Universität Zürich
Institut für Informatik

Rapid Secure Development

Ein Verfahren zur Definition eines Internet–Sicherheitskonzeptes

Projektbericht SINUS – Sichere Nutzung von Online–Diensten

Daniela Damm
Philipp Kirsch
Thomas Schlienger
Stephanie Teufel
Harald Weidner
Urs Zurfluh

Version 1.1
Februar 1999

Zusammenfassung

Die Nutzung von Online–Diensten des Internet wird für immer mehr Unternehmen zu einem wichtigen Geschäfts- und Werbeinstrument. Mit der zunehmenden kommerziellen Nutzung gewinnen auch Sicherheitsaspekte im Internet zunehmend an Bedeutung. Obwohl viele Lösungen zu einzelnen Sicherheitsproblemen existieren, werden diese in der Praxis kaum umgesetzt. Grund dafür sind die Schwierigkeiten bei der Auswahl der einzelnen Sicherheitsmassnahmen und ihre EInbettung in die organisatorischen Abläufe eines Unternehmens.

Rapid Secure Development (RSD) ist ein Verfahren zur Konzeption eines gesicherten Internet–Anschlusses für kleinere und mittlere Unternehmen. Es führt den Benutzer in fünf Schritten von der Definition der Internet–Nutzungsszenarien über Auswahl von Internet–Diensten und Gefahrenanalyse bis hin zur Auswahl und Realisierung von Sicherheitsmassnahmen. Das Verfahren entstand als Resultat des SPP–IuK–Projektes „Sichere Nutzung von Online–Diensten“ (SINUS).

Neben dem eigentlichen Verfahren beinhaltet dieser Bericht eine Einführung in das Informationssicherheitsmanagement und der Internet–Technologie. Im Anhang werden einzelne Sicherheitsmassnahmen, insbesondere der im SINUS–Projekt entstandene Firewall–Prototyp, näher beschrieben. Am Ende wird das Verfahren anhand eines (fiktiven) Beispiels exemplarisch durchgeführt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Teil I

Grundlagen

Kapitel 1

Problemstellung

1.1 Einleitung

Die Nutzung von Online–Diensten des Internet wird von immer mehr Unternehmen als ein wichtiges zukünftiges Werbe- und Geschäftsinstrument angesehen. Sowohl die Präsenz in einem schnell wachsenden Markt als auch die Möglichkeit, von den Informationen und Dienstleistungen anderer Anbieter zu profitieren, versprechen ein wichtiges Standbein des Unternehmenserfolges zu werden.

Mit der zukünftigen Kommerzialisierung des Internet, das ursprünglich als Kommunikationsinstrument zu akademischen Zwecken entwickelt wurde, spielen dabei Sicherheitsaspekte eine immer grössere Rolle. Obwohl die Notwendigkeit von Sicherheitsmassnahmen unumstritten ist und viele technische Lösungen zu Sicherheitsproblemen existieren, werden diese in der Realität kaum umgesetzt. Dies hängt nicht nur mit mangelndem Gefahrenbewusstsein und der Tatsache zusammen, dass Sicherheit Geld kostet und keinen direkten finanziellen Gewinn erbringt. Auch die Schwierigkeiten beim Erkennen und Bekämpfen von Sicherheitslücken, die auf mangelnde Kenntnis der Materie sowie Angst vor Inkompatibilität zu bestehenden Lösungen und Verlust an Funktionalität und Bequemlichkeit zurückzuführen sind, tragen dazu bei.

1.2 Das SINUS–Projekt

Das Projekt SINUS (Sichere Nutzung von Online–Diensten) wird vom Institut für Informatik der Universität Zürich in Zusammenarbeit mit SWITCH (Swiss Academic and Research Network) und der Telekurs Logistik AG Zürich durchgeführt [TZL95]. Es beschäftigt sich mit den Sicherheitsaspekten beim Anschluss eines Unternehmens an das Internet auf organisatorischer, technischer und personeller Ebene. Im Einzelnen wird untersucht, inwieweit Nutzung von Online–Diensten, Angebot von Informationen und Durchführung von Geschäftsbeziehungen über das Internet in das Sicherheitskonzept und die informationstechnische Infrastruktur eines Unternehmens eingebettet werden kann. Ferner soll eine Aussage darüber gewonnen werden, ob frei verfügbare Sicherheitssoftware mit offener Spezifi-

kation dem Anspruch gerecht wird, zur Abwehr von Angriffen aus dem Internet geeignet zu sein.

Wichtigstes Ergebnis des Projektes ist die SINUS–Sicherheitskonzeption. Sie dient zur Unterstützung der Internet–Anbindung für kleinere und mittlere Unternehmen und wird in diesem Bericht ausführlich beschrieben. Eine Implementierung der Sicherheitskonzeption stellt das Expertensystem RSD–XPS dar (Kapitel 10). Im Bereich der technischen Massnahmen wurde im Rahmen des Projektes ein Firewall entwickelt, der auf Paketfilterung beruht (Abschnitt 12.5).

1.3 Ziel dieses Berichtes und Aufbau

Dieser Bericht dokumentiert den theoretischen Basisteil des Projektes. Die praktischen Konzepte und Erfahrungen, sowie die ergänzenden Untersuchungen der Verlängerungsphase werden per Ende 1999 (Projektende) in Anhängen zu diesem Bericht publiziert.

Der Bericht ist in vier Abschnitte unterteilt. Im ersten Abschnitt werden Grundlagen des Sicherheitsmanagements, des Internet und der Internet–Sicherheitskonzeption erläutert. Der zweite Teil ist die Sicherheitskonzeption selbst. Der Leser wird hier anhand eines Verfahrens von der Definition der Internet–Nutzungsszenarien bis hin zur Realisierung seines Internet–Anschlusses geführt. Der dritte Teil geht genauer auf die Sicherheitsmechanismen ein, insbesondere auf den im SINUS–Projekt entwickelten Firewall–Prototyp. Im vierten Teil wird der Ablauf der Sicherheitskonzeption an einem fiktiven Beispiel durchgeführt.

Einige Details im Bereich der technischen Umsetzung, insbesondere im dritten Teil, hängen stark vom verwendeten Betriebssystem ab. Dieser Bericht konzentriert sich im Wesentlichen auf Unix/Linux. Linux ist ein freies Betriebssystem, dessen Sourcen offenliegen. Daher lassen sich neue Techniken im Bereich der Sicherheit hier besonders gut implementieren und austesten. Auch der SINUS–Firewall, der im dritten Teil dieses Berichtes detailliert vorgestellt wird, läuft unter Linux. Die vorgestellten technischen Sicherheitsmassnahmen gelten prinzipiell auch für andere Systeme, wobei sich in den Details Abweichungen ergeben können.

Kapitel 2

Unternehmensziele und Sicherheitsmanagement

In diesem Kapitel erfolgen grundlegende Feststellungen zu Unternehmenszielen und deren Umsetzung durch die Nutzung von Internet–Diensten. Diesen Nutzungszielen stehen Bedrohungen des Internet gegenüber, die mit Hilfe eines Sicherheitsmanagements erkannt und entschärft werden müssen. Ein Kernbestandteil des Sicherheitsmanagements ist die Sicherheitskonzeption. Diese generelle Einführung in die Thematik behandelt grundlegende Aspekte und Definitionen. Im übernächsten Kapitel werden diese für das Gebiet der Internet–Sicherheitskonzeption präzisiert.

2.1 Einführung

Unternehmen verfolgen in unserem Wirtschafts- und Gesellschaftssystem unterschiedliche Ziele, die effizient erreicht werden müssen [vgl. Rüh96, S. 58f]; dies impliziert, dass Unternehmensziele und Effizienzanforderungen eng gekoppelt sind. Die Ziele können durch unterschiedliche Geschäftsstrategien verfolgt werden, wobei sie durch folgende fünf Faktoren massgeblich beeinflusst werden (vgl. Abb. 2.1, [Rüh96, S. 58f] und [Sch94, S. 48f]):

- *Zeit:* der Faktor Zeit (z.B. Entwicklungszeit, Produktionszeit) kann durch kurze Übermittlungszeiten von Nachrichten im Internet wesentlich verkürzt werden. Eine weitere Effizienzsteigerung kann dadurch erreicht werden, dass Informationen für IT–Systeme interpretierbar übertragen werden. Dies erleichtert die Weiterverarbeitung.
- *Qualität:* die Qualität der Produkte bzw. der Arbeit lässt sich neben der gewonnen Zeit und der Kostenreduktion, die direkt in qualitätssichernde Massnahmen einfliessen kann, durch neue Formen der Zusammenarbeit in Unternehmensprozessen verbessern.
- *Kosten:* eine Kostenreduzierung wird zum einen durch Zeiteinsparung und Qualitätsverbesserungen, zum anderen aber auch durch die Nutzung adäquater Informationssysteme zur Unterstützung der Geschäftsprozesse erreicht.

- *Leistung:* eine Erweiterung der Leistungen (z.B. neue Dienstleistungen oder Value Added Services) können u.U. nur durch die Nutzung neuer Technologien realisiert werden.
- *Image:* Besitzt die Informationstechnologie für ein Unternehmen eine strategische Bedeutung, so ist die Nutzung neuer Technologien für diese Unternehmen eine Image–Frage.

Die Optimierung dieser fünf Faktoren sind entscheidend für den unternehmerischen Erfolg. Die Nutzung des Internet kann all diese Faktoren, wie oben angedeutet, positiv beeinflussen.

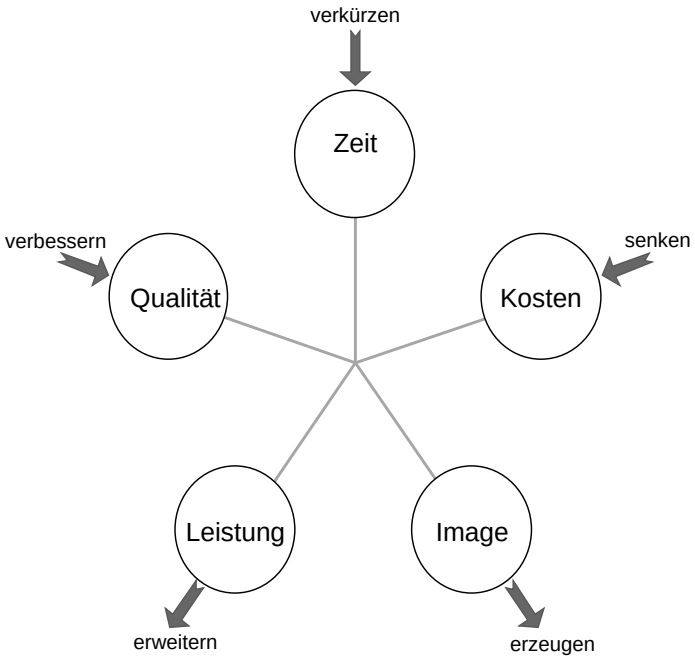

Abbildung 2.1: Einflussfaktoren auf Unternehmensziele

Eine empirische Untersuchung von Unternehmen, die sich neu an das Internet angeschlossen haben, bestätigt zu einem grossen Teil die oben angedeuteten Optimierungsmöglichkeiten der fünf Faktoren durch die Nutzung des Internet. Nach [KS96, S. 32] lässt sich folgendes Muster bei der Nutzung des World Wide Web als herausragenden Internet–Dienst feststellen:

1. Der Einstieg erfolgt mit der Bereitstellung einer eigenen Homepage mit PR- und Marketinginformationen.
2. Gleichzeitig wird das Internet zur Informationsbeschaffung genutzt. E–Mail wird zur Kommunikation genutzt.
3. Es setzt ein Erfolg bezüglich der Qualität und Leistung ein. Überraschend entstehen darüber hinaus positive Effekte bezüglich Zeit und Kosten; die Studie beschränkte sich auf eine Betrachtung dieser Faktoren.

4. Internet–Dienste werden in Unternehmensprozesse integriert.

5. Neue, bisher nicht existierende Geschäftsfelder werden erkannt.

Die Resultate dieser Studie dürften nicht untypisch für eine Entwicklung von Unternehmen im Internet sein. Es stellt sich somit die Frage, wie Unternehmen Nutzungspotentiale des Internet identifizieren und erschliessen können. Bevor auf den Kern dieser Frage im Rahmen der Sicherheitskonzeption (Kapitel 4) eingegangen wird, sollen in diesem Kapitel Unternehmensanforderungen und Fragen des Sicherheitsmanagements erläutert werden, um die Sicherheitskonzeption besser in bestehende Prozesse integrieren zu können.

2.2 Unternehmensanforderungen: Ziele – Strategien – Grundsätze

Die Planung und Entwicklung neuer Informationssysteme und -techniken, in unserem Zusammenhang insbesondere auch des Internet, sollte der Erfüllung unternehmerischer Ziele dienen. Auf der anderen Seite kann der Einsatz von Informationssystemen und -techniken aus einem Erprobungsstadium heraus auch Einfluss auf unternehmerische Ziele nehmen. Dies trifft vor allem auf die Nutzung des Internet als Schlüssel- und Schrittmachertechnologie zu. (Zum Technologiebegriff siehe [Hei96, S. 157f].)

Vor der eigentlichen Zielformulierung steht eine Vision. Von diesen Visionen können dann Ziele (engl. objectives) abgeleitet werden. Die Formulierung der Ziele reicht nicht aus; als nächster wesentlicher Schritt muss die Zielerreichung, d.h. die Strategie (engl. strategy) formuliert werden. Aus dieser Strategie wiederum müssen Handlungsanleitungen bzw. Grundsätze (engl. policies) formuliert werden (vgl. [WGW90, S. 93ff]). Diese dienen dann als Basis für die Systemplanung und -entwicklung (vgl. [Alt92, S. 631ff] und [Sch94, S. 13ff]).

Abbildung 2.2 illustriert den Prozess von der ersten Vision über die Formulierung von Unternehmenszielen bis zu deren Unterstützung durch Informationssysteme. In die Zielformulierung gehen insbesondere Einflussfaktoren wie die derzeitige Informationsinfrastruktur, die verfügbaren Informationssysteme und -techniken, Umwelteinflüsse und die aktuelle Situation ein (vgl. [Alt92, S. 590ff und S. 631ff] sowie [WGW90, S. 95ff]).

Diese Ziele müssen operationalisiert werden, d.h. es muss eine Strategie zur Zielerreichung entworfen werden. Als Ergebnis der Strategie ergeben sich Grundsätze, die in Form von Handlungsanleitungen die Zielerreichung ermöglichen sollen. Bedingt diese Handlungsanleitung den Entwurf eines Informationssystems, so muss dies im folgenden geplant und entwickelt werden.

Die dargestellte Prozesskette läuft natürlich nicht derart sequentiell ab, wie in Abbildung 2.2 dargestellt. Es entstehen Rückflüsse an vielen Punkten. Insbesondere beeinflussen die entworfenen IT–Systeme neue Planungsprozesse.

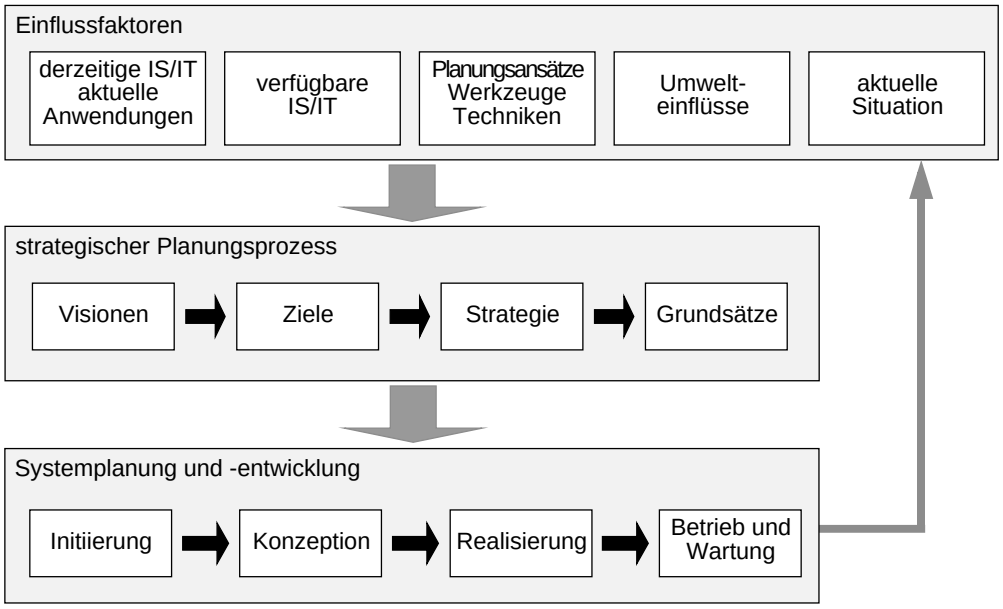

Abbildung 2.2: Von den Zielen zum Informationssystem

2.3 Möglichkeiten des Internet zur Unterstützung der Unternehmensziele

Das Internet ist ein grosses weltumspannendes Kommunikationsnetzwerk, auf dem immer mehr Nachrichten ausgetauscht werden. Als eine entscheidende Dimension der Strukturierung von Informationssystemen im Internet bietet sich daher die Unterteilung nach dem angestrebten Grad der Kooperation an. Der Kooperationsgrad hängt einerseits von der Art der Interaktionsbeziehung und andererseits von der Intensität der Interaktionsbeziehung ab.

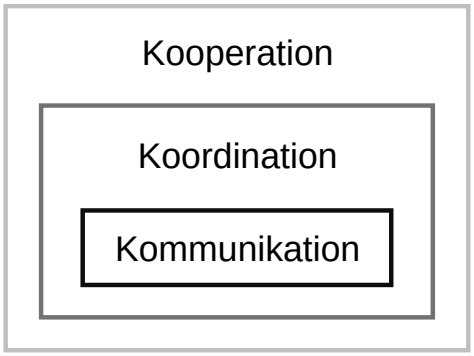

Abbildung 2.3: Kommunikation, Koordination, Kooperation

Grundformen unterschiedlicher Interaktionsbeziehungen setzen sich aus der Möglichkeit zur Kommunikation, Kooperation und Koordination zusammen (vgl. Abbildung 2.3, [TSMB95, S. 10ff]). Die Begriffe entstammen aus dem Kommunikations- bzw. dem CSCW–Bereich (Computer Supported Cooperative Work). Der Fokus der zugrunde liegenden IT–Systeme liegt somit mehr auf dynamischen als auf statischen Gesichtspunkten.Aus organi-

satorischer Sicht ergibt sich die Notwendigkeit, die eingesetzte Informationsinfrastruktur unternehmensübergreifend zu verwenden. Dies hat Auswirkungen auf Kommunikations- und Datenformate.

Mit dem Internet als weltumspannendes Kommunikationsnetzwerk ist derzeit eine grösstmögliche unternehmensübergreifende Zusammenarbeit möglich. Dies erklärt den grossen Erfolg des Systems. Über standardisierte Anwendungsprotokolle bzw. -dienste (z.B. HTTP, FTP, SMTP, etc.) (vgl. Kapitel 6) stehen darüber hinaus viele Dienste bereit, die zur Prozessunterstützung benötigt werden. Die Schaffung gemeinsamer Datenformate, die ein dokumentenzentriertes Arbeiten ermöglichen sollen [Wey95], ist eine weitere wichtige technische Voraussetzung, um Informationen firmenübergreifend auszutauschen. In diesem Bereich konkurrieren nach wie vor mehrere sogenannte Compound Document Architekturen miteinander. Als grundlegendes Dokumentenformat scheint sich derzeit HTML als Rahmenarchitektur durchzusetzen [Kir96].

Aus Sicht eines einzelnen Unternehmens, aber auch aus technischer Sicht, ist die Frage nach den an der Arbeit beteiligte Personengruppen bzw. der Zielgruppe äusserst wichtig.

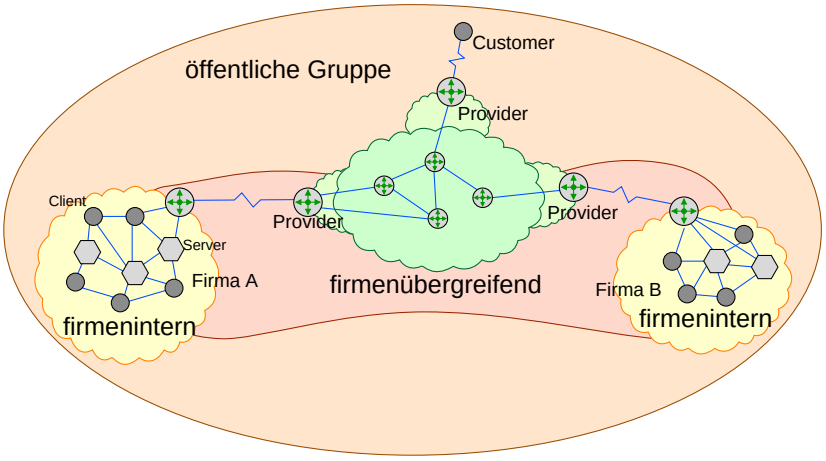

Abbildung 2.4: Gruppen im Internet

Zunächst lassen sich vier grosse Teilnehmerklassen unterscheiden (vgl. Abbildung 2.4, in Anlehnung an [Alt92, S. 123f]): :

- individuumfirmeninterne Gruppen,
- firmenübergreifende Gruppen und
- öffentliche Gruppen.

Unternehmensorganisation geht i.d.R. davon aus, dass Aufgaben bzw. Prozesse, die bearbeitet werden müssen, firmenintern gelöst werden. D.h. nur Mitarbeitende der eigenen Firma werden involviert.

In den letzten Jahren wurde dieses Prinzip durch eine Öffnung des Unternehmens nach aussen immer mehr gelockert. Diese Veränderung geht einher mit Begriffen wie Outsourcing,

virtuelle Unternehmen, etc. Allen mit diesen Begriffen verbundenen Prozessen ist gemein, dass sie firmenübergreifend gelöst werden müssen.

Schliesslich kann man noch die Kommunikation von Mitgliedern eines Unternehmens mit der Aussenwelt als Arbeitsgruppierung betrachten. In diesen Bereich fallen insbesondere alle Beziehungen zum „Endkunden“ (z.B. Marketingaktivitäten) aber auch Beziehungen zum „Ursprungslieferant“ (z.B. Produktforschung).

Das Internet und dessen Anwendungsdienste werden aber auch zu rein persönlichen Zwecken genutzt, ohne das Potential der Gruppenarbeit wirklich ausschöpfen zu wollen. In diesem Fall kann man von einer individuellen Nutzung der Internet–Dienste sprechen.

Die Notwendigkeit bzw. der Bedarf zur Abgrenzung von internen, externen und öffentlichen Gruppen wird durch die verbreiteten Schlagworte Intranet, Extranet und Internet zum Ausdruck gebracht. Die technologische Realisierung ist jeweils identisch. D.h. die drei Schlagworte werden mit unterschiedlichen Anwendungsdiensten bzw. einer unterschiedlichen Nutzung der Dienste verbunden. In diesem Kontext handelt es sich bei den Schlagworten nicht um technische Begriffe, sondern um organisatorische Abgrenzungen bezüglich der Benutzergruppen. Durch den technischen Ursprung der Begriffe entstehen allerdings viele Konfusionen; die Schlagworte bedürfen einer genaueren Klärung:

- Der Begriff *Internet* wird für den ursprünglichen und offenen Netzverkehr verwendet. Hier findet ein öffentlicher Informationsaustausch statt. Das Internet ist darüber hinaus ein Medium zur Kooperation mit dem Endkunden oder dem Ursprungslieferanten. In diesem Bereich sind auch alle Dienste rund um den elektronischen Markt anzusiedeln.
- Technisch betrachtet handelt es sich beim Begriff des *Intranet* ebenfalls um das Internet. Der Einsatzzweck ist jedoch auf nichtöffentliche Informationen und firmeninterner Gruppenunterstützung beschränkt.
- Unter dem Schlagwort *Extranet* werden alle Dienste verstanden, die geschlossene Benutzergruppen (z.B. virtuelle Unternehmen, firmenübergreifende Teams, Interessengemeinschaften, etc.) in ihrer Kooperation und ihrem Informationsaustausch unterstützen.

Diese Betrachtung der Möglichkeiten der Internet–Nutzung deutet bereits auf Nutzungspotentiale hin, die in Kapitel 5 näher betrachtet werden sollen.

2.4 Bedrohungen bei der Unterstützung dieser Unternehmensziele

Die unternehmerischen Möglichkeiten werden allerdings durch vielfältige Bedrohungen im Internet in Frage gestellt. Um diesen Bedrohungen geeignet begegnen zu können, müssen die Schutzziele bei der Nutzung des Internet festgelegt werden. Im Internet beziehen sich

Schutzziele i.d.R. auf Informationen, deren Austausch (Kommunikation), die in den Informationsaustausch involvierten Menschen und Systeme und die verwendeten Technologien. Die Formulierung der Schutzziele erfolgt i.d.R. über die Identifikation von Grundbedrohungen. Informationen sind durch sechs Grundbedrohungen gefährdet, die sich folgenden Kategorien zuordnen lassen (in Anlehnung an [ISO96, Ker91]):

Vertraulichkeit (confidentiality): Sicherstellung, dass Informationen nicht durch unautho-risierte Personen, Instanzen oder Prozesse eingesehen werden können.

Integrität (integrity): Integrität bezieht sich auf Integrität von Daten und Systemen. Datenintegrität ist die Sicherstellung, dass Daten nicht in einer unauthorisierten Art und Weise verändert oder zerstört wurden. Systemintegrität ist die Sicherstellung, dass ein System unbeeinträchtigt mit der gewünschten Performance zur Verfügung steht und nicht durch unauthorisierten Zugang manipuliert wurde.

Verfügbarkeit (availability): Sicherstellung des Zugriffs und der Funktionsfähigkeit bei Zugriffswunsch einer authorisierten Instanz Die Verfügbarkeit wird durch drei Faktoren beeinflusst: die Verfügbarkeit des Netzes, der Endsysteme und der Daten bzw. Anwendungen.

Verbindlichkeit (accountability, non–repudiation): Sicherstellung, dass die Aktionen einer Instanz (Benutzer, Prozess, System, Information etc.) ausschliesslich dieser Instanz zugeordnet werden können und dass die Kommunikationsbeziehung bzw. der Informationsaustausch nicht geleugnet werden kann.

Authentizität (authenticity): Sicherstellung, dass die Identität eines Subjekts auch die ist, für die sich das Subjekt ausgibt. Ein Subjekt kann in diesem Zusammenhang ein Benutzer, ein Prozess, ein System oder eine Information sein. Authentizität ist die Voraussetzung für Verbindlichkeit.

Betriebssicherheit (reliability): Sicherstellung eines konsistenten und gewünschten Verhaltens und Ergebnisses. Voraussetzung für Integrität und Verbindlichkeit.

Betriebssicherheit wird häufig auch unter Safety als eigenständige Disziplin aufgeführt. Dies ist vor allem im ingenieurwissenschaftlichen Bereich sinnvoll, bei dem der sichere Betrieb von Anlagen im Vordergrund steht.

Wie bereits in der Aufzählung angedeutet, sind diese Grundbedrohungen nicht überschneidungsfrei. Streng genommen käme man mit den ersten drei Grundbedrohungen Vertraulichkeit, Integrität und Verfügbarkeit aus; zur Verdeutlichung unterschiedlicher Bedrohungsaspekte ist die Betrachtung aller 6 Sicherheitsanforderungen jedoch sinnvoll. Z.B. ist für das Nutzungsszenario elektronischer Handel über das Internet der Aspekt der Verbindlichkeit immens wichtig.

Diese Grundbedrohungen ergeben das Bedrohungspotential, dass der Nutzung des Internet entgegen steht. Trifft nun eine konkrete Bedrohung auf eine Schwachstelle im Unternehmen (organisatorisch, menschlich oder technisch), dann entsteht für das Unternehmen eine

Gefahr. Aus dieser Gefahr entsteht schliesslich in Zusammenwirkung mit dem Schadenspotential und einer Eintrittswahrscheinlichkeit das Risiko der Nutzung (vgl. Abbildung 2.5). Nachfolgend werden die mit dieser Risikoanalyse verbundenen Begriffe detaillierter erläutert.

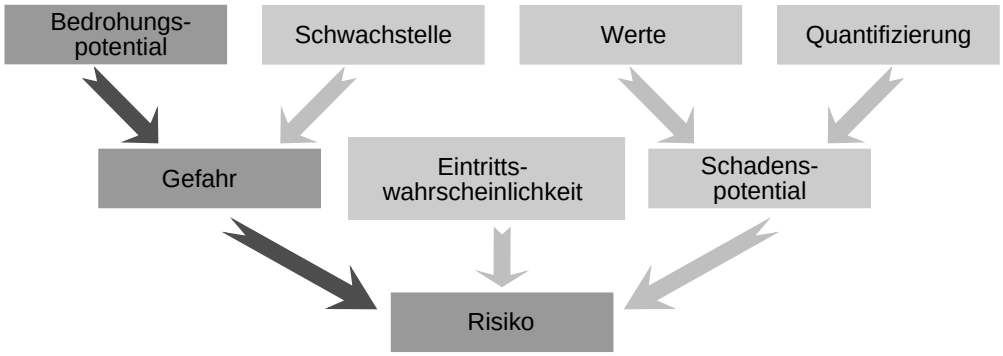

Abbildung 2.5: Bedrohungen, Schwachstellen, Werte, Risiken

Bedrohungen bzw. Bedrohungspotentiale (threats) sind potentielle negative Einwirkungen auf ein System von aussen. Schwachstellen sind Sicherheitslücken innerhalb des Systems. Die Existenz einer Bedrohung alleine stellt noch keine Gefahr für das System dar. Erst wenn eine Bedrohung auf eine Schwachstelle trifft, handelt es sich um eine Gefahr für das System (in Anlehnung an [Bor96, S. 8] und [Dev93, S. 44ff]). Im weiteren wird somit der Begriff Gefahr als eine konkretisierte Bedrohung verstanden, da zum Eintreten einer Gefahr eine Bedrohung auf eine Schwachstelle treffen muss.

Auf der anderen Seite sind innerhalb eines Systems Werte (im Sinne von Informationen, Einrichtungen, etc.) anzutreffen. Diese Werte müssen quantifiziert werden, d.h. es muss eine Summe für diese Werte gefunden werden (z.B.: Was sind uns die Informationen zu unserem neuen Produkt wert?). Um diese Summe zu bestimmen, können mögliche Bedrohungsarten betrachtet werden.

Nach der Ermittlung von Werten und Gefahren kann der Schaden ermittelt werden, der entsteht, wenn ein solcher Wert einer Gefahr ausgesetzt wird. Das Risiko ergibt sich schliesslich aus dem Schaden und der Wahrscheinlichkeit für den Schadenseintritt (vgl. [Dev93, S. 54] und [SB92, S. 37]).

Die in diesem Abschnitt erfolgten Darstellungen zu generellen Sicherheitszielen, Grundbedrohungen und zur Risikoanalyse legen eine organisatorische Einbettung dieser Aufgaben nahe. Dies führt zur Notwendigkeit des Sicherheitsmanagements, das im weiteren näher erörtert wird.

2.5 Sicherheitsmanagement zur Unterstützung der Unternehmensziele

Das Ziel des Sicherheitsmanagements ist die Sicherung des kontinuierlichen Geschäftsablaufes (business safeguarding). Die Unternehmensziele, -strategien und -

grundsätze (vgl. Abschnitt 2.2) müssen durch das Sicherheitsmanagement abgesichert werden. Ein immer wichtiger werdender Teil des Sicherheitsmanagements umfasst die Absicherung der Ressource Information im Rahmen des Informationssicherheitsmanagements (ISIM). Informationen werden in einzelnen Systemen verarbeitet, die ein adaptiertes ISIM auf Systemebene benötigen. Die Hierarchiebildung aus Visionen zur Bestimmung von Geschäftszielen, deren Umsetzung (Strategie) und die daraus resultierenden Handlungsanleitungen (Politiken) ist in Abbildung 2.6 (in Anlehnung an [ISO96]) dargestellt.

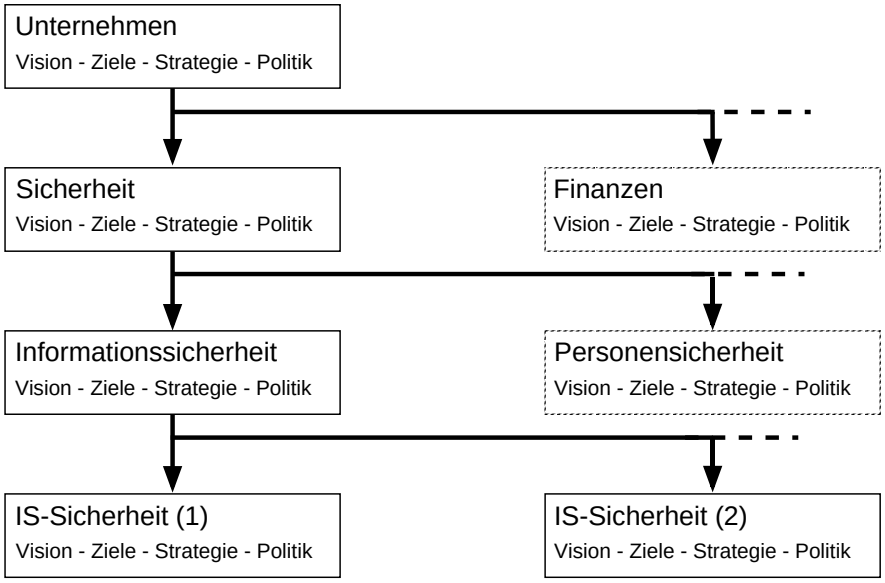

Abbildung 2.6: Ableitung der Informationssicherheit aus Unternehmenszielen

Um die auf ein Unternehmen einwirkenden Bedrohungen handzuhaben, muss sich das ISIM mit folgenden Teilaufgaben beschäftigen [Dev95, Hei96, ISO96, KW97, KWT97, SB92]:

- *Top–Management Unterstützung:* notwendige Voraussetzungen jeder unternehmerischen Aufgabe.
- *Entwicklung von Visionen, Zielen, Strategien, Politiken:* Basis und Leitfaden für das Sicherheitsmanagement ist die Entwicklung einer Sicherheitsstrategie und einer Sicherheitspolitik.
- *Bestimmung von Kompetenzen und Verantwortlichkeiten:* das Sicherheitsmanagement muss organisiert werden.
- *Schaffung eines Sicherheitsbewusstseins:* die entscheidende notwendige Voraussetzung für ein erfolgreiches ISIM ist die Sensibilisierung für Sicherheitsfragen.
- *Abgleich von Sicherheitszielen und Informationszielen:* Die Sicherheitspolitik als Ergebnis der Sicherheitsziele muss im Einklang mit Unternehmenszielen und der Informationspolitik des Unternehmens stehen.
- *Risikomanagement:* Analyse von Werten, Bedrohungen und Schwachstellen als Basis für die Identifizierung von Sicherheitsmassnahmen.

- *Massnahmenentwicklung:* Identifikation und Analyse von Gegenmassnahmen.
- *Implementierung von Sicherheitsmassnahmen:* Realisierung der identifizierten und ausgewählten Massnahmen.
- *Notfallplanung:* Planung der Begegnung unvorhergesehener Fälle. Entwicklung von Sicherheitsbewusstseinsprogrammen.
- *Schulungen:* alle Aufgaben im Rahmen des Sicherheitsmanagement müssen regelmässig vermittelt werden.
- *Konfigurationsmanagement:* Massnahmen müssen betrieben und gemäss der Sicherheitspolitik konfiguriert werden. Änderungen am System müssen bei der Rekonfiguration berücksichtigt werden.
- *Änderungsmanagement:* Veränderungen des Systems, der Sicherheitspolitik etc. müssen erkannt und berücksichtigt werden.
- *Betrieb von Massnahmen:* Massnahmen müssen betrieben und gewartet werden.
- *Monitoring:* alle sicherheitsrelevanten Systeme und Konfigurationen müssen überwacht werden, um ihre Ordnungsmässigkeit überprüfen und Fehler feststellen zu können.
- *Reaktion auf Zwischenfälle:* eingetretene Fehler müssen adäquat behandelt werden. Die Notfallplanung muss entsprechend angewandt werden.
- *Auditing und Controlling:* Alle Aktionen und Aufgaben müssen sowohl während der Durchführung (Controlling) als auch in regelmässigen Abständen (Auditing) überprüft werden.

Diese Aufzählung von Aufgaben kann entsprechend Abbildung 2.6 nach ihrer Langfristigkeit der Planung weiter gegliedert werden. Um diese Unterteilung zu verdeutlichen, bietet sich die Strukturierung nach strategischen, taktischen und operativen Gesichtspunkten an (siehe Abbildung 2.7). Dabei werden manche Aufgaben mehreren Planungshorizonten zugeordnet. Die Aufgaben Auditing und Controlling wurden nicht mit aufgenommen, da sie sich auf den Gesamtprozess des ISIM beziehen und demgemäss als Unterpunkt zu jeder Aufgabe stehen müssten.

Abbildung 2.7 veranschaulicht auch deutlich, dass der Prozess zur Etablierung eines ISIM auf der strategischen Ebene gefällt werden muss. Der Anstoss hierfür kann allerdings woanders stattfinden. Voraussetzung dafür ist die Betrachtung der Information als zu sichernder Produktionsfaktor.

Taktische Aufgaben sind die Initiierung eines Risikomanagements und die Disposition der Massnahmen, die vorgeschlagen werden. Die Massnahmen müssen dann implementiert werden. Schulungen müssen durchgeführt werden und es müssen Änderungen an der Systemumgebung im Rahmen des Änderungsmanagements und des Konfigurationsmanagements berücksichtigt werden.

Abbildung 2.7: Aufteilung der Aufgaben des ISIM nach ihrem Planungshorizont

Die in diesem Bericht eingeführte Sicherheitskonzeption verschmilzt diese taktischen Aufgaben zu einem einheitlichen Prozess. Die Sicherheitskonzeption ist ein Leitfaden zur durchgängigen Realisierung insbesondere der taktischen Aufgaben des Sicherheitsmanagements.

Operative Aufgaben des ISIM bestehen schwerpunktmässig im Betrieb und der Überwachung der eingesetzten Sicherheitsmechanismen. Zudem muss auf (un)vorgesehene Zwischenfälle in geeigneter Weise reagiert werden.

Diese Aufgabenbereiche sind nicht isoliert zu sehen, es herrscht vielmehr ein reger Informationsfluss zwischen den Blöcken. Es finden insbesondere viele Rückkopplungen von konkreteren Teilaufgaben zu abstrakteren Teilaufgaben statt. Zur Unterstützung der Interaktion zwischen den Blöcken dient die Dokumentation der Sicherheitspolitik, die je nach Aufgabenbereich unterschiedlich konkretisiert ist (vgl. [Dev95, S. 69], [Hei96, S. 246f] und [KWT97]).

2.6 Kernprozess des Sicherheitsmanagements nach ISO 13335

Der fortlaufende Sicherheitsmanagementprozess muss somit die funktionalen Anforderungen der jeweiligen Geschäftsumgebung und die Sicherheitsanforderungen aus den Sicherheitspolitiken entsprechend berücksichtigen. Die General Management Guidelines for Information Security der ISO 13335 TR [ISO96] beschreiben ein solches Vorgehen.

Abbildung 2.8 illustriert den Kernprozess in diesem Vorgehen. Ausgehend von einer Informationssicherheitspolitik und einer Geschäftspolitik (Sicherheitsanforderungen und funktionale Anforderungen) kann, bezogen auf ein zu untersuchendes System, eine erste

vorläufige Risikoanalyse durchgeführt werden, die als primäres Ziel der Wertbestimmung der Informationen und der Informationssysteme dient. Diese Wertbestimmung ist wiederum Basis für die Entscheidung, ob eine detaillierte Risikoanalyse durchgeführt werden soll, oder ob eine Grundschutzmassnahme ausreichend erscheint. Unabhängig davon, für welche Variante man sich entschieden hat, müssen anschliessend geeignete Massnahmen identifiziert und schliesslich implementiert werden. Nach einem Durchlauf kann der Prozess wieder von vorne beginnen.

