke über das Internet von einem beliebigen Punkt aus unmöglich ist und nur durch den Zugriff auf die Systeme des Providers des Opfers möglich ist, welches eine sehr große Hürde darstellt. Sollte das Opfer über eine redundante Internet-Anbindung mit mehreren Providern verfügen, was bei größeren Firmen durchaus nicht ungewöhnlich ist, multipliziert sich die Schwierigkeit entsprechend.

4.2 Angriffe und Schwachstellen von SIP-basierten VoIP-Systemen

Obwohl das Design des SIP-Protokolls viele Schritte der Planung, des Entwurfs und der Revision durchlaufen ist, haben Angreifer und Sicherheits-Experten bei Untersuchungen Möglichkeiten gefunden unerwünschte Verhaltensweisen hervorzurufen und damit den Betrieb eines VoIP-Systems zu stören, welche nachfolgend beschrieben werden.

Im Speziellen wird auf folgende Möglichkeiten näher eingegangen:

- Denial-of-Service-Attacken
- Session-Hijacking
- Identitätsdiebstahl

Während es über technische Spezifikationen und andere Details sehr viele, gute Quellen gibt, ist es im Bereich der Informationen über Sicherheitslücken oftmals schwierig gute Quellen zu finden, die einem gewissen Mindeststandard genügen. Generell werden Details über Sicherheitsprobleme oftmals spärlich zur Verfügung gestellt, um eine Verwendung zu vermindern. Entsprechende Quellen, die für die folgenden Angriffe gedient haben, sind in der Einleitung dieses Kapitels beschrieben.

4.2.1 Denial-of-Service-Attacken (DoS)

Wie bei nahezu jedem System ist auch bei einem SIP-basiertem VoIP-System ein DoS-Angriff möglich. Jedoch kann ein Angreifer zusätzlich zu den "normalen" DoS-Attacken mittels Überlastung auch noch durch Ausnutzen des Protokolls selbst eine "DoS-Situation" herstellen.

4.2.1.1 DoS durch Überlastung

Durch die Freiheiten des Nachrichtenformats von SIP ist sehr leicht möglich gültige Nachrichten zu generieren, die aufgrund ihrer Größe das Ziel-System, wenn sie in entsprechender Anzahl geschickt werden, überlasten. Da die Nachrichten aber in diesem Fall wirklich sehr groß sein müssen wird eine hohe Bandbreite benötigt. In diesem Fall ist eine DDoS-Attacke wahrscheinlicher.

Im Grunde genommen funktioniert eine derartige Überlastung gleich wie die Überlastung eines Web-Servers mit HTTP-Anfragen.

4.2.1.2 DoS durch Versenden von SIP-Nachrichten

Eine weitere Form von DoS-Attacken ermöglicht es gezielt ein SIP-Telefon zu sabotieren. Wenn man Zugriff auf den Datenverkehr auf den Netzwerkverkehr eines SIP-Telefons hat (und damit alle Nachrichten mitlesen kann), kann man durch das gezielte Einschleusen von SIP-Nachrichten den ordnungsgemäßen Betrieb dieses Telefons unterbrechen oder stören.

DoS durch "CANCEL"-Nachrichten

Eine weitere Möglichkeit der Störung einer VoIP-Kommunikation ist das Einspielen von gefälschten SIP-Signalisierungs-Meldungen. Dabei ist jedoch zu beachten, dass eine gefälschte Nachricht nur als Reaktion auf eine gültige Nachricht geschickt werden kann und dabei auf ein exaktes Timing zu achten ist. Aus diesem Grund werden Attacken dieser Form normalerweise automatisiert durch eine Applikation ausführt.

Eine Möglichkeit wäre, unmittelbar nachdem das Telefon eine *INVITE*-Nachricht ausgesandt hat (also eine Verbindung aufbauen möchte) dem Empfänger dieser Nachricht ein *CANCEL* mit dem Absender des zu blockierenden Telefons zu senden. Dies lässt den Angerufenen glauben, dass der Anrufer doch keine Verbindung aufbauen möchte und somit kommt einseitig keine Verbindung zustande. Eine Veranschaulichung dieses Szenarios findet sich unter Abbildung 4.1.

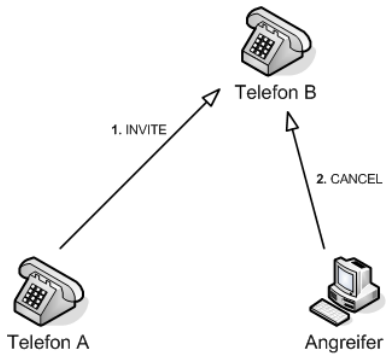

Abb. 4.1: Blockierung aller abgehenden Anrufe eines SIP-Telefons [Ark02a]

Während diese Form der DoS-Attacke nur die abgehenden Gespräche blockiert, kann man

nach dem gleichen Prinzip auch alle ankommenden Anrufe blockieren. Hierbei wird jedem ankommenden INVITE eines anderen Telefons ein CANCEL mit dem (gefälschten) Absender des anrufenden Geräts nachgesandt. Wie schon bei der Variante alle ausgehenden Telefonate zu blockieren, wird hier einseitig das Gespräch beendet. Abbildung 4.2 stellt dieses Szenario dar.

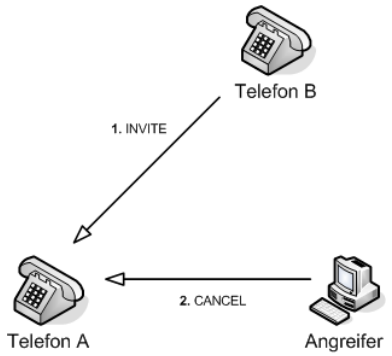

Abb. 4.2: Blockierung aller ankommenden Anrufe eines SIP-Telefons [Ark02a]

DoS durch "BYE"-Nachrichten

Ähnlich der vorher beschriebenen Attacken durch Senden von CANCEL-Nachrichten, kann auch eine BYE-Nachricht, die normalerweise zum Anzeigen des Auflegens/Beenden eines Gesprächs versandt wird, dazu genutzt werden, eine VoIP-Verbindung unbefugt zu beenden. Hierbei wird einfach eine BYE-Nachricht an einen der beiden Teilnehmer mit dem Absender der jeweiligen Gegenstelle geschickt. Während die Attacke mit dem CANCEL beim Telefon, welches unterbrochen wird, keine Spuren hinterlässt, wird der BYE-Angriff jedoch deutlich wahrgenommen.

4.2.2 Session-Hijacking: Unbemerktes Umleiten von Anrufen

Durch die Verwendung von unsignierten SIP-Nachrichten kann es - neben den DoS-Attacken wie unter 4.2.1.2 auf der vorherigen Seite beschrieben - auch zu einer weit schwerwiegenderen Form von Angriff kommen.

Durch das Senden einer Nachricht mit dem Antwort-Code *301 - Moved Permanently* oder *302 - Moved Temporarily*, direkt nach einem versuchten Gesprächsaufbau, kann ein Angreifer vortäuschen, dass das tatsächliche (und legitime) Ziel des Anrufs "umgezogen" ist. Zusätzlich wird mit dieser "Umleitungs-Nachricht" der neue Aufenthalts-Ort (also der Anschluss, unter dem das neue Ziel erreichbar sein soll) mitgesandt.

Diese Information kann nun direkt auf das IP-Telefon des Angreifers zeigen und so wird dem Anrufer eine Verbindung zu einem bestimmten Ziel vorgegaukelt. Auf diesen Weg kann jemand unwissentlich umgelenkt werden. Eine Darstellung dieses Angriffs ist in Abbildung 4.3 zu finden.

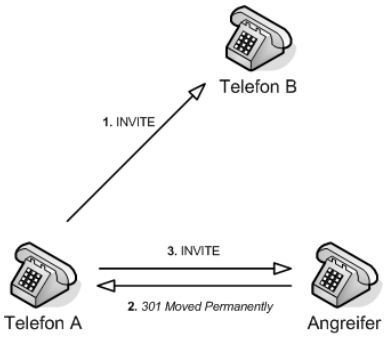

Abb. 4.3: Übernehmen eines Telefongesprächs (Ausgeben als ein anderer Teilnehmer) [Ark02a]

4.2.3 Identitätsdiebstahl durch Fälschen der SIP-Registrierungsinformationen

Obwohl es theoretisch möglich ist SIP ohne Location Service, welcher immer den aktuellen Standort (sprich die IP-Adresse des Clients) kennt, zu betreiben, wird dies, vor allem im Firmen-Umfeld, so nicht gehandhabt, da durch den Einsatz eines derartigen Location Servers die Flexibilität deutlich steigt.

Die Kehrseite der Medaille ist ein möglicher "Identitäts-Diebstahl" einer SIP-Rufnummer. Dazu benötigt ein Angreifer nur die Möglichkeit (z.B. durch erbeutete Zugangsdaten) sich als legitimes SIP-Endgerät auszugeben, welches den aktuellen Standort einer SIP-Rufnummer anmeldet.

Hat der SIP Location Service bereits einen Standort für diese Rufnummer gespeichert, wird diese überschrieben und der neue (gefälschte) Wert auf Anfragen von Anrufern, die diese Endstelle erreichen möchten, zurückgegeben.

Aufgrund der Natur des Angriffs ist das Erkennungspotential hierbei wesentlich geringer als beim Session-Hijacking, wie in Kapitel 4.2.2 auf der vorherigen Seite beschrieben, da das Endgerät des Anrufers keinerlei Möglichkeit hat zwischen echter und gefälschter Information in der Antwort des SIP Location Services zu unterscheiden.

5. Abwehr von Angriffen und Erhöhung der Sicherheit von VoIP-Anlagen

Nachdem in den vorangegangenen Kapiteln ausführlich der Aufbau und die Sicherheitsprobleme von VoIP-Systemen erläutert wurde, widmet sich dieses Kapitel der Abwehr von Angriffen, sowie der generellen Erhöhung der Sicherheit beim Einsatz von VoIP-Systemen.

Wie schon im Kapitel 3 (Seite 17ff.) deutlich wird, besteht ein VoIP-System nicht nur aus direkt für die Telefonie zuständigen Komponenten, sondern basiert auf einer IT-Infrastruktur beziehungsweise ist in diese eingebettet.

Wie bereits mehrfach erwähnt wurde, ist ein VoIP-System von der umgebenden Infrastruktur direkt abhängig und kann nicht ohne diese funktionieren. Daher ist eine explizite Sicherheitsbetrachtung unerlässlich.

Nach dieser Vorstellung von Methoden für die Sicherung der Basis-Infrastruktur werden dann Möglichkeiten zur Erhöhung der Sicherheit von SIP-basierten VoIP-Verbindungen und -Systemen aufgezeigt. Wie auch schon SIP Anleihen bei gut funktionierenden Technologien (wie z.B. HTTP) genommen hat, wurde auch bei der Sicherheit darauf geachtet etablierte und ausgereifte Methoden und Technologien wiederzuverwenden.

An empfehlenswerter Literatur gibt es eine große Auswahl, von denen hier exemplarisch die folgenden Werke genannt und empfohlen werden: [TW05], [Sko02], [Ark02c], [Ark02b], [KWF05], [EC06], [WK05]. Auffallend ist, dass es neben Büchern, die sich rein mit der Sicherheit von IT-Infrastrutkuren beschäftigen (wie z.B. [Sko02]), sämtliche Bücher, die mit die Sicherheit von VoIP behandeln, auch die Sicherheit der Basis-Infrastruktur näher beschreiben.

5.1 Sicherung der Basis-Infrastruktur

Über die Absicherung von VoIP-Systemen nachzudenken, heißt immer auch, über den Schutz der Infrastruktur, in die ein VoIP-Systemn eingebettet ist, nachzudenken. Nur wenn die Basis-Infrastruktur einen gewissen Sicherheits-Standard erfüllt, kann der Schutz eines VoIP-Systems überhaupt gegeben sein.

Wie in Kapitel 2.1 auf Seite 4 dargestellt, ist das Problem der Informationssicherheit immer dagewesen, weshalb schon etliche Methoden entwickelt, angewandt und teilweise wieder überholt wurden. In diesem Kapitel soll daher ein Überblick der mittlerweile gebräuchlichsten Ansätze zur Sicherung gegeben werden. Im Detail sind dies die folgenden Ansätze und Methoden:

- Physische Sicherung
- logische Netzwerktrennung
- Firewalls
- Intrusion Prevention Systeme (IPS)
- Honeypots/-nets

5.1.1 Physische Sicherung

Als elementarste aller Sicherungsmaßnahmen kann der physische Schutz gegen unbefugte Änderung beziehungsweise Verwendung von IT-Systemen gesehen werden.

Durch den direkten Zugriff auf ein Gerät (z.B. ein Rechner oder ein Netzwerk-Gerät) erhält ein potentieller Angreifer die besten Voraussetzungen, um sein Ziel zu erreichen. Im Falle einer Betriebsunterbrechung oder -störung reicht das einfache Abschalten oder Zerstören des Geräts. Sollen Daten gelesen oder verändert werden, so ist dies ebenfalls wesentlich erleichtert, da beim direkten Zugriff auf Speichermedien Kontrollmechanismen (z.B. Authentifizierungen, etc.) umgangen werden.

Die Folgerung aus diesen Bedrohungen ist daher, dass der physische Schutz unumgänglich ist. Dementsprechend ist es schon lange Standard wichtige Netzwerkgeräte und IT-Systeme in separaten Räumen zu versperren, zu denen der Zugang streng reglementiert ist beziehungsweise überwacht wird.