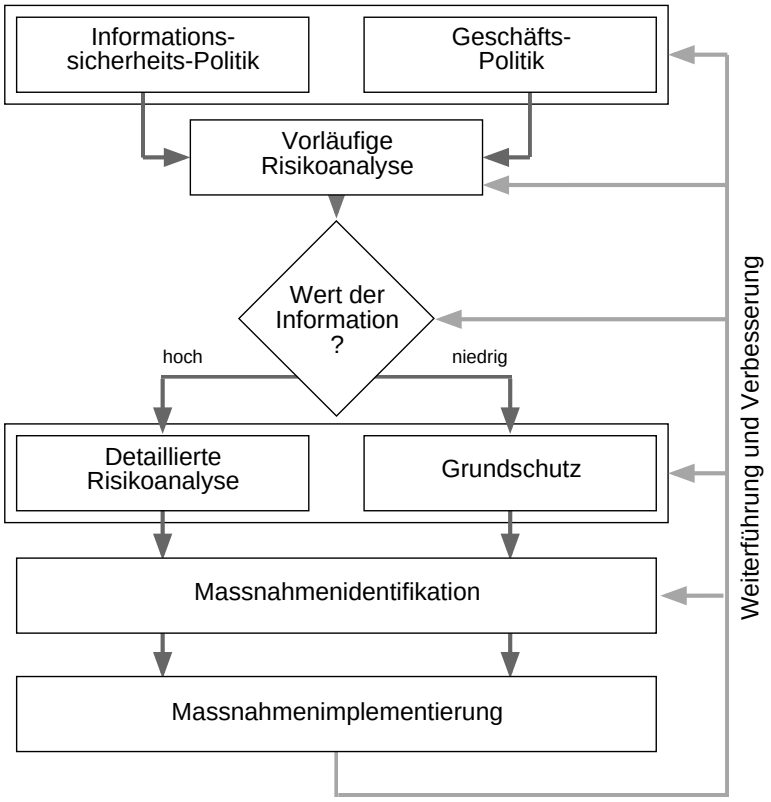

Abbildung 2.8: Kernprozess des Informationssicherheitsmanagements

Hauptaufgaben des Sicherheitsmanagements sind demnach die Harmonisierung von Geschäftspolitik und Sicherheitspolitik, die Risikoanalyse mit der Bestimmung von Werten, Bedrohungen und Schwachstellen sowie eine daraus folgende Bestimmung des Risikos und geeigneter Massnahmen. Diese Massnahmen können organisatorischen, technischen oder personellen (menschlichen) Gesichtspunkten zugeordnet werden.

2.7 Zusammenfassung

Ausgehend von der Analyse der Unternehmensziele und deren Unterstützung durch die Nutzung des Internet wurden mögliche Grundbedrohungen aufgezeigt und in ein generelles Informationssicherheitsmanagement eingeführt. Wesentlicher Bestandteil des ISIM auf taktischer Ebene ist die Sicherheitskonzeption, die die Aufgabenbestandteile Nutzungsidentifikation, Risikoanalyse, Massnahmenidentifikation und -analyse, Implementierung

und Konfiguration von Massnahmen, Betrieb der Massnahmen, Konfigurationsmanagement und Änderungsmanagement zu einem einheitlich strukturierten Prozess zusammenfasst. Dieser Bericht wird schwerpunktmässig die Sicherheitskonzeption als Kern des Sicherheitsmanagements behandeln. Dabei wird jedoch auch auf weitere wichtige Aspekte im Zusammenhang mit der Sicherheitskonzeption eingegangen werden (z.B. Schulung, Awareness etc.).

Die in diesem Abschnitt eingeführten Aspekte des Sicherheitsmanagements werden im folgenden an den geeigneten Stellen weiter ausgeführt und detailliert betrachtet; z.B.: Was sind Ausprägungen der Grundbedrohungen im Internet, welche Nutzungspotentiale gibt es konkret etc.

Kapitel 3

Das Internet

3.1 Geschichte des Internet

Im Rahmen eines Forschungsprojektes der DARPA (Defense Advance Research Project Agency, Unterabteilung des amerikanischen Verteidigungsministeriums) sollte eine Architektur für ein weitverteiltes Netzwerk entworfen werden, das ohne zentrale Steuerungsinstanz auskommt. Ziel war die Entwicklung eines Netzwerkes, das auch dann noch funktioniert, wenn Teile davon zerstört oder ausgefallen sind. Als Ergebnis der Forschungsarbeiten entstand 1969 das ARPANET.

Das ARPANET war in seinen Grundideen dem heutigen Internet sehr ähnlich. Die zentrale Neuerung gegenüber bisherigen Netzen war ein paketorientierter Vermittlungsdienst, auf dem verbindungs- und paketorientierte Transportdienste aufgesetzt werden konnten. Jeder Rechner hatte eine netzweit eindeutige Kennung, die als Adressierungsinformation für die Pakete diente. Kein Knoten brauchte die Topologie des gesamten Netzes zu kennen; jeder kannte nur seine lokalen Nachbarn und tauschte mit ihnen Adressierungsinformationen aus.

Die Transportdienste und die darauf aufbauenden Anwendungsprotokolle waren zunächst sehr primitiv, wurden jedoch Anfang der 70er Jahre weiterentwickelt. 1974 entstand eine erste Spezifikation der TCP / IP–Protokollfamilie, auf der das Internet heute beruht. 1975 wurde TCP / IP in den Kernel des BSD–Unix der University of Berkeley, California, integriert, und kurz darauf ein grösseres Testnetz, das NSFnet (National Science Foundation Network) in Betrieb genommen, an das zunächst vier amerikanische Universitäten angeschlossen war.

Anfang der 80er Jahre waren bereits die meisten amerikanischen Universitäten vernetzt, und auch europäische Universitäten begannen, sich anzuschliessen. Das Netz wurde dabei auch noch für militärische Zwecke genutzt, und erste Sicherheitsprobleme machten sich bemerkbar. Dies führte 1983 zur Abspaltung des MILNET, so dass das restliche ARPANET nur noch für zivile Forschung verwendet wurde.

Für das ARPANET setzte sich allmählich der Begriff Internet durch, was die mittlerweile weltweite Verbreitung des Netzes widerspiegelte. 1992 kam es zur formalen Auflösung des ARPANET. Das Internet lebt jedoch weiter, und die Zuwachsrate ist weiterhin exponentiell.

Mit der Erfindung des World Wide Web 1993 wurde das Netz auch für Laien nutzbar und somit für kommerzielle Zwecke interessant. Heute sind eine Vielzahl von Firmen, Privatorganisationen, Behörden und Ausbildungsstätten im Netz vertreten.

3.2 Das Internet heute

Das Protokoll, das von allen Rechnern des Internet genutzt wird, ist TCP / IP. Dieses Protokoll ist die Minimalvoraussetzung, um einen Rechner an das Internet anzuschliessen.

Prinzipiell ist TCP / IP für Vernetzungen von beliebig vielen Rechnern und beliebig grossen / kleinen Entfernungen geeignet. Dasselbe gilt für die Netzwerkhardware: TCP / IP funktioniert mit allen gängigen Netzwerktopologien (z.B. Ethernet, Token–Ring) und Kombinationen daraus. TCP / IP wird von allen gängigen Betriebssystemen unterstützt.

Das TCP / IP–Schichtenmodell besteht aus vier Schichten (siehe Abbildung 3.1).

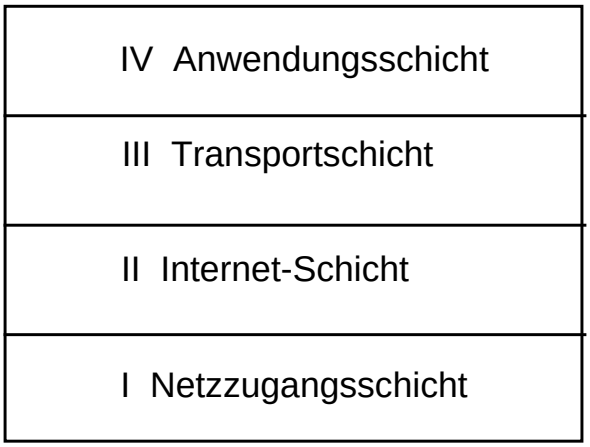

Abbildung 3.1: Das TCP / IP–Schichtenmodell

Im Gegensatz dazu sieht das ISO / OSI Referenzmodell sieben Schichten vor. Abbildung 3.2 zeigt eine mögliche Einordnung von TCP / IP in das OSI–Modell [Hun92, S. 90].

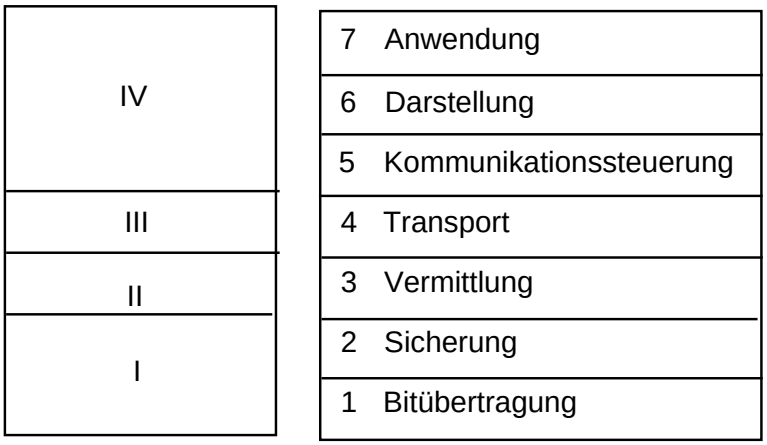

Abbildung 3.2: Einordnung von TCP / IP in das OSI–Schichtenmodell

Die Zuordnung ist nicht eindeutig; insbesondere wird in einigen Büchern auch die Internet–Schicht (II) in die OSI–Schicht 4 eingeordnet.

3.3 Aufgabe und Arbeitsweise der Protokollebenen

Die vier Schichten des TCP / IP–Protokollstacks bilden eine Hierarchie im Bezug auf die Abstraktionsebene und auf den Datenfluss. Jede Schicht kann ”direkt” mit der entsprechenden Schicht auf der Gegenseite kommunizieren, indem sie die Funktionen der darunterliegenden Schicht benutzt (siehe Abbildung 3.3).

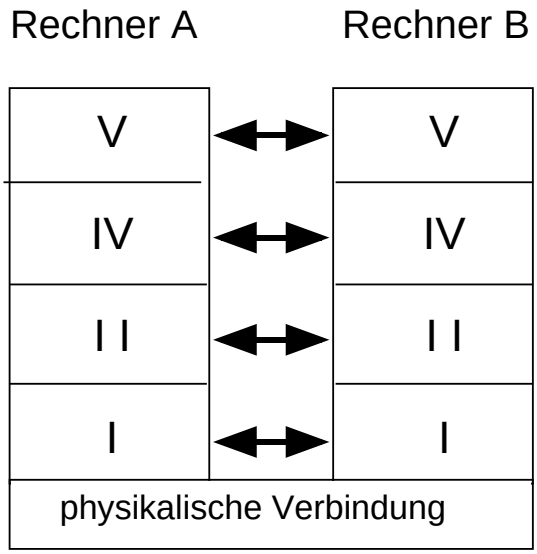

Abbildung 3.3: Abstraktionsebenen des TCP / IP–Protokolls

Eine physikalische Verbindung zu einem anderen Rechner besteht nur unterhalb der Netzzugangsschicht. Ein Protokoll auf Schicht I kann mit dem entsprechenden Protokoll auf der anderen Seite Daten austauschen, indem es die physikalische Verbindung nutzt. Ein Protokoll auf der Schicht II kann dies tun, indem es die Transportfähigkeiten der Schicht I benutzt. Dasselbe gilt sinngemäss für die darüberliegenden Schichten.

Natürlich leiten die einzelnen Schichten nicht nur Daten weiter. Jede Schicht erfüllt eine spezifische Funktion. Die Aufgaben der einzelnen Schichten werden noch genauer erläutert.

Zur Erfüllung ihrer Aufgabe muss eine Schicht im Allgemeinen den Daten zusätzliche Steuerinformationen hinzufügen. Dies können Informationen über das Datenformat, die Beschaffenheit und Qualität der Verbindung oder eine Prüfsumme, die die Integrität der Daten sichern soll, sein. Die Datenmenge wird also um so grösser, je weiter die Daten im Schichtenmodell nach unten wandern. Abbildung 3.4 verdeutlicht dies.

Aus der Abbildung wird mehreres deutlich:

- Die Nutzdaten der Anwendung bilden die Daten (Data, D) der Anwendungsschicht IV. Die Schicht fügt ihre eigenen Kontrollinformationen (Header, H) hinzu und leitet

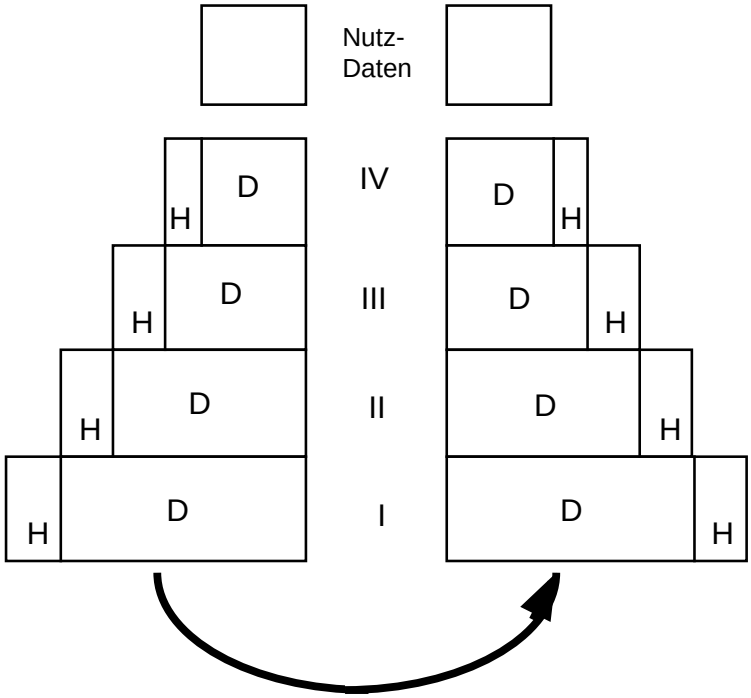

Abbildung 3.4: Daten und Kontrollinformationen (D: Daten, H: Header der einzelnen Schichten)

das ganze Paket an die Schicht III weiter. Die übrigen Schichten verhalten sich entsprechend.

- Eine Schicht interessiert sich nicht für die Unterteilung in Daten und Header der darüberliegenden Schicht, sondern betrachtet das, was sie übergeben bekommt, komplett als zu transportierende Daten.
- Die Datenmenge, die tatsächlich über die Leitung geht, ist häufig erheblich grösser als die Nutzdaten auf der Anwendungsebene (auch wenn dies in der Abbildung übertrieben dargestellt ist).
- Die Header werden meistens, aber nicht immer, an den Anfang der Nutzdaten gehängt. Eine Ausnahme bildet z.B. das Ethernet–Protokoll; dort gibt es Steuerinformationen sowohl vor als auch hinter den Nutzdaten.

In der Literatur zu TCP / IP (z.B. [Hun92, S. 90]) werden die zu transportierenden Daten der Schicht I (Header und Nutzdaten zusammen) Frames, die der Schicht II Datagramme und die der Schicht III je nach Protokoll Pakete oder Datenströme genannt. Auf der Schicht IV gibt es keine einheitlichen Bezeichnungen.

3.4 Die Netzzugangsschicht

Aufgabe der Protokolle der Netzzugangsschicht ist es, sich um den Datentransport auf einem physikalischen Netzwerk zu kümmern. Mit den unterschiedlichen physikalischen

Medien und Übertragungsverfahren unterscheiden sich auch die Protokolle dieser Schicht; dennoch gibt es Aufgaben, die alle Protokolle der Schicht I gemeinsam haben, beispielsweise Kontrolle des Datenflusses auf der Leitung und Absicherung der Daten mittels Prüfsummenverfahren vor unbeabsichtigten Verfälschungen.

Gängige Betriebssysteme bieten Schicht–I–Protokolle mindestens für folgende Übertragungsmedien: V.24 / RS232, Centronics, ISDN, Ethernet, Token Ring.

3.5 Das Internet Protocol (IP)

IP ist das Schicht–II–Protokoll des TCP / IP–Protokollstacks. Es ermöglicht den Datentransport auch zwischen Rechnern, die nicht unmittelbar mit einer physikalischen Leitung verbunden sind. Das Finden eines Weges von einem Quell- zu einem Zielrechner in einem Netz wird als Routing bezeichnet. Eine weitere Aufgabe der Schicht II ist die einheitliche Adressierung aller Rechner eines Netzwerkes, unabhängig von der verwendeten Netzwerktechnologie.

Jeder Rechner in einem TCP / IP–Netzwerk besitzt (mindestens) eine eindeutige Kennung, genannt IP–Nummer. Die IP–Schicht jedes Rechners besitzt Informationen darüber, an welche (direkt verbundenen) Rechner Datagramme für bestimmte Ziel–IP–Nummern weiterzuleiten sind, damit sie irgendwann ihr Ziel erreichen. Aufbau der IP–Nummern und Ablauf des Routings werden unten beschrieben.

Die Funktionalität von IP besteht also darin, Datagramme an bestimmte Rechner im Netz weiterzuleiten. Es findet keine Bestätigung über das Ankommen der Daten statt, die Daten werden nicht vor Verfälschung geschützt und einzelne Datagramme können sich unterwegs überholen. Alle drei Eigenschaften sind häufig unerwünscht; es ist Aufgabe der darüberliegenden Schichten, sich darum zu kümmern.

Der IP–Header kann 20 oder 24 Bytes lang sein; ersteres ist der Normalfall. Er enthält Steuerinformationen, die für das Weiterleiten der Datagramme und die Verarbeitung auf dem Zielrechner wichtig sind. Unter anderem sind dies:

- die IP–Nummer des absenden Rechners
- die IP–Nummer des Zielrechners
- die Protokollnummer des Schicht–III–Protokolles, von dem die Daten kommen (und an das sie weitergereicht werden sollen, wenn sie auf dem Zielrechner angekommen sind)
- die Länge des Datagrammes
- eine Prüfsumme über den Header, um Verfälschungen erkennen zu können
- Flags zum Aktivieren spezieller Optionen; auf eine davon wird später noch näher eingegangen.

3.6 Das Internet Control Message Protocol (ICMP)

ICMP ist ein Protokoll der Schicht III. Das bedeutet, dass ICMP–Datenpakete über IP transportiert werden. Die Aufgabe von ICMP besteht im Wesentlichen darin, Fehler- und andere Statusmeldungen zu versenden, die den IP-Datagrammtransport betreffen.

ICMP–Fehlermeldungen werden zum Beispiel generiert, wenn ein Datagramm empfangen wurde, für dessen Zieladresse keine Routinginformationen bekannt sind, oder wenn der IP–Header fehlerhaft ausgefüllt ist (die Prüfsumme aber korrekt ist). ICMP–Fehler werden nicht generiert, wenn die Prüfsumme im Header fehlerhaft ist.

Die Maxime lautet, dass Fehlermeldungen nur dann verschickt werden, wenn der absendende Rechner schuld an dem Fehler ist. Fehlerhafte Prüfsummen dagegen liegen meistens an Übertragungsfehlern auf dem physikalischen Medium. Der Absender kann dafür nichts, und hat deshalb auch nichts von einer solchen Meldung. Ausserdem steht der Absender nicht zweifelsfrei fest, denn wenn die Header–Prüfsumme fehlerhaft ist, könnte ja auch die Absender–IP–Nummer verfälscht worden sein.

Des weiteren werden niemals Fehlermeldungen über fehlerhafte ICMP–Pakete verschickt (weil dies sonst zu einer endlosen Kette von Fehlermeldungen führen könnte), oder wenn ein fehlerhaftes Datagramm an mehrere Empfänger verschickt wurde (denn sonst würde die Netzlast explodieren).

Neben Fehlermeldungen gibt es auch noch andere Informationen, die über ICMP verschickt werden können. Die wichtigsten davon sind:

echo request und echo reply ermöglichen einen Mechanismus zum Testen, ob ein Rechner im Netzwerk erreichbar ist. Wenn die TCP / IP–Implementierung ein echo request empfängt, schickt sie ein echo reply an den Absender. Dieser Mechanismus wird von dem Kommando ping verwendet.

Ein redirect teilt der IP–Schicht mit, dass ein Rechner oder ein Netzwerk über eine neue Route erreichbar ist, ändert also die Routing–Tabelle der IP–Schicht.

3.7 Das Transmission Control Protocol (TCP)

TCP ist das zentrale Protokoll auf Schicht III des TCP / IP–Protokollstacks. Es ermöglicht einen verbindungsorientierten, reihenfolgetreuen, bidirektionalen und gesicherten Datenstrom zwischen zwei Programmen auf verschiedenen Rechnern (oder auch dem gleichen Rechner). Die Aufgaben von TCP sind also

- Aufteilen eines beliebig langen Datenstromes in einzelne Segmente;
- Verschicken dieser Segmente über IP;
- Wiederholtes Verschicken von Segmenten, die defekt oder gar nicht beim Empfänger angekommen sind;

- Eliminieren duplizierter Segmente;
- Reihenfolgetreues Zusammensetzen des Datenstromes;
- Weitergabe des Datenstromes an das dazugehörige Programm;
- Management von Verbindungen zwischen Programmen.

Der Ablauf einer TCP–Verbindung sieht, stark vereinfacht, etwa so aus wie in Abbildung 3.5 dargestellt.

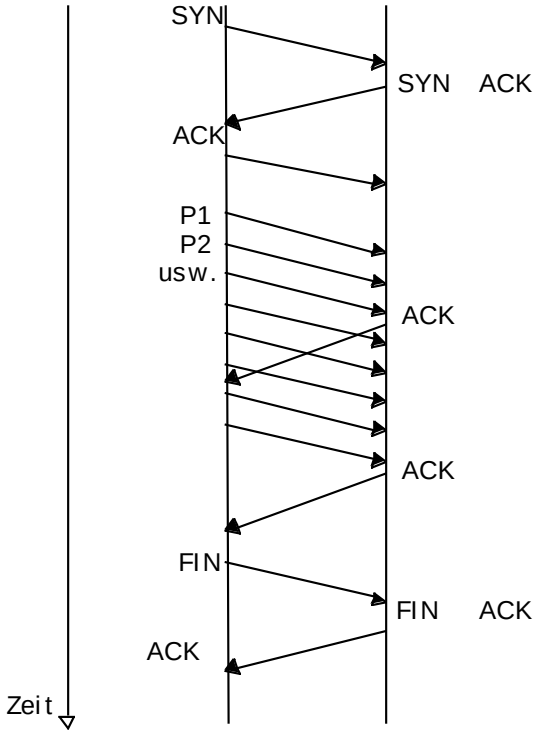

Abbildung 3.5: TCP–Datentransport (vereinfacht)

Der Verbindungsaufbau besteht aus drei TCP–Steuerpaketen (die überhaupt keine Nutzdaten enthalten): SYN (Synchronise), SYN ACK (SYN Acknowledge) und ACK. Darauf folgt die Übertragung von Datenpaketen (in der Abbildung zur Vereinfachung nur in eine Richtung dargestellt) und Bestätigungen korrekt empfangener Daten. Der Verbindungsabbau wird mit FIN eingeleitet; die Verbindung ist beendet, wenn die Gegenseite das FIN bestätigt hat.

Korrekt empfangene Datenpakete werden bestätigt. Eine Bestätigungsnachricht (ACK) kann mehrere Datenpakete bestätigen. Dadurch wird zum einen die Netzlast gesenkt, zum anderen kann die sendende Seite neue Daten sofort nachschieben, auch wenn die letzten Pakete noch nicht bestätigt wurden. Pakete, die längere Zeit nicht bestätigt wurden, werden neu verschickt. Auf diese Weise werden Paketverluste und Verfälschungen korrigiert.

Die Reihenfolgetreue wird dadurch erreicht, dass die einzelnen Pakete mit Sequenznummern versehen sind. Aufeinanderfolgende Pakete in eine Richtung haben fortlaufende Sequenznummern; für beide Übertragungsrichtungen gibt es getrennte Numerierungen. Da-

durch kann der Datenstrom in seiner ursprünglichen Reihenfolge zusammengesetzt werden, auch wenn sich einzelne Pakete unterwegs überholen.

Um Datenverfälschung zu vermeiden, wird eine Prüfsumme über Header und Nutzdaten berechnet und beim Empfänger überprüft. Die Prüfsumme ist 16 Bit lang, so dass Übertragungsfehler mit hoher Wahrscheinlichkeit erkannt werden.

Um die Zuordnung der einzelnen Datenpakete zu den Datenströmen der Programme zu ermöglichen, zwischen denen die Verbindung besteht, enthält der TCP–Header zwei Zahlen, die eine eindeutige Zuordnung der Pakete zu den Datenströmen ermöglicht: Absender- und Empfänger–Portnummer. Genauer: Eine TCP–Verbindung zwischen zwei Programmen auf unterschiedlichen Rechnern ist eindeutig gekennzeichnet durch die Angaben

- Absender–IP–Nummer,
- Absender–Portnummer,
- Empfänger–IP–Nummer,
- Empfänger–Portnummer.

Der Rechner, der eine TCP–Verbindung öffnet, wählt als Absender–Portnummer immer eine Zahl, der noch keine ausgehende TCP–Verbindung zugeordnet ist. Auf der Gegenseite können mehrere Verbindungen auf die gleiche Portnummer geöffnet werden, ohne dass die Eindeutigkeit verletzt wird. Portnummern liegen zwischen 0 und 65536.

Der Header eines TCP–Paketes enthält also unter anderem

- Absender–Portnummer
- Empfänger–Portnummer
- Sequenznummern (getrennt für beide Übertragungsrichtungen)
- Prüfsumme
- Kennung, um was für eine Art von Paket es sich handelt (Daten, SYN, ACK, ...)

Weiterhin enthält er noch eine Längenangabe und Flags für Optionen, auf die teilweise im Kapitel über Paketfilter noch eingegangen wird.

3.8 Das User Datagram Protocol (UDP)

UDP ist ein weiteres Transportprotokoll auf Schicht III. Es bietet eine einfache Alternative zu TCP. UDP ermöglicht einen ungesicherten, verbindungslosen Transport zwischen zwei Programmen und garantiert auch keine Reihenfolgetreue.

Der Unterschied zum direkten Datentransport über IP besteht also lediglich darin, dass die Daten nicht zwischen zwei Rechnern, sondern zwischen Programmen transportiert werden. Um die Zuordnung zu den Programmen zu gewährleisten, wird das Portnummernkonzept von TCP verwendet. UDP überprüft die Integrität der Datenpakete mittels einer Prüfsumme. Einen Schutz gegen Paketverlust, Überholung oder Duplizierung gibt es dagegen nicht; es ist Aufgabe der Anwendungen, sich darum zu kümmern.

Der Vorteil von UDP gegenüber TCP besteht in der kleineren Datenmenge. Der UDP–Header ist lediglich 8 Bytes lang, der TCP–Header dagegen mindestens 20 Bytes. Der TCP–Verbindungsauf- und -abbau erfordert zudem, dass mindestens 5 IP–Datagramme über die Leitung gehen. UDP wird daher häufig für Dienste eingesetzt, bei denen kleine Datenpakete in unregelmässigen Abständen verschickt werden müssen.

3.9 Adressierung in TCP/IP–Netzen

3.9.1 IP–Nummern

Jeder Rechner in einem TCP / IP–Netz besitzt (mindestens) eine IP–Nummer; diese Nummer muss netzweit eindeutig sein. Am besten sollte diese Nummer weltweit eindeutig gewählt werden, damit es keinen Umstellungsaufwand beim Anschluss an das weltweite Internet gibt.

IP–Nummern sind immer von der Form $a.b.c.d$, wobei $0 \leq a, b, c, d \leq 255$ sind. Genauer gesagt ist eine IP–Nummer eine 32–Bit–Zahl, die üblicherweise durch vier 8–Bit–Zahlen dargestellt wird. Häufig ist es auch nützlich, die 8–Bit–Zahlen als Binärzahl darzustellen, also durch 8 Ziffern, die nur ”0” oder ”1” sein können. Zum Beispiel ist 110001102 = 128+64+4+2 = 198.

Eine IP–Nummer besteht aus zwei Teilen, dem Netzwerkteil und dem Rechnerteil. Sinn dieser Unterteilung ist, dass mehrere Rechner, die an dem selben physikalischen Netzwerkstrang hängen, IP–Nummern mit gleichem Netzwerk- und unterschiedlichem Rechnerteil bekommen sollen. Diese Zuordnung ist nicht bindend, sollte jedoch aus Übersichtlichkeits- und Effizienzgründen eingehalten werden.

Die möglichen IP–Nummernbereiche sind in verschiedene Klassen unterteilt.

Class–A–Netze: Die IP–Nummer ist von der Form $a.x.y.z$; dabei ist $1 \leq a \leq 126$. Das bedeutet, dass die Binärdarstellung von a mit einer 0 beginnt. In einem Class–A–Netz geben die ersten 8 Bits der IP–Nummer (also die Zahl a) die Netzwerknummer an; die restlichen 24 Bit enthalten die Rechnernummer. Es gibt also nur 126 verschiedene Class–A–Netze. Jedes dieser Netze kann aber über 16 Millionen Rechner enthalten.

Class–B–Netze: $a.b.x.y$ mit $128 \leq a \leq 191$ Die Binärdarstellung von a beginnt also mit binär 10. Bei einem Class–B–Netz besteht der Netzwerkteil der IP–Nummer aus den vorderen 16 Bit, der Rechnerteil aus den hinteren 16 Bit. Es gibt über 16000 verschiedene Class–B–Netze. Jedes davon kann über 65000 Rechner enthalten.

Class–C–Netze: $a.b.c.x$ mit $192 \leq a \leq 223$. Die Binärdarstellung von a beginnt mit 110. Hier ist der Netzwerkteil 24 Bit, der Rechnerteil 8 Bit lang. Somit gibt es über 2 Millionen Class–C–Netze. Jedes davon kann aber nur 254 Rechner enthalten.

Alle IP–Nummern, die nicht in eine dieser drei Kategorien passen, sind reserviert und dürfen nicht verwendet werden. Insbesondere sind reserviert:

0.x.y.z: Das Netz mit der Nummer Null am Anfang wird meist als Synonym für das Default–Netz genommen, also ein Netz, dass nicht in eine andere Beschreibung passt.

127.x.y.z: Dieses Netz ist das sog. loopback–Netz. Jeder Rechner in einem TCP/IP–Netz kann sich selbst (neben seiner regulären IP–Nummer) auch über die loopback–Nummer 127.0.0.1 ansprechen. Dies gilt selbst dann, wenn der Rechner gar keine Netzwerkkarte eingebaut hat.

a.b.c.d mit $a \geq 224$**:** Diese Netze sind für spezielle Erweiterungen reserviert. Beispielsweise werden sie für Multicasts (paketorientierter, unbestätigter Datentransport zu mehreren Empfängern) verwendet.

Innerhalb eines Netzes dürfen die folgenden IP–Nummern nicht für Rechner vergeben werden, da sie eine besondere Bedeutung haben:

- Die IP–Nummer, deren Rechnerteil nur aus (binären) Nullen besteht: Sie gibt die Netzwerkadresse an.
- Die IP–Nummer, deren Rechnerteil nur aus (binären) Einsen besteht: Sie gibt die sog. Broadcast–Adresse an; dies ist eine IP–Nummer, unter der alle Rechner innerhalb eines Netzwerkes zu erreichen sind. Voraussetzung dafür ist aber, dass die verwendete Netzwerktechnologie dies unterstützt. Dies ist beispielsweise bei Ethernet der Fall.

Beispiele für IP–Nummern

132.17.1.95: Es handelt sich um eine Class–B–Adresse, also um den Rechner 1.95 im Netz 132.17 (häufig als 132.17.0.0 geschrieben).

194.101.76.12: Eine Class–C–Adresse: Rechner 12 im Netz 194.101.76 (bzw. 194.101.76.0)

10.255.240.65: Eine Class–A–Adresse: Rechner 255.240.65 im Netz 10 (bzw. 10.0.0.0)

192.13.7.255: Dies ist keine Rechnernummer, sondern die Broadcast–Adresse im Class–C–Netz 192.13.7

202.12.95.0: Dies ist ebenfalls keine Rechnernummer, sondern die Netzwerkadresse des Class–C–Netzes 202.12.95

176.12.95.0: Dies ist dagegen wieder eine Rechnernummer, nämlich die des Rechners 95.0 im Class–B–Netz 176.12

Abschliessend noch eine Bemerkung zur Terminologie. In der Literatur werden unterschiedliche, teils auch widersprüchliche Bezeichnungen für IP–Nummern und verwandte Konzepte verwendet. Daher sollen die Begriffe hier noch einmal abgegrenzt werden: IP–Nummern sind 32–Bit–Zahlen, die von IP zur Adressierung verwendet werden. Eine IP–Nummer kann eine Rechnernummer, eine Netzwerknummer oder eine Broadcast–Adresse sein. Jede IP–Nummer besteht aus einem Netzwerkteil und einem Rechnerteil. Eine Rechnernummer ist also nicht dasselbe wie der Rechnerteil einer IP–Nummer!

3.9.2 Routing in TCP/IP–Netzen

Routing bedeutet das Zustellen eines IP–Datagrammes an den Rechner, dessen IP–Nummer im Datagramm–Header als Ziel angegeben ist. Jeder Rechner im Netzwerk muss Informationen darüber besitzen, auf welche Weise er Datagramme zu allen anderen gültigen Rechnernummern weiterleiten muss, damit sie ihr Ziel erreichen.

Im Allgemeinen sollen, wie bereits erwähnt, Rechner innerhalb eines physikalischen Netzwerkes Rechnernummern mit dem selben Netzwerkteil bekommen. Das bedeutet, dass alle IP–Datagramme direkt an den Empfänger zugestellt werden können, wenn die Ziel–IP–Nummer den selben Netzwerkteil hat wie der eigene Rechner. Unterscheidet sich der Netzwerkteil dagegen, kann das Datagramm nicht direkt zugestellt werden, sondern wird an einen Router weitergegeben, der dem Ziel ein Stückchen näher ist.

Diese Unterscheidung ist nicht unbedingt bindend; es kann auch Ausnahmen geben. Diese sind jedoch betriebssystemabhängig und machen die Vernetzung kompliziert und unübersichtlich. Bei der Planung einer Rechnervernetzung sollte deshalb immer nach obigem Grundsatz vorgegangen werden.

Rechner, die als Router fungieren, müssen daher an mehrere Netze angeschlossen sein und in jedem Netz eine eigene IP–Nummer besitzen. Es hat also nicht jeder Rechner, sondern jeder Netzwerkanschluss in einem Rechner seine eigene IP–Nummer.

Informationen darüber, wie Ziel–IP–Nummern zu erreichen sind, ist in der Routing–Tabelle eines Rechners abgelegt. Abbildung 3.6 zeigt ein Beispiel für eine Netzanordnung, Abbildung 3.7 die dazugehörige Routingtabelle des schraffierten Rechners.

Jeder Rechner besitzt zunächst einmal eine Route zu sich selber über das loopback–Device lo0. Der schraffierte Rechner hat zwei Ethernet–Anschlüsse (eth0 und eth1), die mit den Netzen 192.86.175.0 und 130.60.0.0 verbunden sind, und somit auch zwei IP–Nummern. Er kann also alle Rechner in diesen Netzen direkt erreichen. Das Netz 202.15.180.0 ist über den Router 130.60.1.20 zu erreichen, der wiederum lokal erreichbar ist. Alle anderen Netze im Internet sind über den Router 130.60.10.64 zu erreichen.

3.9.3 Subnetze und Subnetz–Masken

Wie bereits erwähnt, passen in ein Class–A–Netz 16 Millionen Rechner, in ein Class–B–Netz immer noch 65000 Rechner. Es ist im allgemeinen nicht möglich bzw. sinnvoll, so viele

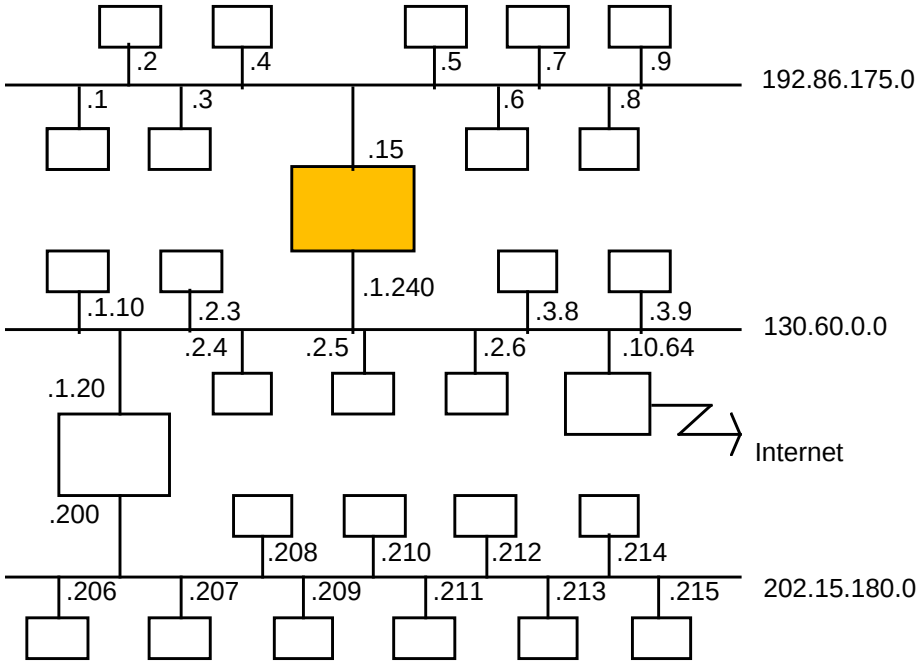

Abbildung 3.6: Beispiel für eine TCP / IP–Vernetzung

Ziel	über	Interface
127.0.0.0	direkt	lo0
192.86.175.0	direkt	eth0
130.60.0.0	direkt	eth1
202.15.180.0	130.60.1.20	eth1
0.0.0.0	130.60.10.64	eth1

Abbildung 3.7: Routingtabelle eines Routers

Rechner in einem physikalischen Netz unterzubringen. Umgekehrt wäre es allerdings Verschwendung, wenn man z.B. ein Class–B–Netz aus nur 10 Rechnern aufbauen würde.

Die Lösung für dieses Problem besteht in einer weiteren Unterteilung einer gegebenen Netzadresse. Mit anderen Worten, es sollen mehr Bits der IP–Nummer für die Netzadresse und weniger für die Rechnernummer verwendet werden.

Zur Angabe, welche Bits zur Netz- und welche zur Rechneradresse gehören soll, dient die Subnetz–Maske. Sie ist ebenfalls eine 32–Bit–Zahl und wird meist in der Form a.b.c.d angegeben, wobei a,b,c und d 8–Bit–Zahlen sind.

In Binärdarstellung geschrieben, besteht eine Subnetz–Maske immer aus einer Folge von Einsen, gefolgt von einer Folge von Nullen. Ist ein Bit auf eins gesetzt, dann gehört das entsprechende Bit der IP–Nummer zur Netzadresse; ansonsten gehört es zu Rechnernummer.

Beispiele

- Die IP–Nummer 128.66.12.1 gehört normalerweise zu einem Class–B–Netz; es handelt sich also um Rechner 12.1 im Netz 128.66.0.0.

  Wenn nun die Subnetz–Maske 255.255.255.0 lautet (das ist binär 11111111.11111111.11111111.000000002), dann bedeutet das, dass jetzt die ersten 24 Bit der IP–Nummer die Netzadresse und die letzen 8 Bit die Rechnernummer sind. Also steht obige IP–Nummer für den Rechner 1 im Subnetz 128.66.12.0.

- Schwieriger wird es, wenn der Übergang zwischen Netz- und Rechnernummer nicht mehr an einer Bytegrenze stattfindet:

  Gegeben sei die IP–Nummer 130.97.16.132 mit der Subnetz–Maske 255.255.255.192. Hier empfiehlt es sich, alle beteiligten Nummern in Binärdarstellung aufzuschreiben:

  SN–Maske: 255.255.255.192 11111111.11111111.11111111.110000002

  IP–Nr.: 130.97.16.132 10000010.01100001.00010000.100001002

  Die Netzadresse berechnet sich, indem alle Bits der Rechnernummer auf Null gesetzt werden: 10000010.01100001.00010000.100000002

  also 130.97.16.128

  Die Rechnernummer kann man ablesen, wenn man alle Bits der Netzwerknummer auf Null setzt: 00000000.00000000.00000000.000001002

  also handelt es sich um Rechner 4 im Netz 130.97.16.128.

  Auf ähnliche Weise kann man noch die Broadcast–Adresse bestimmen: Man setzt dazu alle Bits der Rechneradresse auf Eins: 10000010.01100001.00010000.101111112

  Also lautet die Broadcast–Adresse 130.97.16.191.

- Noch ein anderes Beispiel:

  IP–Nr. 132.90.132.5, SN–Maske: 255.255.240.0

  SN–Maske: 255.255.240.0 11111111.11111111.11110000.000000002

(also 20 Bits Netznummer, 12 Bits Rechnernummer)

IP–Nr.: 132.90.132.5 10000100.01011010.10000100.000001012

Netzadresse: 10000100.01011010.10000000.000000002

also 132.90.128.0;

Rechnernummer: 00000000.00000000.00000100.000001012

also 4.5.

3.10 Das Client–Server–Modell

3.10.1 Beschreibung

Wie in Abschnitt 3.7 erläutert, können über eine TCP–Verbindung zwei Prozesse auf unterschiedlichen Rechnern miteinander kommunizieren. Dabei baut der eine Prozess die Verbindung auf, sobald er die Kommunikation mit dem anderen benötigt; der andere muss ständig bereit sein und auf eine Verbindungsanforderung reagieren. Dieses Konstruktionsprinzip wird als Client–Server–System bezeichnet.

Bei der Konstruktion einer Anwendung nach diesem Prinzip wird das Anwendungssystem meist in zwei Komponenten aufgeteilt:

- Der Server bietet eine bestimmte Teilfunktionalität des Anwendungssystems, meist Verarbeitungs- oder Datenhaltungsfunktionen. Er wartet ständig auf eine Anforderung seiner Dienstleistung und reagiert darauf.
- Der Client ist derjenige Teil des Systems, der die Funktionalität des Servers in Anspruch nimmt. Häufig wird dies durch Aktionen eines Benutzers ausgelöst, und der Client ist diejenige Komponente des Systems, die sich um die Darstellung der Ergebnisse kümmert und direkt vom Benutzer bedient wird.

Technisch gesprochen befinden sich Client und Server auf Rechnern, die über das Netzwerk miteinander verbunden sind (oder beide auf dem selben Rechner). Die Zusammenarbeit läuft ab, indem der Client bei Bedarf Dienstanforderungen an den Server stellt. Der Server beantwortet diese und schickt das Ergebnis zurück an den Client (siehe auch [Tan96, S. 4]).

Client–Server–Anwendungen im Internet werden häufig nach einem der drei folgenden Muster realisiert:

- Client und Server kommunizieren über eine TCP–Verbindung. Der Client baut die Verbindung auf, sobald er die Dienstleistung benötigt, und schickt die zur Bearbeitung nötigen Daten an den Server. Dieser wertet die Anfrage aus, schickt das Ergebnis zurück und schliesst die Verbindung. Typische Dienste, die auf diese Weise realisiert sind, sind das HTTP und die Datenverbindung beim FTP.

- Der Client baut am Anfang der Nutzung (typischerweise beim Start des Client–Programms durch den Benutzer) eine Verbindung zum Server auf. Auf dieser Verbindung schickt der Client jeweils seine Anfragen an den Server; der Server antwortet mit seinem Ergebnis. Die Verbindung bleibt offen, bis der Anwender das Programm beendet; dann schliesst der Client die Verbindung. Beispielsweise benutzen die Protokolle NNTP, FTP (Kommandokanal) und Telnet dieses Muster.
- Client und Server kommunizieren über UDP. Der Client schickt seine Anfragen als UDP–Paket an den Server, der Server antwortet mit einem eigenen UDP–Paket, das an die Absender–IP- und -Portnummer der ersten Paketes adressiert wird. Diese Art der Kommunikation geht schneller als die beiden oben genannten, eignet sich aber nur für kleine Datenmengen. Typischer Vertreter dieses Kommunikationsmodelles ist das DNS–Protokoll (siehe Abschnitt 6.7).

Die Zuordnung zwischen Client und Server ist meist keine 1:1 Beziehung. Im allgemeinen kann ein Server mehrere Clients (gleichzeitig) bedienen, in manchen Fällen kann auch ein Client kann Dienste von mehreren Servern nutzen. Technisch gesehen werden Server so programmiert, das sie

- in einer Schleife arbeiten und nach der Ausführung einer (kurzen) Transaktion wieder in den Wartezustand zurückkehren und bereit für eine neue Transaktion sind,
- für jede Anfrage einen neuen, parallelen Prozess starten, der sich nach Auftragserledigung selbst wieder beendet, oder
- aus zwei Prozessen bestehen, von denen der eine die einkommenden Anfragen speichert (und eventuell nach Dringlichkeit sortiert), der andere die gespeicherten Aufträge der Reihe nach abarbeitet.

Serversoftware für das Betriebssystem Unix ist häufig so programmiert, dass sie mehrere dieser Betriebsarten unterstützt. Beispielsweise beherrscht das in Abschnitt 6.4 vorgestellte E–Mail–Transportsystem Sendmail alle drei genannten Betriebsmodi.

3.10.2 Das Middleware–Konzept

Das Client–Server–Modell sieht eine Aufteilung einer Anwendung in zwei Komponenten vor, die auf (maximal) zwei verschiedenen Rechnern ausgeführt werden. In manchen Fällen ist dies noch nicht ausreichend, sondern eine Zerteilung in drei Komponenten wäre sinnvoller. Gründe dafür können sein:

- Die Anwendung besteht aus drei unterschiedlichen Funktionalitäten, die so stark spezialisiert sind, dass ein Betrieb auf drei verschiedenen Rechnern sinnvoll ist. Ein Beispiel wäre ein Data–Mining–System, das auf sehr grossen Datenbeständen arbeitet: die Datenbestände selbst liegen in einer Datenbank auf einem Mainframe, die Suchalgorithmen werden von einer leistungsfähigen Multi–CPU–Workstation durchgeführt,

die Darstellung erfolgt auf einem einfachen Arbeitsplatzrechner mit graphischer Benutzeroberfläche.

- Client und Server benutzen aus historischen Gründen ein zueinander inkompatibles Kommunikationsprotokoll. In diesem Fall kann eine Zwischenschicht dazu eingesetzt werden, die Protokollumwandlung vorzunehmen. Diese Zwischenschicht kann, je nach Bedarf, auf dem Client-, auf dem Serverrecher oder auf einem dritten Rechner im Netz laufen.

3.10.3 Verwaltungsstrukturen im Internet

Das Internet wurde als dezentrales Netz konzipiert, das auch dann noch funktionsfähig sein soll, wenn wesentliche Teile davon zerstört oder ausgefallen sind. Voraussetzung dafür ist, dass die einzelnen Bestandteile autonom arbeiten und keiner zentralen Steuerung bedürfen. Dennoch gibt es ein Minimum an Zentralismus, das für den Betrieb notwendig ist:

IP–Nummern: Im Internet hat jeder Rechner eine eindeutige IP–Nummer, mit der er von allen Rechnern aus angesprochen werden kann. Die Vergabe der IP–Nummern ist allerdings nicht zentral, sondern hierarchisch geregelt: Beispielsweise ist das RIPE in Amsterdam für die Vergabe der IP–Nummern in Europa zuständig. Es besitzt einen bestimmten Pool von Netzen und kann daraus Netze an andere Organisationen abtreten. Einen Teil der RIPE–Netze wurde an SWITCH abgegeben, das wiederum die schweizerischen Provider und Universitäten mit Netzen versorgt. Innerhalb eines SWITCH–Kunden, zum Beispiel der Universität Zürich, kümmert sich das Rechenzentrum um die weitere Unterteilung der Netze.

Domainnamen: Da IP–Nummern schwer zu merken sind und nur schlecht die organisatorische Zusammengehörigkeit von Rechnern widerspiegeln, wurde das Domainnamenssystem erfunden (siehe Abschnitt 6.7). Während das Internet–Protokoll (IP) prinzipiell auch ohne Domainnamen funktioniert, erwarten die meisten Protokollimplementierungen auf höherer Ebene, dass die beteiligten Rechner auch Domainnamen haben. Auch die Vergabe der Domainnamen ist hierarchisch geregelt; Einzelheiten können im oben erwähnten Abschnitt nachgelesen werden.

Protokolle der Internet–Protokollfamilie: Die wichtigsten davon sind ebenfalls unter zentraler Verwaltung. Das Internet ist definiert als Rechnerverbund, in dem alle diese Protokolle beherrschen. Aus technischen und anderen Gründen bleibt die Entwicklung der Protokolle nicht stehen, sondern schreitet stetig voran. Dabei muss jedoch darauf geachtet werden, dass alle Rechner sich bei der Semantik der Protokolle einig sind. Andernfalls verstehen sie sich nicht, und es kommt keine sinnvolle Kommunikation zustande.

Um diese zentralen Aufgaben wahrzunehmen, existieren eine Reihe von Organisationen. Das InterNIC ist die zentrale Instanz, die für die Vergabe der IP–Nummern und Top–Level–Domainnamen zuständig ist. Es sitzt in den USA und war ursprünglich ein Ableger des

Verteidigungsministeriums. Das InterNIC verwaltet ausserdem die Domainnamen unterhalb der ”generischen” Top–Level–Domains .net, .org und .com. Das RIPE in Amsterdam hat einen Teil der IP–Netze vom InterNIC übernommen und vergibt sie an europäische Vergabeorganisationen. Eine davon ist SWITCH, die ausserdem die Domainnamen unterhalb der schweizerischen und liechtensteinischen Top–Level–Domains .ch und .li vergibt. Die IETF (Internet Engineering Task Force) ist ein Gremium, das sich um die Weiterentwicklung und Standardisierung der im Internet benutzten Protokolle kümmert. Aus ihr gehen Untergremien hervor, die sich mit speziellen Aspekten befassen, beispielsweise das World Wide Web Consortium (W3C).

3.11 Zukünftige Entwicklungen

Weltweit sind derzeit fast alle Class–A–Netze, ca. 50 Prozent aller Class–B–Netze und 5 Prozent aller Class–C–Netze vergeben. Wenn das Internet weiterhin so stark anwächst wie zur Zeit, dann wird es keine zwei Jahre mehr dauern, bis alle Netznummern vergeben sind. Schon heute werden einzelnen Organisationen mehrere Class–C–Netze zugewiesen, da ein Class–B–Netz zu gross wäre. Jedes Class–C–Netz benötigt einen Eintrag in den Routingtabellen der Router weltweit. Zentrale Router haben derzeit etwa 30.000 Routen gespeichert.

Eine Erweiterung des Adressraumes ist also dringend nötig und zur Zeit auch in Arbeit: In IP Next Generation (IPng), der Version 6 des Internet–Protokolles [DH96], werden IP–Nummern 128 Bit lang sein. Das bedeutet, dass man jedem Quadratmeter der Erde über 1023 IP–Nummern zuweisen könnte. Es ist sehr wahrscheinlich, dass diese Länge auf absehbare Zeit ausreichend sein wird [Tan96, S. 442f]. Mit dem grossen Adressraum wird es möglich, Netze streng hierarchisch nach organisatorischen und geographischen Gesichtspunkten zu vergeben. Beispielsweise werden alle IP–Nummern in den USA in den höchstwertigsten 8 Bit übereinstimmen. Ein Router in Europa erkennt also eine solche Adresse sofort und braucht für den Transport über den Atlantik nur noch eine oder wenige Routen zu kennen.

Neben der Erweiterung des Adressraumes bietet IPng noch andere Neuerungen, beispielsweise kryptographische Methoden zur Gewährleistung von Authentizität und Vertraulichkeit, Massnahmen für schnelleres Routing und eine verbesserte Behandlung überlanger IP–Datagramme.

IPng befindet sich noch in der Entwicklungsphase. Ein produktiver Einsatz bei einem Unternehmen ist derzeit noch nicht sinnvoll. Das SINUS–Projekt wird sich in der Verlängerungsphase mit den Sicherheitsaspekten befassen, die bei einer Migration von IPv4 nach IPv6 relevant sind. In diesem Bericht soll darauf nicht weiter eingegangen werden.

Kapitel 4

Einführung in die Sicherheitskonzeption

Die im SINUS–Projekt entwickelte Sicherheitskonzeption stell eine strukturierte Methode dar, mit der Online–Dienste im Internet sicher gestaltet werden können [KWT97, KW97]. Das Kapitel führt in die Sicherheitskonzeption, als grundlegende Methode zur sicheren Gestaltung von Online–Diensten im Internet, ein. Das hier vorgestellte Verfahren ist die Grundlage zum Verständnis der im Teil II und Teil III näher erläuterten Schritte der Sicherheitskonzeption.

4.1 Einführung

Dieses Kapitel führt in die Sicherheitskonzeption als wesentlicher Bestandteil des Sicherheitsmanagements ein (vgl. Kapitel 2). Die Anwendung der Methode erlaubt eine inkrementelle Steigerung sowohl der angebotenen bzw. verwendeten Dienste als auch deren sichere Nutzung. Sie eignet sich daher insbesondere für kleinere und mittlere Unternehmen (KMU), die mit einer vollständigen Durchführung von Sicherheitsmassnahmen häufig überfordert sind; sie können sich so mit der Thematik vertraut machen und sukzessive in das Sicherheitsmanagement einsteigen.

Viele Sicherheitskonzeptionen gehen von der nachträglichen Absicherung von Diensten bzw. Informationssystemen aus, d.h. Sicherheit wird als „Add On“ zur primären Systemfunktion betrachtet (diese Ansätze finden sich z.B. in [BFI95, BSI96]). Durch diese organisatorische Trennung von Anforderungsanalyse und Bedrohungsanalyse sind sicherheitsrelevante Funktionen häufig unzureichend in Informationssysteme integriert. Ein typisches Beispiel ist das Zusammenspiel von E–Mail Systemen mit Verschlüsselungs- und Signatursystemen. Die Folge ist, dass Sicherheitsaspekte aufgrund mangelnder Integration in die Systeme vernachlässigt werden.

Die hier vorgeschlagene Konzeptionierung versucht diese Trennung durch eine Integration von Anforderungs- und Risikoanalyse zu einem gemeinsamen Gestaltungsprozess zu

vermeiden. Dieses Verfahren besitzt drei wesentliche Vorteile gegenüber einem getrennten Vorgehen:

Sicherheit von Anfang an: Gerade bei der Nutzung von Internet–Diensten, ist die nachträgliche Absicherung des Systems nicht zu verantworten. Die Öffnung des Unternehmens muss von Anfang an abgesichert werden.

Schnelle Bereitstellung von Diensten: Im Umfeld sich schnell entwickelnder Systeme, zu denen Internet zweifellos gehört, ist die Durchführung einer abgetrennten Sicherheitskonzeption sehr zeitaufwendig. Dies ist der Hauptgrund, weshalb das Sicherheitsmanagement vernachlässigt wird.

Integration von Sicherheitsmassnahmen: Sicherheitsmassnahmen werden häufig aufgrund des Zusatzaufwandes bei der Arbeit umgangen (z.B. verschicken verschlüsselter E–Mails); eine Integration könnte Sicherheitsmassnahmen benutzerfreundlicher gestalten.

Das vorgeschlagene Verfahren besitzt eine grosse Ähnlichkeit zum Rapid Product Development Ansatz im Software–Engineering (vgl. [Bla94, HI88]). In Anlehnung an diesen Begriff bezeichnen wir diese Sicherheitskonzeption auch als Rapid Secure Development (RSD), d.h. als schnelle und sichere Implementierung von Informationssystemen. Implementierung bezieht sich dabei mehr auf die organisatorische Einführung von Informationssystemen (in unserem Szenario z.B. Internet–Dienste) und weniger auf einen Softwareentwurfsprozess.

4.2 Vorgehen – Rapid Secure Development

Bei der Anbindung eines Unternehmens an das Internet oder auch bei der Neugestaltung der Internetnutzung zu Unternehmenszwecken empfiehlt sich ein Vorgehen, das sowohl Nutzungspotentiale als auch Bedrohungspotentiale gleichermassen zu einer sicheren Gestaltung von Internet–Diensten verschmelzen lässt.

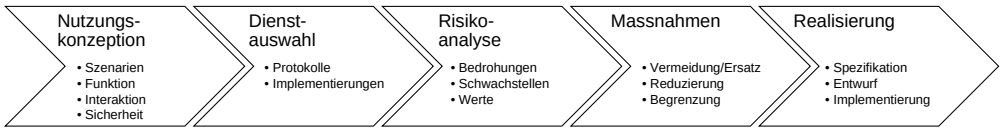

Abbildung 4.1: Sicherheitskonzeption

Im RSD–Ansatz lassen sich fünf Prozessschritte unterscheiden; die Aufgaben und Probleme, die in den jeweiligen Schritten zu lösen sind, lassen sich durch fünf Fragen beschreiben (vgl. Abbildung 4.1):

Nutzungskonzeption (Szenarien): Welcher Nutzen soll aus dem Internet gezogen werden? Hier müssen Szenarien durchgespielt werden, die zur Bestimmung funktionaler Anforderungen und Sicherheitsanforderungen führen.

Dienstauswahl (Dienste, Protokolle): Welche Dienste sind für diese Anforderungen geeignet? Bei der Dienstauswahl ist die Berücksichtigung von Dienstprotokollen und deren Implementierungen entscheidend.

Risikoanalyse (Bedrohungen, Schwachstellen): Welche Risiken entstehen bei der Verwendung der Dienste? Die Risiken ergeben sich aus dem Zusammentreffen von Bedrohungen und Schwachstellen unter Berücksichtigung des Nutzungsszenarios, welches den Wert des Dienstes und der im Dienst verarbeiteten Informationen bestimmt.

Massnahmen: Welche Massnahmen schützen vor diesen Risiken? Massnahmen lassen sich im wesentlichen in die Kategorien Vermeidung, Reduzierung und Begrenzung aufteilen.

Realisierung (Ersatz, sichere Ausgestaltung): Wie können diese Dienste und Massnahmen realisiert werden? Nach der konzeptuellen Bestimmung des Nutzens mit den damit verbundenen Risiken und den Gegenmassnahmen muss das System nun sicher gestaltet werden.

Die in Abbildung 4.1 dargestellten Schritte ähneln in weiten Teilen einem Softwareentwurfsprozess (z.B. bei [LL95, S. 270]); allerdings werden bei der sicheren Realisierung von Internet–Diensten i.d.R. keine neuen Systeme implementiert sondern vielmehr bestehende Systeme und Massnahmen in geeigneter Weise zusammengebracht. Die hier vorgestellte Sicherheitskonzeption wird im Weiteren die Schwerpunkte der Nutzung und der Sicherheit bei der Nutzung betrachten.

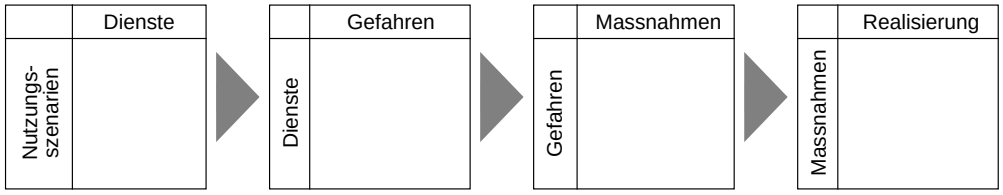

Abbildung 4.2: Beziehungstabellen von den Szenarien zu den Massnahmen

Die Generierung von Beziehungen zwischen den einzelnen Prozessschritten wird mittels folgender Tabellen unterstützt (vgl. Abbildung 4.2):

Szenarien – Dienste bzw. Protokolle: Zuordnung der Nutzungskonzepte zu Internet–Diensten. Welche Dienste sind für meinen Nutzungswunsch geeignet?

Dienste – Gefahren: Zuordnung der Risiken zu den Diensten unter Berücksichtigung der Dienstnutzung. Welche Gefahren können sich aus der Dienstnutzung in Zusammenhang mit meinem speziellen Nutzungswunsch ergeben?

Gefahren – Massnahmen: Zuordnung von Diensten und den mit diesen verbundenen Gefahren zu Massnahmen, die getroffen werden müssen. Welche Massnahmen kann ich gegen diese Gefahr einsetzen?

Massnahmen – Realisierung: Zuordnung der konzeptuellen Massnahmen zu geeigneten organisatorischen oder technischen Massnahmen. Für die technischen Massnahmen werden an dieser Stelle geeignete Protokolle oder Produkte angegeben. Für organisatorische Massnahmen werden Handlungsempfehlungen und Leitfäden gegeben. Wie kann ich diese Massnahmen umsetzen?

Anhand eines einfachen Beispiels könnte die Sicherheitskonzeption für einen Durchlauf folgendermassen aussehen: Ausgehend von dem Wunsch, sich Informationen über das Internet zu beschaffen (Nutzungsszenario), kann ein geeigneter Dienst zur Erfüllung dieses Wunsches ausgewählt werden, z.B. Nutzung des HTTP–Protokolls als Basis des World Wide Web Dienstes. Durch diese Nutzung können Gefahren entstehen, z.B. das Einschleusen und Ausführen unerwünschter Programme (z.B. durch das Herunterladen von ActiveX Applets). Gegen diese Gefahr lassen sich Filtermechanismen auf Anwendungsebene einsetzen. Eine Implementierung dieser Massnahme muss aus der Auswahl eines Application Gateways (Firewall auf Anwendungsebene) bestehen, der in der Lage ist, ActiveX Applets aus dem HTTP–Datenstrom herauszufiltern.

4.3 Kontinuierliche Verbesserung

Verbesserungen bei der Durchführung der Sicherheitskonzeption treten in zwei Formen auf:

Überprüfung: Der aktuelle Durchlauf der Sicherheitskonzeption bedarf einer Rückkopplung im Sinne einer Verifikations- und Validierungsphase. Haben wir das Gewünschte erreicht?

Weiterführung: Nach Beendigung der Realisierung eines Nutzungsszenarios stellt sich die Frage, welche Szenarien jetzt in Angriff genommen werden sollen. Welches sind unsere nächsten Schritte? Hat sich unser Bedarf geändert?

Beide Schritte haben das Ziel, das System kontinuierlich zu verbessern. Während dieser kontinuierlichen Verbesserung wächst dabei das Wissen um weitere Nutzungspotentiale und deren sichere Umsetzung, so dass der Wissenszuwachs gemeinsam mit der Nutzung wächst.

4.3.1 Überprüfung – Verifikation und Validierung

Der in 4.1 dargestellte Entwurfsprozess sieht noch keinerlei Rückkopplung vor. Diese treten allerdings in der Realität zu jeder Zeit auf. Die Sicherheitskonzeption in Abbildung 4.3 trägt diesem Umstand Rechnung.

Neben dem normalen Top–Down Vorgang entstehen gerade in der organisatorischen Gestaltung dieses für viele Unternehmungen recht jungen Betätigungsfeldes viele Rückkopplungen. Auch wenn Rückkopplungen in möglichst frühen Phasen vorgenommen

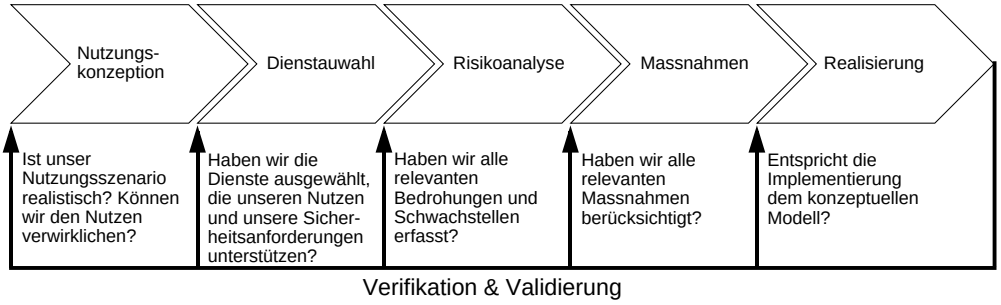

Abbildung 4.3: Rückkopplungen in der Sicherheitskonzeption

werden sollten, ist es nicht auszuschliessen, dass erst bei der Realisierung grundlegende Fehler oder Probleme in der Nutzungskonzeption oder der Risikobetrachtung gefunden werden können.

Die Rückkopplungen entsprechen Verifikations- und Validierungsphasen des aktuellen Prozesses. Die in Abbildung 4.3 angegebenen Fragestellungen der einzelnen Rückkopplungsschritte geben einen Überblick über die unterschiedlichen Ausprägung der Verifikation bzw. Validierung.

4.3.2 Weiterführung

Nach der erfolgreichen Durchführung der Sicherheitskonzeption, d.h. nach der erfolgreichen sicheren Bereitstellung einer Anwendung im Internet, stellt sich die Frage, wie es weitergehen soll. Meist erwachsen bereits während der Realisierung einer Anwendung Ideen für weitere Anwendungen, oder es ergibt sich die Notwendigkeit, weitere Anwendungen bereitzustellen. Es muss daher nach der Durchführung eines Schrittes eine neue Planungsphase erfolgen, in der eine genaue Situations- und Problemanalyse durchgeführt wird [Art93]. Es müssen wieder neue Nutzungsszenarien durchgespielt werden und im Rahmen der Sicherheitskonzeption umgesetzt werden. Die neuen Anforderungen müssen zudem in bisherige Lösungen integriert werden, und es ist bei den installierten Diensten zu überprüfen, ob sich durch neue Entwicklungen neue Sicherheitsprobleme ergeben haben.

Ändert man die Darstellung aus Abbildung 4.1 so um, dass die Prozessschritte in einem Kreis angeordnet werden, dann ergibt sich für die Weiterführung der Sicherheitskonzeption eine Spiralstruktur, wie sie in Abbildung 4.4 dargestellt ist.

Das resultierende Vorgehensmodell (Abbildung 4.4) ist aus dem Total Quality Management (TQM) Ansatz abgeleitet (vgl. [Art93, Dev95, OA90]). Ausgehend von der Nutzungskonzeption, die eine Anforderungsanalyse bezüglich funktionaler Anforderungen und Sicherheitsanforderungen enthält (Planungs- und Entscheidungsphase), wird ein Internet–Dienst oder ein beliebiges Informationssystem ausgewählt. Es folgt eine Risikoanalyse und eine konzeptuelle Massnahmenplanung, die als Basis für die eigentliche Realisierung dient (Ausführen). Das Ergebnis muss (spätestens) anschliessend verifiziert und validiert werden (Prüfen). Anschliessend müssen neue Ziele durch eine Situations- und Problemanalysen vorbereitet werden (Agieren). Diese Phase kann zu einer Entscheidung führen, das Angebot an Diensten

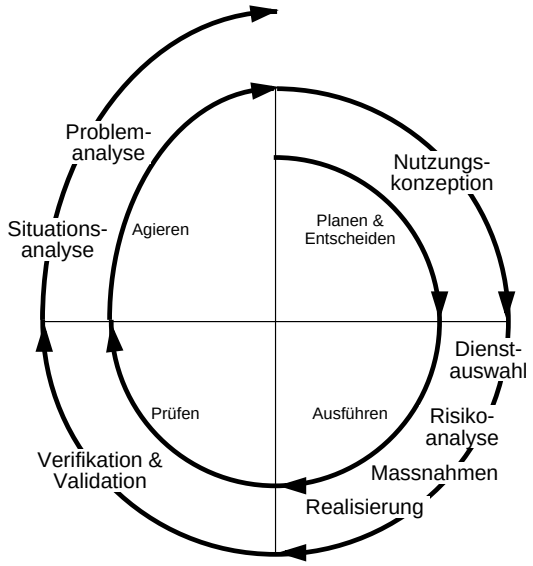

Abbildung 4.4: Weiterführung der Sicherheitskonzeption

im Internet zu erweitern. Diese Dienste müssen dann in einer neuen Sicherheitskonzeptionsphase bereitgestellt werden.

4.3.3 Gesamtsicht auf die kontinuierliche Verbesserung

Fasst man die Aspekte der Weiterentwicklung der sicheren Nutzung von Internet–Diensten und die Verbesserung bzw. Anpassung bestehender Internet–Dienste zusammen, dann ergibt sich eine Prozesskette gemäss Abbildung 4.5. Zur Veranschaulichung des Vorgehens wurde die Darstellung des TQM–Ansatzes gewählt, der allerdings an die speziellen Aufgaben der Sicherheitskonzeption angepasst wurde. Die wesentliche Änderung ergibt sich in der Phase des Agierens. Hier muss entschieden werden, inwieweit ein neues Nutzungsszenario in Angriff genommen werden soll und inwieweit der bestehende Internet–Dienst verbessert werden soll.

Um die kontinuierliche Verbesserung eines bestehenden Internet–Dienstes gewährleisten zu können, müssen Sicherheitsfaktoren messbar gemacht werden (Wurden Angriffe durchgeführt? Wieviele? Hatten Angriffe Erfolg?). Ebenso müssen neue Entwicklungen in der Anpassung von Internet–Diensten berücksichtigt werden. Diese treten i.d.R. entweder durch neue Dienste, neue Bedrohungen, Entdeckung neuer Schwachstellen oder durch eine Veränderung der Systemumgebung durch neue Nutzungsszenarien auf.

Bei der Vorbereitung neuer Nutzungsszenarien wird in der Handlungsphase der bestehende Kreislauf verlassen und ein neuer Zyklus begonnen. Grundsätzlich ist es in der Handlungsphase möglich, sowohl bestehende Internet–Dienste zu verbessern als auch neue Nutzungsszenarien zu erforschen. Allerdings sollte der Arbeitsaufwand den Informatikressourcen entsprechen.

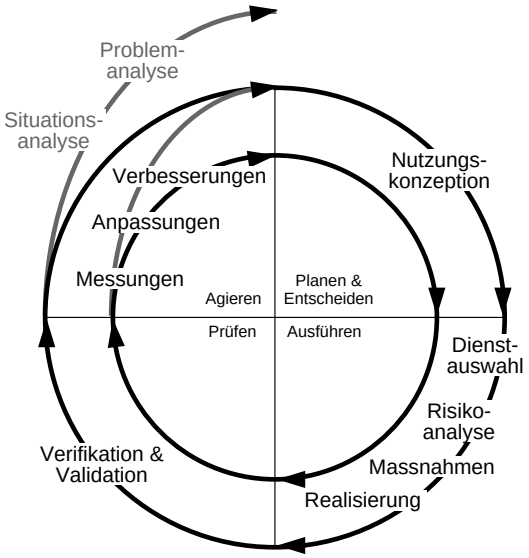

Abbildung 4.5: TQM–Ansatz zur kontinuierlichen Verbesserung

4.3.4 Durchführung und Verfeinerung der Sicherheitskonzeption

In den folgenden Teilen II und III wird die Sicherheitskonzeption nach dem RSD–Ansatz weiter verfeinert und näher erläutert. Die eingeführten Beziehungstabellen werden mit Inhalt gefüllt werden und bieten ein Framework zur Durchführung der Sicherheitskonzeption. Abbildung 4.6 zeigt eine Übersicht der Kapitelaufteilung in Zusammenhang mit den einzelnen Prozessschritten der Sicherheitskonzeption.

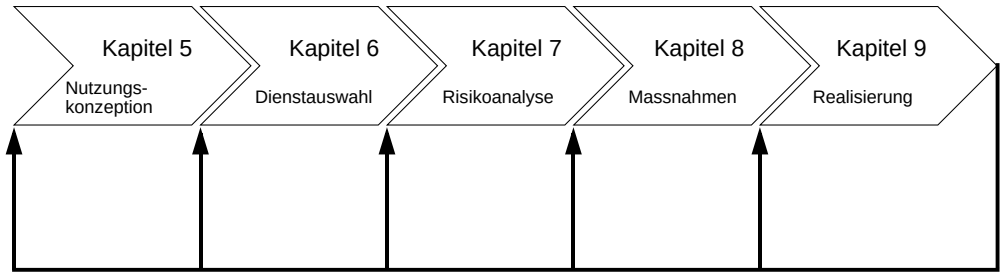

Abbildung 4.6: Übersicht über die Kapitel zur Sicherheitskonzeption

Kapitel 5 wird in die Nutzungskonzeption einführen. Welche Szenarien sind vorstellbar? Bei der Identifizierung relevanter Nutzungspotentiale zur Erfüllung funktionaler Anforderungen und Sicherheitsanforderungen soll Unterstützung geboten werden.

Kapitel 6 führt in die relevanten Internet–Dienste zur Erfüllung der Nutzungsszenarien ein. Am Ende des Kapitels wird eine ausgefüllte Zuordnungstabelle von Nutzungsszenarien zu Internet–Diensten erstellt.

Kapitel 7 hat die Risikoanalyse der ausgewählten Internet–Dienste zum Inhalt. Ziel ist eine Zuordnung von Internet–Diensten zu Bedrohungen. Zudem sollen Hilfestellungen gegeben werden, wie aus der Bestimmung der Bedrohungen unter Berücksichtigung von Schwachstellen und Nutzungsszenario Risiken bestimmt werden können.

Kapitel 8 betrachtet die Massnahmen, die getroffen werden können, um sich gegen die identifizierten Bedrohungen zu schützen. Die Massnahmen bieten an dieser Stelle noch keine konkreten Produkte oder Handlungsanleitungen, sie sind vielmehr konzeptueller Natur. Als Ergebnis entsteht eine Tabelle mit der Zuordnung von Risiken zu Massnahmen.

Kapitel 9 konkretisiert diese Massnahmen durch eine erste Zuordnung zu Produkten bzw. Handlungsanleitungen. Ein wesentlicher Bestandteil des Kapitels ist eine Zuordnungstabelle von konzeptuellen Massnahmen zu Massnahmenrealisierungen.

Im dritten Teil des Berichts wird auf die Realisierung der Massnahmen näher eingegangen. Die in Kapitel 8 aufgestellte Matrix von konzeptuellen Massnahmen zur Realisierung der Massnahmen bildet die Grundlage für eine Zuordnung konkreter Massnahmen zu konzeptuellen Massnahmen.

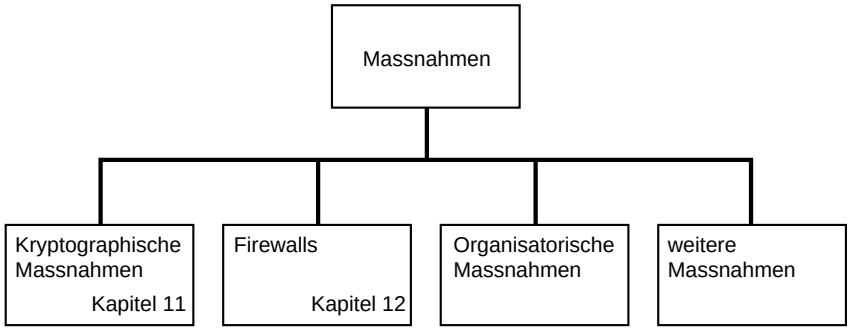

Abbildung 4.7: Aufteilung der konzeptuellen Massnahmen

Die Realisierung der Massnahmen ist gemäss Abbildung 4.7 aufgeteilt in kryptographische Massnahmen (Kapitel 11), Firewalls (Kapitel 12) sowie organisatorische Massnahmen und sonstige (z.B. bauliche) Massnahmen. Im Rahmen dieses Berichtes wird auf die kryptographische Massnahmen und Firewalls näher eingegangen. In Abschnitt 12.5 wird ausserdem der SINUS–Firewall vorgestellt, der im Rahmen des Projektes entstanden ist.

4.4 Einbettung in das Sicherheitsmanagement

Die Sicherheitskonzeption von Internet–Diensten ist ein wesentlicher Teil des Sicherheitsmanagements im Unternehmen. Dabei beschränkt sich die Sicherheitskonzeption auf die Aspekte der sicheren Nutzung von Netztechnologien im Internet.

Wie aus Kapitel 2 angesprochen, bestehen die Kernprozesse des Sicherheitsmanagements nach ISO 13335 aus der Risikoanalyse mit der Bestimmung von Werten, Bedrohungen und Schwachstellen. Je nach geschätzter Höhe des Wertes wird entweder eine detailliertere Betrachtung vorgenommen oder es werden direkt Grundschutzmassnahmen durchgeführt. Es folgt die Identifikation von Massnahmen und deren Implementierung. Anschliessend erfolgt die Weiterführung im Sicherheitsmanagementzyklus.

Die Sicherheitskonzeption füllt genau diese Kernprozesse inhaltlich ideal aus. Vorgaben aus der Sicherheitspolitik und der Geschäftspolitik finden sich in den Phase der Nutzungskonzeption und der Dienstauswahl wieder und werden dort in Bezug auf die Netznutzung

verfeinert. Nach ISO folgt danach eine Risikoanalyse, die je nach Wert der Information unterschiedlich detailliert ausfällt. Dies wird durch die Risikoanalysephase abgedeckt. Die Risikoanalyse der Sicherheitskonzeption unterscheidet allerdings nur implizit nach dem Wert der Informationen. Die Sicherheitskonzeption wie oben beschrieben basiert auf zwei Annahmen:

- Der Wert der verarbeiteten Informationen steigt mit der Kenntnis der Nutzung des Internet.
- Der Wert der Information wird bereits in Zusammenhang mit der Nutzungskonzeption ermittelt; denn hier fliessen Geschäftsanforderungen unmittelbar ein.

Durch diese beiden Annahmen, die sich in unseren Testumgebungen als richtig erwiesen haben, wird die Risikoanalyse der Sicherheitskonzeption automatisch unterschiedlich intensiv durchgeführt. Im Fall des Grundschutzes, können ggf. einfach die Tabellen umgesetzt werden. Im detaillierten Ansatz muss jeder einzelne Punkt noch genauer hinterfragt werden.

Anschliessend werden die Phasen der Massnahmenidentifikation und -realisierung durchgeführt. Es folgt der Weiterführungsprozess, der in diesem Kapitel bereits detailliert beschrieben wurde.

4.5 Zusammenfassung

Die sichere Nutzung von Internet–Diensten muss einerseits sorgfältig geplant werden, andererseits ist das vollständige Abarbeiten von Anforderungsspezifikation und anschliessender Initialisierung eines Sicherheitsmanagementprozesses in der Praxis gerade für kleinere Unternehmen zu komplex. Insbesondere dann, wenn die Nutzungspotentiale und damit die Nutzungsszenarien noch nicht definiert sind, ergeben sich zu häufig Änderungen, so dass es nicht sinnvoll ist, jedesmal eine komplette Risikoanalyse durchzuführen. Durch die integrative Betrachtung von Anforderungen und Risiken können jedoch sowohl die Funktionalität als auch die Sicherheit angemessen berücksichtigt werden. Dies schliesst eine nachträgliche Überarbeitung und Harmonisierung mit einem unternehmensweiten Sicherheitsansatz nicht aus, sondern bereitet sie vielmehr besser vor.

Die hier vorgestellte Sicherheitskonzeption ist ein Lösungsansatz für dieses Problem. Die aufgezeigten Prozessschritte werden im Folgenden weiter verfeinert, so dass Internet–Dienste nach diesem Muster sicher genutzt werden können. Die Konzeption eignet sich grundsätzlich auch für weitere, in dieser Arbeit nicht erwähnte Dienste; es müssen dann allerdings die Tabellen und konkrete Inhalte auf ihre Gültigkeit hin überprüft und ggf. angepasst werden.

Das Modell, das zu diesem RSD–Ansatz führt, ist aus einer Betrachtung von Sicherheitsmanagement–Ansätzen im SINUS Projekt entstanden. Es erwies sich als ein praktikabler Weg, der in der Praxis häufig beschritten wird. Durch eine Strukturierung dieser Betrachtung entstand das Modell.