5.1.2 logische Netzwerktrennung

Während bei ISDN- bzw. POTS-Lösungen immer eine eigene Infrastruktur erforderten, könnte VoIP in einem Netzwerk über die selben Komponenten und Kabeln, wie auch das übrige Firmen-LAN, laufen (da in beiden Fällen Ethernet und IP die Basis sämtlichen Daten-Transports bilden). Doch wie sich gezeigt hat ergibt dies auch einen großen Teil der Unsicherheit und Angreifbarkeit von solchen Installationen, weshalb die oben genannten Sicherungsmethoden notwendig sind.

Neben all den Maßnahmen, welche durch das Hinzufügen von Technologien und Komponenten die Sicherheit erhöhen, gibt es noch den Ansatz sämtliche VoIP-Komponenten von der übrigen Netzwerk-Infrastruktur zu trennen. Im Wesentlichen entspricht dies der Herstellung eines ähnlichen Zustands, wie es schon beim Einsatz von herkömmlicher Telefonie der Fall war beziehungsweise ist.

Der große Unterschied ist nun aber, dass durch den Einsatz von entsprechenden Netzwerkgeräten eine rein logische Trennung zwischen VoIP- und "normalem" Daten-Verkehr erfolgen kann, weshalb eine physikalische Trennung nicht mehr zwingend notwendig ist. Ermöglicht wird dies durch den Einsatz der so genannte VLAN-Technologie, die mittlerweile quasi durchgehend verbreitet ist und von allen Geräten, die sich für den Firmen-Einsatz eignen, beherrscht wird.

VLANs

Die Abkürzung *VLAN* steht "virtuelle LANs" und meint eben diese logische Trennung eines LANs in unterschiedliche virtuelle LANs. Geräte im Anschlussbereich eines Netzwerks (also die Switches) sind für die Identifikation und Zuweisung des Datenverkehrs zu bestimmten VLANs zuständig. Verkehr von bestimmten Ports oder bestimmten Geräten (z.B. anhand ihrer MAC-Adresse) wird dabei einem VLAN zugeordnet und kann dieses nur mit Hilfe eines Routers wieder verlassen.

Realisiert wird VLAN durch eine Erweiterung der Ethernet-Frames um ein Feld mit der VLAN-ID (siehe Abbildung 5.1 auf der nächsten Seite). Dieses Feld wird nun beim Transport im Netzwerk-Kern (also zwischen den Switches und Routern) hinzugefügt, wodurch jedes weitere Gerät weiß, welchem VLAN dieser bestimmte Ethernet-Frame zugeordnet ist.

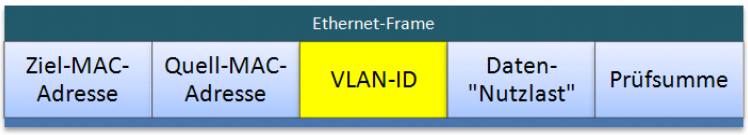

Abb. 5.1: Aufbau eines Ethernet-Frames mit VLAN-Feld

Normale Endgeräte (PCs, VoIP-Phones, etc.) erhalten keine Ethernet-Frames mit VLAN-ID, weil diese vor der Übertragung an diese wieder entfernt wird. Konsequenterweise wird eine VLAN-ID, die von einem Endgerät gesetzt wird, auch ignoriert (wenn der Switch nicht anders konfiguriert ist), weshalb ein "Überspringen" in ein anderes VLAN nicht möglich ist.

Mit Hilfe dieser Technologie passiert der VoIP- und der normale Netzwerk-Verkehr nun die selben Geräte, ohne jedoch für Geräte und Teilnehmer des jeweiligen anderen virtuellen LANs sichtbar zu sein.

5.1.3 Firewalls

Unter einer Firewall versteht man eine Netzwerk-Komponente, die als Filter für den Netzwerk-Verkehr dient. Aus diesem Grund wird eine Firewall auch so in das Netzwerk integriert, dass sämtlicher Verkehr, der überwacht und gegebenenfalls hinausgefiltert werden soll, unweigerlich "durch sie hindurch" übertragen werden muss. Abbildung 5.2 veranschaulicht dies für den Einsatzzweck einer Firewall als Schutz gegen Angriffe von außerhalb des Netzwerks (z.B. aus dem Internet).

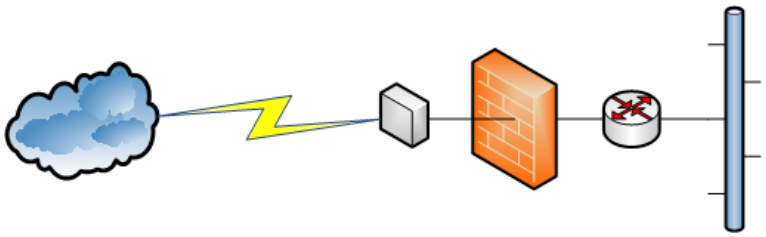

Abb. 5.2: Einbettung einer Firewall in ein Netzwerk

5.1.3.1 Typen von Firewalls

Um die eingangs erwähnte Aufgabe der Kontrolle und Filterung des Netzwerkverkehrs erfüllen zu können, kommen in modernen Firewalls meist mehrere Methoden zum Einsatz. Man unterscheidet hierbei zwischen folgenden Methoden beziehungsweise Typen von Firewalls, welche im Anschluss näher erläutert werden:

- Paketfilter
- Zustandsgesteuerte Paket-Filterung (Stateful Packet Inspection)
- Application Layer Firewall / Application Layer Gateway
- Proxies

Paketfilter

Der einfachste Ansatz für die Kontrolle des Netzwerkverkehrs ist eine Filterung nach den grundlegenden Parametern der verwendeten Netzwerkprotokollen. Im Falle von TCP/IP-Netzwerken sind dies die Informationen, die IP, UDP und TCP zur Verfügung stellen:

- IP-Adresse des Absenders
- IP-Adresse des Ziels
- Quell-Port (bei UDP und TCP)
- Ziel-Port (bei UDP und TCP)

Diese obigen 4 Parameter verknüpft mit einer Handlungsanweisung (was zu tun ist, wenn ein Paket alle Kriterien erfüllt), ergeben nun eine so genannte *Firewall-Regel* (manchmal auch *Filter-Regel* genannt). Als mögliche Aktionen für Pakete stehen normalerweise diese 3 Varianten zur Auswahl:

1. zulassen/weiterleiten (*permit*/*forward*)
2. verweigern/verwerfen (*deny*/*drop*)
3. zurückweisen (*reject*)

Der Unterschied zwischen 2. und 3. ist hierbei, dass bei 3. eine Fehler-Meldung an den Absender versandt wird und bei 2. das Paket ohne weitere Benachrichtigung des Absenders verworfen wird.

Ein Beispiel für eine solche Filter-Regel, die alle beschriebenen Elemente enthält (in der Syntax für eine Cisco PIX-Firewall) ist:

```
access-list beispiel deny ip 172.67.110.0 255.255.255.0 host 10.1.1.1
```

Dieses Kommando erstellt nun eine Regel, welche sämtlichen IP-Verkehr, der aus dem Netzwerk 172.67.110 an den Rechner 10.1.1.1 gesandt wird, abblockt. Der vordere Teil des Kommandos dient lediglich dazu, diese eine Filter-Regel zu einer Liste hinzuzufügen, da diese in Gruppen aktiviert beziehungsweise deaktiviert werden können.

Zusätzlich können mittels Paketfiltern auch auf einfache Art und Weise Abwehrmaßnahmen gegen gefälschte IP-Adressen (wie sie z.B. für DoS-Angriffe eingesetzt werden) getroffen werden. Hierzu erstellt man Regeln, die Pakete mit "unmöglichen" Absender-Adressen einfach verwirft.

Beispiel für Pakete mit offensichtlich gefälschten Adressen wären:

- *ein Paket kommt aus dem Internet, trägt aber die Absende-Adresse aus dem Bereich des internen Netzwerks.*
  In diesem Fall versucht offensichtlich ein Angreifer eine Attacke zu starten, deren ursprüngliche Quelle unbekannt bleiben soll oder Verwirrung zu stiften oder es soll ein Vertrauensverhältnis zwischen 2 Rechnern ausgenützt werden, um gezielt Falsch-Informationen zu einzustreuen.
  Man spricht in diesem Zusammenhang auch von *Ingress Filtering*[Kil00, FS00, FS98].

- *ein Paket wird vom internen Netzwerk ins Internet übertragen, trägt aber eine Absende-Adresse, die nicht dem IP-Adressbereich des internen Netzwerks entspricht.*
  Dies ist für gewöhnlich der Fall, wenn ein Rechner des internen Netzwerks durch einen Virus/Trojaner oder direkt von einem Unbefugtem für einen Angriff benutzt wird.
  Der Fachbegriff für das Entfernen von derartigem Verkehr ist das *Egress Filtering*[Kil00].

Zustandsgesteuerte Paket-Filterung (Stateful Packet Inspection)

Eine spezielle Form eines Paketfilters wäre ein so genannter "zustandsgesteuerter Paketfilter" (auf Englisch "stateful packet inspection", abgekürzt *SPI*). Diese Technologie ist von der Konzeption nur für verbindungsorientierte Protokolle wie TCP zu verwenden, obwohl es mittlerweile für verbindungslose Protokolle wie UDP Ansätze gibt, wie man mit diesen umgeht.

Die wesentliche Funktionsweise von SPI ist nun das Mitprotokollieren von (z.B. TCP-)Verbindungen, die aufgebaut werden. Sämtliche Pakete, die keine Verbindung aufbauen (also einen Three-Way-Handshake einleiten) oder zu einer vorher aufgebauten Verbindung gehören, werden verworfen beziehungsweise zurückgewiesen (je nach Konfiguration der Firewall).

Zusätzlich kann mittels einer SPI-Firewall definiert werden, wer eine Verbindung eröffnen darf. Dies findet z.B. dort Anwendung, wo kein Rechner über das Internet von sich aus eine Verbindung zu einem Rechner des internen Netzwerks eröffnen, sondern höchsten auf eine Verbindungsanfrage eines internen Rechners antworten darf.

Nach diesem Prinzip kann man auch für verbindungslose Protokolle, wie z.B. UDP, eine Art "Pseudo-SPI" betreiben, indem man nur Antworten von außerhalb des internen Netzwerks zulässt, nachdem eine Anfrage von innerhalb des internen Netzwerks an diesen Rechnern gesandt wurde.

Um zusätzlich die Möglichkeiten des Missbrauchs zu verringern arbeiten SPI-Firewalls zusätzlich mit einem Timeout, nach dessen Ablauf eine inaktive Verbindung als abgebaut gilt und keine weiteren Übertragungen zulässig sind. Während dies für UDP-"Verbindungen" unerlässlich ist, wird dieses Verfahren auch für TCP-Verbindungen verwendet, um Verbindungen zu eliminieren, welche nicht ordnungsgemäß abgebaut wurden, aber nicht mehr verwendet werden. Neben der Erhöhung der Sicherheit dient dieser Timeout auch der Minimierung des verwendeten Arbeitsspeichers, da eine Firewall natürlich jede in Verwendung stehende Verbindung speichern muss.

Application Layer Firewall / Application Layer Gateway

Als derzeit komplexeste Form einer Firewall wird eine so genannte *Application Layer Firewall* (oder auch *Application Layer Gateway* genannt) gesehen, welche neben den Möglichkeit von (zustandsgesteuerten) Paketfiltern direkt den Inhalt der Pakete (also die Daten der Anwendungsschicht) kontrollieren und gegebenenfalls manipulieren kann. Zu diesem Zweck muss eine derartige Firewall natürlich über mehr Leistungsfähigkeit verfügen, als

vorher genannte Varianten, bietet aber den größtmöglichen Schutz.

Aufgrund der Komplexität von Anwendungsprotokollen gibt es Application-Layer-Firewalls immer nur für ein bestimmtes Protokoll oder eine Gruppe von artverwandten Protokollen. Beispiele dafür wären

- *Web Application Firewalls*, welche normalerweise die Inspektion von HTTP- und FTP-Übertragungen unterstützen. Oftmals werden in diesem Zusammenhang Viren-Überprüfungen durchgeführt.
- *VoIP Application Gateway*: dienen speziell für die Überwachung und Kontrolle von VoIP beziehungsweise um VoIP-Verbindungen überhaupt zu ermöglichen. Diese werden nachfolgend in Kapitel 5.2.6 auf Seite 97ff. erklärt.

Durch den direkten Zugriff auf die Applikationsdaten sind auch eine Reihe von Zusatz-Überprüfungen möglich, die einer Arbeitsweise als IPS (sieh hierzu Kapitel 5.1.4 auf Seite 75ff.) entsprechen.

Proxies

Das englische Wort *Proxy*, welches "Stellvertreter" bedeutet, ist der Namensgeber eines sehr speziellen Typs von Firewall. Bei dieser Art verbindet sich ein Client nicht direkt mit einem Server, sondern baut über den Proxy eine Verbindung auf (wodurch dieser zum Stellvertreter wird, wie der Name es ausdrückt).

Die Funktionalität eines Proxies ist der einer Application Layer Firewall ähnlich, weshalb es keinen universellen Proxy gibt, sondern nur protokollspezifische. Der wesentliche Unterschied zwischen den beiden Typen ist die "Sichtbarkeit", was bedeutet, dass ein Proxy ganz klar als Stellvertreter eines Clients agiert, während eine Application-Layer-Firewall für den Datenverkehr transparent ist.

Der Vorteil eines Proxies ist auch gleichzeitig sein Nachteil: die explizite Nutzung. Während dadurch auf der einen Seite der Client anonymisiert agieren kann, da ja der Proxy als anfragende Partei bei sämtlichen Server gesehen wird, muss der Client die Verwendung eines Proxies explizit unterstützen und diesen ansprechen.

Ein zusätzliches Problem beim Proxy-Einsatz ist die fehlende Identifikation des Clients für eine Filterung durch eine Firewall. Da auch für die Firewall die Anfragen immer nur vom Proxy kommen und diese nicht wissen kann, welcher Client der Ursprung der Anfrage war, kann diese nicht speziell nach Clients filtern. Diese Filterung müsste dann der Proxy

selbst vornehmen, was einen zusätzlichen Verwaltungsaufwand erfordert, da die Zugriffslisten doppelt konfiguriert werden müssen.

Mittlerweile gibt es aber sogar schon Firewalls, die quasi "das Beste aus beiden Welten" in sich vereinen: die Anonymität für die Clients durch den Einsatz eines Proxies und die Transparenz in der Datenübertragung aus Sicht des Clients. Realisiert wird dies durch das eigenständige Umleiten einer Verbindungsanfrage eines Client auf einen, in der Firewall integrierten, Proxy. Dadurch wird es auch möglich sämtliche Zugriffslisten an einer Stelle zu verwalten, da ja nun Proxy und Firewall in einem Gerät vereint sind.

5.1.4 Instrusion-Prevention-Systeme (IPS)

Während eine Firewall den Zugriff auf IT-Systeme und -Ressourcen generell reglementiert oder verhindert, versucht man mittels so genannter *Instrusion-Detection-Systemen* (kurz: IDS) Angriffe zu erkennen. Ein IDS, welches dann im Bedarfsfall eingreift und für den/die Angreifer den Zugriff sperrt, wird *Instrusion-Prevention-System* (kurz: IPS) genannt. Natürlich ist ein IPS weder ein Garant für die Sicherheit, noch kann eine 100%ige Erfolgsquote gewährleistet werden, aber gegen eine Vielzahl von Attacken bzw. Angriffsmustern bringt ein IPS eine deutliche Erhöhung der System-Sicherheit.

5.1.4.1 Arbeitsweise von IPS

Die generelle Funktion eines IPS ist die Erkennung eines (potentiellen) Angriffs auf ein IT-System, um dann sofort entsprechende Abwehrmaßnahmen (Firewall-Regeln, etc.) einzuleiten und gegebenenfalls direkt die Netzwerk-Administration per E-Mail/SMS/Pager/etc. zu benachrichtigen, sodass im besten Fall eine direkte Zurückverfolgung eingeleitet werden kann.

Die Erkennung von Angriffen erfolgt üblicherweise mittels Signaturen oder Verhaltens-Analysen. Der Einsatz von Signaturen ist hinlänglich von Antivirenprogrammen und dergleichen bekannt und kommt auch in IPS in ähnlicher Weise zum Einsatz (nur in diesem Fall werden Datenpakete mittels Signaturen geprüft).

Schwieriger ist jedoch der Einsatz von Verhaltens-Analysen. Diese basieren meist auf der Definition des "normalen Verhaltens" (z.B. Anzahl, Art und Frequenz der Zugriffe von beziehungsweise auf einen bestimmten Rechner, Typ des Netzwerk-Verkehrs, etc.). Stellt das IPS nun eine Änderung des "gewöhnlichen Verhaltens" fest, so gibt es Alarm und zeichnet verdächtige Aktionen für Nachforschungszwecke auf. Das Problem hierbei, dass

sich Verhaltensmuster ändern können oder ungewöhnliche Ereignisse (z.B. ein sprunghafter Anstieg der Nachfrage der Firmen-Website mit entsprechend höherem Zugriffsaufkommen) zu Fehl-Alarmen führen können, welche - wenn sie häufig auftreten - die Zuverlässigkeit des IPS in Frage stellen und so eventuelle echte Alarme untergehen könnten.

Generell wird zwischen zwei unterschiedlichen Typen von IPS unterschieden:

- netzwerkbasierten IPS (NIPS)
- host-basierten IPS (HIPS)

Während erstere der Absicherung eines Netzwerks beziehungsweise eines Netzwerk-Segments dienen, bieten zweitere nur den Schutz eines spezifischen Hosts (z.B. ein Webserver). Im Folgenden werden beide Typen erklärt und die wesentlichen Eigenschaften und Arbeitsweisen vorgestellt.

Netzwerk-basierte IPS (NIPS)

Netzwerkbasierte IPS (NIPS) dienen der Sicherung und Überwachung von Netzwerken beziehungsweise Teilen (Segmenten) dieser. Um seinen Dienst überhaupt durchführen zu können, muss so ein IPS in dem Netzwerk(-Segment) entsprechend platziert werden. Typischerweise setzt man NIPS in der Umgebung von Übergängen zu anderen Netzwerken (wie z.B. dem Internet) ein. Eine sehr gute Wahl für eine derartige Positionierung wäre hierbei ein Einsatz noch vor dem ersten Router (der meist "Perimeter-Router" genannt wird). Eine entsprechende Darstellung für dieses Szenario wäre die Abbildung 5.3.

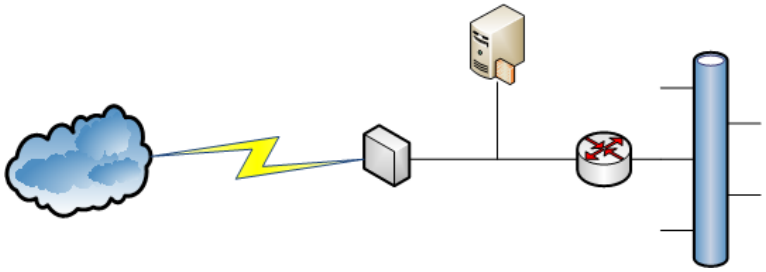

Abb. 5.3: Einbettung eines IPS in ein Netzwerk (vor dem Perimeter-Router)

Beim Einsatz eines NIPS ist vor allem darauf zu achten, dass der Host beziehungsweise das System, welcher das NIPS beherbergt, nicht selbst Ziel eines Angriffs werden kann. Dazu

wird im Idealfall sämtliche Kommunikation des NIPS über das zu überwachende Netzwerk unterbunden, um im Falle eines erfolgreichen Einbruchs weiterhin unsichtbar zu sein.

Host-basierte IPS (HIPS)

Während NIPS der Überwachung von Netzwerk-Verkehr und den ausgetauschten Daten dienen, dient ein host-basiertes IPS (HIPS) dem Schutz eines einzelnen Geräts. Damit erhöht es den Schutz, der schon durch ein NIPS gegeben ist, aber ersetzt dieses nicht.

Üblicherweise arbeiten HIPS mit einer Datenbank, die den Zustand aller wichtigen System-Objekte (Betriebssystem-Dateien, Konfigurationsdateien, etc.) beinhaltet. Sämtliche Veränderungen an diesen werden nun überwacht und im Falle einer unrechtmäßigen Änderung wird diese verhindert beziehungsweise als Angriff gewertet, wodurch entsprechende Maßnahmen eingeleitet werden. Zusätzlich werden sämtliche Protokolle und der System-Zustand laufend überwacht, um eventuelle Angriffe sofort zu erkennen.

Während NIPS vollkommen transparent in ein Netzwerk eingebunden werden können, besteht bei HIPS immer das potentielle Risiko einer Entdeckung beziehungsweise Deaktivierung durch einen Angreifer. Protokolle und Überwachungsergebnisse können im Nachhinein immer noch verändert, gelöscht oder gefälscht werden, weshalb "im Ernstfall" auf ein HIPS weniger Verlass ist beziehungsweise die Informationen, die es sammelt nicht immer absolut akkurat sein müssen. Doch trotz dieses Nachteils verzichtet man nicht gänzlich auf HIPS, sondern nutzt sie als Zusatzmaßnahme zum Einsatz von NIPS.

5.1.5 Honeypots/-nets

Während die vorangegangenen Methoden die Sicherheit von potentiellen Zielen für Angriffe erhöhen, geht man mit dem Einsatz von *Honeypots* einen völlig anderen Weg: Hierbei wird versucht Angreifer auf speziell präparierte Ziele quasi "umzulenken", sie dienen also als Falle. Detaillierte Informationen zu diesem Thema finden sich unter anderem in [Spi02].

Eine erweiterte Technologie, welche auf Honeypots aufbaut, sind so genannte "Honeynets", auf welche hier jedoch nicht näher eingegangen wird. Weitere Informationen zu diesem Thema finden sich in [Hon04].

Die große Schwierigkeit im Einsatz von Honeypots besteht nun in der Vorbereitung der Ziele. Diese müssen auf jeden Fall folgende Kriterien erfüllen:

- Sie müssen für den legitimen Netzwerkbenutzer vollständig unsichtbar sind, sodass dieser auf keinen Fall Dienste von ihnen in Anspruch nimmt oder auch nur legitimer

Verkehr den Weg zum Honeypot findet. Dies ist notwendig, da jede Interaktion mit einem Honeypot als Angriff anzusehen und diese in diesem Fall sofort Alarm auslösen und eine Protokollierung starten.

- Sie müssen für einen Angreifer wie Produktivsysteme wirken, um nicht als Falle zu wirken und einen bestimmten Wert darstellen, sonst werden sie nicht Ziel eines Angriffs und sind damit nutzlos.

- Die wirklichen Produktivsysteme sollten für einen Angreifer am besten unsichtbar sein, sodass die Verwechslung zwischen Honeypot und Produktivsystemen funktionieren kann. Auf jeden Fall muss eine entsprechende Barriere zwischen dem Honeypot und den Produktivsystemen sein, sodass ein Angreifer diese nicht als Plattform für einen Angriff auf die Produktivsysteme nutzen kann.

5.1.5.1 Typen von Honeypots

Generell unterscheidet man aufgrund der Funktionalität 2 Typen von Honeypots:

- *Low-Interaction Honeypots*
- *High-Interaction Honeypots*

Low-Interaction Honeypots

Die einfachste Möglichkeit Honeypots zu etablieren ist die Emulation von Servern beziehungsweise Netzwerken mit Hilfe von Software. Auf diese Weise können komplexe Strukturen als Angriffsziel präsentiert werden, ohne jedoch diese wirklich physisch aufbauen und betreiben zu müssen.

Auf der anderen Seite besteht hierbei das Problem, dass eine derartige Emulation unvollständig ist. Um ein System vollständig zu emulieren muss man die gleichen Ressourcen aufwenden, die für dessen Einsatz nötig wären, wodurch sich keinerlei Vorteil mehr durch die Emulation ergibt.

Aus diesem Grund sind Low-Interaction Honeypots meist so konzipiert, dass sie die wesentlichen bekannten Schwachstellen eines Systems imitieren können, um bei automatisierten Tests als Angriffsziel identifiziert zu werden. Versucht nun ein Angreifer das ausgemachte Ziel zu attackieren wurde schon längst Alarm ausgelöst, eine entsprechende Protokollierung

eingeleitet, um ihn zurückzuverfolgen und der Zugriff auf die Produktivsysteme entsprechend erschwert (z.B. durch Erweiterung der Firewall um zusätzliche Regeln, etc.). Man kann also einen Low-Interaction Honeypot als eine Form eines Intrusion-Detection-Systems bezeichnen.

Eine zweite Funktion, die ein Low-Interaction Honeypot erfüllen kann, ist die Emulation von großen Netzwerk-Strukturen, die einen Angreifer potentiell verlangsamen können, da es ihn Zeit kostet entsprechende Aufklärungsmaßnahmen über alle (emulierten) Netzwerke durchzuführen.

High-Interaction Honeypots

Im Gegensatz zu den vorhin beschriebenen Low-Interaction Honeypots sind High-Interaction Honeypots üblicherweise richtige Server, die für den speziellen Zweck des Einsatzes als Honeypots abgezogen werden. Dies ist eher eine Variante für größere Institutionen, da durch den Einsatz von richtigen Maschinen, die den Produktivsystemen weitestgehend entsprechen sollten, um authentisch zu wirken, der Aufwand höher ist.

Überwacht werden derartige Honeypot-Systeme meistens durch spezielle, tief ins Betriebssystem verwobene, Programme, die für einen Angreifer so gut wie unsichtbar sind. Diese Form birgt jedoch das Risiko doch entdeckt zu werden. Alternativ werden Netzwerkkomponenten eingesetzt, die für jeglichen Netzwerkverkehr transparent und unansprechbar sind, aber jede Kommunikation mit einem Honeypot protokollieren und auch Alarm auslösen können.

Einsatz von Honeypots/-nets

Während andere Technologien, wie Firewalls (Abschnitt 5.1.3 auf Seite 70), IPS (Abschnitt 5.1.4 auf Seite 75) und ähnlichen, primär Einbrüche verhindern sollen, bieten Honeypots zusätzlich zur erhöhten Sicherheit auch Untersuchungsmöglichkeiten von Angriffen. Dieser Umstand kann benutzt werden, um neue Viren und Würmer frühzeitig zu erkennen, beziehungsweise zu studieren. Gerade wegen dieses Umstands, verbunden mit den Kosten und dem Aufwand für den Betrieb von Honeypots/-netzen, ist deren Einsatz üblicherweise auf Sicherheitsfirmen und Universitäten beschränkt. Nur wenige, meist sehr große, Unternehmen, die den Aufwand der Rückverfolgung von Angriffen auf sich nehmen, setzen auch Honeypots ein.