Die RSD–Sicherheitskonzeption erhebt nicht den Anspruch, eine umfassende Lösung für alle Sicherheitsprobleme im Zusammenhang mit dem Internet zu sein! Sie ermöglicht kleinen und mittleren Unternehmen ohne eigenem, auf Internet–Sicherheit spezialisiertem Mitarbeiter, die Schritte von der Dienstauswahl bis hin zur Massnahmenidentifikation und -realisierung selbständig durchzuführen. Für die Überprüfung der getroffenen Massnahmen ist es jedoch dringend empfehlenswert, einen (externen) Spezialisten hinzuzuziehen.

Teil II

Die RSD–Sicherheitskonzeption

Kapitel 5

Nutzungspotentiale des Internet

Dieses Kapitel beinhaltet eine Analyse der Nutzungspotentiale des Internet für unternehmerische Zwecke. Es lassen sich zwei wesentliche Einflussgrössen – der Interaktionsgrad und die Zielgruppe – identifizieren. Diese Aspekte können zu zwei Dimensionen zusammengefasst werden, die ein Framework zur Klassifizierung von Internet–Diensten bilden. Es ergeben sich vier grundlegende Nutzungsklassen im Internet: Informationsnachfrage, Informationsangebot, elektronischer Handel und Gruppenunterstützung. Dieses Framework unterstützt zum einen die Umsetzung unternehmerischer Ziele und Strategien im Internet, zum anderen ermöglicht es als morphologischer Kasten eine kreative Ausgestaltung und Erweiterung des Angebots im Internet.

5.1 Einführung

Bei einer ersten Präsentation der Sicherheitskonzeption auf der Tagung VIS'97 [KTW97] ergab sich folgender Einwand eines Teilnehmers: „Wozu brauche ich überhaupt die Betrachtung von Nutzungsszenarien und Risiken, wenn ich doch heute ein Komplettpaket inkl. Sicherheitseinrichtungen kaufen kann?“ In der Tat: wozu eine Nutzungsbetrachtung, wenn es doch „klar“ ist, dass das Unternehmen sich an das Internet anschliessen will?

Diese Einstellung zur Beschaffung von Informationssystemen ist zwar weit verbreitet, birgt allerdings den grossen Nachteil, dass nicht die Systeme beschafft und betrieben werden, die den Unternehmenszielen dienen, vielmehr bestimmen oder begrenzen die beschafften Systeme die Unternehmensziele. Anders ausgedrückt: um Unternehmensziele und Geschäftsprozesse effizient zu unterstützen, ist vor der eigentlichen Beschaffung der Systeme eine genaue Analyse der Unternehmenssituation, der Anforderungen und des Bedarfs erforderlich (vgl [Sch94, S. 38ff]).

Um diesen Bedarf bestimmen zu können, soll in diesem Kapitel ausgehend von Unternehmenszielen der mögliche Nutzen des Internet bzw. der Anwendungsdienste im Internet analysiert werden. Hierzu müssen zunächst die Rahmenbedingungen und die Motivation betrachtet werden, die ein Unternehmen dazu veranlassen kann, sich mit dem Thema

Nutzungspotentiale des Internet zu beschäftigen. Darauf aufbauend können spezielle Nutzungsszenarios für das Internet abgeleitet werden. Es erfolgt eine Klassifikation der Informationssysteme im Internet anhand der zwei Dimensionen Interaktionsbeziehung und Zielgruppe.

5.2 Die Dimensionen – Interaktion und Zielgruppe

Die erste Dimension ergibt sich, wie in Kapitel 4 bereits angedeutet, aus dem Grad der Interaktion. Mit Grad ist bei der Nutzung von Internet–Diensten vor allem die Intensität der Zusammenarbeit gemeint. Diese Intensität kann am Besten durch die Begriffe Kooperation, Koordination und Kommunikation ausgedrückt werden. Bei einer reinen Kommunikationsbeziehung ist der Grad der Interaktion am geringsten, während bei der Kooperation ein hoher Interaktionsgrad notwendig ist.

Die zweite Dimension ergibt sich wie in Kapitel 4 angedeutet aus der Zielgruppe eines Unternehmens. Die Zielgruppe hat einen direkten Einfluss auf die Intensität der Interaktion. Besteht nur der Wunsch des Informationsaustauschs, soll ein Ziel erreicht werden oder besteht gar Konsens in der Zielfindung? Dies führt auf den Begriff der Gruppe; unter Berücksichtigung von [DL87, S. 143ff], [Jen97, S. 444] und [TSMB95, S. 52ff] lassen sich folgende unterschiedliche Intensitäten der Interaktionsbeziehung festhalten:

Gruppe: die Teilnehmer bilden eine freie Gruppe, die miteinander interagiert und sich gegenseitig beeinflusst.

Arbeitsgruppe: zusätzlich zur normalen Gruppe arbeitet eine Arbeitsgruppe an einer gemeinsamen Aufgabe und besitzt somit ein konkretes gemeinsames Ziel.

Team: zusätzlich zu einem gemeinsamen Ziel existiert ein Konsens in der Zielerreichung und ein gemeinsames Vorgehen bei der Zielerreichung. Ein Team besitzt dadurch eine sehr spezielle Organisationsform, die sie von einer Arbeitsgruppe unterscheidet.

Die Intensität der Interaktionsbeziehung beeinflusst in Folge massgeblich die Intensität der Interaktionsbeziehung. Abbildung 5.1 stellt den Zusammenhang zwischen Gruppenausprägung und Interaktionsbeziehung dar (in Anlehnung an [NBM+97]). Die in Abbildung 5.1 dunkel schraffierten Flächen sind als Übereinstimmung zu lesen. Eine lose Gruppe benötigt demnach die Möglichkeit zum Austausch von Informationen untereinander (Kommunikation). Die Arbeitsgruppe hat aufgrund ihres gemeinsamen Ziels darüber hinaus Koordinationsbedarf zur Erreichung des gemeinsamen Ziels. Ein Team besitzt nicht nur ein gemeinsames Ziel, es erarbeitet diese Ziele und Aufgaben darüber hinaus gemeinsam, es entsteht die Notwendigkeit zur Kooperation.

Ergänzt manAbbildung 5.1 um individualistische Aspekte und die Aufteilung in unternehmensübergreifende und unternehmensinterne Gruppen, kann man darüber hinaus noch besondere organisatorische Anforderungen bei der Internetnutzung ausdrücken. Eine allgemeine Gruppe kann somit über das Internet interagieren, während eine Arbeitsgruppe

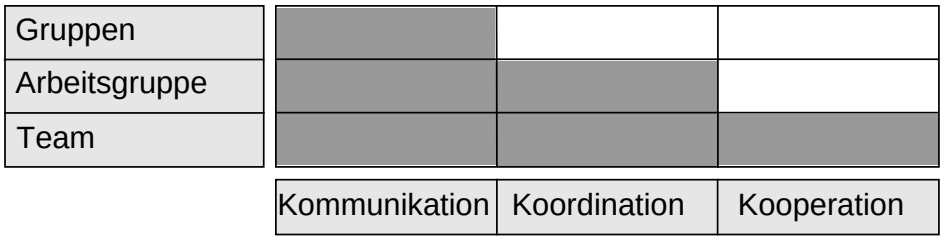

Abbildung 5.1: Zusammenhang zwischen Interaktionsbeziehung und Gruppenausprägung

je nach organisatorischer Verankerung zur Interaktion über ein Intranet oder ein Extranet neigen wird (vgl. Kapitel 8).

Der Übersicht halber wird im folgenden Framework nicht mehr explizit zwischen Team und Arbeitsgruppe unterschieden.

5.3 Ein Framework zur Klassifikation von Nutzungsszenarien

Setzt man die in Abschnitt 5.2 betrachteten Dimensionen zusammen, so ergibt sich ein Framework (bzw. ein morphologischer Kasten) zur Klassifikation und Bestimmung von Nutzungsszenarien im Internet (unter Verwendung von [Alt92, S. 125f]).

Abbildung 5.2: Framework zur Klassifikation von Nutzungsszenarien

Die Nutzungspotentiale bzw. die unterschiedlichen Klassen von Informationssystemen lassen sich in dieses Framework einordnen. Anwendungsdienste im Internet können nach ihrer Nutzungsart der Kooperation, Kommunikation und Koordination und zum anderen nach den Gruppen, die diese Systeme zu unterschiedlichen Zwecken einsetzen, strukturiert werden (siehe Abbildung 5.2). Diese Betrachtung nach Nutzungsart und Zweck ist nicht nur zur Identifikation geeigneter Internet–Dienste sinnvoll, sondern hat auch Auswirkungen auf die auftretenden Bedrohungen und in Folge auf die zu treffenden Sicherheitsmassnahmen (siehe Kapitel 7 und 8).

In dieses Framework lassen sich vier Nutzungsklassen der Informationssysteme im Internet identifizieren (vgl. Abbildung 5.3). Im einzelnen sind dies (unter Berücksichtigung von [Bor96, S. 26ff]):

- Informationsnachfrage (information demand)

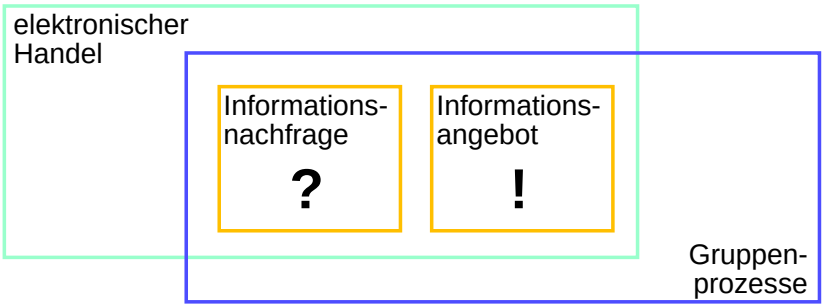

Abbildung 5.3: Nutzungsklassen

- Informationsangebot (information supply)
- Elektronischer Handel (electronic commerce)
- Gruppenunterstützungssysteme (group support systems)

Abbildung 5.3 illustriert zudem noch die Abhängigkeiten der vier Nutzungsklassen. So ist die Informationsnachfrage bzw. das Informationsangebot Basis für elektronischen Handel als auch für die Unterstützung von Gruppenprozessen.

Wie sich später zeigen wird, bedeutet die Identifikation von unterschiedlichen Nutzungsklassen im Allgemeinen nicht, dass diese vier Klassen die Verwendung unterschiedlicher Dienste erzwingen (vgl. [Alt92, S. 126]); es gibt allerdings Internet–Dienste, die die eine oder andere Nutzungsart besser unterstützen als andere. Dies gilt sowohl für funktionale Anforderungen als auch für Sicherheitsanforderungen.

Unter *Informationsnachfrage* versteht man alle Prozesse, die der Informationsversorgung bzw. -beschaffung von Personen dienen. Informationen, die über das Internet beschafft werden können, können je nach Kontext Dokumente, E–Mails aber auch Programme oder Datenbankausschnitte sein. Die Informationsbeschaffung kann durch sogenannte Informations–Broker (d.h. im Auftrag von jemanden) unterstützt werden oder sie erfolgt ausschliesslich über Eigeninitiative.

Mit der Informationsnachfrage eng verknüpft ist die Bereitstellung eigener Informationen, dem *Informationsangebot*. Das Angebot kann von PR- und Marketinginformationen bis hin zu detaillierten Informationen (z.B. White Papers, Institutsberichte, etc.) gehen. Informationen können dynamisch (z.B. über Datenbanksysteme) oder statisch zugänglich gemacht werden. Zudem können Informationen kostenpflichtig, was i.d.R. weitere Prozesse im Bereich des elektronischen Handels nach sich ziehen wird, oder gratis zur Verfügung gestellt werden.

Systeme, die den *elektronischen Handel* unterstützen sollen, müssen in der Lage sein, Kunden–Lieferanten–Beziehungen abzubilden. Sie müssen sich zudem an rechtlichen Vorgaben (z.B. Vertragsrecht, Handelsrecht, etc.) orientieren. Über ein elektronisches Medium können insbesondere immaterielle Werte gehandelt werden (z.B. Finanzdienstleistungen, Informationen, Softwareprodukte, etc.), aber auch materielle Güter können über elektroni-

sche Medien gehandelt werden; die Logistik kann allerdings nicht innerhalb des Systems erfolgen.

Die vierte Nutzungsklasse sind alle Systeme zur *Gruppenunterstützung*. Diese Systeme dienen der Unterstützung und Gestaltung von Prozessen, die bei der Erstellung von Produkten anfallen. Während der Produktion müssen Personen koordiniert werden, es müssen Logistikinformationen verarbeitet werden, die Gruppenarbeit muss unterstützt werden, etc. Produkte sind in dem hier verwendeten Sinn nicht notwendigerweise materielle Güter. Ein Produkt ist im weitesten Sinn das gemeinsame Ziel (Resultat), das eine Verfeinerung eines Unternehmensziels darstellt.

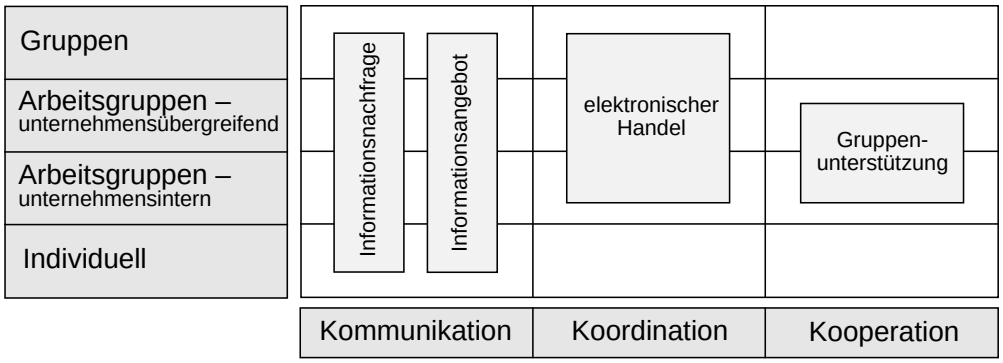

Abbildung 5.4: Einordnung der Nutzungsklassen in das Framework

Wir betrachten nun die Einordnung der identifizierten Nutzungsklassen in das Framework; diese Einordnung ist in Abbildung 5.4 dargestellt. Die Zuordnung, die nachfolgend näher ausgeführt wird, stellt dabei einen Schwerpunkt der Systemverwendung dar. Gemäss Abbildung 2.3 stellt z.B. eine Einordnung in die Kooperation die Möglichkeit der Kommunikation voraus.

Die Nutzungsklassen Informationsangebot und -nachfrage lassen sich im wesentlichen der Kommunikation als Form der Zusammenarbeit zuordnen. Die Nachfrage nach Informationen bzw. das Angebot von Informationen kann individuellen Zwecken, firmeninternen, firmenübergreifenden oder öffentlichen Gruppenzwecken dienen.

Elektronischer Handel baut direkt auf den Informationsprozessen auf, bereichert diese allerdings um vertragsrechtliche und logistische Aspekte. Handel benötigt somit zumindest ein gewisses Mass an Koordination (z.B. Zahlung und Auslieferung der Ware). Elektronischer Handel wird hauptsächlich zwischen Firmen und Endkunden (öffentlicher Personenkreis) betrachtet, kann aber auch firmenintern oder -übergreifend stattfinden (vgl. [Han90, S. 386ff]).

Die Gruppenunterstützung benötigt neben Koordinationsfunktionen insbesondere auch die Möglichkeit zur Unterstützung von Kooperationsprozessen. Eine Zusammenarbeit findet in diesem Kontext nur innerhalb oder zwischen Firmen statt.

5.4 Auswahl eines Nutzungsszenarios

Im Abschnitt 5.3 wurde ein Framework zur Klassifikation von Internet–Diensten entworfen und vier grundlegende Klassen von Internet–Diensten eingeführt. Es stellt sich die Frage, wie nun aufgrund des Frameworks und der Klassen ein Unternehmensziel und dessen strategische Umsetzung unterstützt werden kann (vgl. Abschnitt 2.5)?

In diesem Kapitel wurden bisher die potentiellen Nutzungsmöglichkeiten des Internet und dessen Internet–Dienste aufgezeigt. Diese umfassen den Kooperationsgrad mit den Grundformen der Interaktion und der Intensität der Interaktion und die Zielgruppe der Interaktionsteilnehmer.

Es müssen in einer unternehmensweiten strategischen Zielplanung (vgl. [Hei96, S. 109]) Ziele und Prozesse identifiziert werden, die durch internetbasierte Online–Dienste erfüllt werden sollen. Das Framework kann hier zu einer differenzierten Ausgestaltung der Anforderungen beitragen.

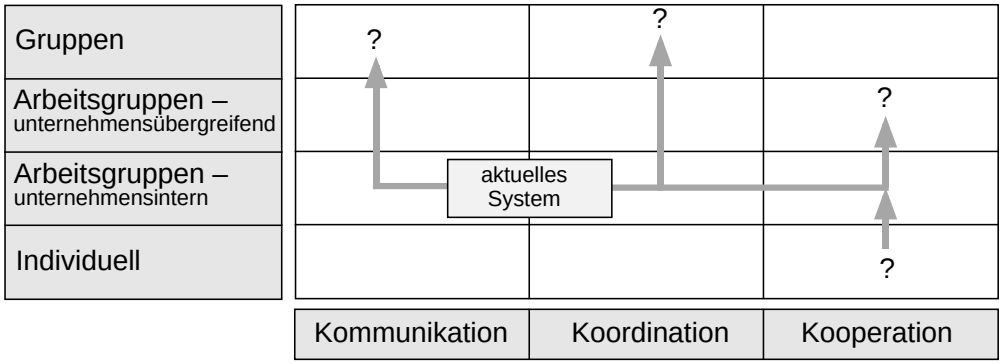

Abbildung 5.5: Möglichkeiten der Erweiterung des Angebots an Internet–Diensten

Auf der anderen Seite kann von einem bestehenden Dienstleistungsangebot im Internet anhand dieses Frameworks Entwicklungsalternativen zur Erweiterung oder Verbesserung des Angebots erkundet werden. In Abbildung 5.5 ist dies am Beispiel eines bestehenden Systems, das die Kommunikation und z.T. auch die Kooperation von firmeninternen Arbeitsgruppen unterstützt, dargestellt. Sollte das System um die Unterstützung firmenübergreifender Teams erweitert werden? Sollte die Kommunikation mit öffentlichen Gruppen ermöglicht werden? Dies sind alles denkbare Entwicklungsszenarien, die entweder auf strategischen Überlegungen beruhen (z.B. Bildung virtueller Unternehmen) oder aber als Anforderungen der täglichen Arbeit entstehen (z.B. innerhalb einer Arbeitsgruppe entstehen Teams, deren Arbeit durch die Unterstützung von Internet–Diensten effizienter gestaltet werden kann).

5.5 Neben den vier Nutzungsklassen

Neben den in Abschnitt 5.3 eingeführten Nutzungsklassen ergeben sich noch eine Reihe weiterer wichtiger Dienste im Internet, die nicht in das Schema der vier Nutzungsklassen

hineinpassen. Diese Dienste sind Voraussetzungen zur Realisierung der vier Grundtypen. Sie bergen somit keinen direkten Nutzen sind, aber aus administrativen Gründen notwendig. Es ergeben sich zwei grundlegende Varianten:

- Administrationsdienste, die zum Betrieb unbedingt notwendig sind (z.B. das eigentliche Internet–Protokoll), und
- Administrationsdienste, die den Betrieb der Anwendungsdienste erleichtern (z.B. die Übersetzung der Rechnernummern des Internet–Protokolls in sprechende Namen).

Es handelt sich in beiden Fällen um Unterstützungsdienste zur Erfüllung der Nutzungsszenarien. Im weiteren werden diese Dienste daher als Administrationsdienste aufgeführt.

Eine Betrachtung der Administrationsdienste ist zur Auswahl einer Nutzungsmöglichkeit nicht erforderlich, muss allerdings bei der konkreten Auswahl eines Dienstes (siehe Kapitel 6) aus Installations- und Sicherheitsgründen berücksichtigt werden.

5.6 Zusammenfassung

In diesem Kapitel wurden die Nutzungspotentiale von Online–Diensten im Internet betrachtet. Ausgehend von den zwei Dimensionen Kooperationsgrad und Zielgruppe lässt sich ein Framework zur Klassifizierung von Diensten im Internet erstellen. Das Internet unterstützt primär vier Nutzungsarten oder -klassen: die Informationsnachfrage, das Informationsangebot, den elektronischen Handel und die Gruppenarbeit. Dies kann aus einem individuellen Informationsangebot oder -bedürfnis heraus erfolgen, Gruppen bei ihrer allgemeinen Kommunikation unterstützen oder aber Gruppen in Unternehmen oder gar unternehmensübergreifend bei ihrer Arbeit unterstützen. Ein wesentlicher Aspekt der unterschiedlichen Gruppen ist die Intensität ihrer Interaktion. Bilden sie nur eine offene, miteinander kommunizierende Gruppe, verfolgen sie als Arbeitsgruppe ein gemeinsames Ziel oder bilden sie gar ein Team?

Diese Aspekte lassen sich, wie in Abbildung 5.6 dargestellt, zu Nutzungsformen des Internet zusammenfassen. Die Darstellung ergibt sich aus Abbildung 5.3, indem auf Basis der Nutzungsklassen die Dimensionen der Nutzungsklassen erfasst werden.

Als nächster Schritt müssen auf diesen Nutzungsszenarien aufbauend die konkreten Dienste im Internet ausgewählt werden, die diesen Nutzen erfüllen können. Hierzu muss zunächst das Leistungspotential der einzelnen Internet–Dienste und der administrativen Dienste im Internet bestimmt werden. Dann kann eine geeignete Zuordnung von dieser Internet–Dienste zu den Nutzungspotentialen erfolgen. Dies wird Gegenstand des nächsten Kapitels sein.

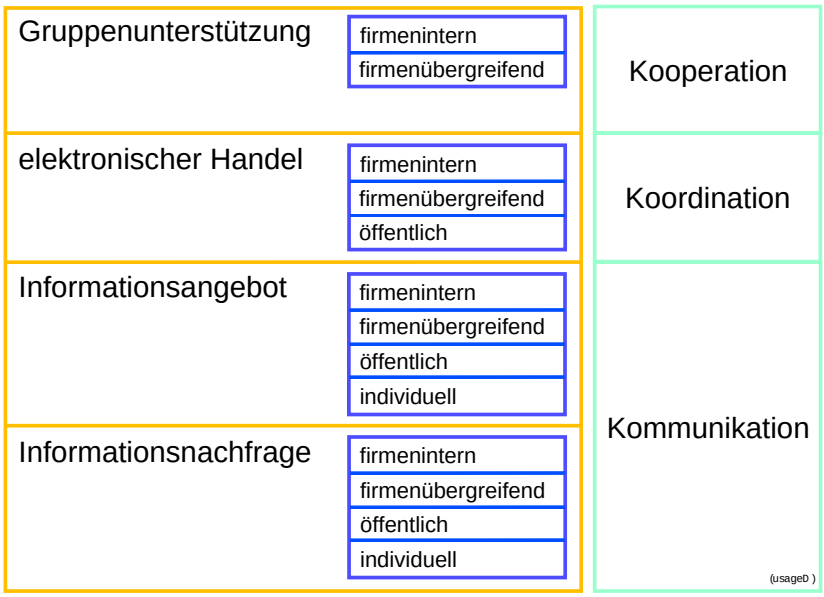

Abbildung 5.6: Nutzungsformen

Kapitel 6

Nutzungsszenarien und Internet–Dienste

6.1 Von Anforderungen zu Internet–Diensten

Den im letzten Kapitel dargestellten Nutzungsszenarien stehen etliche Internetdienste gegenüber, die zu einem grossen Teil die mit den Szenarien verbundenen Anforderungen erfüllen können. Im einzelnen sollen hier die am meisten verbreiteten Internet–Dienste die Dienste Telnet, FTP, E–Mail, WWW, Berkeley r–Tools, SSH, Sendfile, News und IRC betrachtet werden. Andere Internet–Dienste wie z.B. DNS sind eher administrativer Art und nicht direkt für den Einsatz der Prozessunterstützung geeignet. Da DNS jedoch Voraussetzung für die Realisierung der meisten anderen Internet–Dienste ist, wird es hier ebenfalls kurz beschrieben.

Wie aus Abbildung 6.1 zu entnehmen ist, existieren zu den meisten der im letzten Kapitel erarbeiteten Nutzungsklassen mehrere Internet–Dienste, die die definierten Anforderungen erfüllen können. Die nur schwach markierten Punkte deuten auf Spezialfälle hin. Die Dienste sind entweder nur eingeschränkt tauglich, oder sie sind nicht auf allen Systemen einsetzbar. Die Nummern beziehen sich auf die Abschnittsnummern in diesem Kapitel, in denen der jeweilige Dienst näher beschrieben ist.

Bei der Auswahl der hier dargestellten Internet–Dienste haben wir uns auf die wichtigsten beschränkt. Von diesen Diensten sind Implementierungen für alle wichtigen Betriebssysteme verfügbar. Speziell die Server–Teile mancher Dienste (Telnet, FTP) sind jedoch nur in Multitasking- und Multiuser–Betriebssystemen sinnvoll nutzbar. Unter dem Begriff World Wide Web (WWW) wird eine Reihe von weiteren Diensten aufgezählt, die das WWW um weitere wichtige Funktionalitäten erweitern.

Nutzungsszenario		Telnet		FTP		E-Mail		WWW						Berkeley r-Tools		DNS		Sendfile		IRC		Net News	
Internet-Dienst		Client	Server	Client	Server	Client	Server	HTML	Plug-Ins	Javascript	Java	ActiveX	CGI	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server
Informationsbeschaffung	intern	○			●	●		●	○		●	○		○		●		○		○		○	
	extern	○			●	●		●	○		●	○		○		●		○		○		○	
	öffentlich	○			●	●		●			●					●		○				○	
	individuell	○			●	●		●			●			○		●		○					
Informations-bereitstellung	intern		○		●		●	●	○		●	○	●				●		○		○		○
	extern		○		●		●	●	○		●	○	●				●		○		○		○
	öffentlich		○		●		●	●			●		●				●		○		○		○
	individuell		○		●		●	●			●		●				●		○				
elektronischer Handel	intern					●	●	●			●	○	●					○	○				
	extern					●	●	●			●	○	●					○	○				
	öffentlich					●	●	●			●		●					○	○				
Gruppenunterstützung	intern	●	●	●	●	●	●	○	○		●	○	○	○	○			○	○	○	○	○	○
	extern	●	●	●	●	●	●	○	○		●	○	○	○	○			○	○	○	○	○	○

● geeignet
○ nur für spezielle Zwecke, bzw. nicht überall verfügbar

Abbildung 6.1: Nutzungsszenarien und Internet–Dienste

6.2 Telnet

Steckbrief

Zweck: Einloggen auf einem entfernten Rechner über das Netzwerk

Funktionsweise: Client–Server–Prinzip; benutzt TCP Port telnet (23)

Anwendungsszenarien:

- Zugriff von auswärts auf die eigene Rechneranlage (z.B. zum Lesen von E–Mail)
- Fernadministration von Rechnern
- Zugriff auf öffentlich verfügbare Informationsquellen (z.B. Bibliotheksinformationssysteme)
- Zugriff auf Netzwerkspiele im Internet

6.2.1 Beschreibung

Telnet (Teletype Network) dient zum Einloggen auf einem Rechner A von einem anderen Rechner B aus. Telnet ist, wie die meisten der hier besprochenen Dienste, eine Client–Server–Applikation: Auf A muss ein Telnet–Server (telnet daemon, telnetd) laufen, der einkommende Verbindungen entgegennimmt, auf B muss dem Benutzer ein Telnet–Client zur Verfügung stehen. Ein telnetd ist nur auf Systemen, die ein Benutzer- und Loginkonzept

haben, sinnvoll, zum Beispiel Unix, VMS oder MVS. Clients gibt es für jedes gängige Betriebssystem.

Telnet arbeitet über TCP. Der Telnet–Server wird auf dem Rechner A über den Port 23 angesprochen. Das bedeutet, wenn auf A eine TCP–Verbindungsanforderung mit dem Zielport 23 ankommt, wird die TCP–Implementierung die einkommenden Daten an den telnetd weiterleiten. In TCP / IP–Terminologie heisst es, der telnetd lauscht auf Port 23. Die Portnummer auf der Absenderseite ist beliebig; der Absender muss nur sicherstellen, dass nicht mehrere Verbindungen mit gleichen Verbindungsparametern gleichzeitig benutzt werden.

Sobald die Verbindung aufgebaut ist, bietet der Server die übliche Loginprozedur über die Telnet–Verbindung an. Sie besteht im Allgemeinen aus einer Eingabeaufforderung für Benutzernamen und Passwort; stimmt beides mit den dem Server bekannten Informationen überein, ist der Benutzer eingeloggt.

6.2.2 Nutzen von Telnet zur Informationsbeschaffung

Eine Reihe von Informations- und Auskunftssystemen, vor allem älterer Bauart, kann über einen Telnet–Client angesprochen werden. Beispielsweise ist das Bibliotheks–Auskunftssystem über das Kommando telnet rzubiz.unizh.ch angesprochen werden (Login rzubiz). Obwohl viele Betreiber solcher Informationssysteme inzwischen dazu übergehen, eine WWW–Bedienung zu ermöglichen, werden Telnet–basierte Auskunftssysteme auf absehbare Zeit weiter existieren.

6.2.3 Nutzen von Telnet zum Informationsangebot

Analog dazu ist es möglich, Auskunftssysteme zu realisieren, die über Telnet zu erreichen sind. Vorteilhaft ist dabei, dass sich solche Systeme relativ einfach und sicher realisieren lassen. Nachteilig ist aber, dass sie mit einer reinen Textdarstellung auskommen müssen. Die Darstellung von Grafiken ist bei einem solchen System nicht möglich, auch der Einsatz von farbigem Text funktioniert mangels Standardisierung nur in den wenigsten Fällen. Beim Selbstbau von öffentlich zugänglichen Informationssystemen ist daher, ausser in ganz wenigen Spezialfällen, von der Verwendung einer Telnet–Oberfläche abzusehen.

6.2.4 Nutzen von Telnet für Geschäftsbeziehungen

Telnet–Server und -Client besitzen von Hause aus die Möglichkeit zu einer Benutzerauthentifizierung. Dazu wird die Eingabe eines Benutzernamens und eines Passwortes verlangt. Daher eignet sich Telnet prinzipiell als Werkzeug zur Realisierung von Geschäftsbeziehungs–Anwendungen. Problematisch ist aber, dass diese Form des Passwortschutzes nicht sehr sicher ist. Ausserdem gelten auch hier die Einschränkungen aus Abschnitt 6.2.3 bezüglich der Darstellungsmöglichkeiten eines Telnet–basierten Informationssystems. Daher sollte, abgesehen von Spezialfällen, von Telnet abgesehen werden.

6.2.5 Nutzen von Telnet für die Gruppenarbeit

Für seinen eigentlichen Zweck, Logins in Mehrbenutzer–Rechnersysteme über ein Netzwerk zu ermöglichen, kann Telnet hier gut Verwendung finden. Allerdings sind die Sicherheitsprobleme zu beachten.

6.3 FTP

Steckbrief

Zweck: Übertragen von Dateien in TCP / IP–Netzwerken

Funktionsweise: Client–Server–Prinzip; benutzt TCP Port ftp (21) für Steuerinformationen und Port ftp–data (20) oder beliebigen anderen Port für die zu übertragenden Daten

Anwendungsbeispiele:

- Übertragung von Daten von / zu dem eigenen Account
- Herunterladen von öffentlich verfügbaren Daten
- Bereitstellen von öffentlich verfügbaren Daten

6.3.1 Beschreibung

Das File Transfer Protocol (FTP) dient zum Übertragen ganzer Dateien über das Netz. Das Grundprinzip ist ähnlich wie bei Telnet: Der Benutzer braucht einen FTP–Client, auf der Gegenseite läuft ein FTP–Server (auf Port 23). Der Client öffnet eine TCP–Verbindung, der Server führt eine Authentifizierung durch und erlaubt dann das Ablegen oder Herunterladen von Dateien.

Eine Besonderheit besteht darin, dass der Dateitransport meist über getrennte TCP–Verbindungen abgewickelt wird; die ursprüngliche Verbindung wird nur zum Transport von Kommandos benutzt. Das ermöglicht einige zusätzliche Angriffsvarianten, die aber auch eine Abhörmöglichkeit der Leitung voraussetzen und somit auch einfacher zu realisieren sind. Die Sicherheitsprobleme sind prinzipiell die gleichen wie bei Telnet. Erschwerend kommt aber hinzu, dass in den gängigen FTP–Server–Programmen relativ häufig sicherheitsrelevante Programmierfehler gefunden werden. Das liegt daran, dass FTP–Server viele Funktionen enthalten und damit komplexe und umfangreiche Programme sind.

6.3.2 Nutzen von FTP zur Informationsbeschaffung

Viele Organisationen bieten im Internet Dateien zum Herunterladen mittels FTP an. Meistens handelt es sich dabei um Texte in Postscript oder ASCII, oder um Programme im Binärformat oder im Sourcecode. Das meiste an freier Software unter GPL (siehe http: / / www.fsf.org / copyleft / gpl.html) oder anderen freien Lizenzen wird ausschliesslich

über FTP–Server bereitgestellt. Ein FTP–Client ist also ein äusserst nützliches Werkzeug zur Informationsbeschaffung.

6.3.3 Nutzen von FTP für eigene Informationsangebot

Auch wenn nach Meinung einiger Autoren FTP in Zukunft durch HTTP (siehe Abschnitt 6.5) verdrängt werden soll, nutzen viele Organisationen derzeit weiterhin FTP, um Dateien weltweit zum Abruf bereit zu stellen. Da FTP das ältere Protokoll ist, existiert mehr und bessere Client–Software, die beispielsweise das automatische Nachführen von Veränderungen auf einem Server erlaubt. Die meisten Distributionen des frei verfügbaren Betriebssystems Linux (siehe http://www.linux.org/) bieten eine Option, das Betriebssystem direkt von einem FTP–Server zu installieren.

Wer ausschliesslich textuelle und graphische Information öffentlich verfügbar machen möchte, kann sich mit HTTP begnügen. Wer aber auch Dokumente in Binärformaten (z.B. Postscript oder komprimierte Dateien) oder Programmdateien anbieten möchte, sollte den Betrieb eines FTP–Servers in Erwägung ziehen.

6.3.4 Nutzen von FTP für Geschäftsbeziehungen

Die gängige FTP–Server- und Client–Software bietet die Möglichkeit einer Benutzerauthentifizierung. Dadurch wird es möglich, einzelne Dateien nur bestimmten Benutzern individuell verfügbar zu machen. Da dies aber bei anderen Protokollen wie HTTP ebenso möglich ist, sollte FTP nur in den oben genannten Fällen zum Einsatz kommen.

6.3.5 Nutzen von FTP für die Gruppenarbeit

Der ursprüngliche Zweck von FTP war es, Dateien zwischen Accounts einer Person auf mehreren Rechnern zu transportieren. Dazu erfolgt eine Benutzerauthentifizierung, bei der allerdings Sicherheitsbedenken zu beachten sind (siehe Abschnitt 7.4).

6.4 Electronic Mail

Steckbrief

Zweck: Kommunikation mittels Textdateien

Funktionsweise: Übertragung der Mails mittels der Protokolle SMTP (TCP Port 23), POP 3 (TCP Port 110) und IMAP (TCP Port 143)

Anwendungsbeispiele:

- Offline–Kommunikation zweier Benutzer
- Diskussion unter mehreren Benutzern (Mailingliste)
- Offline–Datenbankabfragen

6.4.1 Beschreibung

Das Simple Mail Transfer Protocol (SMTP) ist das Protokoll, mit dem E–Mail im Internet transportiert wird. Es basiert auf TCP und liesse sich im Prinzip ebenfalls nach dem Client–Server–Prinzip betreiben, dass also der Absender eine Mail mit seinem Client direkt eine Verbindung zu einem SMTP–Server auf dem Rechner des Mail–Empfängers eröffnet. Dies ist jedoch aus verschiedenen Gründen nicht üblich: Erstens soll E–Mail häufig auf Rechnern empfangen werden, die an Arbeitsplätzen stehen und nicht ständig eingeschaltet sind oder keine permanente Verbindung zum Internet haben, zweitens kennt der Absender einer Mail im Allgemeinen auch nicht die IP–Nummer oder den Rechnernamen der Maschine, auf die eine Mail ausgeliefert werden soll.

Die Architektur eines E–Mail–Systems besteht daher meistens aus einer Unterteilung in Mail User Agents (MUA) und Mail Transport Agents (MTA). Abbildung 6.2 zeigt eine mögliche Anordnung.

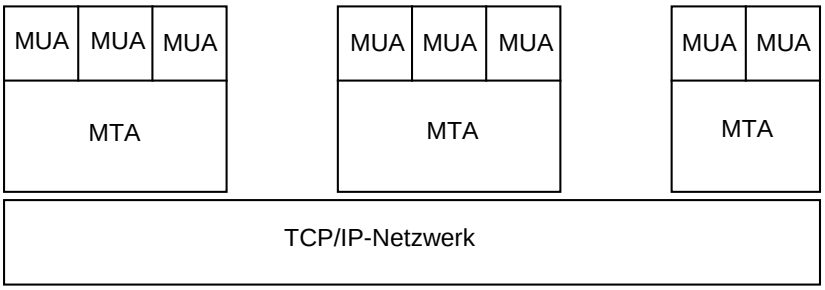

Abbildung 6.2: Aufbau von E–Mail–Systemen im Internet

Die einzelnen MTA's kommunizieren über das SMTP–Protokoll über TCP miteinander. Dabei baut jeweils derjenige MTA die Verbindung auf, der eine Mail versenden möchte.

Der MTA und die MUA's können auf dem selben oder auf verschiedenen Rechnern laufen; die Kommunikation zwischen ihnen erfolgt entweder über das lokale Dateisystem eines Rechners oder Rechnerverbundes, oder nach dem Client–Server–Prinzip über eine TCP–Verbindung. Der MUA ist dabei der Client, er baut die Verbindungen auf und holt Mail ab bzw. übergibt Mail an den MTA. Gängige Protokolle zum Abholen von Mail sind das Post Office Protocol (POP) oder das Internet Mail Agent Protocol (IMAP). Zum Versenden von Mail wird meist SMTP benutzt.

Die Zuordnung zwischen MTA und MUA's kann technisch auf vielfältige Weise gemacht werden. Da die Kommunikation über TCP erfolgen kann, können MUA und MTA beliebig weit im Netz entfernt sein. Die Planung erfolgt meist nach organisatorischen oder Netzlast–Gesichtspunkten. Kleinere Organisationen können möglicherweise nur einen MTA verwenden, der für die gesamte E–Mail zuständig ist; in grösseren Unternehmen kann ein MTA pro Netzstrang, Gebäude, Stockwerk oder Abteilung Verwendung finden.

Der Einsatz von E–Mail ist nicht zwingend mit einer Internet–Anbindung verbunden. E–Mail ist sogar wesentlich älter als das Internet und wird sowohl in anderen Online–Transportsystemen (Bitnet, IPX) als auch in Offline–Systemen (UUCP, FTN) verwendet.

Dank spezieller Gateways ist es möglich, dass Benutzer verschiedener Netze per E–Mail miteinander kommunizieren. Ein solches Gateway wird durch einen MTA realisiert, der das Versenden von Mail über (mindestens) zwei verschiedene Transportsysteme beherrscht.

6.4.2 Mailinglisten

Eine Mailingliste ist ein Mailadresse, über die nicht eine einzelne Person, sondern eine Personengruppe angesprochen werden kann. Jede Mail, die an diese Adresse geschickt wird, wird automatisch an alle eingeschriebenen Teilnehmer weitergeleitet. Es gibt verschiedene Formen von Mailinglisten:

offene Mailingliste: Jeder Benutzer kann / darf eine E–Mail an die Listenadresse schreiben, auch wenn er selber nicht in die Liste eingeschrieben ist.

geschlossene Mailingliste: Nur Benutzer, die selbst eingeschrieben sind, dürfen Mail an die Liste schicken.

moderierte Mailingliste: Alle Beiträge werden zuerst dem Moderator der Liste zugestellt. Dieser prüft sie und leitet sie ggf. an alle Teilnehmer weiter. Dies ist z.B. dann sinnvoll, wenn über die Liste keine Diskussionen geführt, sondern nur Ankündigungen verbreitet werden sollen.

teilmoderierte Mailingliste: Eingeschriebene Benutzer dürfen selber E–Mails an die Liste schreiben. Versucht ein Benutzer, der nicht eingeschrieben ist, eine Mail an die Liste zu schicken, dann wird diese Mail dem Moderator zugestellt. Dieser leitet sie ggf. an die Liste weiter.