5.2 Sicherung von VoIP-Verbindungen

Wenn man den Begriff einer "VoIP-Verbindung" genauer betrachtet, dann merkt man schnell, dass es so etwas wie "eine" VoIP-Verbindung gar nicht gibt. Vielmehr setzt sich eine Verbindung aus mehreren tatsächlichen Verbindungen unterschiedlichster Protokolle (z.B. TCP, UDP, RTP, ...) zusammen, die unterschiedliche Aufgaben erfüllen (Signalisierung, Medien-Transport, ...). Daher gliedert sich auch das nachfolgende Kapitel in mehrere Teile auf, die sozusagen den einzelnen Teilen einer VoIP-Verbindung entsprechen.

Wie bei TCP/IP-basierenden Diensten üblich unterliegt auch VoIP dem Schichtenmodell, weswegen man immer eine gemeinsame Basis hat: das Internet Protocol - IP. Der erste Abschnitt beschäftigt sich daher mit der Sicherung von IP-Verbindungen im Allgemeinen. Danach geht es weiter mit der Sicherung von Signalisierungs-Informationen, gefolgt von Maßnahmen zur Sicherung des Datenstroms selbst (also der Sprachdaten bei Telefonie).

IP-Verbindungen

Aufgrund der sehr hohen Verbreitung von IP als Netzwerk-Protokoll haben sich einige Methoden zur Sicherung von IP-Verbindungen etabliert. Da sämtlicher Datenverkehr (sowohl Signalisierung, als auch die Sprach-Übertragung) auf IP basiert, bietet eine Absicherung einer IP-Verbindung zwischen zwei Endpunkten (Teilnehmern) einen umfassenden Schutz der gesamten VoIP-Kommunikation zwischen diesen Endpunkten.

Die zwei gebräuchlichsten Methoden diesbezüglich sind:

- IP Security (IPSec)
- Transport Layer Security (TLS)

SIP-Signalisierungs-Meldungen

Mit Hilfe von Signalisierungs-Protokollen werden in einem Telefonsystem eine große Menge an sensiblen Daten ausgetauscht. Diese Daten beinhalten Informationen, die zum Gesprächsauf- und -abbau, sowie zur Verrechnung und weiterem benötigt werden. Diese Informationen geben zwar keinerlei Informationen über den Inhalt des Gesprächs, sind aber mindestens genau so kritisch. Durch Veränderungen kann das System selbst gestört und sogar lahm gelegt werden oder z.B. die Abrechnung behindert oder verfälscht werden (was z.B. bei Telefon-Betreibern fatale Folgen haben könnte). Zusätzlich geht es auch um den Datenschutz der anrufenden und angerufenen Parteien, der gewährleistet sein muss.

SIP bietet auf Protokoll-Ebene einige Möglichkeiten an den Signalisierungs-Datenstrom zu sichern und Authentizität, Vertraulichkeit und Integrität zu gewährleisten. Dazu wurden, um eine "Neuerfindung des Rades" zu vermeiden, folgende - bereits bekannte und beherrschte - Protokolle in SIP integriert:

- HTTP Authentifizierung
- S/MIME

RTP-Medienströme

VoIP verwendet für die Sprachübertragung RTP, welches bereits ausführlich in Kapitel 3.4.1 auf S. 32ff. erklärt wurde. Diese Übertragung findet normalerweise unverschlüsselt und ohne weitere Sicherung statt - das Mithören bzw. Verändern wären also theoretisch möglich. Nachdem dies aber keinesfalls gewünscht ist, gibt es auch für RTP spezielle Mechanismen zur Absicherung der Medienströme, welche in den nachfolgenden Unterkapiteln vorgestellt werden.

5.2.1 IP Security (IPSec)

Als derzeit sicherste Variante für die Daten-Übertragung via IP, wird IPSec angesehen. IPSec ist - entgegen dem ersten Anscheins - kein einzelner Standard, sondern vielmehr ein System, welches durch die RFCs 2401 bis 2412 [KA98c, KA98a, MG98a, MG98b, MD98, KA98b, Pip98, MSST98, HC98, GK98, TDG98, Orm98] beschrieben wird. Im Wesentlichen versucht IPSec die Integrität (siehe 2.2.1.2 auf Seite 6), Vertraulichkeit (siehe 2.2.1.3 auf Seite 6) und die Authentizität (siehe 2.2.1.5 auf Seite 7) bei einer Datenübertragung zu gewährleisten.

IPSec kann im TCP/IP-Schichtenmodell unmittelbar unterhalb der Transport-Schicht angesiedelt werden (siehe Abbildung 5.4 auf der nächsten Seite) und setzt auf IP auf. Das ursprüngliche Entwicklungsziel von IPSec war die Bereitstellung von Sicherheitsfunktionen für IPv6. Da jedoch die Verbreitung von IPv6 langsamer stattfand als angenommen, wurde das Protokoll nachträglich für IPv4 angepasst. Dadurch ist IPSec mit IPv4 eine Art "Zwischenlösung", die bis zur Verbreitung von IPv6 aber noch Bestand hat und weiterhin Verwendung finden wird.

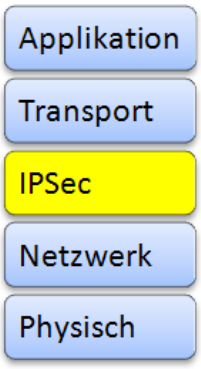

Abb. 5.4: Einordnung von IPSec im TCP/IP-Stack

5.2.1.1 IPSec-Architektur

Die allgemeine Architektur eines IPSec-Systems wird im RFC2401 [KA98c] beschrieben. IPSec setzt direkt auf IP-Pakete auf und erweitert diese um maximal zwei zusätzliche Protokoll-Header, den *Authentication Header (AH)* und den *Encapsulation Security Protocol (ESP)*-Header.

Die Tatsache, dass IPSec nur eine Erweiterung zu IP darstellt, aber die Grundstruktur nicht verändert, ermöglicht es Netzwerk-Komponenten, die IPSec nicht unterstützen, die IP-Pakete trotzdem weiterzuleiten und so den normalen Betrieb aufrechtzuerhalten.

Zum Transport von, mit IPSec geschützten, IP-Paketen unterscheidet man zwischen zwei Arbeitsmodi:

- den Transport-Modus und
- den Tunnel-Modus.

Der Unterschied liegt hierbei im Aufbau der Paket-Ergänzungen und deren Einsatzmöglichkeiten.

Transport-Modus

Der Transport-Modus wird in Endgeräten (Hosts) verwendet, um den Datenaustausch mit anderen Hosts zu sichern. Die dazwischenliegenden Übertragungsgeräte (Hubs, Switches, Router, etc.) sind an der IPSec-Übertragung nicht beteiligt und müssen nur die IP-Pakete weitertransportieren.

Technisch gesehen geschieht dies durch ein Einfügen eines weiteren Headers im IP-Paket. Der originale IP-Header wird nicht verändert - es werden lediglich die Anwendungsdaten authentifiziert und/oder verschlüsselt (siehe Abbildung 5.5).

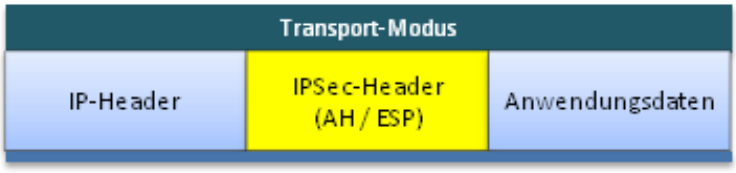

Abb. 5.5: Ein IP-Paket mit IPSec im Transport-Modus

Tunnel-Modus

Der Tunnel-Modus ermöglicht, dass so genannte Security-Gateways (Router oder Firewalls mit IPSec-Funktion) die Sicherung der Datenströme übernehmen. Dies findet vor allem bei Site-to-Site-VPNs Verwendung, da die Endgeräte per LAN vernetzt sind und nur die Kommunikation zwischen den LANs verschlüsselt erfolgen muss. Im Unterschied zu anderen Tunneling-Technologien werden aber keine Layer-2-Informationen mittransportiert, sondern vielmehr die Layer-3-Daten (IP) nochmals in Layer-3-Paketen eingepackt (siehe Abbildung 5.6).

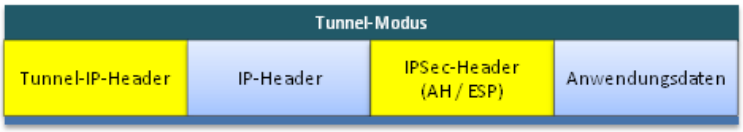

Abb. 5.6: Ein IP-Paket mit IPSec im Tunnel-Modus

5.2.1.2 Authentifizierung mit IPSec

Zur Sicherstellung, dass ein übertragenes Paket auch tatsächlich so beim Empfänger ankommt, wie es beim Sender weggeschickt wurde (ohne Änderungen und ohne ungewollte Wiederholungen) wird bei IPSec der so genannte *Authentication Header* (AH) verwendet, der im RFC2412 [Orm98] spezifiziert ist.

Als AH wird ein spezieller Protokoll-Kopf bezeichnet, der sich zwischen dem ursprünglichen IP-Header und den Layer-4-Elementen, wie z.B. TCP, befindet. Die Nutzdaten (Payload) bleiben durch dieses Voranstellen völlig unberührt, es wird jedoch die Sicherheit erhöht,

da IP-Spoofing und Session-Hijacking-Angriffen entgegengewirkt wird.

Dies wird durch eine Hash-Funktion gewährleistet, die für das zu übertragende Paket einen Hash-Wert berechnet und diesen mit dem Paket mitüberträgt. Da es für einen Angreifer kein Problem darstellen würde bei einer Manipulation des Paketes diesen Hash-Wert ebenfalls zu manipulieren, wird dieser mit einem geheimen Schlüssel bzw. Passwort verknüpft.

Aufbau des AH

Der generelle Aufbau des Headers, welcher vom AH-Protokoll verwendet wird, ist in Abbildung 5.7 dargestellt. Im Wesentlichen besteht dieser aus Längen-Angaben und den Authentifizierungs-Daten. Die Verwendung dieses Headers ist dann in den folgenden zwei Abschnitten genauer beleuchtet.

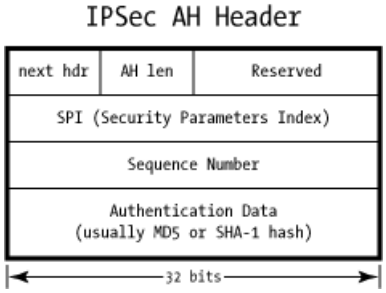

Abb. 5.7: Aufbau des AH-Headers [Fri08]

Verwendung des AH im Transport-Modus

Abbildung 5.8 auf der nächsten Seite zeigt nun die Verwendung des AH für ein IP-Paket, welches ein TCP-Segment beinhaltet.

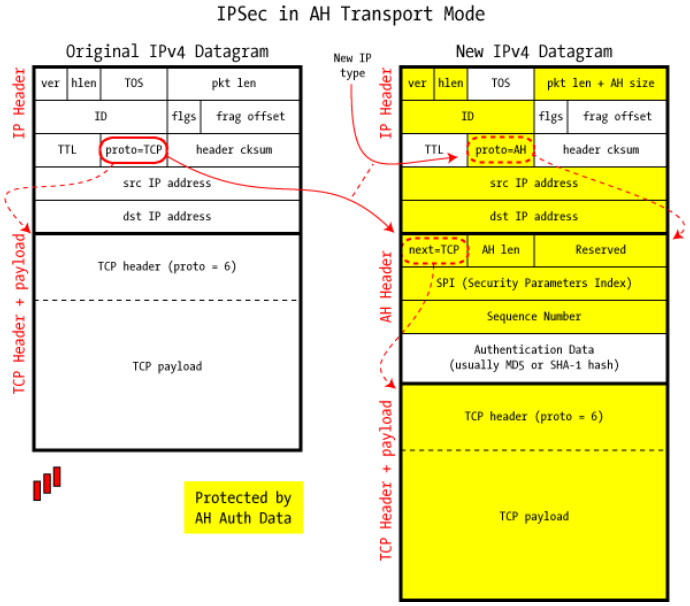

Abb. 5.8: Verwendung des AH im Transport-Modus [Fri08]

Die in der Abbildung farblich hinterlegten Teile zeigen hierbei, die vom AH geschützten Bereiche des neuen IP-Pakets. Weiters erkennt man, dass der Original-Kopf des IP-Pakets nicht verändert wird, sondern lediglich im Verweis auf das eigentliche Paket der "Zwischenschritt" des AH vermerkt wurde. So können Zwischengeräte, die für die Übertragung des IP-Pakets verantwortlich sind auch ohne IPSec-Funktionalitäten das Paket zuverlässig ans Ziel weiterleiten.

Verwendung des AH im Tunnel-Modus

Im Unterschied zum vorangegangenen Transport-Modus passieren beim Tunnel-Modus weitläufigere Änderungen. Abbildung 5.9 auf der nächsten Seite zeigt hierbei die Vorgangsweise bei der Sicherung eines IP-Pakets, welches ein TCP-Segment überträgt.