Auch bei der Art, wie Benutzer sich einschreiben und austragen, gibt es unterschiedliche Vorgehensweisen:

offene Einschreibung: Jeder Benutzer kann sich selber in die Mailingliste eintragen. Dies geschieht in der Regel automatisch über eine spezielle E–Mail–Adresse, die sogenannte request–Adresse. Mails an diese Adresse bekommt ein Programm, das sie automatisch auswertet und den Benutzer ein-/austrägt.

geschlossene Einschreibung: Es gibt einen Listenverantwortlichen, der entscheidet, welche Benutzer in die Liste aufgenommen werden dürfen. Mails an die request–Adresse werden automatisch an den Listenverantwortlichen geleitet.

Offene Mailinglisten mit geschlossener Einschreibung sind am einfachsten zu realisieren. Diese Möglichkeit biete fast jeder MTA. Für die anderen Formen werden Zusatzprogramme benötigt. Diese Programme bieten meist noch eine grosse Fülle weiterer Einstellungsmöglichkeiten. Beispielsweise können sie jeder E–Mail, die an die Liste geschickt wird, zusätzliche Textzeilen anhängen oder das Subject (eine Zeile, in der das Thema der E–Mail angegeben wird, und die von MUAs besonders hervorgehoben wird) verändern.

6.4.3 Nutzen von E–Mail zur Informationsbeschaffung

Die Betreiber vieler Informationsangebote im Internet bieten die Möglichkeit, den Interessenten der E–Mail weitere Informationen zukommen zu lassen. Dazu ist entweder eine Möglichkeit vorhanden, seine eigene E–Mail–Adresse zu hinterlassen, um die gewünschten Informationen geschickt zu bekommen, oder es ist die E–Mail–Adresse eines Mitarbeiters angegeben, den man direkt anschreiben kann.

Firmen, die Produkte im EDV–Bereich anbieten, haben häufig Mailinglisten für ihre Produkte eingerichtet. In sie können sich alle Kunden oder Interessierten einschreiben und somit neuste Informationen über die Produkte erhalten und sich bei Problemen gegenseitig unterstützen.

6.4.4 Nutzen von E–Mail für eigene Informationsangebote

Analog zu oben gesagtem kann eine Firma in eigenen Publikationen eine E–Mail–Adresse angeben und somit für Auskunftsanfragen erreichbar sein. Die eingehenden Mails müssen von einem Mitarbeiter beantwortet werden. Eine andere Möglichkeit ist es, einen sogenannten Autoresponder einzurichten. Dies ist ein Programm, das Mails bekommt, die an eine bestimmte Adresse geschickt werden, und dem Absender automatisch einen Standardtext zuschickt. Solche Autoresponder sind häufig unter der Adresse infoFirmenname.com oder info–ProduktnameFirmenname.com zu erreichen.

Wenn zu erwarten ist, dass ein Grossteil der Kunden eine E–Mail–Adresse besitzt (z.B. bei Produkten im EDV–Bereich), dann kann eine Mailingliste eingerichtet werden, in die sich Kunden oder Interessierte eintragen können. An diese Liste können aktuelle Informationen über das jeweilige Produkt geschickt werden, und die Liste kann von ihren Mitgliedern zu Diskussion und Problemberichten über das Produkt genutzt werden.

6.4.5 Nutzen von E–Mail für Geschäftsbeziehungen

Da per E–Mail die direkt Kommunikation zwischen zwei Personen ermöglicht, können darüber geschäftliche Beziehungen mit Kunden abgewickelt werden. Manche Firmen bieten ihren Kunden bereits Beratung, Produktbestellungen, oder Produktsupport per E–Mail an. Ebenfalls denkbar wäre eine Mailingliste mit geschlossener Einschreibung, an die aktuelle, wertvolle Informationen (z.B. aktuelle Wertpapierkurse oder Anlagetips) verschickt werden kann. Für die Teilnahme an diesem Dienst könnte auch eine Gebühr verlangt werden.

6.4.6 Nutzen von E–Mail für Gruppenarbeit

E–Mail eignet sich sehr gut für die Überbrückung räumlicher oder zeitlicher Distanzen bei der computergestützten Gruppenarbeit. Neben persönlicher Mail können geschlossene Mailinglisten zum Einsatz kommen.

6.5 Das World Wide Web

Steckbrief

Zweck: Publikation von Dokumenten vielfältiger Typen im Internet.

Funktionsweise: Übertragung von Dateien, meist Hypertext–Dokumente, über das HTTP–Protokoll (TCP, meist Port 80)

Anwendungsbeispiele:

- Abruf öffentlich zugänglicher Informationen
- Bereitstellung öffentlich abrufbarer Informationen
- Informationsangebot für geschlossene Benutzergruppen (Intranet)

6.5.1 Beschreibung

Das World Wide Web (WWW) ist ein System zum einfachen Abrufen und Darstellen von Dokumenten und Dateien in TCP/IP–Netzen. Im Prinzip ist es nichts weiter als die Möglichkeit, Dateien in mehreren standardisierten Formaten (Textdokumente, Bilder, Audio, Video) über verschiedene, teils altbekannte Protokolle der TCP/IP–Protokollfamilie zu übertragen und mit einem einzigen Programm, dem Web–Client oder WWW–Browser genannt, darstellen zu können.

Das zentrale Dateiformat ist die Hypertext Markup Language (HTML). HTML–Dokumente sind Textdateien mit Steueranweisungen (Tags). Der Text wird vom Client als Fliesstext betrachtet und entsprechend den Eigenschaften des Ausgabemediums (Bildschirmgrösse, Druckformat, Farbanzahl) und den Wünschen des Benutzers (Zeichensatz, Fenstergrösse) entsprechend dargestellt. Die Tags dienen dazu, Strukturelemente wie Überschriften, Listen oder Trennlinien in den Text einzubringen, Grafiken nachzuladen und einzelne Seitenelemente als Verweise (Links) auf andere Dokumente zu kennzeichnen. Daneben gibt es noch Tags, mit denen sich Eingabemasken realisieren lassen.

Ein im Web verfügbares Dokument wird über eine Universal Resource Locater (URL) angesprochen. Eine URL besteht aus einer Angabe zum Übertragungsprotokoll, der Quelle, über die das Dokument zu beziehen ist, sowie optional einer Reihe von Parametern. Die Übergabe von Parametern ist nur bei bestimmten Protokollen möglich und ermöglicht eine Interaktion des Benutzers mit dem Server.

HTML–Dokumente werden meist im Hypertext Transmission Protocol (HTTP), dem zentralen Protokoll des World Wide Web, übertragen. Die URL eines über HTTP verfügbaren Dokumentes kann zum Beispiel lauten:

```
http://www.ifi.unizh.ch/events/dexa.html
```

Dabei bezeichnet `http` den Protokolltyp und `www.ifi.unizh.ch` den DNS–Namen des Rechners, der das Dokument anbietet. `dexa.html` ist der Name des Dokumentes, `events/` das Verzeichnis, in dem das Dokument auf dem Server liegt. Auf dem genannten Rechner muss ein HTTP–Server (httpd) laufen. Die Portnummer für HTTP ist üblicherweise 80. Wegen der zentralen Bedeutung von HTTP für das WWW wird ein httpd häufig auch als WWW–Server bezeichnet.

Andere Übertragungsprotokolle, die über einen WWW–Browser genutzt werden können, sind das oben beschriebene FTP, gohper (das Protokoll eines älteren, WWW–ähnlichen Informationssystems) oder S–HTTP (secure HTTP, ein HTTP–ähnliches Übertragungsprotokoll mit verbesserten Sicherheitseigenschaften).

Das Common Gateway Interface (CGI)

HTML–Seiten müssen nicht immer statische Dokumente sein, sondern können (einfache) Interaktionen zwischen Benutzer und Server erlauben. Zu diesem Zweck ist es möglich, in ein HTML–Dokument Textfelder zum Ausfüllen oder Boxen zum Ankreuzen einzubauen. Zur Auswertung dieser Benutzereingaben dient das Common Gateway Interface (CGI). Es definiert einen Standard, mit dem Webserver mit externen Programmen kommunizieren können. Die Idee besteht darin, dass der HTTP–Server bei einer Client–Anfrage nicht ein (statisches) Dokument zurückliefert, sondern statt dessen das Dokument von einem externen Programm dynamisch generieren lässt. Der CGI–Standard legt die Form fest, in das Programm das erstellte Dokument zurückliefert und in dem eventuell auch Parameter an das Programm übergeben werden können, die zum Beispiel durch die interaktiven Elemente einer HTML–Seite vom Benutzer abgefragt wurden.

Aus der Sicht des WWW–Benutzers ist ein von einem CGI–Programm generiertes Dokument prinzipiell nicht von einem statischen Dokument zu unterscheiden. Häufig erkennt man CGI–Programme zwar daran, dass sie sich in einem Unterverzeichnis cgi–bin o.ä. befinden, aber das ist lediglich eine Konvention und keinesfalls bindend. Auch entstehen durch den Aufruf von CGI–Programmen keine unmittelbaren Sicherheitsgefahren für den Benutzer. Der Typ des zurückgelieferten Dokumentes ist allerdings a–priori unbekannt.

6.5.2 Java–Applets, Javascript und Active X

Die Programmiersprache Java wurde von Sun Microsystems entwickelt. Es handelt sich um eine objektorientierte Programmiersprache mit strengem Typenkonzept und Typüberprüfung zur Laufzeit. Darüber hinaus ist Java dafür konzipiert, eine Sprache für mobilen Code zu sein: Der Java–Compiler übersetzt den Sourcecode in einen maschinenunabhängigen Zwischencode, den Bytecode. Dieser Code wird auf dem Zielsystem interpretiert und setzt keine speziellen Hardware- oder Betriebssystemeigenschaften voraus, sondern lediglich das Vorhandensein der standardisierten Java Virtual Machine (JVM).

Das HTML–Dokumentenformat unterstützt die Einbindung von Java–Programmen, sog. Applets, in HTML–Dokumente. Mit einem speziellen Tag wird ein Applet von einem

WWW–Server geladen und ein rechteckiger Bereich in der Darstellung des HTML–Dokumentes reserviert, in dem die Ausgaben des Applets angezeigt werden.

Das automatische Ausführen von Programmen, die aus dem Netzwerk bezogen werden, ermöglicht interessante neue Anwendungen. So lassen sich beispielsweise komplexere Interaktionen zwischen Anwender und Server realisieren, als dies mit den einfachen HTML–Formularen möglich wäre. Diese Flexibilität hat jedoch ihren Preis: Vom Standpunkt der Sicherheit her ist das Laden und Ausführen von fremden Programmcode auf das eigene System eine erhebliche Gefahr, denn es liegt ausschliesslich in der Hand des Programmierers, was das Programm tatsächlich macht.

Javascript ist eine Erweiterung von der Netscape Corporation entwickelte Erweiterung des HTML–Dokumentenformates um eine (einfache) Programmiersprache. Trotz der Namensähnlichkeit und der Idee des mobilen Codes hat Javascript keine Gemeinsamkeit mit Java. Neben dem Netscape Navigator unterstützt auch der Internet Explorer von Microsoft Javascript.

Javascript–Funktionen werden direkt im HTML–Dokument definiert und aufgerufen. Sie ermöglichen einfache interaktive Anwendungen und die Steuerung vieler Funktionen des WWW–Browsers. Damit können harmlose Dinge wie das Anzeigen von Laufbändern in der Statuszeile erzeugt werden, allerdings auch E–Mails an beliebige Empfänger verschickt oder Mausbewegungen des Benutzers aufgezeichnet werden.

Bei Microsofts ActiveX handelt es sich um ein Konkurrenzprodukt zu Java und den Netscape–Plug–Ins, das bislang nur mit dem MS Internet Explorer und nur unter den Windows–Plattformen zu benutzen ist; eine Portierung auf MacIntosh und Unix ist in Arbeit. In Analogie zu Java können kleine Programme, sog. ActiveX–Controls, über das WWW geladen und ausgeführt werden. Dies kann auch automatisch geschehen, indem eine HTML–Seite eine inline–Referenz auf das ActiveX–Control enthält.

6.5.3 Helper Applications und Plug–Ins

Wie oben beschrieben, erfährt ein WWW–Browser beim Laden eines Dokumentes über HTTP dessen Typ. Einige Typen kann der Browser selbst verarbeiten, andere sind ihm unbekannt. Die gängigen Browser können jedoch leicht dazu gebracht werden, neue Dokumentenarten zu verarbeiten: sie verwalten eine Tabelle von Typennamen und dazugehörigen Programmen, die Dokumente dieses Typs weiterverarbeiten können. Diese Tabelle ist vom Benutzer leicht erweiterbar. Beispielsweise kann für den Dokumententyp `text/postscript` ein Programm zum Anzeigen von Postscript–Dokumenten aufgerufen werden, oder `application/framemaker`–Dokumente können direkt an Framemaker übergeben werden.

Ähnlich wie die Helper Applications bieten beim Netscape Navigator auch Plug–Ins die Möglichkeit, die Fähigkeiten des Browsers im Bezug auf unterstützte Dokumententypen zu erweitern. Im Unterschied zu den Helper Applications sind Plug–Ins keine eigenständigen Programme, sondern Module, die vom Browser aufgerufen werden können. Die Ergebnis-

se werden nicht in einem externen Fenster, sondern in einem Bereich des Browserfensters dargestellt. Die Kommunikation zwischen Browser und Plug–In erfolgt über eine genormte, von Netscape veröffentlichte Schnittstelle.

6.5.4 Einsatz des WWW zur Informationsbeschaffung

Die meisten öffentlich zugänglichen Informationen im Internet können gegenwärtig über das WWW abgerufen werden. Der Benutzer braucht dazu lediglich ein einzelnes Client–Programm, das alle im WWW verwendeten Dienste unterstützt: einen WWW–Client, auch WWW–Browser genannt. Der Browser stellt Dokumente in HTML oder anderen Formaten textuell oder graphisch dar; bei exotischeren Dateiformaten, die er selber nicht unterstützt, bedient er sich ggf. externen Programmen. WWW–Browser sind für die gängigen Betriebssysteme kostenlos oder preiswert erhältlich.

6.5.5 Einsatz des WWW für eigene Informationsangebote

Sollen die Informationsangebote des eigenen Unternehmens für einen grossen Leserkreis verfügbar sein, dann führt am Einsatz des WWW kein Weg vorbei. Statische Informationsangebote können in Form von HTML–Dateien auf HTTP–Servern angeboten werden; für dynamische oder interaktive Informationsangebote können die Techniken CGI, Java oder Javascript verwendet werden.

6.5.6 Einsatz des WWW für Geschäftsbeziehungen

Die gängige Software für das WWW bietet die Möglichkeit der Benutzerauthentifizierung. Damit wird es möglich, bestimmte Informationen nur einzelnen Personen oder Institutionen zur Verfügung zu stellen, etwa nur bei Voranmeldung und gegen Entgelt. Beispielsweise könnte ein Finanzdienstleistungs–Unternehmen, das Wertpapierdepots für seine Kunden verwaltet, die Kunden über das WWW auf die eigenen Depots zugreifen lassen.

6.5.7 Einsatz des WWW für die Gruppenarbeit

Für die computerunterstützte Gruppenarbeit kann das WWW bedingt als Werkzeug verwendet werden. Das Lesen von Dokumenten ist mit einem herkömmlichen Browser möglich. Das Publizieren von HTML–Dokumenten erfordert den Zugriff auf das Dateisystem des WWW–Servers. Bei gängiger Serversoftware sind dazu andere Dienste wie NFS, SMB oder FTP nötig. Ausserdem werden zum Erstellen von HTML–Dokumenten entweder Kenntnisse der Sprache HTML oder aber spezielle HTML–Editor–Werkzeuge benötigt. Neuere Entwicklungen im WWW–Bereich unterstützen auch das direkte Verändern von HTML–Dokumenten mit dem Client und zurückschreiben auf den Server. Diese Technik befindet sich jedoch noch in der Entwicklung, und ihr Einsatz ist mit einigen Problemen verbunden.

Gängige WWW–Server bieten Zugriffsbeschränkungen anhand der Herkunft des Zugriffes (IP–Nummer, Rechnername) oder anhand des Benutzernamens an. Somit lassen sich bestimmte Bereiche auf WWW–Servern exklusiv von definierten Benutzergruppen verwenden.

6.6 Die Berkeley r–Tools

Steckbrief

Zweck: Einloggen und Programmausführung auf entfernten Rechnern

Realisierung: Client–Server–System; Kommunikation über TCP Ports exec (512), login(513) und shell (514)

Anwendungsbeispiele:

- Interaktives Arbeiten auf einem entfernten Rechner
- (Automatisierte) Kommandoausführung auf einem entfernten Rechner
- (Automatisiertes) Kopieren von Dateien zwischen unterschiedlichen Rechnern

6.6.1 Beschreibung

Die Kommandos rlogin (remote login), rsh (remote shell) und rcp (remote copy) stammen von der Universität Berkeley und waren ursprünglich Bestandteil von BSD.

rlogin ermöglicht, ähnlich wie Telnet, das Einloggen auf einem anderen Rechner im Netzwerk. Im Unterschied zu Telnet kann dieses Einloggen jedoch auch ohne Passwortschutz erfolgen, wenn der Zielrechner dem Ursprungsrechner vertraut. Die Authentifizierung des Rechners erfolgt dabei über IP–Nummern.

rsh ermöglicht das Ausführen einzelner Kommandos auf einem anderen Rechner, rcp das Kopieren von Dateien über Rechnergrenzen hinweg. Beide Kommandos funktionieren überhaupt nur dann, wenn der Zielrechner dem Ursprungsrechner vertraut; ist dies nicht der Fall, dann wird auch keine Passwort–Authentifizierung vorgenommen, sondern die Programme versagen ihren Dienst.

6.6.2 Einsatz der r–Tools zur Informationsbeschaffung

rlogin bietet eine ähnliche Funktionalität wie Telnet. Im Gegensatz zu Telnet gibt es jedoch, abgesehen von einigen Netzwerkspielen, keine öffentlich verfügbaren Dienstangebote auf Basis von rlogin nutzbar. Somit ist der Nutzen von rlogin zur Informationsbeschaffung sehr eingeschränkt. Die Ausführung von rsh und rcp setzt voraus, dass der Anbieter auf seinem Server eine Benutzerkennung eingerichtet hat, die ohne Passwortschutz zugänglich ist. Dies ist nicht empfehlenswert, und daher gibt es keine öffentlich nutzbaren Informationsangebote, die auf rcp und rsh basieren.

6.6.3 Einsatz der r–Tools zum Anbieten eigener Informationen

Das Anbieten eigener Informationen ist aus den im letzten Abschnitt genannten Gründen nicht sinnvoll und sollte nicht in Betracht gezogen werden.

6.6.4 Einsatz der r–Tools zu Geschäftsbeziehungen

Da rsh und rcp keinen Authentifizierungsmechanismus vorgesehen haben, scheiden sie für den Einsatz personifizierter Kommunikation aus. rlogin kann verwendet werden, um bestimmten Personen ein Einloggen auf einem Rechner mit Mehrbenutzerbetriebssystem zu ermöglichen. Es bietet hier jedoch keine Vorteile gegenüber Telnet. Historisch hat sich für diesen Einsatzbereich Telnet durchgesetzt (siehe Abschnitt 6.2).

6.6.5 Einsatz der r–Tools für die Gruppenarbeit

Die r–Tools eignen sich zum Einloggen auf entfernten Rechnern, zum Kopieren von Dateien zwischen verschiedenen Rechnern und zum Ausführen von Kommandos auf entfernten Rechnern. Für reinen Datenaustausch gibt es jedoch andere Protokolle, die mehr Vorteile bieten. Der Einsatz der r–Tools für die Gruppenarbeit sollte daher nur in sehr speziellen Fällen in Betracht gezogen werden.

6.7 Der Domain Name Service (DNS)

Steckbrief

Zweck: Abfragen der Zuordnung zwischen IP–Nummern und Rechner-/Domainnamen

Realisierung: Verteilter, hierarchisch aufgebauter Dienst; UDP 53

Einsatzszenarien:

- Ein DNS–Server wird für jede Domain benötigt, die einen Domainnamen beantragen möchte.
- Häufig Trennung in öffentlichen DNS für die öffentlich zugänglichen Rechner und versteckten DNS für das interne Netzwerk.

6.7.1 Beschreibung

Der Domain Name Service (DNS) ist der Dienst, der im Internet die Zuordnung zwischen Rechnernamen und IP–Nummern durchführt. Sinn der Rechnernamen ist, dass sich IP–Nummern schwer merken lassen und die Zuordnung zwischen IP–Netznummern zu organisatorischen und geographischen Strukturen nicht offensichtlich ist.

Ein Beispiel: Der IP–Nummer 130.60.48.8 ist der Rechnername frederic.ifi.unizh.ch zugeordnet. Der Rechnername setzt sich dabei aus dem Hostnamen frederic und dem Domainnamen ifi.unizh.ch zusammen. Rechnernamen werden auch als Fully Quantified Domain Name (FQDN) bezeichnet. Das Ermitteln der IP–Nummer zu einem bekannten Rechnernamen nennt man Lookup (Auflösen, Resolving), den umgekehrten Weg Reverse Lookup.

Im Internet gibt es keine zentrale Datenbank, in der alle Rechnernamen mit der dazugehörigen IP–Nummer gespeichert sind. Statt dessen gibt es viele einzelne Rechner, die einen Teil des Datenbestandes kennen, genannt Nameserver. Die Anordnung der Nameserver folgt dabei dem Aufbau der Domainnamen (siehe Abbildung 6.3).

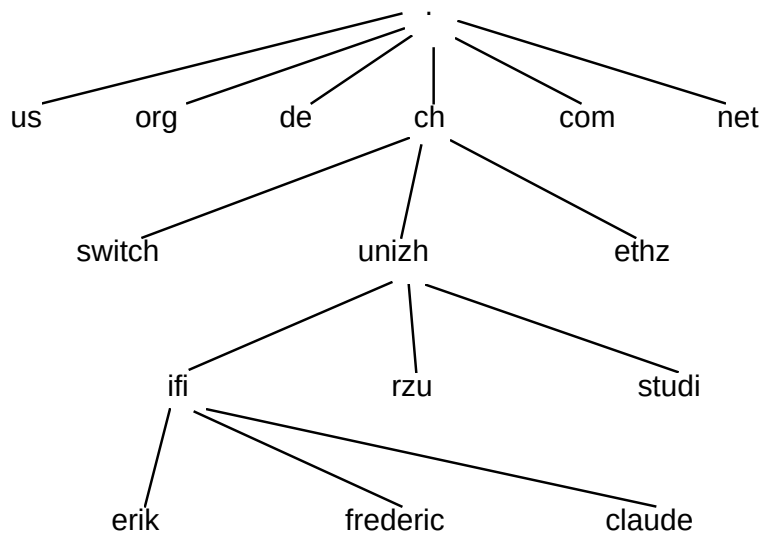

Abbildung 6.3: Serverhierarchie beim Domain Name Service

Angenommen, ein Rechner irgendwo im Internet möchte die IP–Nummer des Rechners `frederic.ifi.unizh.ch` ermitteln. Dazu muss er einen beliebigen Nameserver kennen; üblicherweise betreibt jede Organisation mindestens einen davon. Der Nameserver wird feststellen, dass er den gesuchten Namen nicht kennt (ausser es handelt sich um den Nameserver des Institutes für Informatik der Uni Zürich, was hier aber ausgeschlossen werden soll). Er weiss aber, welches die Root–Nameserver sind, die alle Länderkürzel kennen (`us`, `org`, `ch`, `de`, `com`, `net` usw.). Root–Nameserver gibt es weltweit neun Stück, und alle Nameserver im Internet müssen diese kennen.

Der Nameserver befragt im ersten Schritt die Root–Nameserver nach dem gesuchten Namen `frederic.ifi.unizh.ch`. Aus Gründen der Fehlertoleranz werden alle neun auf einmal befragt; die erste Antwort wird verwendet. Die Antwort lautet, dass der Name selbst nicht bekannt ist, wohl aber die IP–Nummern der Nameserver für die Zone `ch.` Im zweiten Schritt werden diese nach dem gesuchten Namen befragt. Wieder lautet die Antwort, dass der Name selbst nicht bekannt ist, aber die IP–Nummern der Nameserver von `unizh.ch`. Der dritte Schritt liefert die Nameserver von `ifi.unizh.ch`, der vierte schliesslich die gesuchte IP–Nummer zu `frederic.ifi.unizh.ch`.

Für die umgekehrte Richtung existiert ein ähnlicher Mechanismus. Eine Anfrage nach

dem Rechnernamen, der zu der IP–Nummer `130.60.48.8` an den Reverse–Lookup–Root–Nameserver führt zu den IP–Nummern der Server für `130.x.x.x`. Im zweiten Schritt werden Nameserver für den Bereich `130.60.x.x` ermittelt, im dritten für `130.60.48.x`. Die vierte Anfrage liefert schliesslich den gesuchten Rechnernamen.

Neben dem oben beschriebenen „normalen“ Ablauf einer DNS–Anfrage kann es zahlreiche Besonderheiten geben. Beispielsweise könnte der Nameserver für `130.x.x.x` auch direkt den für `130.60.48.x` kennen und somit die Anfrage um einen Schritt verkürzen. Es ist möglich, einem Namen mehrere IP–Nummern zu geben und umgekehrt, und es gibt auch Aliasnamen, denen keine IP–Nummer, sondern ein anderer Name zugeordnet ist; beispielsweise ist `www.ifi.unizh.ch` ein Alias auf `claude.ifi.unizh.ch`.

Der Domain Name Service spielt aus der Sicht der Netzwerksicherheit eine Rolle, wenn die damit ermittelten Informationen zu Authentifizierungs- oder Auditing–Zwecken verwendet wird. Zum Beispiel liegt vielen Unix–Versionen ein sogenannter TCP–Wrapper bei, mit dessen Hilfe TCP–Verbindungen abgelehnt oder angenommen werden können. Entscheidungskriterium hierbei kann der Rechnername sein, von dem die Verbindung kam. Angenommen, ein Rechner erlaubt einkommende TCP–Verbindungen nur, wenn die Rechnernamen der Gegenseite auf *.unizh.ch (also auf alle Rechnernamen mit der Endung .unizh.ch) passen. Von der einkommenden Verbindung ist zunächst nur die IP–Nummer bekannt; über den Domain Name Service wird der zuständige Nameserver ermittelt und nach dem Rechnernamen gefragt. Ein Angreifer ausserhalb der Universität Zürich, der selbst den Nameserver für seine IP–Nummern betreibt, kann ihn so manipulieren, dass er tatsächlich einen Namen unterhalb von .unizh.ch zurückliefert, wenn er weiss, dass dies das Kriterium für den Zugang ist. Diese Technik wird DNS–Spoofing genannt.

Abhilfe gegen DNS–Spoofing ist relativ einfach: Sobald der Rechnername ermittelt ist, wird das Resolving in umgekehrter Reihenfolge durchgeführt, also die zum ermittelten Rechnernamen gehörende IP–Nummer ermittelt. Diese Anfrage landet, sofern nicht gerade die Root–Nameserver oder die Server für ch korrupt sind, bei den Nameservern von unizh.ch und wird mit Sicherheit nicht mehr die ursprüngliche IP–Nummer zurückliefern (da sich diese ja nicht auf dem Campus befindet). In diesem Fall wird die Verbindung nicht erlaubt oder ein Alarm ausgelöst.

Neuere TCP–Wrapper besitzen die Fähigkeit, die ermittelten Rechnernamen zu überprüfen. Jedoch muss diese Eigenschaft extra aktiviert werden; die Begründung vieler Hersteller ist, dass sie zusätzliche Netzlast erzeugt.

6.7.2 Softwarekomponenten im DNS

Zur Nutzung der Funktionalität des Domain Name Service werden verschiedene Softwarekomponenten benötigt. Diese werden üblicherweise unter dem Begriff Nameserver zusammengefasst und als ein Produkt angeboten. Eine Aufteilung der Funktionen auf mehrere Programme ist jedoch ebenso möglich.

Funktionalitäten eines Nameserver (NS) sind:

primary Nameserver: Ein Nameserver, der Informationen über eine Zone / Domain besitzt. Diese Information wird üblicherweise vom Netzwerkadministrator in eine Textdatei eingetragen. Für jede Zone gibt es genau einen primary NS.

secondary Nameserver: Dieser NS kennt ebenfalls die Information über eine Zone. Allerdings wird sie nicht von Hand in eine Datei eingetragen, sondern der secondary NS kopiert sie sich in regelmässigen Zeitabständen vom primary NS (meist alle 1–2 Tage).

resolving Nameserver: Dieser Typ NS ist zuständig für das Erfragen eines Eintrags im DNS. Dazu befragt er hierarchisch die Nameserver wie oben beschrieben, bis er den entsprechenden Eintrag findet. Einmal gefundene Einträge können in einem Cache abgelegt werden, damit das erneute Abfragen des selben Eintrag keine zusätzliche Netzlast erzeugt. Wie lange ein Eintrag im Cache maximal gespeichert bleiben darf, ist in einem entsprechenden Datenfeld der Konfiguration des primary Nameserver angegeben.

forwarding–only Nameserver: Ein NS, der eine NS–Anfrage von einem Client erhält und sie an einen anderen NS weiterleitet. Üblicherweise leitet er sie an einen resolving NS weiter; es ist jedoch auch möglich, sie an einen weiteren forwarding–only NS zu leiten und somit mehrere zu kaskadieren.

6.7.3 Einsatz von DNS zur Informationsbeschaffung

Um auf Informationsangebote zugreifen zu können, die über die gängigen Internet–Dienste wie WWW, FTP oder E–Mail angeboten zu werden, muss der angegebene DNS–Name im DNS abgefragt werden können. Bei E–Mail–Adressen ist das der Teil nach dem „@“–Symbol, bei URLs der Rechnerteil (siehe Abschnitt 6.4). Es ist also der Zugriff auf einen resolving oder forwarding–only Nameserver nötig. Dieser Server kann beim Unternehmen selbst oder beim Provider betrieben werden.

In vielen Fällen ist die Nutzung eines Informationsdienstes im Internet nur möglich, wenn der eigene Rechner, auf dem der Client läuft, ebenfalls im DNS eingetragen ist. Ansonsten kann der angesprochene Server auf der Gegenseite die IP–Nummer nicht zu einem FQDN auflösen. Viele Server sind so konfiguriert, dass sie in diesem Fall den Dienst verweigern oder zumindest stark einschränken.

6.7.4 Einsatz von DNS zum Angebot eigener Informationen

Um selbst Informationen anbieten zu können, braucht man in der Regel einen Server, der über eines der dazu geeigneten Protokolle ansprechbar ist. Dieser Server benötigt einen Eintrag im DNS. Beispielsweise ist der WWW–Server des Instituts für Informatik der Universität Zürich unter dem Namen www.ifi.unizh.ch bekannt. Um ihn von aussen ansprechen zu können, muss zunächst die IP–Nummer ermittelt werden. Es wird also ein primary oder secondary Nameserver benötigt, über den diese Zuordnung abgefragt werden kann. Er kann im Unternehmen selbst oder beim Provider betrieben werden.

6.8 Sendfile

Steckbrief

Zweck: Übertragen von Dateien im Internet

Realisierung: Client–Server–Architektur; Aufteilung in Transportsystem und User–Agent ähnlich wie bei E–Mail. Kommunikation über TCP Port 487.

Anwendungsbeispiele:

- Offline–Versenden von Dateien an Benutzer anderer Rechner
- Abrufen öffentlich verfügbarer Informationen

6.8.1 Beschreibung

Sendfile dient zum Übertragen von Dateien im Internet. Es basiert auf SAFT (Simple Asynchronous Filetransfer Protocol), einem vergleichsweise jungem Protokoll der TCP / IP–Protokollfamilie. SAFT und die dazugehörigen Implementierungen sendfile (Client) und sendfiled (Server) orientieren sich stark am send- / receive–Mechanismus des Bitnet und vereinigen die Vorteile von FTP und E–Mail. Ähnlich wie bei E–Mail können sendfile einzelnen Benutzern auf anderen Rechnern Dateien oder ganze Unterverzeichnisse geschickt werden. Der sendende Benutzer braucht keinen Account auf dem entfernten Rechner, und der empfangene muss zum Zeitpunkt des Transfers nicht eingeloggt sein. sendfile bietet Mechanismen zum Verschlüsseln und Signieren der zu übertragenen Dateien. Die Übertragung erfolgt ähnlich wie bei FTP oder HTTP über einen 8–bit–cleanen TCP–Datenstrom. Eine 7–bit Codierung wie bei E–Mail ist daher nicht notwendig.

Auf der Empfängerseite nimmt der sendfiled die Dateien entgegen und lagert sie in einem Zwischenbereich. Beim Einloggen erfährt der Benutzer, dass Dateien für ihn eingetroffen sind, und kann sie annehmen oder den Empfang verweigern. Der sendfiled limitiert verweigert automatisch den Empfang weiterer Dateien, wenn der Plattenplatz knapp wird.

Mit SAFT können ausserdem frei verfügbare Dateien im Internet abgerufen werden. Der Einsatzbereich ist hier sehr ähnlich dem HTTP, allerdings ist SAFT nicht sehr weit verbreitet. Implementierungen von SAFT Clients und Servern gibt es bislang nur für Unix.

6.8.2 Einsatz von SAFT zur Informationsbeschaffung

Mit SAFT können, ähnlich wie mit FTP oder HTTP, Dateien zum Herunterladen öffentlich angeboten werden. Allerdings gibt es nicht viele Angebote im Netz, die das SAFT–Protokoll verwenden.

6.8.3 Einsatz von SAFT zum Angebot eigener Informationen

Ähnlich wie bei FTP können mit SAFT Dateien im Netz zum Herunterladen bereitgestellt werden. Die gängigen WWW–Browser unterstützen jedoch (noch?) kein SAFT, so dass FTP oder HTTP hier vorzuziehen ist.

6.8.4 Einsatz von SAFT für Geschäftsbeziehungen

Mit SAFT können einzelne Dateien gezielt an Benutzer zugestellt werden. Damit ist es möglich, individuell an bestimmte Benutzer oder Benutzergruppen zuzustellen. Aufgrund der geringen Verbreitung von SAFT ist hier allerdings E–Mail vorzuziehen, auch wenn es Nachteile in puncto Performanz hat.

6.8.5 Einsatz von SAFT für die Gruppenarbeit

Für den Einsatz in der Gruppenarbeit ist SAFT sehr gut geeignet. Es ermöglicht es, dass Benutzer anderen Benutzern oder Benutzergruppen gezielt Dateien zuzuschicken. Die Empfänger brauchen dabei nicht gerade eingeloggt zu sein, sondern bekommen beim nächsten Login eine Meldung, dass Dateien für sie zum Abruf bereitstehen. Da es SAFT–Clients allerdings bisher nur für Unix gibt, nutzt es nur Gruppen, die auf UNIX–Systemen arbeiten. Andere Arbeitsgruppen müssen auf E–Mail ausweichen, obwohl SAFT zahlreiche Vorteile bietet.

6.9 Internet Relay Chat

Steckbrief

Zweck: Online–Textkommunikation zwischen Benutzern weltweit

Realisierung: Client–Server–Architektur; Servernetz mit Servern in grösseren Städten. Server–Server–Kommunikation über Port TCP Port 194, Client–Server–Kommunikation meist über TCP Port 6667

Einsatzbereiche:

- direkte Textkommunikation zwischen zwei oder mehreren Benutzern
- Informationsroboter, die auf Befehl bestimmte Informationen ausgeben

6.9.1 Beschreibung

Chat–Systeme sind Systeme, bei denen ein Benutzer direkt am Terminal mit anderen Benutzern auf der Welt sprechen kann. Nach Eingabe einer Zeile dauert es üblicherweise nur Sekunden, bis die Nachricht bei den Kommunikationspartnern angekommen ist. Die Benutzer identifizieren sich durch Spitznamen.

Das Internet Relay Chat (IRC) ist das wohl am weitesten verbreitete Chat–System. Im IRC bilden ein oder mehrere Server zusammen ein Netz. Innerhalb eines Netzes ist es für den Benutzer egal, mit welchem Server er Kontakt aufnimmt. Da die Server ständig die Informationen über die Benutzer und deren Aktivitäten austauschen, hat man von allen Servern aus die gleiche Sicht auf das Netz.

Jeder Benutzer identifiziert sich im IRC durch seinen Spitznamen. Der Spitzname kann frei gewählt und beliebig geändert werden; jedoch muss er stets netzweit eindeutig sein, d.h. es können nicht zwei verschiedene Benutzer gleichzeitig den selben Spitznamen verwenden. Ein Spitzname ist auch nicht reservierbar; sobald ein Benutzer das Netz verlässt, kann ein anderer dessen Spitznamen weiterverwenden.

Zusätzlich zum Spitznamen zeigen IRC–Clients auf Wunsch auch den DNS–Namen des Rechners sowie den Login–Namen des Benutzers an. Dies erlaubt manchmal, jedoch längst nicht immer, einen Rückschluss auf die Identität oder den Aufenthaltsort eines IRC–Teilnehmers.

Um mit einem Benutzer zu kommunizieren, kann man ihm direkt einzeilige Nachrichten schicken. Als Adressierungsinformation dient der Spitzname. Weiterhin ist es möglich, einem Channel beizutreten, auf dem sich mehrere Benutzer befinden. Jeder Channel hat einen Namen und evtl. einen Titel, der die dort erwünschten Gesprächsthemen näher charakterisiert. Innerhalb eines Channels können Nachrichten ohne spezielle Adressierungsinformation abgesetzt werden; sie gehen dann an alle Teilnehmer des Channels. Viele Channels sind allgemeiner Natur, und es wird dort nur Smalltalk betrieben. Manche Channels bieten jedoch sehr fundierte Diskussionen, beispielsweise über Netzwerke oder Betriebssysteme. Einzelne Organisationen bieten sogar in regelmässigen Zeitabständen Online–Diskussionen über spezielle Themen im IRC an.

In den Anfangszeiten des IRC waren alle Server zu einem gemeinsamen Netz verbunden. Mittlerweile haben sich mehrere Netze gebildet, die untereinander nicht verbunden sind. Das hat zum Teil politische Gründe. Die Benimmregeln im ursprünglichen IRC–Netz, in dem sich überwiegend Server an Universitäten befinden, waren recht restriktiv, sowohl bezüglich technischer Vorschriften als auch bezüglich der Inhalte. Es entstanden alternative Netze mit freierer Policy.

Um am IRC teilnehmen zu können, benötigt man einen IRC–Client und den Namen eines Servers in seiner Nähe, der mit dem gewünschten Netz verbunden ist. IRC Clients gibt es u.a. für Unix, VMS, MacOS, Windows und OS/2. IRC–Serversoftware gibt es nur für Unix. Um einen eigenen Server an ein grösseres IRC–Netz anzuschliessen, sind aufwendige organisatorische Massnahmen nötig. Sie umfassen auch den Nachweis, dass man genügend Benutzer hat und dass man sich hinreichend gut in der Materie auskennt. IRC–Server stehen zum grössten Teil an Universitäten.

6.9.2 Nutzen von IRC zur Informationsbeschaffung

In manchen IRC–Servernetzen gibt es spezielle Kanäle, die sich mit festgelegten Themen beschäftigen. Diese Themen sind grösstenteils technischer Natur, z.B. Unix, OS/2, ISDN oder ähnliches. Jedoch sind auch andere Themen wie Politik oder Datenschutz vertreten. Manche Interessengruppen bieten einmalig oder regelmässig Chat–Sessions zu bestimmten Themen an, häufig mit Prominenten. Bekanntes Beispiel ist der Online–Chat mit dem deutschen Bundestagsabgeordneten Jörg Tauss.

6.9.3 Nutzen von IRC zum Angebot eigener Informationen

Auch wenn die meisten IRC–Netze nicht kommerziell sind, ist es erlaubt, (kostenlosen) Online–Support für Produkte der eigenen Firma zu bieten. Auch das Veranstalten von Online–Chats zu Themen oder Produkten der eigenen Firma bietet eine Möglichkeit, die Bekanntheit zu steigern.

6.9.4 Nutzung von IRC für Geschäftsbeziehungen und für die Gruppenarbeit

Das IRC bietet prinzipiell die Möglichkeit, direkt mit einem einzelnen bekannten Partner oder einer geschlossenen Benutzergruppe zu kommunizieren. Allerdings ist die Identifizierung der Teilnehmer und die Kommunikation selbst umständlich. Da die Beteiligten alle zu diesem Zeitpunkt anwesend sein müssen, ist eine Telefonkonferenz in der Regel vorzuziehen.

6.10 Net News

Steckbrief

Zweck: Diskussionsforen über spezielle Themen, z.T. weltweit

Realisierung: kein eigentlicher Internet–Dienst, aber Transport über Internet üblich. Client–Server–Architektur; Servernetz mit einem Server pro (grösserer) Organisation; Server–Server–Kommunikation über NNTP; Client–Server–Kommunikation über NNRP (beide TCP Port 119).

Einsatzszenarien:

- weltweite Diskussionen über bestimmte Themen, Produkte, Anwendungen
- unternehmensinterne Kommunikation, Bekanntmachungen
- Kundensupport für eigene Produkte über eigene Newsgruppen

6.10.1 Beschreibung

Die NetNews (auch Usenet–News, Usenet oder News genannt) sind im Internet abrufbare Diskussionsforen zu einer Vielzahl unterschiedlicher Themen. Die News sind in Hierarchien untergliedert. Einige Hierarchien werden weltweit verbreitet, die Diskussionssprache ist dort üblicherweise englisch. Andere sind lokal begrenzt, und es wird die jeweilige Landessprache benutzt. Innerhalb der Hierarchien gibt es ein hierarchisches System von Newsgruppen (kurz: Gruppen) zu den unterschiedlichen Themen.

Die 8 offiziellen internationalen News–Hierarchien (genannt Big–8) sind comp für computerbezogene Themen, sci für (natur-)wissenschaftliche Themen, rec für Freizeit (recreation), soc für gesellschaftliche Themen, humanities für Geisteswissenschaften, talk für weniger ernste Gespräche, news für Administrativa zum Medium NetNews an sich, sowie misc für sonstige Themen, die in keine der anderen Kategorien passen. Die Newsgruppen innerhalb der Big–8 werden von einer zentralen Instanz verwaltet und (weitgehend) weltweit synchron gehalten. Somit wird sichergestellt, dass jeder Teilnehmer auf dem News–Server seines Providers oder seiner Organisation stets die gleichen Newsgruppen vorfindet.

Beispiele für Gruppennamen aus den Big–8 sind:

`comp.os.linux.networking:` Netzwerk–Aspekte des Betriebssystems Linux

`rec.music.classical.guitar:` Diskussionen über klassische Gitarrenmusik

`soc.culture.australian:` Kultur in Australien

`news.announce.newgroups:` Ankündigungen von neuen Newsgruppen

`sci.crypt:` Kryptographie

Neben den internationalen News–Hierarchien gibt es regionale. Beispiele sind ch für die Schweiz, at für Österreich, de für den deutschsprachigen Raum, stgt für den Grossraum Stuttgart oder ifi für das Institut für Informatik der Universität Zürich. Einige davon werden weltweit verteilt, sind aber aufgrund der verwendeten Sprache nur für ein beschränktes Publikum interessant (z.B. de). Andere existieren nur innerhalb geschlossener Organisationen und werden nicht nach aussen weitergegeben (z.B. `ifi`).

Für die Einrichtung neuer Gruppen oder die Löschung bestehender Gruppen gibt es in den meisten Hierarchien feste Regeln. Meist sehen sie einen festen Ablauf vor, der mit einem Diskussionsaufruf (Request for Discussion, RfD) beginnt und einer Wahl (Call for Votes, CfV) endet. Teilnahmeberechtigt ist jeder interessierte Teilnehmer, der in der Lage ist, eine Stimme per E–Mail einzusenden. Kleinere Hierarchien haben z.T. weniger formale Regeln. Bei geschlossenen Hierarchien innerhalb eines News–Servers ist es üblich, das der News–Administrator oder sein Vorgesetzter alleine über die Einrichtung oder Löschung einer Gruppe entscheidet.

Technisch gesehen werden die NetNews dadurch realisiert, dass jeder Artikel an alle News–Server transportiert werden. Dazu gibt jeder Server neu einkommende Artikel an alle Server

weiter, mit denen er eine Verbindung unterhält. Jeder Server verewigt sich selbst ausserdem in den Kontrollinformationen des Artikel. Ein Server leitet Artikel nur an seine Nachbarn weiter, wenn deren Namen noch nicht im Artikel vermerkt sind. Dennoch kann es passieren, dass Artikel mehrfach bei einem Server eintreffen. Ein typisches Szenario zeigt Abbildung 6.4.

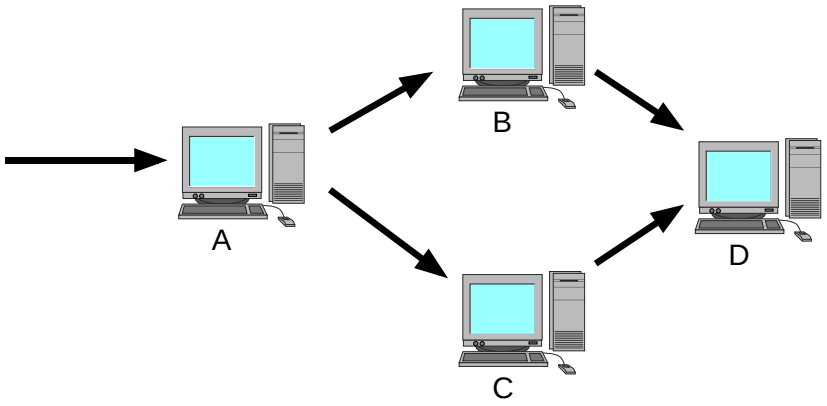

Abbildung 6.4: Redundante News–Server–Vernetzung

In der Abbildung sind die News–Server redundant vernetzt. Artikel, die auf Server A geschrieben werden, werden auf zwei Wegen nach D weitergeleitet, nämlich über B und über C. Dabei weiss Server B nicht, dass der Artikel auch von A über C nach D gelangt und umgekehrt. Daher bekommt D den Artikel zweimal. Eine Datenbank der vorhandenen Artikel auf jedem Server sorgt dafür, dass identische Artikel nur einmal gespeichert und dem Benutzer zum Lesen angeboten werden. Fällt Server B oder C aus, gelangt der Artikel dennoch von A nach D. Daher ist es sinnvoll, mehrere Server–Verbindungen zu haben, um einen stabileren Artikelfluss zu haben.

Eine der Aufgaben von News–Servern ist es also, einkommende Artikel an andere Server weiterzuleiten. Dies geschieht im Internet meist über das Protokoll NNTP (News to News Transport Protocol), es sind jedoch auch andere Transportmechanismen möglich (z.B. UUCP, das Unix–To–Unix–Copy Protokoll). Eine weitere Aufgabe ist es, Verbindungen mit den Client–Programmen (genannt Newsreader) unterhalten zu können, ihnen Artikel zum Lesen anzubieten und neu geschriebene Artikel entgegenzunehmen. Die gängige News–Server–Software erfüllt beide Aufgaben, lässt sich also zur Kommunikation sowohl mit Clients, als auch mit anderen Servern einsetzen. Manche Produkte erfüllen jedoch nur eine der beiden Aufgaben: News–Server, die ausschliesslich mit anderen Servern kommunizieren kann, wird News–Router genannt. News–Router werden von grossen Internet–Providern eingesetzt, um einen schnellen Artikelaustausch zwischen Providern zu ermöglichen. Andere News–Server können nur mit Clients sowie mit einem einzigen anderen Server kommunizieren. Sie beziehen die Artikel von News–Routern und bieten spezialisierte Funktionen für den Zugriff auf die Artikel durch die Clients.

6.10.2 Einsatz der News zur Informationsbeschaffung

Neben dem WWW sind die News die bedeutendste Quelle von Informationen im Netz. In den zahlreichen lokalen und internationalen Newsgruppen werden eine grosse Zahl von Themen abgedeckt. Wer in den News eine Frage stellt, bekommt in den allermeisten Fällen innerhalb von 24 Stunden eine Antwort. Darüber hinaus gibt es Listen von „Frequently Asked Questions“ (FAQ), in denen die am häufigsten gestellten Fragen zu bestimmten Themen abgehandelt werden. Vor allem für Unternehmen im EDV–Bereich, aber auch für andere Branchen, sind die News eine wertvolle Informationsquelle.

6.10.3 Einsatz der News zum Angebot eigener Informationen

Die offiziellen, internationalen und nationalen News–Hierarchien sind streng nicht kommerziell. Das bedeutet, dass es nicht erwünscht und auch nicht sinnvoll ist, Artikel mit Produktwerbung in die News zu setzen. Missachtung dieser Regel kann im schlimmsten Fall die „Usenet Death Penalty“ (UDP, nicht zu verwechseln mit dem User Datagram Protocol) zur Folge haben; eine Massnahme, bei der kein Server im Netz mehr Artikel vom News–Server des betreffenden Unternehmens annimmt. Aber auch in weniger schlimmen Fällen kann das Ansehen der eigenen Firma leiden, wenn sie Werbung im Usenet plaziert.

Was dagegen erlaubt ist, ist Produktsupport in den News, sofern er nicht in Werbung ausartet. Zahlreiche Firmen haben Mitarbeiter, die die thematisch passenden Newsgruppen verfolgen und Fragen bezüglich der eigenen Produkte beantworten. Eine andere Möglichkeit ist die Einrichtung einer eigenen News–Hierarchie zur Firma, in der dann natürlich auch geworben werden kann. Den Interessenten muss der Zugriff per NNRP auf den eigenen Server geöffnet werden, auf dem die Hierarchie lokalisiert ist.

6.10.4 Einsatz der News zu Geschäftsbeziehungen

Aufgrund des nichtkommerziellen Charakters der offiziellen News–Hierarchien eignen sich diese nicht zum Abschluss von Geschäften oder dem Marketing. Das Einrichten eigener News–Hierarchien zum gezielten Support für Geschäftspartner ist dagegen möglich. Die gängige Serversoftware unterstützt rudimentäre Authentifikationsmechanismen, so dass eine Zugriffsbeschränkung realisiert werden kann.

6.10.5 Einsatz der News für die Gruppenarbeit

Eine eigene News–Hierarchie kann innerhalb eines Unternehmens, einer Abteilung oder einer Arbeitsgruppe als produktives Hilfsmittel zur unstrukturierten Diskussion und zu Ankündigungen eingesetzt werden. Der Zugriff auf diese Hierarchie kann entweder auf das lokale Firmennetz beschränkt werden, oder es können die in der gängigen Serversoftware eingebauten Authentifizierungsmechanismen zum Einsatz kommen, um den Zugriff auf bestimmte Benutzer zu beschränken.

Kapitel 7

Bedrohungen/Gefahren und Dienste

7.1 Einführung

Wie in Kapitel 3 bereits erwähnt, wurde der Grundstein zur Erfindung der Internet-Protokolle bereits Ende vor 30 Jahren gelegt. Bei der Entwicklung der Internet-Protokolle stand die Fehlertoleranz und die Eignung für grosse, geographisch weiträumige Netze im Vordergrund. Sicherheitsaspekte spielten dagegen eine untergeordnete Rolle. Im Laufe der Zeit gab es zwar immer wieder Verbesserungen der einzelnen Protokolle im Bezug auf Sicherheit; jedoch sollten die neuen Protokolle kompatibel zu ihren Vorgängern bleiben, was für die Innovation hemmend war.

In diesem Kapitel werden die Schwachstellen der einzelnen Internet–Protokolle erläutert. Abschnitt 7.2 beschreibt die Sicherheitsprobleme der unteren drei Schichten im TCP/IP–Schichtenmodell. Sie sind für jede Organisation relevant, die die Internet–Protokolle in sicherer Weise einsetzen möchte. In den restlichen Abschnitten werden die in Kapitel 6 eingeführten Dienste der Anwendungsebene behandelt.

Ein Überblick über die Zuordnung zwischen Internet–Diensten und Gefahren gibt Abbildung 7.1. In den Zellen ist angegeben, ob der jeweilige Dienst von der genannten Gefahr betroffen ist, und in welchem Abschnitt dieses Kapitels der Zusammenhang näher erläutert wird.

7.2 Gefahren der unteren drei Schichten der Internet–Protokollfamilie

7.2.1 Vertraulichkeit der transportierten Daten

Üblicherweise werden die Daten auf den Schichten I–III unverschlüsselt übertragen. Das bedeutet, dass jeder, der Zugriff auf das physikalische Übertragungsmedium oder einen der beteiligten Router hat, die Daten abhören kann.

Internet-Dienst \ Bedrohung		Verlust der Verfügbarkeit	Verlust der Vertraulichkeit	Verlust der Verbindlichkeit	Verlust der Authentizität	Verlust der Integrität	Verlust der Betriebssicherheit
Telnet	Client	●	7.3.2		●		7.3.2
	Server	●	7.3.2/7.3.3		●	7.3.3	7.3.2/7.3.3
FTP	Client	●	7.4.2		●		7.4.2
	Server	●	7.4.2/7.4.3		●	7.4.3	7.4.2/7.4.3
E-Mail	Client	7.5.2	7.5.2	7.5.3	7.5.3	●	7.5.2
	Server	●	7.5.2/7.5.3		7.5.3	7.5.3	7.5.2/7.5.3
WWW	HTML	●	7.6.2				7.6.2
	Helper Appl.	●	7.6.4		●	7.6.4	7.6.4
	Plug-Ins	●	7.6.4		●	7.6.4	7.6.4
	Javascript	●	7.6.2			7.6.2	7.6.2
	Java-Applets	●	7.6.2		●	7.6.2	7.6.2
	ActiveX	●	7.6.4		●	7.6.4	7.6.4
	CGI	●	7.6.3		●	7.6.3	7.6.3
Berkeley	Client	●	7.7.3		●		7.7.3
r-Tools	Server	●	7.7.2/7.7.3		7.7.2	7.7.2	7.7.2/7.7.3
DNS	Client	●	●		7.8.2		●
	Server	7.8.3	7.8.3		7.8.2	7.8.3	7.8.3
Sendfile	Client	●	7.9.2		7.9.2		7.9.2
	Server	7.9.2	7.9.2/7.9.3	7.9.2	7.9.2	7.9.3	7.9.2/7.9.3
IRC	Client	7.10.2	7.10.2	7.10.2	7.10.2	7.10.2	7.10.2
	Server	●	●				●
Net News	Client	●	7.11.2	7.11.2	7.11.2		7.11.2
	Server	●	7.11.2		7.11.2		7.11.2

Abbildung 7.1: Internet–Dienste und Gefahren

Je nach Übertragungsmedium ist dies jedoch unterschiedlich schwierig. Eine Modemstrecke über eine Telefonleitung abzuhören, gilt als sehr schwer, insbesondere dann, wenn der Angreifer die Datenübertragung selbst nicht beeinflussen möchte oder kann. Die Kommunikation über Analogmodems erfolgt verbindungsorientiert; zu Beginn einer Verbindung werden mehrere Verbindungsparameter ausgehandelt, die sich jedoch im Laufe der Übertragung ändern können. Wenn sich der Angreifer erst später zuschaltet, hat er es sehr schwer, diese Verbindungsdaten zu rekonstruieren. Bekommt ein empfangendes Modem einzelne Teile der Daten nicht mit (z.B. wegen einer kurzen Leitungsstörung), fordert es diese Daten neu an. Der Angreifer kann dagegen nicht so einfach diese Daten neu anfordern. Ein Angriff, der auf blossem Mithören beruht, wird passiver Angriff genannt.

Wenn der Angreifer die Leitung nicht nur abhören, sondern auch beeinflussen oder unterbrechen kann, hat er es einfacher. Er kann dann gezielt Störungen auf die Leitung geben, so dass die Modems gezwungen werden, die Verbindungsdaten neu auszuhandeln. Auf diese Weise bekommt er die fehlenden Informationen nachgeliefert. Noch einfacher geht es, wenn er die Verbindung ganz unterbricht und sich dazwischenschaltet. Er gibt sich dann beiden Teilnehmern gegenüber als der jeweils andere aus und liest die Daten mit, während er sie weiterleitet. Ein Angriff, der auf einen Eingriff in die Datenübertragung oder den Protokollablauf beruht, nennt man aktiven Angriff. Er ist im Allgemeinen effektiver als ein passiver Angriff, kann aber auch leichter bemerkt werden.

Im Gegensatz zu einer Modemverbindung ist es sehr einfach, einen Ethernet–Strang oder ein Token–Ring–Netzwerk abzuhören. Frames werden auf diesen Medien in kleinen Blöcken übertragen; Blockanfang und -Ende sind klar definiert, und auszuhandelnde Verbindungsparameter gibt es nicht. Zum reihenfolgetreuen Zusammensetzen der einzelnen Blöcke werden natürlich Kenntnisse der Protokolle auf den höheren Schichten benötigt.

7.2.2 Integrität der Daten

Auf den unteren drei Schichten der TCP/IP Protokollhierarchie werden Daten gegen Übertragungsfehler geschützt, indem jeweils Prüfsummen mit übertragen werden. TCP und UDP sichern Daten und Schicht–III–Header, IP den Schicht–II–Header; je nach Protokoll und Übertragungstechnik werden auf Schicht I ganze Frames, nur Header oder auch überhaupt nichts gesichert.

Es ist möglich, aber unwahrscheinlich, dass Daten durch zufällige Fehler verfälscht wurden, die Prüfsumme aber dennoch korrekt ist. Die Algorithmen sind so gewählt, dass typische Fehler sich möglichst nicht gegenseitig aufheben. Gegen absichtliche Verfälschungen besteht jedoch kein Schutz; die Prüfsummenverfahren sind öffentlich bekannt, so dass ein Angreifer, der Zugriff auf die physikalische Verbindung oder einen der Router hat, die Daten verändern und die Prüfsumme neu berechnen kann.

7.2.3 Authentizität

Auf IP–Ebene ist es prinzipiell kein Problem, eine falsche Absender–IP–Nummer in die Datagramme einzutragen (IP Spoofing). Der Weitertransport erfolgt bei den gängigen Implementierungen ausschliesslich anhand der Ziel–IP–Nummer. Transport anhand Absender- und Zieladresse (und eventuell noch Protokollkennung, Portnummern und sonstiger Optionen) gibt es zwar für manche Betriebssysteme auch (sog. Policy–Based Routing), das ist aber aufwendig zu konfigurieren und kostet CPU–Zeit. Daher wird es selten eingesetzt.

Das blosse Fälschen der Absender–IP–Nummer bringt jedoch keinen direkten Gewinn, abgesehen davon, dass Angriffe schwerer rückverfolgbar sind. Voraussetzung dafür, dass man auf diese Weise wirklich an mehr Privilegien kommen kann, sind Dienste auf höherer Ebene, die bestimmten IP–Nummern mehr Privilegien einräumen als anderen, oder anders ausgedrückt, die eine Authentifizierung nach IP–Nummern vornehmen. Leider gibt es solche Dienste tatsächlich: Die Berkeley r–Tools und der Line Printer Daemon sind bekannte Beispiele.

Für den Angreifer taucht jedoch hier noch ein zweites Problem auf. Die oben genannten Dienste arbeiten mit TCP–Verbindungen, benötigen also alleine zum Aufbau einer Verbindung bereits Pakettransport in beide Richtungen. Ist aber die Absenderadresse des ersten Paketes gefälscht, so wird das Antwortpaket an die vermeintliche IP–Nummer des Kommunikationspartners geschickt. Für den Angreifer ist es jedoch essentiell wichtig, das Antwortpaket zu bekommen. Denn es enthält die initiale Sequenznummer der Rückrichtung, ohne deren Bestätigung keine Verbindung zustande kommen kann. Er kann das Paket nur abhören, wenn er entweder Zugriff auf einen Router hat, über den diese Antworten transportiert werden (also sich hinreichend nahe an dem anzugreifenden Rechner befindet und somit leicht zu entdecken ist), oder indem er sich eines Tricks bedient: Dem Source–Routing. Dies ist eine spezielle IP–Option, mit der der Absender eines Datagrammes explizit die IP–Nummern von Routern angeben kann, über die das Datagramm laufen soll. Die Antwort wird ebenfalls über diese Knoten geroutet und somit dem Angreifer zugestellt.

Die Source–Routing–Option dient der Fehlersuche in IP–Netzen und sollte eigentlich nicht für TCP–Datentransport benutzbar ein. Bei den meisten kommerziellen Routern kann sie abgeschaltet werden, so dass TCP–Pakete mit eingeschaltetem Source–Routing einfach verworfen werden. Leider machen noch nicht alle Organisationen an ihren Eingangsroutern von dieser Möglichkeit Gebrauch.

Falls Source Routing nicht möglich ist, z.B. weil es von einem Router nicht unterstützt wird oder im TCP/IP–Code des anzugreifenden Rechners nicht implementiert ist, kann durch ICMP–redirect–Meldungen dafür gesorgt werden, dass Antwortpakete an gefälschte IP–Nummern zum Absender zurückgelangen. Eine redirect–Meldung veranlasst einen Rechner oder Router dazu, eine Route in seiner Routingtabelle zu ändern. Daher sollten solche Meldungen nur von vertrauenswürdigen Routern akzeptiert werden, z.B. den Routern des Providers, was aber wegen der Fälschbarkeit von Absender–IP–Nummern schwierig zu realisieren ist.

Für den Fall, dass der Angreifer keine Möglichkeit hat, die Antworten zu erhalten, gibt es

noch einen weiteren Trick, die richtige Sequenznummer zu bekommen: Sequence Number Guessing [Bel89], Erraten der Sequenznummern. Manche Betriebssysteme vergeben ihre initialen TCP–Sequenznummern nach einem leicht nachvollziehbaren Schema, zum Beispiel in sequentieller Reihenfolge. In diesem Fall braucht er also lediglich vorher eine ”harmlose” TCP–Verbindung an den anzugreifenden Rechner zu eröffnen, zum Beispiel ein finger, und sich die verwendeten Sequenznummern zu merken. Wenn dieser Rechner zwischenzeitlich keine andere Verbindungsanforderung bedient hat, dann ist die initiale Sequenznummer der Angriffsverbindung bekannt, auch wenn das Antwortpaket nicht empfangen werden kann.

Bei Protokollen, die UDP verwenden, existiert für den Angreifer das Problem mit den Sequenznummern nicht (ausser, wenn eine höhere Schicht einen ähnlichen Mechanismus benutzt). Protokolle, bei denen ein einzelnes UDP–Paket von einer ”vertrauenswürdigen” IP–Nummer eine sicherheitsrelevante Aktion auslösen kann, sind somit prinzipiell zum Scheitern verurteilt.

Die Fälschbarkeit von IP–Nummern sollte dazu führen, dass IP–Nummern niemals zu Authentisierungszwecken benutzt werden sollen. Dienste, die bestimmten Benutzern oder Rechnern mehr Privilegien als anderen gewähren, sollten statt dessen auf einer höheren Ebene kryptographische Authentifizierungsverfahren benutzen. Zukünftige Erweiterungen der TCP / IP–Protokollfamilie sehen kryptographische Verfahren auch auf Schicht II und III vor (siehe Kap. 3.11). Standardisierung und Implementierung dieser Verfahren stehen jedoch in absehbarer Zeit noch nicht zur Verfügung.

7.2.4 Verlust der Verfügbarkeit von Rechnern

Die gängigen Implementierungen von TCP / IP bieten einige Ansatzpunkte für Angreifer, Rechner, Netze oder Router im Betrieb zu stören. Die meisten dieser Angriffe basieren auf der Ausnutzung von Fehlern in der Implementierung des TCP / IP–Systems, und es hängt vom System ab, ob eine bestimmte Sicherheitslücke vorhanden ist. Häufig zeigt sich jedoch, dass verschiedene Entwickler die gleichen Fehler machen, obwohl sie unabhängig voneinander arbeiten oder zumindest vorgeben, dies zu tun. Die nachfolgende Auflistung erhebt bei weitem keinen Anspruch auf Vollständigkeit, sondern liefert einige Beispiele für diese Form von Attacken.

Ein altbekannter Angriff sind sogenannte SYN-Floods. Dabei werden einem einzelnen Rechner innerhalb kurzer Zeit viele TCP-Pakete mit gesetzter SYN-Kennung geschickt; das sind Pakete, mit denen das Öffnen einer neuen TCP–Verbindung eingeleitet wird. Der Rechner beantwortet diese Pakete alle mit einem SYN ACK und merkt sich den Vorfall, damit später das erwartete ACK–Paket richtig zugeordnet werden kann (siehe Abschnitt 3.6). Erst wenn diese nach Ablauf einer festgelegten Zeitspanne immer noch nicht eingetroffen sind, werden die gespeicherten Informationen wieder gelöscht. Viele TCP / IP–Implementierungen haben jedoch Schwierigkeiten, wenn mehrere tausend Verbindungsanforderungen gespeichert werden müssen. Es können dann keine sinnvollen Verbindungen mehr aufgebaut werden, weil Puffer überlaufen, oder der Überlauf kann sogar zum Absturz führen, weil die-

ser Fall bei der Implementierung nicht bedacht oder getestet wurde. Tragen diese Pakete gefälschte Absender–IP–Nummern, so sind sie auch sehr schwer von ërnsthaften" Verbindungsanforderungen zu unterscheiden. SYN–Floods sind in der letzten Zeit wieder in die öffentliche Diskussion geraten [Tea96b], weil im Usenet ein Programm veröffentlicht wurde, das diesen Angriff tatsächlich ausführt.

Eine andere Angriffsvariante ist das Verschicken überlanger ping–Datenpakete [Tea96a]. Sie basiert auf einem Fehler in der Implementierung der IP–Schicht vieler Betriebssysteme. Bekommt ein Rechner ein Datagramm mit einer Länge von über 64 Kilobytes, dann läuft ein interner Puffer über; es können wichtige Variablen des Betriebssystems überschrieben und das System zum Absturz gebracht werden. Obwohl dies ein reiner Implementierungsfehler ist, sind viele verschiedene Systeme davon betroffen, nicht nur Rechnerbetriebssysteme, sondern auch Router, Netzwerkdrucker und ähnliches. Diese Lücke existiert schon seit Jahrzehnten. Da IP–Datagramme über 64 Kilobytes verboten sind und von den meisten Betriebssystemen gar nicht verschickt werden können, ist sie aber lange Zeit nicht aufgefallen. Erst die Implementierungen des ping–Programmes unter Windows 95 und Windows NT ermöglichten ein Verschicken solch langer Pakete.

Das Verschicken von irreführenden Fehlermeldungen [Bel89] bezüglich der Erreichbarkeit von Rechnern ist eine weitere Angriffsvariante auf die Verfügbarkeit von Datenverarbeitungseinrichtungen. Mit einer speziellen ICMP–Meldung, host unreachable, zeigt ein IP–Router an, dass er für IP–Datagramme an eine bestimmte IP–Nummer keine Route kennt. Daraufhin löscht der absendende Rechner die entsprechende Route ebenfalls aus seiner Routingtabelle. Durch gezieltes Verschicken solcher Meldungen können fehlerhafte Informationen in Routingtabellen erzeugt werden, die zur Unerreichbarkeit anderer Rechner oder Netze führen können.

7.3 Telnet

Steckbrief

Schwachstellen Client:

- Abhörbarkeit des Datenverkehrs
- Abhörbarkeit der Passwörter

Schwachstellen Server:

- hohe Privilegien, daher Implementierungsfehler fatal
- Erratbarkeit von Passwörtern

7.3.1 Schwachstellen des Telnet–Clients

Daten, die bei einer Telnet–Sitzung eingetippt werden oder auf dem Bildschirm erscheinen, gehen unverschlüsselt über die TCP–Verbindung. Insbesondere wird auch das Passwort

beim Einloggen unverschlüsselt übertragen. Jeder, der Zugriff auf die physikalische Leitung oder einen der Router hat, kann es mit wenig Mühe auslesen und dann selber damit einloggen. Aus diesem Grund sollte Telnet, wenn überhaupt, nur innerhalb eines gesicherten Bereiches, in dem sich Benutzer und Administratoren gegenseitig vertrauen, benutzt werden. Telnet von aussen auf die Rechner der eigenen Organisation sollte nicht zugelassen werden. Leider ist die Realität davon weit entfernt. Das liegt auch daran, dass die Alternativen teilweise umständlich zu bedienen und nicht für alle Plattformen verfügbar sind.

7.3.2 Schwachstellen des Telnet–Servers

Der Telnet–Server auf Rechner A läuft mit relativ hohen Privilegien. Dies ist nötig, da er nach erfolgreicher Authentifizierung auf die Rechte umschalten kann, die der einloggende Benutzer besitzt. Der telnetd muss also mindestens die Vereinigung der Rechte besitzen, die alle von aussen erreichbaren Benutzeraccounts besitzen.

Unter Unix läuft der telnetd unter der Benutzerkennung des Superusers, root. Nur dieser Benutzer hat die Möglichkeit, auf jede andere Benutzerkennung umzuschalten. Probleme ergeben sich, wenn der telnetd Programmierfehler hat. Denn nichts hindert ihn daran, beliebige Teile des Dateisystems oder des Speichers auszulesen oder zu verändern oder andere Programme auszuführen, die dann ebenfalls mit sehr vielen Rechten ablaufen.

Auch ohne ein Passwort abgefangen zu haben, kann man mit Telnet unter Umständen unerlaubten Zugriff auf fremde Rechner bekommen: durch das Raten von Passwörtern. Die Benutzernamen sind in der Regel öffentlich bekannt; sie kommen in Mail–Adressen vor, werden in die News gepostet oder auf WWW–Seiten veröffentlicht. Mit etwas Phantasie kann man Passwörter häufig erraten. Viele Benutzer verwenden Namen von Freunden, geographische Bezeichnungen oder sonstige Begriffe, die sie sich leicht merken können. Wörterbücher mit gängigen Passwörtern sind frei verfügbar, Programme zum Durchtesten über eine Telnet–Verbindung ebenfalls.

Abhilfe können die Benutzer selber schaffen, indem sie Passwörter verwenden, die nicht so leicht zu erraten sind. Das Bewusstsein dafür ist allerdings kaum vorhanden. Besser ist es, wenn der Systemadministrator Software installiert, die darauf achtet, dass die Benutzer nur Passwörter wählen können, die hinreichend ”kryptisch” sind, zum Beispiel weil sie Ziffern, gemischt Gross-/Kleinschreibung oder Sonderzeichen enthalten. Ausserdem sollte der telnetd die Eigenschaft haben, bei Fehleingaben von Benutzername oder Passwort etwas Zeit verstreichen zu lassen, bevor ein neuer Login–Versuch gemacht werden kann. Zusätzlich sollte die maximale Zahl der gleichzeitig einkommenden Telnet–Verbindungen limitiert werden. Software, die dies leistet, ist für die meisten Unix–Varianten verfügbar, für andere Systeme allerdings weniger.

7.4 File Transfer Protocol (FTP)

7.4.1 Steckbrief

Schwachstellen Client:

- Abhörbarkeit der übertragenen Dateien
- Abhörbarkeit der Passwörter

Schwachstellen Server:

- hohe Privilegien, daher Implementierungsfehler fatal
- Erratbarkeit von Passwörtern

7.4.2 Schwachstellen des FTP–Clients

Ähnlich wie bei Telnet werden auch bei FTP die Nutzdaten unverschlüsselt übertragen. Dies gilt sowohl für die Authentifizierungsinformation, wie auch für die Kommandos und die übertragenen Dateien. Jeder, der physischen Zugriff auf die Router oder die Netzwerkleitungen hat, ist also in der Lage, den Datenstrom abzuhören.

Wird FTP zur Übertragung von Daten von einem Benutzer–Account eines mehrbenutzerfähigen Betriebssystems verwendet, dann ist das Passwort meist das selbe wie für den Zugang mit Telnet. Wird das Passwort auf dem Netz abgelauscht, kann sich der Angreifer also meist in das System einloggen und dort Kommandos ausführen.

7.4.3 Schwachstellen des FTP–Servers

Die FTP–Serversoftware hat die selben Schwachstellen die ein Telnet–Server. Da ein FTP–Server Zugriff auf zahlreiche Daten verschiedener Benutzerkennungen haben muss, läuft er auf Unix–Systemen unter der Kennung des Superusers, root. Daher ist bei Implementierungsfehlern das System besonders gefährdet: Gelingt es einem Angreifer, durch Ausnutzung eines Programmierfehlers im FTP–Server ein Kommando auszuführen, so geschieht dies mit den höchsten Privilegien.

Bei nicht–anonymem FTP gilt bezüglich der Wahl der Passwörter das selbe wie bei Telnet: sie sollen schwer zu erraten sein und daher nicht aus Wörtern bestehen, die in Wörterbüchern vorkommen.

7.5 E–Mail

7.5.1 Steckbrief

Schwachstellen E–Mail allgemein:

- Abhörbarkeit des Datenverkehrs
- Mailbomben

Schwachstellen SMTP:

- Fälschbarkeit der Absenderadressen
- Implementierungsfehler der Server

Schwachstellen POP/IMAP:

- Abhörbarkeit der Passwörter
- Implementierungsfehler der Server

7.5.2 Schwachstellen von E–Mail

Die zur Übertragung von E–Mail im Internet benutzten Protokolle SMTP, POP und IMAP sehen keine Verschlüsselung vor. Damit sind Verletzungen der Vertraulichkeit und der Integrität einer Mail durch jeden möglich, der Zugriff auf das physikalische Übertragungsmedium oder einen Netzknoten besitzt. Dieses Problem lässt sich durch den Einsatz kryptographischer Verfahren auf höherer Ebene lösen. Implementierungen dazu gibt es (siehe Kap. 11), jedoch sind diese teilweise komplex in der Handhabung, und es gibt ungeklärte Rechtsfragen.

Unter einer Mailbombe versteht man eine sehr grosse Mail, deren Zweck es ist, die Leitungen auf der Empfängerseite zu blockieren oder die Mailplatte des Empfängers zum Überlaufen zu bringen und somit die Verfügbarkeit des E–Mail–Dienstes oder eines ganzen Rechners auf der Empfängerseite zu beeinträchtigen. Manche MTA's bieten einen Schutz vor solchen Mails, indem sie Mails oberhalb einer bestimmten Maximalgrösse nicht annehmen. Jedoch ist das Ziel einer Mailbombe auch mit vielen kleinen Mails zu erreichen.

7.5.3 Schwachstellen von SMTP

Ein grosses Problem ist die Fälschbarkeit der Absenderadresse. Die Mailadresse des Absenders lässt sich, analog zu Absender–IP–Nummern, beliebig fälschen; sie hat keine Funktion für den Mail–Transport. Manche MTA's protokollieren die IP–Nummer des Rechners, von dem eine Mail eingeliefert wurde, die meisten tun dies jedoch nicht. Damit ist Mail nicht nur ohne kryptographische Absicherung für verbindliche Absprachen oder geschäftliche Transaktionen ungeeignet, sondern birgt auch die Gefahr, dass die beiden nachfolgend beschriebenen Angriffsvarianten völlig anonym durchgeführt werden können.

Die gängigen Implementierungen der Mail Transport Agents sind sehr komplizierte und umfangreiche Programme. Daher werden relativ häufig Programmierfehler gefunden, die sicherheitskritische Folgen haben. Beispielsweise ist es mit älteren Versionen von Sendmail unter Unix möglich, per Mail ferngesteuert Programme auszuführen. Man musste einfach die Empfängeradresse mit einem „—“ beginnen lassen; der nachfolgende Text wurde dann

als Unix–Kommando interpretiert und ausgeführt. Speziell bei Sendmail stehen meist sofort nach Bekanntwerden eines Fehlers neue Versionen zur Verfügung, bei denen das Problem beseitigt ist. Allerdings installieren nicht alle Administratoren die neuen Programmversionen sofort, so dass man noch lange Zeit verwundbare Mailserver finden kann.

7.5.4 Schwachstellen von POP und IMAP

Die Protokolle POP und IMAP ermöglichen das Abholen bzw. Manipulieren der Mailbox eines Benutzers. Dazu authentisiert sich der Benutzer mit einem Passwort. Dieses Passwort wird, wie bei Telnet oder FTP, unverschlüsselt übertragen. Ebenso werden die Mails unverschlüsselt an den Client übertragen.

7.6 Das World Wide Web

Steckbrief

- Dokumente können bei der Übertragung zum Benutzer eingesehen und verändert werden.
- Parameter des Benutzers können bei der Übertragung zum Server eingesehen und manipuliert werden.
- Es können unbemerkt Dateien mit gefährlichem Inhalt (z.B. Viren) geladen werden.
- Java–Applets können durch Konzeptions-. oder Implementierungsfehler der Browser Sicherheitslücken ausnutzen.
- Mit Javascript–Programmen kann der Autor von HTML–Seiten den Browser fernsteuern.
- ActiveX–Controls werden von geeigneten Browsern ohne jegliche Sicherheitsvorkehrungen ausgeführt.
- Schwachstellen im HTTP–Server oder in CGI–Programmen können dazu genutzt werden, Kommandos von aussen auf dem Server–Rechner auszuführen.

7.6.1 Schwachstellen von WWW–Clients

Durch das reine Herunterladen von öffentlich verfügbaren Dokumenten kann für den Betreiber des WWW–Servers kaum ein Schaden entstehen. Daher bietet das WWW vergleichsweise wenig Sicherheitsprobleme. Ganz ungefährlich ist es dennoch nicht.

Durch die Übergabe von Parametern beim Abruf eines Dokumentes hat der Benutzer die Möglichkeit, mit dem Server zu interagieren. Dies geschieht in den meisten Fällen durch Ausfüllen und Absenden eines HTML–Formulares. Die Information wird dabei unverschlüsselt übertragen; manche WWW–Browser ermahnen den Benutzer daher vor dem

Durchführen einer solchen Transaktion, keine sensitiven Daten wie Passwörter oder Kreditkartennummern zu übertragen. Erweiterungen von HTTP (z.B. S–HTTP) oder für das WWW geeignete Verschlüsselungsprotokolle auf niedriger Netzwerkebene (z.B. Secure Socket Layer, SSL) beheben dieses Problem teilweise.

7.6.2 Java und Javascript

Eine jüngere Entwicklung im WWW–Bereich ist ein Dokumentenformat, das nicht Text oder Grafik enthält, sondern Programmcode, der vom Browser ausgeführt wird. Die bekannteste Programmiersprache dieser Gattung ist Java von Sun Microsystems. Die Entwickler von Java waren sich durchaus dessen bewusst, dass es eine grosse Sicherheitslücke sein kann, Programme automatisch auszuführen, die über unsichere Quellen aus dem Netz bezogen wird. Daher ist Java so konzipiert, dass solche Programme in einer abgeschotteten Umgebung ablaufen und keinen Schaden am System anrichten können. Zum einen ist dies jedoch nur teilweise gelungen, zum anderen sind auch bei der Implementierung der gängigen Java–Interpreter Fehler gemacht worden, so dass Java derzeit alles andere als sicher ist. Die Ausführung von Java–Netz–Programmen, sog. Applets, kann bei den gängigen Web–Clients unterbunden werden. Da jedoch immer mehr Dienstangebote im WWW einen Java–Interpreter voraussetzen, ist dies keine zufriedenstellende Lösung.

Eine (nicht standardisierte) Eigenschaft zweier kommerzieller Web–Browser, dem Netscape Navigator und dem Explorer von Microsoft, ist die Möglichkeit, Programmcode ausführen zu können, der direkt im HTML–Dokument steht. Die Programmiersprache wird Javascript genannt, was eine irreführende Bezeichnung ist, denn abgesehen von der Idee des mobilen Codes hat diese Sprache nichts mit Java zu tun. Javascript beinhaltet nicht nur viele Sicherheitslücken (u.a. die Möglichkeit, unbemerkt beliebige Dateien des lokalen Rechners per Mail an eine angegebene Adresse zu schicken), sondern auch datenschutzrechtlich umstrittene Funktionen wie das Aufzeichnen der Mausbewegungen eines Benutzers. Bei neueren Versionen der genannten Clients lässt sich diese Funktion abschalten.