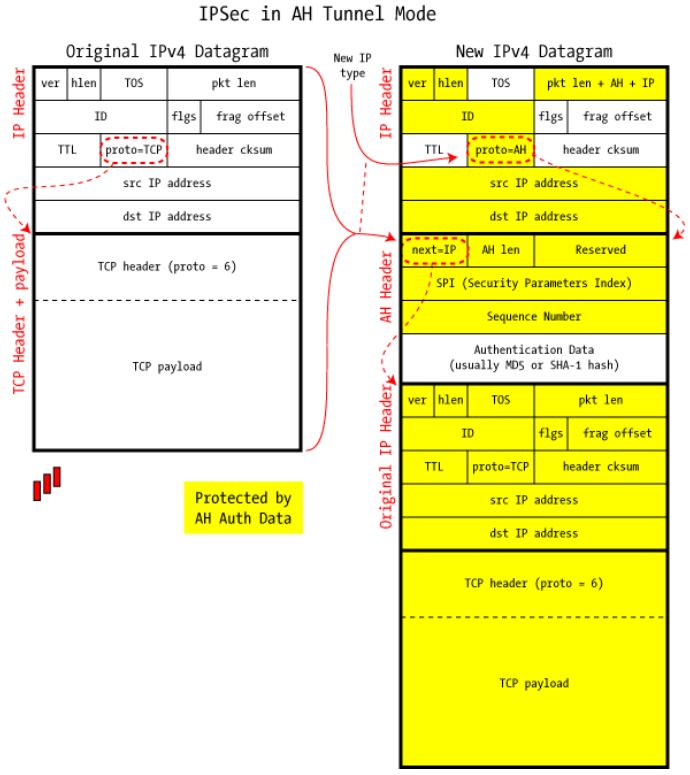

Abb. 5.9: Verwendung des AH im Tunnel-Modus [Fri08]

Wie auch schon beim vorigen Beispiel zeigt die farbige Hinterlegung auch hier die Bereiche an, die vom AH geschützt werden. Auffallend ist zusätzlich noch, dass beim Tunnel-Modus ein neuer IP-Header mitsamt dem AH-Header dem vollständigen IP-Paket (also mit intaktem Header) vorangestellt wird. Dadurch wird das Gesamt-Paket wesentlich vergrößert, was aber für die Verwendung notwendig ist.

5.2.1.3 Vertraulichkeit mit IPSec

Wie aus dem vorigen Punkt ersichtlich ist, hat IPSec geeignete Mechanismen, um zu gewährleisten, dass die Daten unverfälscht sind und nur von authentischen Quellen stammen. Außerdem kann eine Verletzung einer dieser Kriterien sofort von IPSec erkannt werden.

Im Rahmen von IPSec ermöglicht das *Encapsulation Security Payload* (ESP) genannte Verfahren die Anwendung von Verschlüsselungsverfahren zum Schutz der Daten. Obwohl im RFC2406 [KA98b] (quasi als Mindestanforderung) für IPSec DES spezifiziert wurde,

kann theoretisch jedes beliebige symmetrische Verschlüsselungsverfahren verwendet werden, solange es alle an der Übertragung beteiligten Stellen unterstützen.

Zusätzlich zur Verschlüsselung unterstützt ESP auch Authentifizierung, welche nach der Verschlüsselung stattfinden kann. Auf der Empfängerseite wird dann - in umgekehrter Reihenfolge - zuerst die Authentifizierung überprüft und dann die Daten entschlüsselt. Als Authentifizierungs-Mechanismen können alle für den AH verfügbaren Verfahren eingesetzt werden. Theoretisch wäre es auch möglich ESP nur zur Authentisierung zu nützen, indem man kein Verschlüsselungsverfahren auswählt (technisch gesehen wird *NULL* als Verfahren im Paket vermerkt). Da aber mit dem AH eine bessere Authentisierung möglich ist, ist es für die Praxis irrelevant.

Aufbau des ESP

Im Vergleich zu dem bei AH verwendeten Header (siehe voriges Kapitel), sieht ESP-Header deutlich anders aus. Abbildung 5.10 zeigt hierbei den allgemeinen Aufbau. Der eingezeichnete Teil für die Authentifizierungs-Daten muss natürlich nicht immer notwendigerweise vorhanden sein, ist an dieser Stelle aus Gründen der Vollständigkeit eingezeichnet. Anschließend wird nun der Einsatz des ESP in den zwei möglichen Einsatzarten Transport- und Tunnel-Modus exemplarisch gezeigt und näher erläutert.

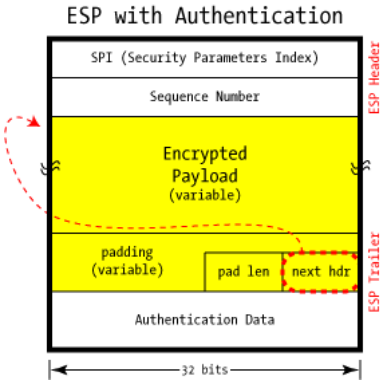

Abb. 5.10: Aufbau des ESP-Headers [Fri08]

Verwendung des ESP im Transport-Modus

Wie schon bei der Verwendung von AH, wird auch beim Einsatz von ESP im Transport-Modus der Original-Header des zu übertragenden IP-Pakets erhalten. Danach folgt der obligatorische ESP-Header und eine verschlüsselte Version der Payload des Original-Pakets.

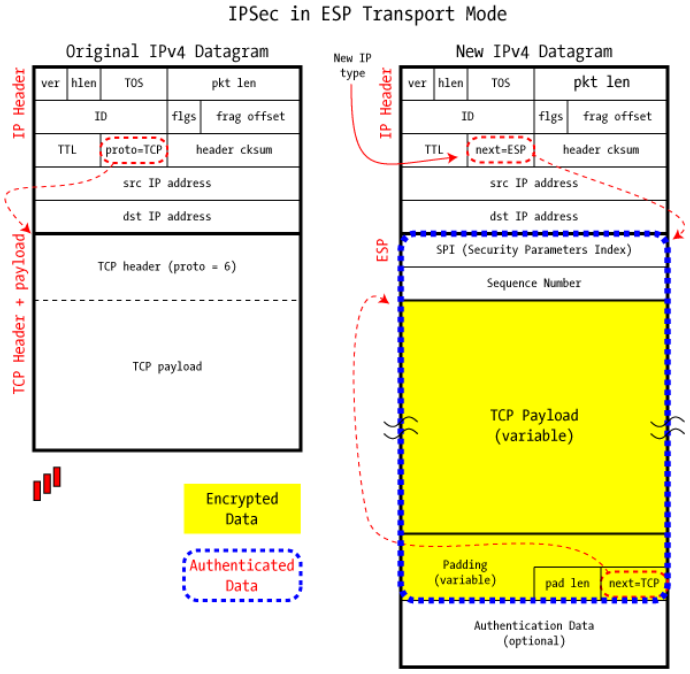

Abb. 5.11: Verwendung des ESP im Transport-Modus [Fri08]

Welcher Teil hierbei genau verschlüsselt und gegebenenfalls authentifiziert wird, stellt Abbildung 5.11 sehr gut dar. Auffallend hierbei ist, dass die Authentifizierung von ESP nicht das gesamte Paket (speziell den Header nicht) umfasst und somit einen weitaus weniger wirksamen Schutz bietet, als dies mittels AH der Fall wäre.

Verwendung des ESP im Tunnel-Modus

Nach der Darstellung von ESP unter Verwendung des Transport-Modus folgt nun die Betrachtung beim Einsatz des Tunnel-Modus. Hierbei wird ein IP-Paket vollständig verpackt und als Payload eines anderen Pakets übertragen. Bei der Verwendung von ESP ist es, dank der Verschlüsselung, für eine Zwischenstelle nicht möglich, herauszufinden, welcher Sender, Absender oder Inhalt in dem originalen IP-Paket eingetragen ist. Zur Veranschaulichung des entstehenden IP-Pakets dient hierbei Abbildung 5.12 auf der nächsten Seite.

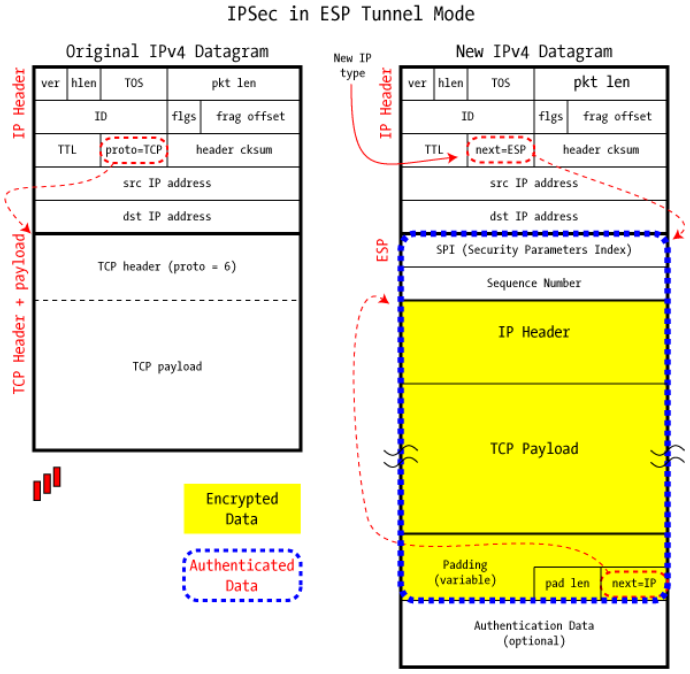

Abb. 5.12: Verwendung des ESP im Tunnel-Modus [Fri08]

5.2.2 Transport Layer Security (TLS)

In den Anfangsjahren des Internets und des World Wide Webs wurde von der Firma "Netscape" das Protokoll *SSL* ("Secure Sockets Layer") entwickelt. Die dritte Version von SSL (SSL 3.0) wird nunmehr unter dem Namen TLS 1.0 (Transport Layer Security) weiterentwickelt und ist im RFC2246 [DA99] standardisiert. Die letztgültige Version ist TLS 1.1 (beschrieben im RFC4346 [DR06]), eine Version 1.2 ist gerade in Vorbereitung.

TLS setzt direkt auf der Transport-Schicht auf (siehe Abbildung 5.13 auf der nächsten Seite) und funktioniert nur in Zusammenhang mit TCP. Trotz dieser Einschränkung ist TLS die derzeit am häufigsten verwendete Sicherungsmethode im Internet.

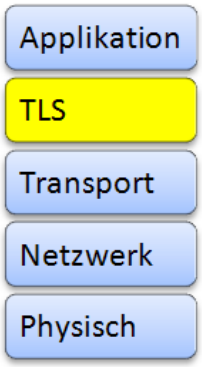

Abb. 5.13: Einordnung von TLS im TCP/IP-Stack

5.2.2.1 Aufbau von TLS

Das TLS-Protokoll selbst setzt sich aus zwei Schichten zusammen:

- TLS Handshake Protocol
- TLS Record Protocol

Diese beiden Protokolle werden in den nachfolgenden Absätzen ausführlicher erklärt.

TLS Handshake Protocol

Bevor eine Verbindung durch TLS geschützt werden kann, muss zuerst ein so genannter "Handshake" erfolgen. Die Durchführung von diesem wird vom TLS Handshake Protocol übernommen.

Ein TLS Handshake besteht grundsätzlich aus mehreren Phasen (siehe Abbildung 5.14 auf der nächsten Seite):

- **Phase 1** ist die Aushandlung der zu verwendenden kryptographischen Methoden (Verschlüsselungsalgorithmen) und die Übermittlung einer Session-ID und eines pre-master-secrets (welches später verwendet wird).
- **Phase 2** ist die Authentifizerung des Servers gegenüber des Clients mittels X509-Zertifikat. Obwohl dieser Schritt optional ist, wird er normalerweise durchgeführt.
- **Phase 3** ist die Authentifizierung des Clients gegenüber des Server. Dieser Schritt ist genau so wie Phase 2 optional und in der Praxis unüblich.

- **Phase 4** schließt den Handshake ab. Hierbei wird aus dem in Phase 1 übermittelten pre-master-secret das Master-Secret gebildet, welches den Schlüssel zur (symmetrischen) Verschlüsselung des Datenstroms bildet.

Ein derartiger TLS Handshake kann jederzeit erneut durchgeführt werden, um beispielsweise die Sicherheits-Einstellungen (Verschlüsselungsalgorithmen) zu ändern. In diesem Fall wird solange mit den bestehenden Einstellungen weitergearbeitet, bis der Handshake abgeschlossen ist.

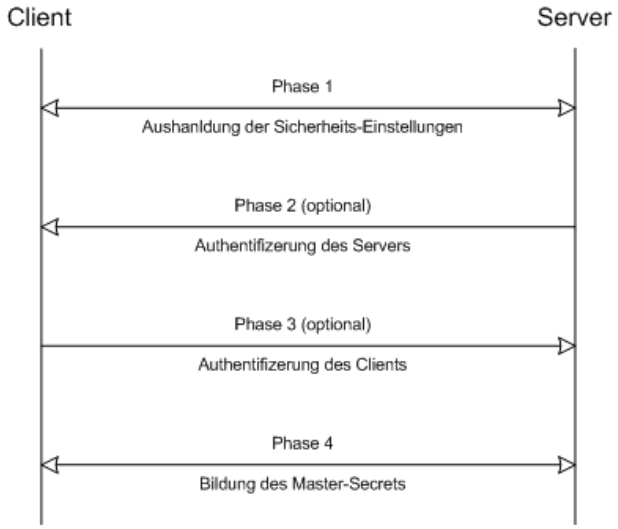

Abb. 5.14: TLS-Handshake

TLS Record Protocol

Das TLS Record Protocol dient zur Kapselung von Anwendungsdaten auf sichere Art und Weise. Dazu werden die vom TLS Handshake Protocol ausgehandelten Einstellungen angewandt und die Daten, welche von der übergeordneten Schicht geliefert werden, entsprechend verschlüsselt bzw. signiert. Je nach Einstellung können Verschlüsselung und Signatur wahlweise oder gemeinsam genutzt werden.

5.2.2.2 Datagram Transport Layer Security (DTLS)

Da TLS, wie in der Einleitung von Kapitel 5.2.2 auf S. 89 erwähnt, nur für die Verwendung mit TCP konzipiert wurde, musste es für den Einsatz mit UDP angepasst werden. Die adaptierte Version wird dann DTLS genannt und ist in RFC 4347 [RM06] standardisiert.

Während sich TLS auf TCP verlässt, dass keine Pakete verloren geht, kann bei DTLS diese

Annahme nicht getroffen werden. Dies ist jedoch besonders wichtig für die Funktion von TLS, da sonst der Handshake nicht funktionieren kann und weiters bei der verschlüsselten Übertragung, aufgrund der Zusammengehörigkeit der einzelnen Teile des Datenstroms, dieser nicht mehr entschlüsselt werden kann, wenn Pakete fehlen oder in falscher Reihenfolge ankommen. Gelöst werden die Probleme durch erneutes Senden der Pakete beim Handshake und durch explizite Nummerierung (bei TLS passiert dies implizit durch TCP).

5.2.3 HTTP Authentifizierung

HTTP ist ein seit vielen Jahren bekanntes und auch weiterentwickeltes Protokoll, welches auch um Authentifizierungs-Mechanismen erweitert wurde. Man unterscheidet hierbei zwischen "Basic" und "Digest" Authentifizierung.

5.2.3.1 Basic Authentication

Die einfachste Methode der Authentifizierung bei HTTP ist die so genannte "Basic Authentication" [FHBH+99]. Diese arbeitet nach dem Prinzip, dass der Server eine Authentifizierung verlangt und gleichzeitig einen so genannten "Realm-Namen" hinzufügt (also für welchen Bereich die Authentifizierung erfolgen soll - sozusagen als Orientierungshilfe für den Benutzer). Nach Eingabe der Authentifzierungsdaten werden diese Base64-codiert (also im Wesentlichen im Klartext) an den Server gesandt, der dann den Zugriff gewährt oder verweigert.

Eine beispielhafte Kommunikation zwischen einem Client und einem HTTP-Server wäre wie folgt:

```
GET /private/index.html HTTP/1.0
Host: localhost
```

Mit dieser Nachricht fordert ein Client, wie in jedem Fall, den Zugriff auf eine Datei (hier *http://localhost/private/index.html*) an.

```
HTTP/1.0 401 UNAUTHORIZED
Server: SokEvo/1.0
Date: Sat, 27 Nov 2004 10:18:15 GMT
WWW-Authenticate: Basic realm="SokEvo"
Content-Type: text/html
Content-Length: 311
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<HTML>
  <HEAD>
    <TITLE>Error</TITLE>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
  </HEAD>
  <BODY><H1>401 Unauthorised.</H1></BODY>
</HTML>
```

Aufgrund der Server-Einstellungen, die eine Authentifizierung nach "Basic Authentication" erfordert, antwortet der Server mit dem Fehler-Code *401* und sendet mittels des Parameters *WWW-Authenticate* den Typ der Authentifizierung (hier "Basic") und den so genannten "realm" mit. Ein Realm dient zur Unterscheidung von "Authentifizierungs-Bereichen" auf dem gleichen Server, worauf hier jedoch nicht näher eingegangen wird.

```
GET /private/index.html HTTP/1.0
Host: localhost
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

Nachdem der Web-Browser den Benutzer zur Eingabe der Login-Informationen aufgefordert hat, wird eine neuerliche Anfrage an den Server gesandt, diesmal mit der Zusatzangabe "Authorization", die den Benutzernamen und das Passwort BASE64-codiert enthält, welches sich problemlos in Klartext rückumwandeln lässt.

```
HTTP/1.0 200 OK
Server: SokEvo/1.0
Date: Sat, 27 Nov 2004 10:19:07 GMT
Content-Type: text/html
Content-Length: 10476

.....
```

War die Benutzername/Passwort-Kombination richtig, so antwortet der Server mit dem Status-Code *200* und liefert im Anschluss das gewünschte HTML-Dokument aus.

5.2.3.2 Digest Authentication

Verbesserung fand die HTTP-Authentifizierung in Form der so genannten "Digest Authentication" [FHBH+99]. Dabei sendet der Server zusätzlich zur der Authentifizierungs-

Aufforderung eine zufällige Zeichenfolge. Der Client berechnet dann mittels MD5 eine Checksumme aus Benutzername, Passwort und der erhaltenen Zeichenfolge. Danach wird nur mehr dieser berechnete Wert übertragen. Auf der Server-Seite passiert nun der selbe Vorgang und ein Vergleich der erhaltenen Checksummen gibt dann Aufschluss, ob die Authentifizierung erfolgreich war. Der Vorteil hierbei ist, dass eine Replay-Attacke vollständig verhindert wird, ebenso wie einfaches Mitlesen, da die Authentifizierungsinformationen (Benutzername und Passwort) nicht übertragen werden.

```
GET /private/index.html HTTP/1.0
Host: localhost
```

Mit dieser Nachricht fordert ein Client, wie in jedem Fall, den Zugriff auf eine Datei (hier *http://localhost/private/index.html*) an.

```
HTTP/1.0 401 Unauthorised
Server: SokEvo/0.9
Date: Sun, 10 Apr 2005 20:26:47 GMT
WWW-Authenticate: Digest realm="testrealm@host.com",
                         qop="auth,auth-int",
                         nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
                         opaque="5ccc069c403ebaf9f0171e9517f40e41"
Content-Type: text/html
Content-Length: 311

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<HTML>
  <HEAD>
    <TITLE>Error</TITLE>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
  </HEAD>
  <BODY><H1>401 Unauthorised.</H1></BODY>
</HTML>
```

Aufgrund der Server-Einstellungen, die eine Authentifizierung nach "Digest Authentication" erfordert, antwortet der Server mit dem Fehler-Code *401* und sendet mittels des Parameters *WWW-Authenticate* den Typ der Authentifizierung (hier "Basic"), den "realm" und die zufällige Zeichenfolge, welche in die Client-Antwort eingebaut wird, mit.

```
GET /private/index.html HTTP/1.0
Host: localhost
Authorization: Digest username="Mufasa",
                      realm="testrealm@host.com",
                      nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
                      uri="/dir/index.html",
                      qop=auth,
                      nc=00000001,
                      cnonce="0a4f113b",
                      response="6629fae49393a05397450978507c4ef1",
                      opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Nachdem der Web-Browser den Benutzer nach den Login-Informationen gefragt hat, verarbeitet er diese mittels MD5 zu einer Zeichenkette, die als "response" in obiger Nachricht an den Server zurückgesandt wird. Damit der Server überhaupt eine Überprüfung durchführen kann, muss der Benutzername als Klartext mitgeschickt werden, was aber aufgrund der nicht-reversiblen "Verschlüsselung" des Passworts hier keine Beeinträchtigung der Sicherheit der Übertragung bedeutet.

```
HTTP/1.0 200 OK
Server: SokEvo/0.9
Date: Sun, 10 Apr 2005 20:27:03 GMT
Content-Type: text/html
Content-Length: 7984
```

War die Benutzername/Passwort-Kombination richtig, so antwortet der Server mit dem Status-Code *200* und liefert im Anschluss das gewünschte HTML-Dokument aus.

5.2.3.3 Zusätzlicher Schutz

Wie auch beim Einsatz von HTTP kann die Übertragung der Authentifizierungs-Informationen zusätzlich durch den Einsatz von SSL/TLS (beschrieben in 5.2.2 auf Seite 89) gesichert werden. Dies ist besonders bei der Basic Authentication sinnvoll, da hierbei die Informationen im Klartext übertragen werden.

5.2.4 S/MIME

Während in der ursprünglichen SIP-Spezifikation das von Phil Zimmermann entwickelte PGP [ASZ96] zur Verschlüsselung der SIP-Daten verwendet werden sollte, wurde dies - aufgrund der aufwändigen Verwaltung der öffentlichen Schlüssel (welche auch bei der Verbreitung von PGP sehr hinderlich waren beziehungsweise sind) - zugunsten von S/MIME aufgegeben.

Das in RFC 3851 [Ram04] spezifizierte S/MIME (*Secure/Multipurpose Internet Mail Extensions*) bietet Mechanismen zur Verschlüsselung und Signierung von SIP-Nachrichten und -Headern. Realisiert wird dies durch den Einsatz von X.509-Zertifikaten.

Die S/MIME-Spezifikation und die Implementierung derer ist sehr komplex, weswegen einiges an Rechenleistung für die Unterstützung von S/MIME benötigt wird. Daher ist der Einsatz auf mobilen bzw. kleinen Geräten oftmals nicht möglich, da hier oft nicht die benötigte Rechenleistung zur Verfügung steht.

5.2.5 Secure RTP (SRTP)

Wie schon eingangs besprochen ist die Übertragung mittels RTP völlig ohne Sicherheitsaspekte. Um diesem Umstand Abhilfe zu verschaffen wurde *Secure RTP* (SRTP) als Erweiterung von RTP in RFC 3711 [BMN+04] spezifiziert. SRTP fügt damit RTP Methoden für die Gewährleistung von Integrität, Authentizität und Vertraulichkeit mit Hilfe von symmetrischer Verschlüsselung (AES-CM) und Hashingverfahren (SHA-1) hinzu. Durch dieses Hinzufügen eines Hash-Wertes ist die Größe von SRTP-Paketen geringfügig höher als von RTP-Paketen. Zusätzlich zur sicheren Variante von RTP gibt es auch eine sichere Variante des dazugehörigen Kontrollprotokolls RTCP, das *Secure RTCP* (SRTCP).

Schlüssel-Verwaltung bei SRTP

Während im Zusammenhang mit SRTP die Verschlüsselung und dergleichen geregelt ist, gibt SRTP kein Schlüssel-Austausch-Verfahren vor, welches aber für eine symmetrische Verschlüsselung zwingend benötigt wird. In diesem Zusammenhang können dann zusätzliche Verfahren wie z.B. MIKEY (Multimedia Internet Keying, RFC 3830 [ACL+04]) verwendet werden. Auf diese wird jedoch in der vorliegenden Arbeit nicht näher eingegangen.

ZRTP

Im Zusammenhang mit der Schlüsselverteilung für SRTP-Sitzungen hat sich auch PGP-Entwickler Phil Zimmermann Überlegungen angestellt und das so genannte *ZRTP* entwickelt, welches bereits als Draft an die IETF eingereicht wurde [ZJC06].

Im Wesentlichen wird das *Diffie-Hellman-Verfahren* zum Schlüsselaustausch verwendet, jedoch erweitert um Abwehrmaßnahmen gegen mögliche Man-in-the-Middle-Attacken. Dazu gehört die Verwendung von Passwörtern (so genannte "Short Authentications Strings"), die sich die Gesprächsteilnehmer vorlesen - sollten sie übereinstimmen, ist die Wahrscheinlichkeit sehr hoch, dass keine MitM-Attacke stattgefunden hat. Zusätzlich werden die Schlüssel der vorangegangenen Sitzungen in einem Cache gespeichert - wenn also ein MitM-Angreifer eine (oder die erste) Sitzung auslässt, so kann er die folgenden Sitzungen nicht mehr belauschen.

5.2.6 VoIP Application Gateways

Eine Paketfilter-Firewall trifft anhand von Kriterien wie Quell-/Ziel-IP-Adresse oder Quell-/Ziel-Port der Pakete Entscheidungen, ob die entsprechenden Pakete weiterzuleiten oder abzuweisen sind. Beim Einsatz von VoIP ergibt sich nun die unangenehme Situation, dass die Ports und IP-Adressen der Kommunikationspartner vorab nicht bekannt sind. Dadurch würde eine generelle Sperre von vorab nicht bekannten Ports VoIP nicht ermöglichen. Auf der anderen Seite würde ein generelles Öffnen die Sicherheit des gesamtes Netzwerks beeinträchtigen, weshalb dies ebenfalls nicht in Frage kommt.

Der Lösungsansatz hierbei ist eine Firewall, welche die abgehenden und ankommenden VoIP-Pakete (im Speziellen die Signalisierungs-Mitteilungen, wie sie z.B. mittels SDP ausgetauscht werden) mitliest und aufgrund der darin enthaltenen Informationen gezielt einzelne VoIP-Verbindungen passieren lässt, während die Gesamt-Sicherheit nicht beeinträchtigt wird.

Natürlich gibt es für diesen Ansatz keine Universal-Lösung, sondern das verwendete Application Gateway muss genau die eingesetzten Protokolle unterstützen, um seine Funktion erfüllen zu können, weshalb beim Wechsel der VoIP-Infrastruktur oftmals das Gateway angepasst werden muss.

5.3 Konsequenzen von verschärften Sicherheitsmaßnahmen

Maßnahmen, die der Erhöhung der Sicherheit dienen, haben neben den Vorteilen auch Nachteile, die bei einem Einsatz in Kauf genommen werden müssen. Durch den Einsatz modernerer Geräte (Rechner, Endgeräte, Netzwerkkomponenten) und Internetverbindungen fällt der Mehraufwand für Sicherheit zwar geringer aus, aber kann trotzdem nicht vollständig ignoriert werden.