7.6.3 Helper Applications, PlugIns und ActiveX

Dokumente, deren Datentyp dem Browser selbst unbekannt sind, können an Helper Applications übergeben werden. Die Flexibilität, dadurch beliebige Dokumententypen verarbeiten zu können, birgt jedoch grosse Gefahren für die Sicherheit des Systems. So ist es leicht einzusehen, dass Dokumente des Typs application / sh nicht wirklich von einer Shell abgearbeitet werden sollen. Dies würde einem Angreifer ermöglichen, beliebige Shell–Programme auf dem Rechner des Benutzers ausführen zu können. Weniger bekannt ist dagegen die Tatsache, dass auch Postscript–Dokumente Kommandos enthalten können, die bei der Interpretation durch einen Postscript–Previewer tatsächlich auf dem Rechner ausgeführt werden. Auch der bekannte Word–Makrovirus [WDR96], bei dem ein Textdokument ein schadhaftes Programm enthielt, das in der Makro–Programmiersprache von MS Word geschrieben war, fällt unter diese Kategorie.

Natürlich können Dokumente oder Programme auch über Web–Browser geladen und manuell mit einem geeigneten Programm weiterverarbeitet werden. Dies ist technisch kaum zu verhindern und kann nur durch eine geeignete Sicherheitspolitik reglementiert werden. Dennoch ergibt sich durch den Einsatz von Helper–Applications für „gefährliche“ Dokumententypen eine neue Dimension der Gefahr, denn der Schaden kann völlig automatisch herbeigeführt werden und auch für einen erfahrenen Benutzer völlig unbemerkt ablaufen. Eine Helper–Application sollte daher nur installiert werden, wenn sichergestellt ist, dass dieses Programm keine ungewollten Aktionen durchführt, selbst wenn das zu verarbeitende Dokument defekt ist oder sogar böswillige Teile enthält.

Das Gefahrenpotential von PlugIns ist ähnlich dem der Helper–Applications. Die PlugIns haben die selben Rechte wie der Browser selbst, kann also im Normalfall auf sämtliche Dateien des Benutzers zugreifen und beliebige weitere Programme ausführen. Das Fehlerpotential mag geringer sein als bei den Helper–Applications, da PlugIns speziell für die WWW–Nutzung entwickelt wurden. Dafür ist bei den meisten angebotenen PlugIns der Sourcecode und die genaue Spezifikation nicht bekannt. Auch können PlugIns nicht wie externe Programme eingehend getestet werden, bevor sie auf WWW–Dokumente angewendet werden.

Das vermeintliche Gefahrenpotential eines PlugIns hängt nicht zuletzt auch davon ab, welche Art von Dokumenten es verarbeiten soll. Je mächtiger die Dokumentensprache ist, desto eher kann das Plug–In als gefährlich eingestuft werden. PlugIns für die Darstellung bewegte Bilder oder Musikstücke sind sicherlich ungefährlicher als IRC–Clients oder gar das Tcl / Tk–PlugIn von Sun Microsystems.

Noch schlechtere Sicherheitseigenschaften hat ActiveX, das nur im Microsoft Internet Explorer implementiert ist. Im Gegensatz zu Java–Applets läuft ein ActiveX–Control nicht in einer eingeschränkten Umgebung ab, sondern besitzt die selben Rechte wie der Web–Browser. Unter Betriebssystemen wie Windows 95, bei denen es kein Schutzkonzept gibt, kann ein ActiveX–Control also alle Funktionen des Rechners steuern: Beliebige Netzwerkverbindungen öffnen, sämtliche Dateien lesen oder verändern oder den Rechner herunterfahren. So ist es beispielsweise findigen Programmieren gelungen, mittels eines ActiveX–Controls der ebenfalls auf dem Rechner installierten Online–Banking–Software eine Überweisung unterzujubeln. Diese Überweisung wurde auch tatsächlich ausgeführt, ohne dass der Benutzer es bemerkte [Don97, Plu97].

Das Ausführen von ActiveX–Controls sollte, falls überhaupt, nur auf einem Rechner möglich sein, der keine wichtigen Daten enthält und keine Netzwerkverbindungen öffnen kann. Da es jedoch abgesehen von Grafikspielereien kaum interessante ActiveX–Controls gibt, dürfte der Verlust zu verschmerzen sein.

7.6.4 Schwachstellen von WWW–Servern

Viele HTTP–Server erlauben die Angabe von URL’s, die nicht direkt ein Dokument zurückliefern, sondern ein Programm auf dem Server ausführen. Zum Beispiel sind die oben genannten URL’s mit Parametern meist von dieser Art. Die Kommunikation zwischen

dem Server und einem solchen Programm erfolgt nach dem Common Gateway Interface (CGI) Standard, weshalb die Programme CGI–Programme oder CGI–Skripten genannt werden. Ein CGI–Programm dient dazu, die Parameter auszuwerten und in Abhängigkeit ein neues Dokument zu erzeugen, das an den Client übertragen wird. Es kann jedoch auch Nebeneffekte haben, beispielsweise die Ablage der Parameter in einer Datenbank oder das Absenden einer Mail an den Server–Verwalter. Die Sicherheitsproblematik besteht hier bei der Programmierung der CGI–Programme. Da der Benutzer durch die Wahl der Parameter Einfluss auf die Programmausführung hat, muss das Programm alle möglichen Fälle berücksichtigen, um sich auch bei unsinnigen oder fehlerhaften Parametern noch definiert zu verhalten. Dies wird häufig bei der CGI–Programmierung vernachlässigt. Einfache CGI–Programme lassen sich mit geeigneten Skriptsprachen schnell und kurz erstellen, so dass Sicherheitsaspekte dabei leicht zu kurz kommen. Manche HTTP–Server wurden sogar zeitweise mit CGI–Programmen ausgeliefert, die Sicherheitslücken enthielten.

7.7 Die Berkeley r–tools

Steckbrief

- Übertragung der Nutzdaten im Klartext
- Übertragung eventueller Passwörter im Klartext
- Authentifizierung anhand IP–Nummer bzw. Domainnamen sowie Benutzernamen

Analog zu Telnet und FTP werden auch bei den r–Tools die Nutzdaten im Klartext über das Netz übertragen. Daher kann die Kommunikation von Personen mit physischem Zugriff auf die Leitungen oder die Netzwerkknoten abgehört werden.

Die Berkeley r–Tools bieten neben dem Authentifikationsmechanismus über ein Passwort auch die Möglichkeit, eine Authentifikation über die IP–Nummer bzw. den Domainnamen des Client–Rechners, sowie über den Benutzernamen auf dem Client–Rechner durchzuführen. Dieser Schutz lässt sich durch DNS–Spoofing (siehe Abschnitt 7.8) oder einer Kombination aus IP–Spoofing und Routing–Attacke oder Sequence Number Guessing (Abschnitt 7.2) umgehen. Sie ist ohne zusätzliche Massnahmen als sehr unsicher einzustufen.

Bei dieser Art der Authentifizierung wird nicht nur den IP–Nummern (und damit dem Netzwerk) vertraut, sondern ausserdem noch den Benutzernamen auf dem Client–Rechner. Daher ist sie nur sinnvoll, wenn der Server dem Client–Rechner vertrauen kann, d.h. wenn auf dem Client–Rechner ein Mehrbenutzerbetriebssystem läuft und das es von einer vertrauenswürdigen Person administriert wird. Auf einem Einbenutzersystem kann sich dagegen jeder Benutzer einen beliebigen Namen zuordnen und somit den Schutz umgehen.

Jeder Benutzer des Server–Rechners kann für sich selber einstellen, welchen Rechnern (IP–Nummern und Benutzernamen) er vertraut. Dies kann der Systemadministrator nur unterbinden, indem er den Sourcecode der r–Tools verändert.

7.7.1 Sicherheitsprobleme von rlogin

rlogin verfügt ähnlich wie Telnet über die Möglichkeit, den Benutzer mittels Passwort zu authentifizieren. Dieses Passwort wird im Klartext über das Netz übertragen, kann also abgehört werden.

7.7.2 Sicherheitsprobleme von rsh und rcp

rsh und rcp funktionieren ausschliesslich ohne Passwort, also mit Authentifikation anhand der IP–Nummer bzw. des FQDN. Die Benutzung dieser Dienste setzt also ein gewisses Vertrauen in das Netzwerk und die zugelassenen Client–Rechner voraus.

7.8 Das Domain Name System (DNS)

Steckbrief

- Zuordnungen zwischen IP–Nummern und Rechnernamen fälschbar
- DNS–Server vertrauen unverlangten Informationen aus unzuverlässigen Quellen
- DNS–Server laufen mit hohen Privilegien, sind daher anfällig für Implementierungsfehler

7.8.1 Prinzipielle Sicherheitsprobleme

Wie in Abschnitt 6.7 beschrieben, erlauben (primäre und sekundäre) DNS–Server die Zuordnung eines Rechnernamens zu einer IP–Nummer und umgekehrt. Dabei findet keine Überprüfung statt, ob diese Zuordnung tatsächlich legal ist. In beiden Richtungen können absichtlich Schäden durch falsch eingetragene Werte herbeigeführt werden. Dieses Angriffsverfahren wird DNS–Spoofing genannt.

Angenommen, eine Organisation hat einen begehrten und einprägsamen Domainnamen (z.B. super.ch) organisiert. Dann ist zu erwarten, dass HTTP–Verbindungen auf den Host www.super.ch aufgebaut werden. Wird nun der DNS–Eintrag von www.super.ch auf einen WWW–Server einer ganz anderen Organisation umgelenkt, dann kann sich diese zwar über zusätzliche Serverzugriffe freuen, bekommt jedoch zusätzlichen Datenverkehr, den sie je nach Vertragsgestaltung mit ihrem Internet–Provider auch zusätzlich bezahlen darf.

Der umgekehrte Fall kann noch schädlicher werden. Angenommen, eine Firma, die den Domainnamen firma.com für ihr internes Netz benutzt, setzt die in Abschnitt 6.6 vorgestellten Berkeley r–Tools ein und erlaubt die Nutzung innerhalb der Domain firma.com. Dann kann ein Angreifer die Dienste ebenfalls nutzen, indem er in seinem DNS–Server der von ihm benutzten IP–Nummer den Domain–Name host1.firma.com zuordnet.

DNS–Spoofing lässt sich im Prinzip auf einfache Weise nachweisen: Wenn, wie im letzten Beispiel, ein Hacker seine eigene IP–Nummer im DNS auf den Rechnernamen

host1.firma.com zeigen lässt, dann kann durch eine umgekehrte Abfrage nach der IP–Nummer von host1.firma.com festgestellt werden, dass die Zuordnung gefälscht ist. Diese Anfrage wird nämlich vom lokalen Nameserver der entsprechenden Firma beantwortet; der Rechnername existiert entweder gar nicht, oder die IP–Nummer zeigt zumindest auf einen Rechner aus dem internen Firmennetz, nicht jedoch auf den Rechner des Angreifers. Es ist jedoch nicht unproblematisch, zu jeder DNS–Abfrage auch die umgekehrte Abfrage zu machen. Erstens erzeugt dies die doppelte Netzlast, zweitens sind durch Konfigurationsfehler in vielen DNS–Servern gar nicht beide Abfragerichtungen möglich. Benutzer, deren Provider seinen DNS–Server nicht richtig konfiguriert hat, bliebe somit der Zugriff zu vielen öffentlich verfügbaren Internet–Diensten verwehrt.

7.8.2 Implementierungsprobleme

Die mit Abstand am häufigsten im Internet eingesetzte DNS–Serversoftware ist bind (Berkeley Internet Name Daemon). bind zeichnet sich durch eine hohe Geschwindigkeit und Zuverlässigkeit aus. Jedoch werden in regelmässigen Abständen Sicherheitsprobleme in aktuellen bind–Versionen gefunden.

Erst vor kurzem wurde ein altes Problem in der Implementierung des bind entdeckt [Tea97]: wird er benutzt, um eine Zuordnung Hostname–zu–IP–Nummer herauszufinden, dann wertet er nach und nach die Informationen aus, von den root–Nameservern bis zu den zuständigen Nameservern für die genannte Domain bekommt. Dabei können die befragten Nameserver auch Informationen schicken, nach denen überhaupt nicht gefragt wurde. Ältere bind–Versionen trauen diesen ungefragten Informationen und legen sie sogar im lokalen Cache ab. Wird später eine Abfrage nach ihnen gemacht, dann wird sie lokal aus dem Cache beantwortet. Neuere Versionen von bind haben dieses Problem nicht mehr. Sie trauen grundsätzlich keinen unverlangt zugesandten Informationen mehr.

Weitere Implementierungsprobleme in bind beziehen sich darauf, dass die Software unter der Benutzerkennung root läuft. Dies ist nötig, damit sie UDP–Pakete von der privilegierten Absender–Portnummer 53 aus absenden kann. Es macht den Daemon jedoch besonders anfällig gegenüber Implementierungsfehler. In jüngster Zeit wurde von mehreren solchen Fehlern berichtet, durch die ein Angreifer Kommandos auf einem System ausführen oder die Maschine lahmlegen kann [Tea98].

7.9 Sendfile

Steckbrief

- Übertragung der Daten möglicherweise im Klartext
- Verfälschen der Absenderadresse
- Überfluten mit Nachrichten

- Implementierungsprobleme

7.9.1 Sicherheitsprobleme von Sendfile

Die SAFT–Implementierung Sendfile unter Unix beherrscht das Versenden von Dateien, die mit dem Public–Key–Kryptosystem PGP verschlüsselt oder signiert wurden. Der Einsatz setzt jedoch die Installation von PGP auf Sender- und Empfängerseite sowie ein funktionierendes Schlüsselmanagement voraus (siehe Abschnitt 11.4.1). Wird auf PGP–Unterstützung verzichtet, dann werden die Daten bei der Übertragung mit Sendfile offen übertragen, können also an den Leitungen und Netzwerkknoten abgehört werden.

Der Benutzername des Absenders wird im SAFT–Protokoll mit übertragen. Dies macht der Sendfile–Client automatisch. Durch Verändern des Source–Codes ist es jedoch möglich, einen falschen Benutzernamen einzutragen. Der Empfänger hat keine Möglichkeit, die Echtheit des Benutzernamens zu überprüfen.

Ähnlich wie bei E–Mail ist es möglich, einem Benutzer sehr grosse und / oder viele Dateien zu schicken. Der Sendfile–Daemon verweigert zwar automatisch beim Erknappen des Plattenplatzes den Empfang weiterer Dateien. Somit kann die Platte nicht zum Überlauf gebracht werden. Allerdings können danach auch keine sinnvollen Dateien mehr empfangen werden.

Sendfile ist ein relativ neuer Internet–Dienst. Zwar arbeiten etwa ein Dutzend Leute an der Programmierung und überprüfen dabei auch die Sicherheitsfunktionen. Jedoch ist Sendfile lange nicht so weit verbreitet und so gut getestet wie zum Beispiel der E–Mail–Server Sendmail, der News–Server INN oder der DNS–Server bind. Der Sendfile–Daemon läuft unter der Benutzerkennung root, Implementierungsfehler können also leicht sicherheitsrelevant sein.

7.10 Internet Relay Chat

Steckbrief

- Authentizität der Kommunikationspartner nicht gesichert
- IRC–Clients sind anfällig gegenüber Angriffen durch Protokollverletzungen
- Beim Dateitransport können vorhandene Dateien überschrieben werden

7.10.1 Sicherheitsprobleme des IRC

Das IRC–Protokoll bietet keinen verlässlichen Mechanismus zur Authentifizierung der Teilnehmer. Der Benutzername wird vom Client des jeweiligen Benutzers übertragen. Bei IRC–Clients auf Einbenutzersystemen ist er meist beliebig einstellbar, bei Mehrbenutzersystemen

durch eine kleine Manipulation am Sourcecode des Client ebenfalls. Die IRC–Server versuchen zwar eine Verbindung zum Identifikations–Server nach RFC1413 [Weidner, 1997 #63; Johns, 1993 #62] aufzubauen. Dessen Antwort ist jedoch nur vertrauenswürdig, wenn der Rechner des Clients unter einem Mehrbenutzerbetriebssystem läuft und der Administrator vertrauenswürdig ist. Maschinen, auf denen kein Identifikations–Server läuft, werden i.d.R. nicht vom IRC–Dienst ausgesperrt.

Das IRC basiert auf einem recht komplexen Protokoll. Einen Client dafür zu programmieren, der sich in allen Zuständen korrekt verhält, ist alles andere als trivial. Die gängigen Clients lassen sich durch gezielte Protokollverletzung zu unerwünschten Reaktionen bewegen. Meist ist nur ein Programmabsturz die Folge, im schlimmsten Fall können jedoch auch Dateien gelesen oder zerstört werden.

Das IRC bietet neben der reinen textorientierten Kommunikation auch die Möglichkeit, Dateien zu versenden und zu empfangen. Auf diese Weise können sich z.B. Benutzer, die sich noch nie persönlich begegnet sind, auf einfache Weise ein Bild von sich zukommen lassen. Problematisch ist allerdings, dass nur der Sender den Dateinamen festlegen kann; wählt er einen Dateinamen, der auf dem Zielrechner im Home–Verzeichnis des Benutzers bereits vorhanden ist, dann kann er die Datei überschreiben, sobald der Empfänger sein Einverständnis für den Datentransport gegeben hat. Erschwerend kommt hinzu, dass bei den gängigen IRC–Clients der Name der Datei nicht angezeigt, sondern die Datei direkt geschrieben wird.

7.11 Net News

Steckbrief

- Authentizität und Integrität der News–Artikel nicht gesichert
- Artikel können Programmcode enthalten, der direkt auf dem Client–Rechner ausgeführt wird
- IP–Nummer / DNS–basierte Zugangsrestriktion
- Passwortübertragung im Klartext bei Benutzerauthentifizierung

7.11.1 Sicherheitsprobleme des Usenet

Ähnlich wie bei E–Mails können bei News–Artikeln die Namen und E–Mail–Adressen der Absender leicht gefälscht werden. Die gängige Serversoftware nimmt keine Überprüfung vor. Auf diese Weise lassen sich sehr leicht bestimmte Personen in Misskredit bringen. Unabhängig davon sollte den aus den News gewonnen Informationen prinzipiell ein gewisses Misstrauen entgegengebracht werden. In den Diskussionsgruppen gibt es viele Experten, aber dennoch auch häufig Falschinformationen.

Die gängige News–Serversoftware erlaubt es, den Zugriff auf ihre Server auf bestimmte Rechner einzuschränken. Die Rechner werden anhand ihrer IP–Nummer oder ihres DNS–Namens identifiziert. Dieser Mechanismus kann mittels DNS–Spoofing [Wei97b, S. 58f] oder einer Kombination aus IP–Spoofing und Routing–Tricks oder Sequence Number Guessing [Wei97b, S. 44f] umgangen werden. Für sicherheitsrelevante Diskussionen in geschlossenen Benutzergruppen sollten daher zusätzliche Sicherheitsmassnahmen eingeleitet werden.

Viele News–Server erlauben ausserdem, einzelnen Personen den Zugriff freizuschalten. Die Zuordnung geschieht anhand der Benutzername / Passwort–Kombination. Beide Angaben werden hierbei im Klartext übertragen. Sie können also von Personen mit Zugriff auf die physischen Leitungen oder die Netzwerkkomponenten abgehört werden. Die Datei mit den Passwörtern wird darüber hinaus auf dem Server–Rechner im Klartext abgelegt.

Kapitel 8

Massnahmen gegen die Bedrohungen

Nach der Beschreibung möglicher Bedrohungen und Schwachstellen des Internet und der Internet–Dienste im letzten Kapitel führt dieser Teil auf einer konzeptuellen Ebene in die Gegenmassnahmen ein, die getroffen werden können. Nicht alle der möglichen Massnahmen sind auch sinnvoll; die Massnahmen verursachen einen gewissen Aufwand, der im Einzelfall gegen den zu erwartenden Nutzen abzuwägen ist.

Neben der rein wirtschaftlichen Sicht auf die Massnahmen spielen arbeitspsychologische Aspekte ebenfalls eine grosse Rolle bei der Auswahl von Sicherheitsmassnahmen. Sicherheit ist auch eine Vertrauensfrage; vertraue ich einem Mitarbeiter zu wenig, fühlt sich dieser gegängelt und neigt evtl. eher zum Missbrauch eines Systems; ist er auf der anderen Seite völlig frei, kann er das System ebenfalls leicht durch Unerfahrenheit oder Unwissenheit missbrauchen.

Diese Aspekte sollen im Folgenden aufgegriffen werden und mit organisatorischen, personellen und technischen Massnahmenansätzen belegt werden. Welche Massnahmen ausgewählt werden hängt mit dem Wert der Information, dem gewählten Dienst, der Dienstnutzung, den Bedrohungen und der Unternehmenskultur zusammen.

8.1 Massnahmendisposition

Die Disposition von Massnahmen ist ein wichtiges Element, um geeignete Massnahmen identifizieren zu können. Ein Massnahmendispositiv kann als Framework betrachtet werden, das gewährleistet, dass bei der Auswahl von Massnahmen an alle Aspekte gedacht wird. Es verhindert somit eine zu frühe Fokussierung auf eine Massnahme. Häufig finden sich Aussagen wie „unser Netz wird durch einen Firewall gesichert“ oder ähnliches. Firewalls sind im Bereich Netzwerksicherheit sicherlich eine wichtige technische Massnahme, aber eben nicht die einzige.

Eine Möglichkeit der Massnahmendisposition basiert auf der stufenweisen Minimierung des Risikos bis zu einem Restrisiko (vgl. Abbildung 8.1). Grundsätzlich gibt es mehrere Möglichkeiten, Risiken zu minimieren (vgl. [Mau98, S. 33f] und [SB92, S. 30f]):

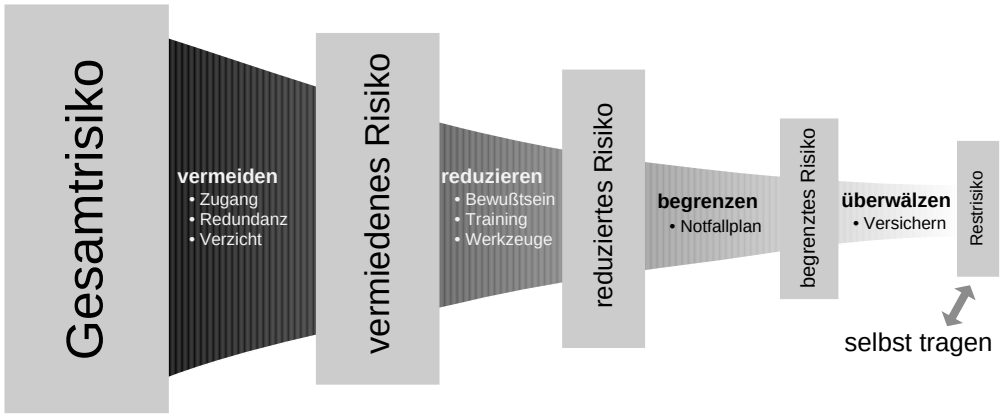

Abbildung 8.1: Stufenweise Minimierung des Gesamtrisiko

Vermeidung: Ein Dienst wird nicht eingesetzt oder durch einen alternativen aber sicheren Dienst ersetzt.

Reduzierung: Trotz eines Risikos bei der Nutzung eines Dienstes kann das Risiko reduziert werden, indem zusätzliche Sicherheitsvorkehrungen getroffen werden.

Begrenzung: Im Schadensfall ist eine Begrenzung des Schadens erforderlich (z.B. durch Datensicherung, Ermittlung des Verursachers, etc.).

Überwälzen: Schliesslich kann ein Risiko auch versichert werden.

Die Reihenfolge in der Massnahmenanordnung muss nicht zwingend so durchlaufen werden, dennoch sind im Normalfall die angegebenen Schritte in dieser Reihenfolge sinnvoll. Die Vermeidung eines Risikos ist grundsätzlich der Reduzierung des Risikos vorzuziehen, wobei allerdings funktionale Anforderungen oder die Unternehmenskultur angemessen berücksichtigt werden müssen. Eine wesentliche Erkenntnis, die man der Sicherheitsdisposition in Abbildung 8.1 entnehmen kann, ist, dass immer ein Restrisiko verbleiben wird. Das Restrisiko ist übrigens das Risiko, das nicht erfasst werden konnte. Das Risiko, das in Kauf genommen wird oder für das Notfallpläne existieren, kann nicht als Restrisiko bezeichnet werden, da es berücksichtigt wurde.

Eine weitere sinnvolle Disposition ergibt sich aus der Betrachtung der Bedrohungs- und Schwachstellenquellen. Bedrohungen oder Gefahren entstehen durch die drei Faktoren Mensch (Personal), Organisation und Technik. Im Folgenden soll ein Überblick über Massnahmen gegeben werden, die hier getroffen werden können. Abbildung 8.2 fasst diese Massnahmen grafisch zusammen. Auch wenn im Rahmen dieses Berichts die technischen Massnahmen weiter ausgebreitet werden als die menschlichen (bzw. personellen) und organisatorischen, kommt es hier doch auf ein Zusammenspiel und eine ausgewogene Mischung der Massnahmen an.

Gegenmassnahme / Bedrohung	personelle			organisatorische			technische			
	Schulung	Sicherheits-bewusstsein	Handlungs-anleitungen	Netztopologie	situative Rechtevergabe	Zuständigkeiten und Verantwortlichkeiten	kryptographische Methoden	Firewalls	redundante Einrichtungen	Auditwerkzeuge
Verlust der Verfügbarkeit				8.3.1		8.3.3		8.4.2	8.4.3	
Verlust der Vertraulichkeit	8.2.1	8.2.2	8.2.3	8.3.1	8.3.2	8.3.3	8.4.1	8.4.2		
Verlust der Verbindlichkeit						8.3.3	8.4.1			8.4.4
Verlust der Authentizität						8.3.3	8.4.1	8.4.2		
Verlust der Integrität	8.2.1	8.2.2	8.2.3	8.3.1	8.3.2	8.3.3	8.4.1	8.4.2		
Verlust der Betriebssicherheit	8.2.1	8.2.2	8.2.3	8.3.1	8.3.2	8.3.3		8.4.2		8.4.4

Abbildung 8.2: Gefahren und Gegenmassnahmen

8.2 Personelle Massnahmen

In diesen Bereich gehören alle Massnahmen, die den einzelnen Mitarbeiter in das Zentrum von Sicherheitsmassnahmen rücken. Die Sicherheitspolitik des Unternehmens muss in diesem Bereich mit der Unternehmenskultur übereinstimmen. Klaffen Kultur und Sicherheitspolitik auseinander, entstehen Spannungen, die letztlich neben anderen unerwünschten Wirkungen auch die Sicherheit gefährden können. Besteht in einem Unternehmen z.B. eine grosszügige Informationspolitik, darf die Einführung neuer IT–Systeme nicht zur Begrenzung der Informationspolitik beitragen.

Neben diesen kulturellen Aspekten, die im Rahmen dieses Berichts nicht weiter vertieft werden sollen, ergeben sich drei Hauptgebiete personeller Massnahmen: Schulung, Sicherheitsbewusstsein und Handlungsanleitungen (Policies).

8.2.1 Schulung

Viele Bedrohungen und Schwachstellen entstehen durch fehlerhafte Konfigurationen und Unkenntnis in der Bedienung der Systeme. Dies trifft insbesondere auf die Vernetzungstechnik und die Bereitstellung von Internet–Servern zu.

Die Schulung soll dazu beitragen, den verantwortlichen Mitarbeitern genügend Wissen zu vermitteln, damit sie überhaupt in der Lage sind, ihre Aufgaben unter Sicherheitsaspekten zu erfüllen. Ein weiteres Ausbildungsziel ist die Schulung im richtigen Verhalten auf unvorhergesehene Fälle und die Sensibilisierung zu Sicherheitsthemen.

Leider wird die Schulung in Sicherheitsmassnahmen häufig vernachlässigt, da sie zunächst keinen unmittelbaren Nutzen erbringt. Erst wenn ein Unternehmen durch personelle

Schwachstellen einen Schaden erlitten hat, steigt die Bereitschaft zu Schulungsmassnahmen (vgl. [Mau98, S. 34] und [SB92, S. 261ff]).

8.2.2 Sicherheitsbewusstsein

Die Schaffung eines Sicherheitsbewusstsein ist eine wesentliche Voraussetzung für den Erfolg von Sicherungsmassnahmen. Studien belegen immer wieder, dass von Mitarbeitern durch mangelndes Sicherheitsbewusstsein die grösste Gefahr ausgeht (vgl. z.B. [Hei96, S. 251]). Wesentliche Schwachpunkte zeigen sich bei der Vergabe von Passworten und dem vertraulichen Umgang mit Informationen.

Ein weiteres wichtiges Problemgebiet wird mit dem des „Social Engineering“[1] beschrieben. Hier geht es um Versuche von Fremden, durch Ausspionieren von persönlichen Daten und Vortäuschen einer Bekanntheit mit dem Kommunikationspartner Zugriff zu IT–Systemen zu erhalten. Z.B. könnte ein Anrufer sich als Firmenmitarbeiter ausgeben, der sein Passwort vergessen hat und jetzt dringend Zugang zum System benötigt.

Eine weitere Quelle für Bedrohungen aufgrund mangelnden Sicherheitsbewusstsein ist das sorglose Kopieren und Ausführen von Programmen, die z.B. über das Internet bezogen wurden.

In all diesen Fällen ist die Aufklärung in Verbindung mit einer Schulung die billigste und effektivste Massnahme die getroffen werden kann. Der Umgang mit Bedrohungen aus dem Internet sollte in Form von Handlungsanleitungen (bzw. Policies) geregelt werden.

8.2.3 Handlungsanleitungen – Policies

Sicherheitsrelevante Regelungen (Gebote, Verbote, Erläuterungen etc.) sollten in Form eines kurzen, allgemein verständlichen Dokumentes beschrieben werden. Diese Dokumentation sollte von Mitarbeitern gelesen und akzeptiert werden. Handlungsanleitungen dienen somit der Schaffung eines Sicherheitsbewusstseins und sind als Nachschlagewerke für bestimmte Sicherheitsaspekte brauchbar.

Inhaltlich sollten auf operativer Ebene alle relevanten Regelungen aufgeführt werden. Konkrete Inhalte könnten Regelungen zur Passwortvergabe (z.B. Gültigkeitsdauer, Aufbau etc.), Netzzugriffe (z.B. kein Aufruf von Webseiten mit zweifelhaften Inhalten) oder Regelungen im Umgang mit überlassener Software sein. Ob bei Missachtung Sanktionen drohen, hängt sicherlich von der Schwere des Vergehens und von der Unternehmenskultur ab. Falls bestimmte Policies rechtliche Ursachen haben, sollte darauf auch hingewiesen werden (z.B. Datenschutzrecht).

[1] ftp://ftp.cert.org/pub/cert_advisories/CA–91%3A04.social.engineering

8.3 Organisatorische Massnahmen

Organisatorische Massnahmen sind Massnahmen, die sich aus Organisationsprinzipien heraus ergeben. Im Einzelfall können diese Massnahmen allerdings durchaus durch technische Einrichtungen oder durch personelle Massnahmen unterstützt werden. Zu den organisatorischen Massnahmen werden folgende Massnahmentypen gezählt:

Logische Netztopologie: Unter der Bestimmung der Netztopologie versteht man den organisatorischen Netzaufbau, so dass Unternehmensprozesse optimal unterstützt werden. Die Aufteilung oder gar Trennung von Netzen kann durch unterschiedliche Informationswerte der einzelnen Prozesse begründet sein.

Situative Rechtevergabe: Durch feine Rechtegranulate und flexiblere Rechteverteilung kann die Sicherheit vor Manipulation von Daten erheblich gesteigert werden. Rechte müssen aufgabenspezifisch vergeben werden können und nicht durch einen Superuser (vgl. auch Need–To–Know Ansatz bei [Hol96]).

Zuständigkeiten und Verantwortlichkeiten: Die Festlegung von Zuständigkeiten und damit verbunden Verantwortungen ist ein zentraler Bestandteil jeder organisatorischen Einbettung einer Aufgabe.

Im folgenden sollen diese drei Punkte weiter ausgeführt werden.

8.3.1 Netztopologie

Obwohl der Aufbau und Betrieb von Netzen gerne als rein technische Angelegenheit betrachtet wird, ist die Netzkonzeption hauptsächlich von organisatorischen Randbedingungen abhängig. Oder anders ausgedrückt: Das Netzwerk muss zu den Geschäftsprozessen im Unternehmen passen. Reicht für kleinere Unternehmen häufig ein gemeinsames Netzwerk aus, so kann dies bei grösseren Unternehmen oder bei speziellen Anforderungen nicht mehr als ausreichend erachtet werden.

Aspekte zur Aufteilung eines Unternehmensnetzes in mehrere Teilnetze entstammen funktionalen Anforderungen und Sicherheitsanforderungen. Nachfolgend sind einige Hauptkriterien genannt, die eine Aufteilung des Netzwerkes beeinflussen:

Performanz: Sind besondere Geschwindigkeitsanforderungen oder gar Echtzeitanforderungen in Teilen eines Unternehmens notwendig, kann es notwendig werden, diese Netze aus Geschwindigkeitsgründen zu trennen.

Testbetrieb: Entwicklungsabteilungen oder Softwareabteilungen die insbesondere auch Netzanwendungen (z.B. Client / Server–Programme) erstellen, sollten ein eigenes Teilnetz erhalten.

Produktion: Produktive Server–Umgebungen in Rechenzentren oder bei der Prozesssteuerung werden aus Sicherheits- und Verfügbarkeitsgründen ebenfalls gerne in eigene Teilnetze gelegt.

Aufgabenorientierung: Abteilungen oder Gruppen mit gleichem Aufgabenspektrum können aus Sicherheitsgründen in eigene Netze gelegt werden. Dadurch gelangen Informationen, die innerhalb des Aufgabenbereichs liegen sollten, nicht nach aussen. Es kommt hier auf eine geschickte Auswahl der Granularität an, damit das Konzept sinnvoll eingesetzt werden kann.

Internet–Anbindung: Die Anbindung des Unternehmens an das Internet erfordert ebenfalls besondere Massnahmen. Aus Sicherheitsgründen sollte ein direkter Zugriff auf das Internet von Unternehmensrechnern vermieden werden. Alle Zugriffe sollten über spezielle Application–Firewalls (Proxies) geleitet werden (s. Kapitel 12.3.2).

Physische Trennung: Gelegentlich kann auch die vollständige Trennung von Netzen sinnvoll sein. Verfügt ein Unternehmen z.B. noch über keine Internet–Erfahrung, dann kann es zunächst sinnvoll sein, ein eigenes Netz mit Internet–Anschluss zu legen. Eine Bedrohung des Unternehmensnetzes ist damit ausgeschlossen. Dies wird durch Unternehmen bei ihrer ersten Bekanntschaft mit dem Internet auch häufig so angewandt.

Die Segmentierung eines Unternehmensnetzes kann nach unterschiedlichen Kriterien erfolgen. Im Zusammenhang mit der Nutzung von Internet–Diensten wird auch die Unternehmensübergreifende Kooperation von zunehmender Bedeutung. Abbildung 8.3 zeigt eine mögliche Aufteilung eines Unternehmensnetzes mit Internet–Anschluss, über das unterschiedliche Interaktionen mit Kunden und anderen Unternehmen stattfindet. Im Beispiel wurde das Rechenzentrumsnetz vom Hausnetz durch einen Firewall getrennt. Ebenso wurden der Finanzabteilung und der Konstruktionsabteilungen unterschiedliche Netze zugeordnet. Die Finanzabteilung ist ebenfalls über einen Firewall geschützt, da hier besonders vertrauliche Daten verarbeitet werden. Die Konstruktion wurde in ein eigenes Netz gelegt, damit der dort stattfindende Tests nicht zu Ausfällen im Unternehmensnetz führen. Ein Internet–Zugang wird durch Proxy–Server, die zwischen zwei Firewalls liegen, hergestellt. Diese Architektur wird häufig als demilitarisierte Zone (DMZ) oder Transit–Netz bezeichnet (vgl. [CZ96, S. 72ff]).

Zudem ist die Verbindung mit Kooperationspartnern anderer Firmen angedeutet. Es stellt sich die Frage, welche Sicherheitsanforderungen an die andere Firma gestellt werden müssen, damit der Zugang zum eigenen System erlaubt werden kann. Dies kann die Netztopologie ebenfalls beeinflussen.

Die Netztopologie orientiert sich somit an Werteklassen und / oder Vertraulichkeitsklassen von Informationen, die in einem Netz verarbeitet werden. Bei der Bildung dieser Klassen müssen daher drei Gesichtspunkte beachtet werden:

Informationen: Welche Informationen werden hier verarbeitet? Lassen sich Cluster von gleichwertigen Informationsverarbeitungsprozessen bezüglich des Informationswertes bzw. der Vertraulichkeit bilden?

Anwendungen: Welche Anwendungen werden in diesem Bereich besonders eingesetzt? Diese Analyse kann bei Spezialanwendungen Hinweise geben, welche Informationstypen in einem Bereich auftreten.

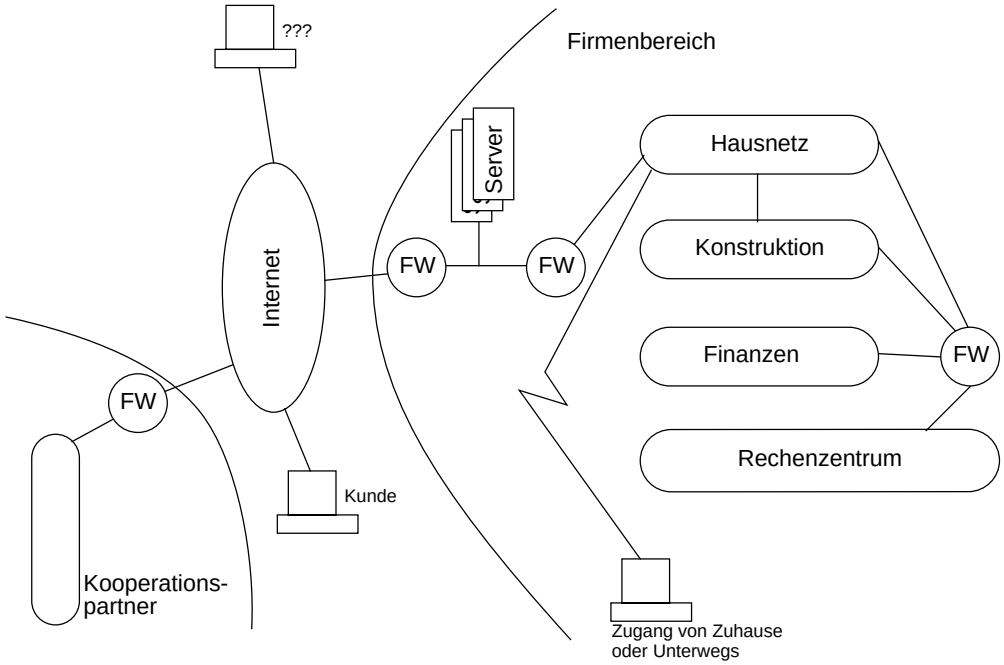

Abbildung 8.3: Szenario eines aufgabenorientierten Netzaufbaus

Endsysteme: Auf welchen Systemen werden die Informationsprozesse abgewickelt? Auf welche Systeme muss ein Zugriff gewährleistet sein?

Diese drei technischen Gesichtspunkte konkretisieren die allgemein gehaltenen Auswahlkriterien. Es kann durchaus sein, dass sich nicht immer eine adäquate Zuordnung finden lässt, da Personen in unterschiedlichsten Rollen tätig sind. Dies ist aber auch eine Frage der Granularität, die, um die Arbeit nicht zu behindern, nicht zu gering sein sollte.