Gerade bei der Sprachtelefonie, bei der sich jeder Benutzer einen gewissen Mindeststandard (wie z.B. verzögerungsfreie Übertragung) erwartet, ist es für VoIP schwer diesen immer zu erfüllen. "Alte" Telefonnetze waren dediziert für den Zweck der Telefonie konzipiert und gebaut, während VoIP auf einem offenen, im Falle des Internets sogar unkontrollierten, Netz aufsetzt. Die Erfüllung der Anforderungen der Benutzer ist hierbei natürlich weitaus schwieriger und sicherlich nicht immer möglich.

Nachfolgend soll kurz auf mögliche Nachteile der vorher genannten Sicherheitsmethoden eingegangen werden.

5.3.1 Rechenaufwand

Jede Form des Hinzufügens von Sicherheits-Informationen (wie z.B. Signierung, Verschlüsselung, usw.) benötigt eine größere oder kleinere Zahl an Rechenoperationen. Obwohl diese Operationen normalerweise keinen großen Rechenaufwand darstellen, so fallen sie in speziellen Situationen doch sehr ins Gewicht:

- **Das Gerät ist überlastet**

  Bei modernen Multitasking-Betriebssystemen kann nie davon ausgegangen werden, dass die volle Rechenleistung nur einem Programm zur Verfügung steht. In Situationen, wo andere rechenaufwändige Tasks gerade arbeiten, kann ein zusätzliches Telefonat die Möglichkeiten der Rechenkapazität sprengen.

- **Endgerät mit geringer Rechenleistung**

  Da für ein Telefonat immer schon ein extra Apparat verwendet wurde, liegt es nahe spezielle VoIP-Endgeräte zu bauen. Diese Geräte belasten auch den normalen Arbeitsplatz-Rechner nicht weiter und können problemloser gewartet, ausgetauscht, usw. werden. Der Nachteil solcher Geräte ist die geringe Rechenleistung und meist geringere Flexibilität. So gibt es für "VoIP-Telefone" derzeit kaum Unterstützung für

Sicherheits-Features und ein Upgraden ist meist nur durch einen Austausch, welcher mit großen Kosten verbunden ist, möglich.

5.3.2 Vergrößerung der Datenmengen und des Bandbreitenbedarfs

Gerade beim Einsatz von Verschlüsselung werden nicht nur die zu übertragenden Daten verändert, sondern auch die Größe der Daten ändert sich. Je größer die Daten werden (also die Nutzlast der einzelnen Pakete), desto mehr Pakete werden dann tatsächlich gebraucht, um diese zu transportieren. Dadurch erhöht sich der Aufwand beim Senden bzw. Empfangen der einzelnen Pakete und zusätzlich der Bandbreitenbedarf für die Übertragung.

Bei der momentanen Bandbreitenverfügbarkeit ist dies für den Einzelnen normalerweise kein Problem, jedoch im Backbone-Bereich, wo mehrere Tausend Gespräche gleichzeitig übertragen werden, kann eine geringe Erhöhung des Bandbreitenbedarfs der jeweiligen Gespräche einen drastischen Anstieg des Gesamt-Bandbreitenbedarfs nach sich ziehen. Dadurch kann die zur Verfügung stehende Bandbreite dann schlichtweg zu klein werden und die Übertragung ist nicht mehr reibungslos.

5.3.3 Steigender Administrationsaufwand

Wie jedes IT-System benötigt VoIP auch ein gewisses Maß an Verwaltungsaufwand. Durch den Einsatz von Sicherheits-Maßnahmen wird dieser jedoch entsprechend vergrößert. Methoden zur Authentifizierung oder Verschlüsselung von Daten benötigen für deren Funktion Schlüssel beziehungsweise Zertifikate, die entsprechend den jeweiligen Kommunikationsteilnehmern bekannt sein müssen oder auf Abruf zur Verfügung stehen müssen (z.B. durch eine so genannte *Certification Authority* für Zertifikate). Eine derartige (zusätzliche) Infrastruktur benötigt dann natürlich auch (zusätzliche) Verwaltungsressourcen.

5.3.4 Fehlende Kompatibilität von einzelnen Komponenten

Ein weiterer Punkt, der sich durch den Einsatz von Sicherheits-Merkmalen ergibt, ist eine mögliche Inkompatibilität zu gewissen Gesprächsteilnehmern. Da VoIP kein einzelner Standard ist, sondern es eine Vielzahl von Protokollen gibt, die wiederum eine Vielzahl von Erweiterungen haben, ist es nicht unwahrscheinlich für einen Anruf ein bestimmtes Protokoll oder eine bestimmte Protokoll-Erweiterung (für die Sicherung der Verbindung)

verwenden zu wollen, welche von der Gegenstelle nicht unterstützt wird. Im Umfeld von großen Firmen, wo der Einsatz von Verschlüsselung oder zumindest Authentifizierung aus Geschäfts- und Sicherheitsinteressen meist zwingend sind, kann so eine Kommunikation mittels VoIP unmöglich sein.

6. Zusammenfassung

6.1 Fazit

Voice over IP ist ein Trend, der sich in den letzten Jahren immer stärker entwickelt hat und auch in den kommenden Jahren noch weiter verstärken wird. Experten rechnen in den kommenden Jahren mit einem VoIP-Anteil von bis zu 70%.

In der heutigen Zeit spielt die Sicherheit eine zentrale Rolle. Mit der fortschreitenden Vernetzung ergeben sich neue Möglichkeiten die Produktivität zu erhöhen, aber auch neue Möglichkeiten kriminelle Delikte durchzuführen.

Ein Hauptproblem bei der Sicherheit von VoIP ist - wie man an den möglichen Angriffen auch sehr deutlich sieht - die Verwendung des firmeneigenen LANs - es gibt also keine dedizierte Verkabelung/Infrastruktur. Dadurch ist es, im Unterschied zu herkömmlichen Telefon-Systemen, um ein vielfaches einfacher die Datenübertragung mitzuverfolgen beziehungsweise zu verfälschen. Viele Firmen-LANs sind unzureichend gegen Fremd-Zugriffe geschützt und bieten oftmals auch Gästen die Möglichkeit die eigene Infrastruktur zu verwenden. Daraus resultieren eine Menge von Attacken, die in herkömmlichen Telefon-Systemen undenkbar oder um ein Vielfaches schwieriger zu vollziehen sind.

Diese Problematik stellt in Zukunft erhöhte Ansprüche an ein LAN in dem VoIP und "normale" Datenkommunikation stattfinden soll. Viele Sicherheits-Experten raten in dem Zusammenhang überhaupt zu einer getrennten Vernetzung. Möglichkeiten wie VLANs und dergleichen bieten hierbei eine gute Basis, um den VoIP-Verkehr vom übrigen "Datennetz" innerhalb einer Firma abzuschotten.

Allerdings bedingt eine konsequente Trennung von VoIP-Verkehr und dem übrigen LAN einen sehr kostenintensiven Aufwand, der z.B. Authentifizierung auf Layer2-Ebene, um MAC-Adressen-Spoofing zu umgehen umfasst und rückt damit VoIP auch wieder ein anderes Licht. Dabei wird VoIP oft mit den Schlagwörtern "Kosteneffizienz", "Vereinfachung der Kommunikations-Landschaft" und Ähnlichen in Verbindung gebracht. Wie im Rahmen der vorliegenden Arbeit aufgezeigt wurde, sind bei Einhaltung entsprechender Sicherheits-

vorkehrungen diese Vorstellungen allerdings nicht haltbar.

6.2 Ausblick

Nachdem die anfängliche überschwängliche Begeisterung für VoIP nun langsam am Abklingen ist, ist es nun an der Zeit VoIP "alltagstauglich" für den professionellen Einsatz zu machen, sodass auch Sicherheitsbelange entsprechend berücksichtigt werden.

Die Einarbeitung von Sicherheitsmaßnahmen in den Standards bzw die Implementierung und Verbesserung dieser wird die Entwicklung von VoIP in naher Zukunft prägen. Wenn VoIP die Sicherheitsaspekte ausreichend erfüllt wird es seinen Siegeszug, den es seit der Einführung/Entwicklung angetreten hat, sicherlich auch weiterhin konsequent fortsetzen.

Abkürzungsverzeichnis

ACM	Address Complete Message (SS7-Nachricht)
AES	Advanced Encryption Standard
AES-CM	AES counter mode
ANS	Answer (SS7-Nachricht)
ASN.1	Abstract Syntax Notation One
BRI	Basic Rate Interface
CRC	Cyclic Redundancy Check
CSRC	Contributing Source Identifiers
DDoS	Distributed DoS
DNS	Domain Name System
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
FVSt	Fernvermittlungsstelle
HDLC	High-Level Data Link Control
HTTP	HyperText Transfer Protocol
IAM	Initial Address Message (SS7-Nachricht)
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force

IPS Intrusion Prevention System

IPSec IP Security

ISDN Integrated Services Digital Network

ISP Internet Service Provider

ITU International Telecommunication Union

ITU-T ITU Telecommunication Standardization Sector

MG Media Gateway

MGC Media Gateway Controller

MGCP Media Gateway Control Protocol

PRI Primary Rate Interface

PSTN public switched telephone network

QoS Quality of Service

RAS Registration, Admission, Status (Aufgabenbereich eines H.323-Gatekeepers)

REL Release (SS7-Nachricht)

RLC Release Complete (SS7-Nachricht)

RSVP Resource reSerVation Protocol

RTCP RTP Control Protocol

RTP Realtime Transport Protocol

S/MIME Secure/Multipurpose Internet Mail Extensions

SDES Source Description (RTCP-Pakettyp)

SDP Session Description Protocol (Teil-Protokoll von SIP)

SIP Session Initiation Protocol

SMTP Simple Mail Transfer Protocol

SRTP Secure RTP

SS7 Signalisierungs-System 7 / Signaling System 7

SSRC Synchronization Source Identifier (RTP-Header-Feld)

TCP Transmission Control Protocol (Schicht4-Protokoll der TCP/IP-Protokoll-Suie)

TLS Transport Layer Security

TVSt Teilnehmervermittlungsstelle

UDP User Datagram Protocol (Schicht4-Protokoll der TCP/IP-Protokoll-Suie)

URL Uniform Resource Locator

VoIP Voicer over IP

Abbildungsverzeichnis

2.1 Schema eines Man-in-the-Middle-Angriffs, bei dem der Datenstrom umgelenkt wird . . . 15
2.2 Schema eines Sniffing-Angriffs, bei dem der Datenstrom dupliziert wird . . 15

3.1 Telefone im Wandel der Zeit [Bad05] . . . 19
3.2 Schematische Darstellung eines Telefonnetzes [Bad05] . . . 20
3.3 Ablauf der Signalisierung in einem Telefonnetz [Bad05] . . . 22
3.4 Ablauf der Signalisierung bei ISDN [Bad05] . . . 26
3.5 VoIP-Anwendungsszenario: PC-to-PC [Kuc05] . . . 30
3.6 VoIP-Anwendungsszenario: PC-to-Phone [Kuc05] . . . 30
3.7 VoIP-Anwendungsszenario: Phone-to-Phone [Kuc05] . . . 31
3.8 Anwendung des TCP/IP-Schichtenmodells für RTP-Pakete . . . 34
3.9 Aufbau des RTP-Headers [Bad05] . . . 35
3.10 Ein vollständiges RTP-Paket [Bad05] . . . 36
3.11 Netzwerk mit H.323-Komponenten [rad98] . . . 37
3.12 Einordnung der H.323-Subprotokolle in den TCP/IP-Stack [rad98] . . . 39
3.13 Verbindungsaufbau/Signalisierung bei H.323 [KWF05] . . . 41
3.14 einfacher Verbindungsaufbau bei SIP [Bad05] . . . 44
3.15 Einordnung der VoIP-Protokolle in den TCP/IP-Stack . . . 49
3.16 Schematische Darstellung eines Ethernet-Netzwerks . . . 50
3.17 Aufbau einer Ethernet-MAC-Adresse . . . 50
3.18 Aufbau eines Ethernet-Frames . . . 52
3.19 Aufbau eines IP-Pakets . . . 54
3.20 Klassen von IP-Adressen . . . 55
3.21 Aufbau eines UDP-Segments . . . 55
3.22 Aufbau eines TCP-Segments . . . 57
3.23 Drei-Wege-Handshake beim Aufbau einer TCP-Verbindung . . . 59
3.24 Abbau einer TCP-Verbindung . . . 59

4.1 Blockierung aller abgehenden Anrufe eines SIP-Telefons [Ark02a] 64
4.2 Blockierung aller ankommenden Anrufe eines SIP-Telefons [Ark02a] 65
4.3 Übernehmen eines Telefongesprächs (Ausgeben als ein anderer Teilnehmer) [Ark02a] 66

5.1 Aufbau eines Ethernet-Frames mit VLAN-Feld 70
5.2 Einbettung einer Firewall in ein Netzwerk 70
5.3 Einbettung eines IPS in ein Netzwerk (vor dem Perimeter-Router) 76
5.4 Einordnung von IPSec im TCP/IP-Stack 82
5.5 Ein IP-Paket mit IPSec im Transport-Modus 83
5.6 Ein IP-Paket mit IPSec im Tunnel-Modus 83
5.7 Aufbau des AH-Headers [Fri08] 84
5.8 Verwendung des AH im Transport-Modus [Fri08] 85
5.9 Verwendung des AH im Tunnel-Modus [Fri08] 86
5.10 Aufbau des ESP-Headers [Fri08] 87
5.11 Verwendung des ESP im Transport-Modus [Fri08] 88
5.12 Verwendung des ESP im Tunnel-Modus [Fri08] 89
5.13 Einordnung von TLS im TCP/IP-Stack 90
5.14 TLS-Handshake 91

Literaturverzeichnis

[ACL+04] Arkko, J., E. Carrara, F. Lindholm, M. Naslund und K. Norrman: *MIKEY: Multimedia Internet KEYing*. RFC 3830 (Proposed Standard), August 2004. Updated by RFC 4738.

[ADE+99] Arango, M., A. Dugan, I. Elliott, C. Huitema und S. Pickett: *Media Gateway Control Protocol (MGCP) Version 1.0*. RFC 2705 (Informational), Oktober 1999. Obsoleted by RFC 3435, updated by RFC 3660.

[AF03] Andreasen, F. und B. Foster: *Media Gateway Control Protocol (MGCP) Version 1.0*. RFC 3435 (Informational), Januar 2003. Updated by RFC 3661.

[Ark02a] Arkin, O.: *Cracking VoIP Architecture Based on the Session Initiation Protocol (SIP)*. In: *BlackHat USA 2002*, 2002.

[Ark02b] Arkin, O.: *E.T. Can't Phone Home - Security Issues with VoIP*. Technischer Bericht, Blackhat Conference, USA, 2002.

[Ark02c] Arkin, O.: *VoIP - The Next Generation of Phreaking*. Technischer Bericht, Blackhat Conference, USA, 2002.

[ASRS01] Ackermann, R., M. Schumacher, U. Roedig und R. Steinmetz: *Vulnerabilities and Security Limitations of current IP Telephony Systems*. In: *Proceedings of the Conference on Communications and Multimedia Security (CMS 2001), Darmstadt*, Seiten 53–66, Mai 2001.

[ASZ96] Atkins, D., W. Stallings und P. Zimmermann: *PGP Message Exchange Formats*. RFC 1991 (Informational), August 1996.

[Bad05] Badach, A.: *Voice over IP - Die Technik*. Hanser, 2005.

[BLFF96] Berners-Lee, T., R. Fielding und H. Frystyk: *Hypertext Transfer Protocol – HTTP/1.0*. RFC 1945 (Informational), Mai 1996.

[BMN+04] Baugher, M., D. McGrew, M. Naslund, E. Carrara und K. Norrman: *The Secure Real-time Transport Protocol (SRTP).* RFC 3711 (Proposed Standard), März 2004.

[BSI05a] BSI - Bundesamt für Sicherheit in der Informationstechnik: *IT-Grundschutzhandbuch.* 2005.

[BSI05b] BSI - Bundesamt für Sicherheit in der Informationstechnik: *VoIPSEC - Studie zur Sicherheit von Voice over Internet Protocol.* Technischer Bericht, 2005.

[CGR+00] Cuervo, F., N. Greene, A. Rayhan, C. Huitema, B. Rosen und J. Segers: *Megaco Protocol Version 1.0.* RFC 3015 (Proposed Standard), November 2000. Obsoleted by RFC 3525.

[DA99] Dierks, T. und C. Allen: *The TLS Protocol Version 1.0.* RFC 2246 (Proposed Standard), Januar 1999. Obsoleted by RFC 4346, updated by RFC 3546.

[Dem04] Demuth, C.: *Skriptum zur Vorlesung "Computer Networks".* Technischer Bericht, Technische Universtität Wien, 2004.

[DH98] Deering, S. und R. Hinden: *Internet Protocol, Version 6 (IPv6) Specification.* RFC 2460 (Draft Standard), Dezember 1998.

[DR06] Dierks, T. und E. Rescorla: *The Transport Layer Security (TLS) Protocol Version 1.1.* RFC 4346 (Proposed Standard), April 2006. Updated by RFCs 4366, 4680, 4681.

[Dur03] Durkin, J. F.: *Voice-Enabling the Data Network: H.323, MGCP, SIP, QoS, SLAs, and Security.* Cisco Press, 2003.

[EC06] Endler, D. und M. Collier: *Hacking Exposed VoIP.* McGraw-Hill, 2006.

[FCC03] Friedman, T., R. Caceres und A. Clark: *RTP Control Protocol Extended Reports (RTCP XR).* RFC 3611 (Proposed Standard), November 2003.

[FGM+97] Fielding, R., J. Gettys, J. Mogul, H. Frystyk und T. Berners-Lee: *Hypertext Transfer Protocol – HTTP/1.1.* RFC 2068 (Proposed Standard), Januar 1997. Obsoleted by RFC 2616.

[FHBH+99] Franks, J., P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen und L. Stewart: *HTTP Authentication: Basic and Digest Access Authentication.* RFC 2617 (Draft Standard), Juni 1999.

[Fri08] Friedl, S.: *An Illustrated Guide to IPsec*, 2008. `http://www.unixwiz.net/techtips/iguide-ipsec.html` - 5. Mai 2008.

[FS98] Ferguson, P. und D. Senie: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.* RFC 2267 (Informational), Januar 1998. Obsoleted by RFC 2827.

[FS00] Ferguson, P. und D. Senie: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.* RFC 2827 (Best Current Practice), Mai 2000. Updated by RFC 3704.

[GK98] Glenn, R. und S. Kent: *The NULL Encryption Algorithm and Its Use With IPsec.* RFC 2410 (Proposed Standard), November 1998.

[GSC+96] Group, Audio-Video Transport Working, H. Schulzrinne, S. Casner, R. Frederick und V. Jacobson: *RTP: A Transport Protocol for Real-Time Applications.* RFC 1889 (Proposed Standard), Januar 1996. Obsoleted by RFC 3550.

[HC98] Harkins, D. und D. Carrel: *The Internet Key Exchange (IKE).* RFC 2409 (Proposed Standard), November 1998. Obsoleted by RFC 4306, updated by RFC 4109.

[Hon04] Honeynet Project, The: *Know Your Enemy: Learning about Security Threats.* Addison-Wesley Professional, 2004.

[HSSR99] Handley, M., H. Schulzrinne, E. Schooler und J. Rosenberg: *SIP: Session Initiation Protocol.* RFC 2543 (Proposed Standard), März 1999. Obsoleted by RFCs 3261, 3262, 3263, 3264, 3265.

[IAN08] IANA (Internet Assigned Numbers Authority): *Port Numbers*, 2008. `http://www.iana.org/assignments/port-numbers` - 5. Mai 2008.

[Int96] International Telecommunication Union: *Media stream packetization and synchronization on non-guaranteed quality of service LANs.* Recommendation H.225.0, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, November 1996.

[Int98] International Telecommunication Union: *Control protocol for multimedia communication.* Recommendation H.245, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Februar 1998.

[Int00] International Telecommunication Union: *Packet based multimedia communication systems.* Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, November 2000.

[Joh04] Johnston, A. B.: *understanding the Session Initiating Protocol.* Artech House Inc., 2004.

[JP06] Johnston, A. B. und D. M. Piscitello: *Unterstanding Voice over IP Security.* Artech House Inc., 2006.

[KA98a] Kent, S. und R. Atkinson: *IP Authentication Header.* RFC 2402 (Proposed Standard), November 1998. Obsoleted by RFCs 4302, 4305.

[KA98b] Kent, S. und R. Atkinson: *IP Encapsulating Security Payload (ESP).* RFC 2406 (Proposed Standard), November 1998. Obsoleted by RFCs 4303, 4305.

[KA98c] Kent, S. und R. Atkinson: *Security Architecture for the Internet Protocol.* RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[Kan05] Kanbach, A.: *SIP Die Technik.* vieweg, 2005.

[Köh02] Köhler, R.-D.: *Voice over IP.* mitp, 2002.

[Kil00] Killalea, T.: *Recommended Internet Service Provider Security Services and Procedures.* RFC 3013 (Best Current Practice), November 2000.

[Kle01] Klensin, J.: *Simple Mail Transfer Protocol.* RFC 2821 (Proposed Standard), April 2001.

[Kuc05] KUCH, R.: *Folien zur Vorlesung "Telekommunikations-Systeme"*. Technischer Bericht, Technische Universität Graz, 2005.

[KWF05] KUHN, R., T. J. WALSH und S. FRIES: *Security Considerations for Voice over IP Systems*. Technischer Bericht, NIST - National Institute of Standards and Technology, 2005.

[LM00] LIU, H. und P. MOUCHTARIS: *Voice over IP Signaling: H.323 and Beyond*. IEEE Communications Magazine, October:142–148, 2000.

[MD98] MADSON, C. und N. DORASWAMY: *The ESP DES-CBC Cipher Algorithm With Explicit IV*. RFC 2405 (Proposed Standard), November 1998.

[MG98a] MADSON, C. und R. GLENN: *The Use of HMAC-MD5-96 within ESP and AH*. RFC 2403 (Proposed Standard), November 1998.

[MG98b] MADSON, C. und R. GLENN: *The Use of HMAC-SHA-1-96 within ESP and AH*. RFC 2404 (Proposed Standard), November 1998.

[MSST98] MAUGHAN, D., M. SCHERTLER, M. SCHNEIDER und J. TURNER: *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408 (Proposed Standard), November 1998. Obsoleted by RFC 4306.

[MU01] MEHTA, P. und S. UDANI: *Voice over IP Sounding good on the Internet*. IEEE POTENTIALS, OCTOBER/NOVEMBER:36–40, 2001.

[Orm98] ORMAN, H.: *The OAKLEY Key Determination Protocol*. RFC 2412 (Informational), November 1998.

[Pip98] PIPER, D.: *The Internet IP Security Domain of Interpretation for ISAKMP*. RFC 2407 (Proposed Standard), November 1998. Obsoleted by RFC 4306.

[Pos80] POSTEL, J.: *User Datagram Protocol*. RFC 768 (Standard), August 1980.

[Pos81a] POSTEL, J.: *Internet Protocol*. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[Pos81b] POSTEL, J.: *Transmission Control Protocol*. RFC 793 (Standard), September 1981. Updated by RFC 3168.

[Pos82] Postel, J.: *Simple Mail Transfer Protocol.* RFC 821 (Standard), August 1982. Obsoleted by RFC 2821.

[rad98] *H.323 Tutorial.* Technischer Bericht, RADCOM, 1998.

[Ram04] Ramsdell, B.: *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification.* RFC 3851 (Proposed Standard), Juli 2004.

[RM06] Rescorla, E. und N. Modadugu: *Datagram Transport Layer Security.* RFC 4347 (Proposed Standard), April 2006.

[Roa02] Roach, A. B.: *Session Initiation Protocol (SIP)-Specific Event Notification.* RFC 3265 (Proposed Standard), Juni 2002.

[RS02a] Rosenberg, J. und H. Schulzrinne: *An Offer/Answer Model with Session Description Protocol (SDP).* RFC 3264 (Proposed Standard), Juni 2002.

[RS02b] Rosenberg, J. und H. Schulzrinne: *Reliability of Provisional Responses in Session Initiation Protocol (SIP).* RFC 3262 (Proposed Standard), Juni 2002.

[RS02c] Rosenberg, J. und H. Schulzrinne: *Session Initiation Protocol (SIP): Locating SIP Servers.* RFC 3263 (Proposed Standard), Juni 2002.

[RSC+02] Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley und E. Schooler: *SIP: Session Initiation Protocol.* RFC 3261 (Proposed Standard), Juni 2002. Updated by RFCs 3265, 3853, 4320, 4916.

[SC03] Schulzrinne, H. und S. Casner: *RTP Profile for Audio and Video Conferences with Minimal Control.* RFC 3551 (Standard), Juli 2003.

[SCFJ03] Schulzrinne, H., S. Casner, R. Frederick und V. Jacobson: *RTP: A Transport Protocol for Real-Time Applications.* RFC 3550 (Standard), Juli 2003.

[Sch05] Scholz, H.: *VoIP Phreaking - Introduction to SIP Hacking.* Technischer Bericht, 22C3 Conference, 2005.

[Sko02] Skoudis, E.: *Counter Hack*. Prentice Hall Series in Computer Networking and Distributed Systems, 2002.

[Spi02] Spitzner, L.: *Honeypots: Tracking Hackers*. Addison-Wesley Longman, Amsterdam, 2002.

[SS05] Schleife, K. und O. Schmid: *IT-Sicherheit in Unternehmen*. In: *FAZIT-Schriftenreihe: Open-Source Software und IT-Sicherheit*. Zentrum für Europäische Wirtschaftsforschung GmbH (ZEW), 2005.

[TDG98] Thayer, R., N. Doraswamy und R. Glenn: *IP Security Document Roadmap*. RFC 2411 (Informational), November 1998.

[TW05] Trick, U. und F. Weber: *SIP, TCP/IP und Telekommunikationsnetze*. Oldenbourg, 2005.

[Ver01] Vern, P.: *An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks*. Technischer Bericht, AT&T Center for Internet Research at ICSI, Berkeley, CA, 2001.

[WK05] Walsh, T. J. und D. R. Kuhn: *Challenges in Securing Voice over IP*. IEEE SECURITY & PRIVACY, MAY/JUNE:44–49, 2005.

[ZJC06] Zimmermann, P., A. Johnston und J. Callas: *ZRTP: Media Path Key Agreement for Secure RTP*. Technischer Bericht, IETF Internet-Draft, 2006.