Durch Segmentierung mittels Firewalls, können auch einzelne Verstösse gegen die Netztopologie toleriert werden. Sollte z.B. ein Rechenzentrumsmitarbeiter auch Softwareentwickler in der Konstruktion sein, könnte man seinem speziellen Endsystem den freien Zugang zu beiden Netzen gestatten. Hier muss dann aber eine in Abschnitt 8.2 angesprochene Bewusstseinsschulung des betroffenen Mitarbeiters stattfinden.

Sicherheitsanforderungen an unterschiedliche Teilnetze können durch Klassifizierung verallgemeinert werden. Auf dieser Basis ist eine Interaktion mit anderen Unternehmen oder Partnern möglich. Sollen z.B. Konstrukteure unternehmensübergreifend zusammenarbeiten, dann sollten die Informationen, die sie verarbeiten, und die entsprechenden Anwendungen und Endsysteme in einer vergleichbaren Sicherheitsklasse liegen.

Im Rahmen des SINUS Projekts wurde daher das Konzept der DMZ erweitert und verallgemeinert. Die Idee ist es, aufgabenorientierte Netze unterschiedlicher Sicherheitsanforderungen und -einschätzungen in unterschiedliche Bereiche (areas) einzustufen. Die Analyse der Segmentierungsmöglichkeiten in unterschiedliche Teilnetze sollte zu einer Einteilung in Klassen führen [Zur98]. Grundsätzlich lassen sich folgende Netzbereiche identifizieren:

Untrusted Area (UTA): Die untrusted Area ist ein Netzbereich der bezüglich Sicherheitsgarantien im Netz nicht eingeschätzt werden kann. Das heisst nicht, dass der Bereich

wirklich nicht vertrauenswürdig ist, er kann bezüglich seiner Vertrauenswürdigkeit lediglich nicht eingeschätzt werden.

Trusted Area (TA): In diesem Bereich werden sehr wertvolle Informationen verarbeitet und es bestehen sehr hohe Sicherheitsanforderungen. Eine Verletzung der Sicherheit in diesem Bereich hätte gravierende Folgen für das Unternehmen. Typische Beispiele für diese TAs sind Hostsysteme einer Bank, Echtzeitverarbeitungssysteme, etc. Diese Systeme sollten aus dem Internet heraus niemals direkt ansprechbar sein.

Internet Access Area (IAA): Die IAA ist ein Bereich, der das Unternehmensnetz und das Internet in einer gesicherten Weise verbindet. Hierzu werden Kombinationen aus Internet–Firewalls und Internet–Servern aufgebaut (vgl. Kapitel 12). Dieser Begriff ersetzt den Begriff der DMZ.

Zwischenbereiche: In diesen Bereichen werden Informationen mit einem definierten Wert bzw. einer definierten Vertraulichkeit verarbeitet. Es gibt in diesem Bereich bestimmte und definierte Sicherheitsanforderungen, die Konnektivität mit anderen Netzen beeinflussen. Eine Unterteilung nach diesen Vertraulichkeitsanforderungen ist ratsam. Die Granularität sollte allerdings nicht zu fein gewählt werden. Maximal drei weitere Unterteilungen in schwache (low), mittlere (medium) und hohe (high) Sicherheitsanforderungen sollten genügen. Es entstehen somit dir drei Bereiche LTA, MTA und HTA.

In dem Szenario, dass in Abbildung 8.3 wiedergegeben wurde, lassen sich unschwer diese Bereiche unterschiedlicher Sicherheitsanforderungen (bzw. -aussagen) zuordnen.

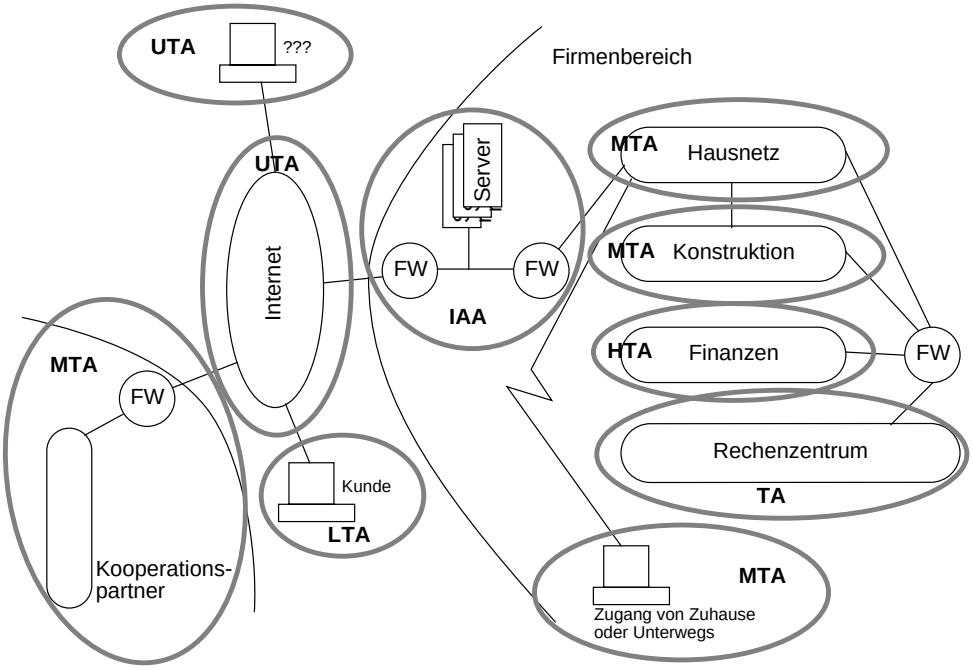

Abbildung 8.4: Abgeleitete, unterschiedlich vertrauenswürdige Teilnetze aus dem Szenario

Abbildung 8.4 illustriert nochmals das Szenario aus Abbildung 8.3. Die verschiedenen Aufgabenbereiche und Kommunikationsteilnehmer wurden nach dem Konzept unterschiedlicher Vertrauensbereiche klassifiziert. In dem Beispiel wurden neben den beiden Klassen TA und UTA insgesamt vier verschiedene Klassen für Netze mit einem mittleren Sicherheitsniveau angenommen.

Aus dem Szenario lassen sich weitere Schlüsse für das Area–Konzept ziehen. So ist aus Unternehmenssicht ein beliebiger Teilnehmer im Internet zunächst nicht vertrauenswürdig; es können keine Aussagen über ihn getroffen werden. Ein Teilnehmer, der dem Unternehmen als Kunde oder auch als Mitarbeiter bekannt ist, kann bezüglich seines Sicherheitsniveaus und Verhaltens besser eingeschätzt werden. Es besteht weiterhin die Notwendigkeit, vertrauenswürdige Netze über nicht vertrauenswürdige Netze (z.B. zur Kooperation) zu unterhalten. Kooperierende Netze zwischen Unternehmen sollten in der gleichen oder zumindest in ähnlichen Sicherheitsklasse liegen.

8.3.2 Situative Rechtevergabe

Die situative Rechtevergabe ist auch bei der Nutzung von Internet–Diensten ein geeignetes Instrument zur Erhöhung der Sicherheit. Ziel ist es, Benutzungsrechte von Systemkomponenten (Daten, Anwendungen und Netzen) an Aufgaben zu binden. D.h. die Rechte an bestimmten Systemressourcen sind an die aktuelle Aufgabensituation gebunden.

Situative Rechte können auch bedeuten, dass Dienste nur zu bestimmten Uhrzeiten oder nach bestimmten Parametern erlaubt sind. Überprüfen und steuern lassen sich z.B. Übertragungsmenge und -inhalte. Eine Überprüfung in diesem Bereich muss sich allerdings mit der Unternehmenskultur und rechtlichen Vorgaben in Einklang bringen lassen.

8.3.3 Zuständigkeiten und Verantwortlichkeiten

Die Aktivitäten im Internet müssen in die bestehenden Organisationsstrukturen eingebunden werden. Falls noch keine Verantwortlichkeiten für das Sicherheitsmanagement definiert wurden, sollte dies jetzt nachgeholt werden.

Die konkreten Ausprägungen der Organisationsstruktur sind von Unternehmen zu Unternehmen verschieden. Es lassen sich nach [BSI96, S. 14ff] jedoch folgende Funktionen definieren:

Informationssicherheits–Ausschuss: Erarbeitung abteilungsübergreifender Sicherheitsbelange. Ausarbeitung von Strategien und Politiken.

Informationssicherheitsbeauftragter: Auch als Chief Security Officer (CSO) bezeichnet. Verantwortlichkeit in allen Informationssicherheitsfragen des Unternehmens oder einer Unternehmenseinheit.

Die Bereitstellung von Internet–Diensten ist i.d.R. im Vollausbau eine bereichsübergreifende Tätigkeit. Sie kann bei der Informationsmanagement–Abteilung des Unternehmens angesiedelt sein. Je nach Grösse des Unternehmens fällt natürlich auch die Grösse der Organisationsstruktur unterschiedlich aus. Es ist nicht unbedingt notwendig, dass der Informationssicherheits–Ausschuss und der CSO getrennte Personen sind. In sehr kleinen Unternehmen kann der CSO und das Informationssicherheits–Ausschuss in einer Person zusammengefasst werden.

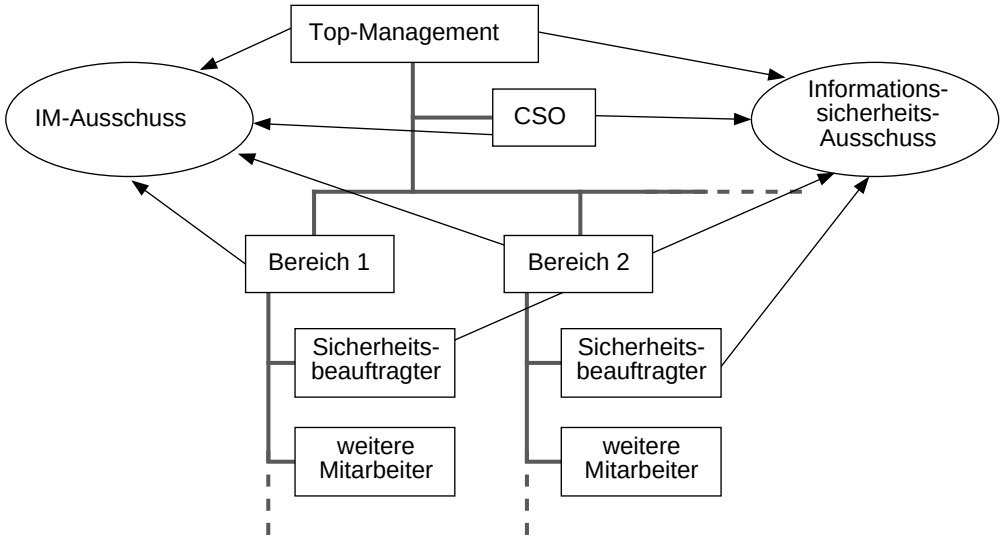

Abbildung 8.5: Organisationsmöglichkeit der Informationssicherheit

Abbildung 8.5 zeigt ein Beispiel zur Organisation in einem mittleren Unternehmen. Es gibt mehrere Möglichkeiten, das Sicherheitsmanagement in die Gesamtorganisation einzufügen. Der CSO kann je nach Bedeutung der Sicherheit für das Unternehmen als Stabsfunktion des Top–Managements oder des Bereichs Informationsmanagement definiert sein. Er sollte eine entscheidende Rolle im Informationssicherheits–Ausschuss und dem IM–Ausschuss des Unternehmens spielen. Bei kleineren und mittleren Unternehmen können IM–Ausschuss und Informationssicherheits–Ausschuss auch zusammenfallen. In diesen Gremien sollten neben dem Top–Management auch Personen aus den Abteilungen als Anwender vertreten sein. Bei grösseren oder international tätigen Unternehmen kann sich diese Organisationsstruktur in den Geschäftseinheiten fortsetzen. Existieren sowohl IM–Ausschuss als auch der Informationssicherheits–Ausschuss, so sollten diese beiden Gremien eng gekoppelt werden.

8.4 Technische Massnahmen

Aus technischer Sicht gibt es vier grundlegende Massnahmen zur Sicherung bei der Verwendung von Internet–Diensten, die getroffen werden können (in Anlehnung an [Wei97b]):

Kryptographische Methoden: Sie helfen bei der Bestimmung der Authentizität von Personen und Dokumenten (digitale Signaturen), der Integrität von Dokumenten und bei der Wahrung der Vertraulichkeit (Verschlüsselung).

Firewalls: Der Abgrenzung von Teilen des Netzes und eine damit verbundene Beschränkung der netzübergreifenden Dienstnutzung ist für die kontrollierte Dienstnutzung sinnvoll.

Redundante Einrichtungen: Die Schaffung von Redundanzen bzgl. Rechnern, Netzverbindungen, etc. kann einen entstandenen Schaden begrenzen. Insbesondere kann die Verfügbarkeit des Netzes dadurch gesichert werden.

Auditwerkzeuge: Hierunter fallen technische Einrichtungen, die eine Überwachungs- oder Kontrollaufgabe wahrnehmen. Typische Vertreter sind Virenschutzprogramme, aber auch Angriffssimulatoren und Protokollauswertungssysteme.

Diese technischen Massnahmen werden im weiteren näher beschrieben.

8.4.1 Kryptographische Methoden

Kryptographische Methoden dienen grundsätzlich zur Realisierung zweier wesentlicher Sicherheitsmassnahmen:

Verschlüsselung: Die Verschlüsselung (Chiffrierung) dient der vertraulichen Übermittlung von Nachrichten über vertrauensunwürdige Wege.

Signatur: Die digitale Signatur dient zum Nachweis der Authentizität des Absenders und zum Nachweis der Integrität im Sinne der Verbindlichkeit; d.h. des Nachweises, das eine Nachricht nach der Unterschrift nicht mehr verändert wurde.

Beide Verfahren lassen sich grundsätzlich getrennt oder gemeinsam benutzen. Im nachfolgenden soll in die Thematik kryptographischer Methoden eingeführt werden. Der Reihe nach werden symmetrische und asymmetrische Verfahren, Zertifikate und typische Anwendungen näher betrachtet.

8.4.1.1 Symmetrische Kryptosysteme (Private Key)

Symmetrische Kryptosysteme dienen streng genommen ausschliesslich zur Verschlüsselung von Informationen. Haben allerdings die Kommunikationspartner die Gewähr, dass nur sie über den Schlüssel verfügen und ist die Anzahl der an der Kommunikation beteiligten Personen gering, können symmetrische Kryptosysteme auch zum Nachweis der Identität des Senders dienen. Dies ist insbesondere der Fall, wenn nur zwei Instanzen miteinander kommunizieren. Abbildung 8.6 zeigt den Prozess der Ver- und Entschlüsselung der symmetrischen Kryptomethode (vgl. [Wei97b, S. 11ff]).

Bekannte symmetrische Kryptosysteme sind der Data Encryption Standard (DES) bzw. der Triple–DES (3DES) und der International Data Encryption Algorithm (IDEA). Zur Bewertung der Sicherheit eines symmetrischen Kryptosystems dient, wie bereits angedeutet, die Schlüssellänge, da der einfachste Weg, einen Schlüssel zu knacken, darin besteht, alle

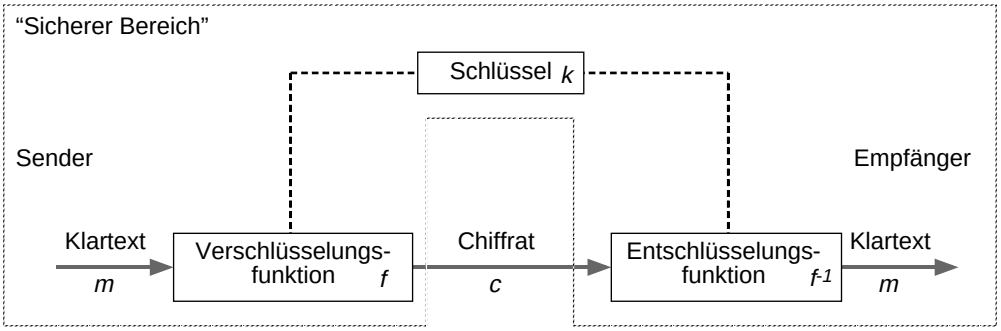

Abbildung 8.6: Symmetrische Ver- und Entschlüsselung

möglichen Schlüsselkombinationen durchzuspielen. Da ein Bit entweder auf 0 oder 1 gesetzt sein kann, kann es somit zwei Werte annehmen. Bei einer Schlüssellänge von x Bit ergeben sich somit 2^x mögliche Kombinationen. Dieser Angriff wird auch als Brute–Force–Attack bezeichnet [Mau97]. Bei der Auswahl eines Algorithmus ist zudem noch zu beachten, dass aufgrund amerikanischer Exportrestriktionen aus Amerika exportierte Software i.d.R. über eine maximale Schlüssellänge von mindestens 40 Bit verfügt (vgl. [Kya96]).

Schlüssellänge	Dauer der Brute-Force-Attack*
2^{40} = 1’099’511’627’776	109’951 s 1 Tag
2^{56} = 72’057’594’037’927’900	7’205’759’404 s 228 Jahre
2^{128} = 340’282’366’920’938’463’374’607’431’768’211’456	34'028'236'692'093'800'000'000'000'000'000 s 1'079'028'307'080'600 Mrd. Jahre

*Annahme: 10 Millionen Versuche pro Sekunde

Abbildung 8.7: Vergleich der Sicherheit bei unterschiedlicher Schlüssellänge

Abbildung 8.7 vergleicht die Sicherheit unterschiedlicher Schlüssellängen bei einer Brute–Force–Attack. Die Berechnung der Werte unterliegt der Annahme, dass der Angreifer pro Sekunde 10 Mio. Schlüssel testen kann. Dies entspricht der derzeitigen Rechenleistung von technisch sehr gut ausgestatteten Angreifern. Geht man davon aus, dass die Rechenleistung weiter steigt und dass vertrauliche Informationen auch über einige Zeit (z.B. 10 Jahre) vertraulich behandelt werden sollten, dann kommt man zu dem Schluss, dass eigentlich nur eine Schlüssellänge von 128 Bit in Frage kommt. Für manche Anwendungen mögen aus heutiger Sicht auch Schlüssellängen von 56 Bit ausreichen.

8.4.1.2 Asymmetrische Kryptosysteme (Public Key)

Asymmetrische Verschlüsselungsverfahren oder auch Public–Key–Kryptosysteme sind eine jüngere Entwicklung. Mit diesen Verfahren lassen sich sowohl Daten verschlüsseln als auch signieren.

Der Name Public–Key–Kryptosystem bezeichnet den Umstand, dass ein Teilnehmer über zwei Schlüssel verfügen muss, von dem der eine öffentlich und der andere geheim ist. Der

öffentliche Schlüssel wird häufig mit e für encryption und der geheime mit d für decryption bezeichnet. Die Bezeichnung geht auf die Funktionsweise des Ver- und Entschlüsselns zurück: Damit ein Sender einem Empfänger eine verschlüsselte Nachricht zukommen lassen kann, muss der Sender den öffentlichen Schlüssel des Empfängers besitzen. Mit diesem verschlüsselt er die Nachricht und sendet sie dem Empfänger zu, der die Nachricht mit seinem geheimen Schlüssel entschlüsseln kann. Anschaulich kann man sich diesen Vorgang wie einen privaten Briefkasten vorstellen; Jeder kann Dokumente in den Briefkasten legen, aber nur der Besitzer des Briefkasten kann die Dokumente herausnehmen.

Der Vorgang des Signierens geht genau anders herum: Ein Absender eines Dokumentes, dass er signieren möchte, verwendet hierzu seinen privaten Schlüssel zum signieren. Der Empfänger kann dann mit dem öffentlichen Schlüssel des Absenders die Authentizität der Signatur überprüfen. Das Analogon ist die herkömmliche Unterschrift unter ein Dokument. Aus technischer Sicht ist die digitale Signatur sicherer als die herkömmliche Unterschrift, da mit der digitalen Signatur auch die Integrität des Dokumentes nach der Unterschrift gewährleistet wird. Dies funktioniert allerdings nur, wenn der geheime Schlüssel des Absenders wirklich nur dem Absender bekannt ist (vgl. [Sei97]).

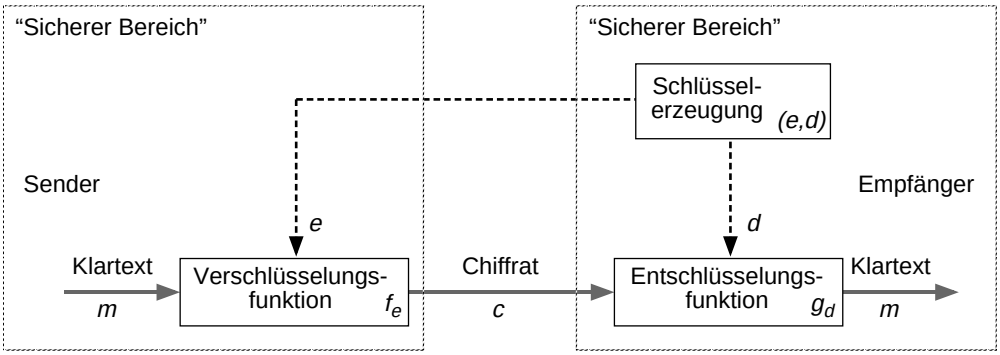

Abbildung 8.8: Ver- und Entschlüsselung mit einem asymmetrischen Kryptosystem

Wie in Abbildung 8.8 angedeutet benötigt das Ver- und Entschlüsseln mittels einem asymmetrischen Verfahren drei Funktionen. Zunächst muss mit einer Schlüsselerzeugungsfunktion das Schlüsselpaar e und d erzeugt werden. Der öffentliche Schlüssel e muss dann verteilt werden. Wie in Abbildung 8.8 angedeutet, muss dieser Schlüssel nicht über einen sicheren Bereich verteilt werden. Der Schlüsselaustausch vereinfacht sich dadurch wesentlich.

Abbildung 8.9 zeigt die Signierung von Dokumenten mittels asymmetrischen Kryptosystemen. Der Vorgang ähnelt den der Ver- und Entschlüsselung. Wesentlicher Unterschied ist jedoch, dass zum Signieren eines Dokumentes der eigene private Schlüssel verwendet wird. Das Überprüfen der Signatur findet mit dem öffentlichen Schlüssel statt.

Eine digitale Signatur bestätigt nicht nur die Authentizität des Absenders sondern auch die Integrität des Inhalts. Sobald auch nur ein Bit nachträglich verändert wird, stimmt das Ergebnis der Signaturprüfungsoperation nicht mehr mit dem Dokument überein.

Praktische Implementierungen von asymmetrischen Kryptosystemen sind der RSA–Algorithmus und das El–Gamal–Signaturverfahren.

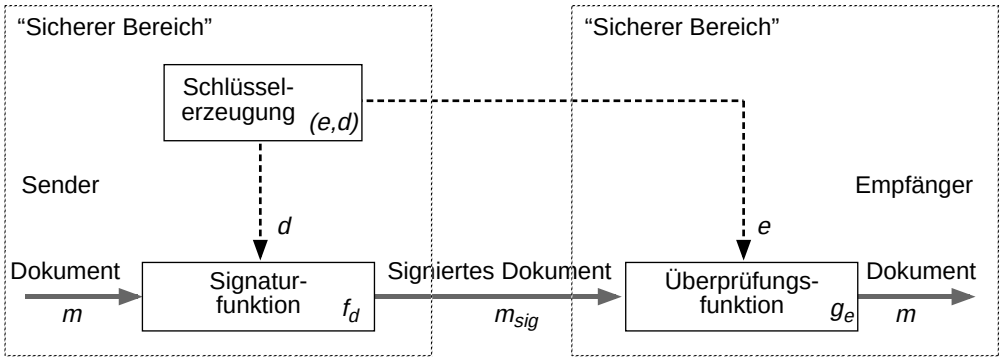

Abbildung 8.9: Signieren und Überprüfen von Dokumenten

Beim RSA–Kryptosystem ist es möglich, gleichzeitig eine Kombination aus Verschlüsselung und Signatur zu erstellen. Somit sind Vertraulichkeit, Integrität und Authentizität eines Dokumentes gleichermassen geschützt.

8.4.2 Zertifikate

Vergleicht man Abbildung 8.6 und Abbildung 8.8 miteinander, stellt man fest, dass im Fall der symmetrischen Kryptographie beide Partner über den gleichen Schlüssel verfügen müssen, den sie sinnvoller Weise über einen vertraulichen Kanal ausgetauscht haben. Bei der asymmetrischen Kryptographie kann der öffentliche Schlüssel auch über einen nicht vertraulichen Kanal ausgetauscht werden, allerdings wer leistet dann noch für die Echtheit des Schlüssels Gewähr?

Hierfür gibt es zwei Ansätze: das Vertrauensnetzwerk (Web of Trust) und eine hierarchische Zertifizierungsinstanz. Beide Systeme bauen darauf, dass ein öffentlicher Schlüssel eines Teilnehmers bei einer vertrauenswürdigen Instanz zertifiziert wird, d.h. der Schlüssel wird von einer authorisierten Stelle zur Verfügung gestellt und mit einem Zertifikat versehen, das die Echtheit des Schlüssels gewährleistet. Falls nun ein Empfänger ein signiertes Dokument von einem Sender erhält, muss er sich nur noch an die Zertifizierungsstelle wenden, um dort den öffentlichen Schlüssel zu erhalten oder sich die Authentizität des Schlüssels bestätigen zu lassen.

Wie entsteht nun aber das Vertrauensverhältnis zur Zertifizierungsstelle? Hier unterscheiden sich die beiden Konzepte deutlich.

Das Web of Trust geht davon aus, dass eine Zertifizierung von Schlüsseln über ein beliebiges Netz geschieht. D.h. Ein unbekannter Sender schickt einem Empfänger ein signiertes Dokument; der Empfänger kennt einen Freund, der den Sender kennt und seinen Schlüssel zertifiziert hat, da der Empfänger seinem Freund vertraut, kann er den Schlüssel des Senders überprüfen. Der Empfänger muss natürlich den öffentlichen Schlüssel seines Freundes kennen. Dieses Spiel kann sich im Prinzip auf beliebig vielen Stufen fortsetzen.

Bei einer hierarchischen Zertifizierung verfügt ein Empfänger einer signierten Nachricht eines unbekannten Senders über einen Zugriffspfad zu einer ihm vertrauten Zertifizierungsinstanz (d.h. er verfügt über deren öffentlichen Schlüssel). Bei dieser Instanz sucht

er nach dem öffentlichen Schlüssel des Absenders. Wird dieser dort nicht gefunden, kann eine nächsthöhere Zertifizierungsinstanz angefragt werden.

Das Problem des authentischen Schlüsselaustauschs wird somit durch die Zertifizierung auf einige wenige Schlüssel vertrauenswürdiger Instanzen beschränkt. In einigen Internet–Dienst–Programmen (z.B. Web–Browser) sind vom Hersteller bereits Zertifikate von bekannten Zertifizierungsinstanzen abgelegt.

Ein übliches Protokoll für hierarchische Zertifizierungsinstanzen ist das X.509 Protokoll. Das Web of Trust wird durch das bekannte PGP (s. Teil III) realisiert [Gar95, S. 235ff].

8.4.3 Kombinierte Anwendung kryptographischer Methoden

Dieser Abschnitt beschreibt neben den grundlegenden Anwendungen kryptographischer Methoden wie das Verschlüsseln und Signieren eine besondere Anwendungsart, bei der die kryptographischen Methoden in Kombination vorkommen. Um z.B. eine vertrauliche, authentische und integere Kommunikation zwischen zwei unbekannten Partnern herstellen zu können, wird man wie in Abbildung 8.10 angedeutet vorgehen.

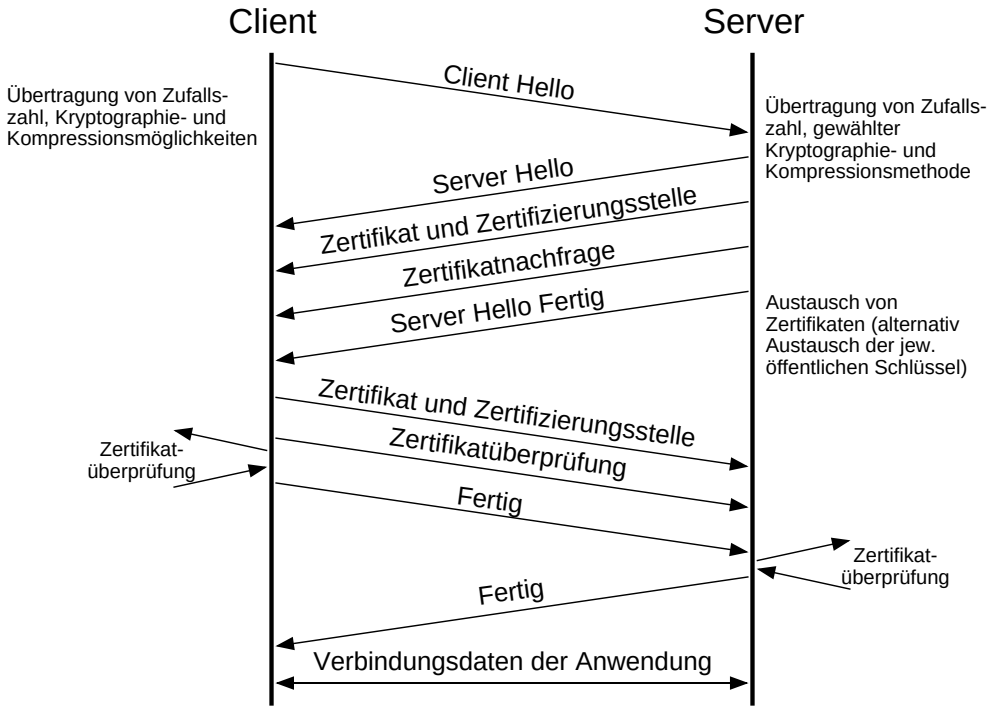

Abbildung 8.10: Verwendung kryptographischer Massnahmen zur Absicherung von Kommunikationsprotokollen

Abbildung 8.10 orientiert sich stark an der Beschreibung des SSL (Secure Socket Layer) Protokolls; dieses Protokoll bildet die Basis für eine Reihe sehr ähnlicher Implementierungen zur Absicherung von Kommuniaktionsprotokollen über das Internet. SSL wurde von Netscape Communications in der Version 3 im März 1996 als Internet Draft veröffentlicht (vgl. [FKK96]). Eine ausführliche Einführung in SSL und Sicherheitsprodukte von Netscape findet sich ausserdem bei [ETC96].

Fordert ein Client–System von einem Server Daten an (wie es in Abbildung 8.10 unterstellt wurde), werden zunächst Sicherheitsmodalitäten (insbesondere Zertifikate, Verschlüsselungs- und Kompressionsmethoden) in einem Handshake–Prozess ausgehandelt (Hello Nachricht in Abbildung 8.10). Erst nach einem erfolgreichen Aushandeln der Konditionen kann die eigentliche Kommunikation beginnen. Das Aushandeln der Konditionen wird durch asymmetrische Kryptographieverfahren unterstützt, während die eigentliche Sitzung mit einem gemeinsamen symmetrischen Sitzungsschlüssel erfolgt.

Stark vereinfacht funktioniert dies nach folgendem Schema (in Anlehnung an [ETC96]):

- Der Client generiert eine Zufallszahl.
- Der Client verschlüsselt die Zufallszahl mit dem öffentlichen Schlüssel des Servers und schickt die Zahl an den Server.
- Der Server enschlüsselt die Zufallszahl mit seinem geheimen Schlüssel.
- Die Zufallszahl kann zur Erstellung eines symmetrischen Schlüssels benutzt werden, um geheime Informationen auszutauschen.

Dieses Verfahren funktioniert nur, wenn sich Client und Server kennen und ihre öffentlichen Schlüssel ausgetauscht haben. Dies trifft möglicherweise für besondere Fälle zu, kann im allgemeinen allerdings nicht unterstellt werden. Die Lösung für dieses Problem ist die Verwendung von Zertifikaten nach X.509.

Zu Beginn einer Sitzung werden Zertifikate ausgetauscht und anhand von angegebenen Zertifizierungsstellen überprüft (dies kann sich bis zur Wurzel, d.h. die Zertifizierungsstelle der Zertifizierungsstellen, fortsetzen). Dies ist in Abbildung 8.10 durch die Zertifizierungsaustauschnachrichten angedeutet. Der Benutzer ist dabei in der Lage eine Auswahl von gültigen Zertifizierungsstellen zu treffen. Die mit dem Zertifikat ausgehändigten öffentlichen Schlüssel werden dann gemäss obigen Ablauf zur Erzeugung eines symmetrischen Schlüssels benutzt. Zudem kann durch die Verwendung von Hashfunktionen die Integrität bei der Übermittlung einer Nachricht gewährleistet werden (vgl. Abschnitt 11.3.3).

Ein optimaler Einsatz zu Kommunikationszwecken besteht also in einem gesicherten Verbindungsaufbau durch asymmetrische Kryptosysteme und eine anschliessende Verbindungsabsicherung durch symmetrische Kryptosysteme.

8.5 Firewall–Systeme

Um das interne Netzwerk vom Internet abzuschirmen oder um die Abgrenzung von Netzsegmenten zu unterstützen, können sogenannte Firewallsysteme eingesetzt werden. Ein Firewallsystem besteht i.d.R. aus Hard- und Softwarekomponenten, die zwischen die zu verbindenden Teilnetze gesetzt werden. Ein Firewall ersetzt somit u.a. die Funktion eines Routers. Die Verbindung zwischen diesen Teilnetzen findet ausschliesslich über diesen Firewall statt, der über unterschiedliche Filter- und Kontrollmechanismen verfügt. Es gibt zwei

grundlegende Firewalltypen: den Paketfilter und das Application–Gateway. Die Unterscheidung findet nach Filterungsebene statt. Der Paketfilter arbeitet auf TCP / IP–Ebene, während das Application–Gateway auf Anwendungsebene arbeitet [Mau98, S. 36ff].

8.5.1 Paketfilter

Der Paketfilter (auch Screening Router, Screening Filter, Packet Filter) ist die einfachste Variante eines Firewalls. Paketfilter analysieren die ein- und ausgehenden Datenpakete auf der Internet- und Transportschicht. Konkret heisst das, dass sie die Protokolltypen ICMP, IGMP, TCP und UDP nach bestimmten Kriterien überprüfen. Datenpakete können auf drei Arten behandelt werden:

Weiterleiten: die Daten werden durch den Firewall geleitet.

Verwerfen (block): die Daten werden einfach verworfen.

Zurückweisen (reject): die Daten werden als unzustellbar an den Absender zurückgeschickt.

Die Pakete können auf unterschiedliche Weise gefiltert werden:

IP–Adresse: Bestimmte IP–Adressen können gesperrt oder durchgelassen werden.

Protokolltyp: Z.B. können nur TCP–Pakete aber keine ICMP–Pakete erlaubt sein.

Portnummer: Zugriffe können auch nach Portnummer erlaubt werden oder nicht; dies entspricht einer Filterung nach Dienst. Der Default–Port für HTTP ist z.B. der Port 80.

Benutzer: Bei einer Zuordnung von Benutzern zu IP–Nummern, oder wenn sich Benutzer erst am Firewall anmelden müssen, kann auch danach gefiltert werden.

Datenmenge: Übergrosse Mails könnten so z.B. verworfen werden.

All diese Regeln können darüber hinaus kombiniert werden. Manch Firewalls, wie z.B. der SINUS–Firewall (siehe Abschnitt 12.5) verfügen darüber hinaus über dynamische Filterregeln, die nur zu bestimmten Zeiten gelten oder sich selbst aktivieren und deaktivieren je nach Systemumgebung.

Abbildung 8.11 zeigt beispielhaft einen Paketfilter zwischen dem Internet und dem Unternehmensnetz. Im Beispiel wird ein Zugriff aus dem Unternehmen auf Port 80 (HTTP) nach aussen zugelassen, der Benutzer Meier darf keine Zugriffe auf das Internet machen. Von aussen darf der Rechner 135.132.133.45 nicht zugreifen. Das Beispiel illustriert ausserdem, dass Regeln immer in zwei Richtungen definiert werden können.

Paketfilter müssen alle Zugriffe Protokollieren und bei Angriffsversuchen unterschiedliche Alarmierungsmechanismen bereitstellen können.

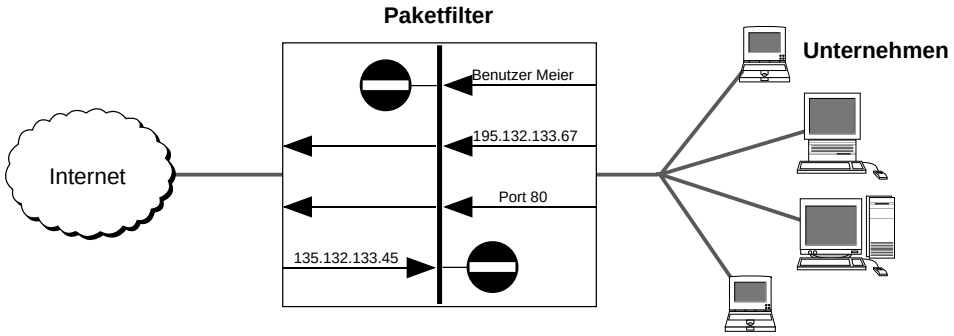

Abbildung 8.11: Paketfilter

8.5.2 Application–Gateway

Bei einem Application–Gateway (auch Proxy–Firewall oder Proxy) handelt es sich um einen Firewalltyp, der Datenpakete auf Anwendungsebene filtern kann [Mau98, S. 37]. Die Bezeichnung Proxy rührt auch daher, dass Anwendungen, die über einen Proxy laufen niemals direkt mit dem gewünschten Server kommunizieren, sondern immer über den Proxy verbunden werden (vgl. Abbildung 8.12).

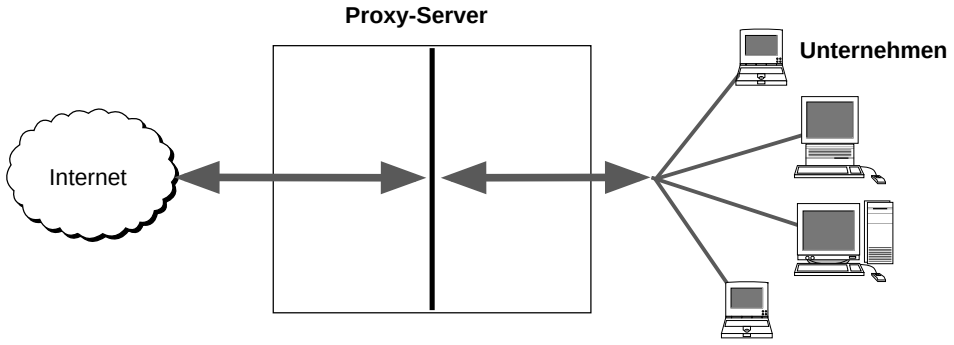

Abbildung 8.12: Application Gateway

Der Vorteil eines Application–Gateways gegenüber einem Paketfilter besteht darin, dass er Datenpakete nicht nur nach Steuerinformationen sondern auch nach Inhalten überprüfen kann. Soll z.B. der WWW–Zugang ermöglicht werden aber die Ausführung von ActiveX Applets verhindert werden, könnten diese Applets aus dem Datenstrom herausgefiltert werden. Eine weitere Möglichkeit ist die automatische Überprüfung auf Viren im eingehenden Datenstrom.

Nachteilig ist auf der anderen Seite, dass für jeden Internet–Dienst ein eigener Anwendungsfilter erstellt werden muss. Die Funktion als Proxy ist zudem für die Anwender nicht transparent, sie müssen ihren jeweiligen Anwendungen die Verwendung des Proxies mitteilen.

8.5.3 Firewall–Architekturen

I.d.R. werden Paketfilter und Application–Gateway bei der Anbindung an das Internet in Kombination eingesetzt. Es entstehen eine Reihe von Architekturmodellen beim Aufbau von Firewalls.

- Bastion–Host
- DMZ
- dual homed Gateway
- single homed Gateway

8.5.4 Redundante Einrichtungen

Wenn Teile eines Systems mehrfach vorhanden sind und bei Ausfall einer Komponente die Funktion von einer anderen Komponente übernommen werden kann, spricht man von redundanten Einrichtungen (vgl. [Mau98, S. 41]). Redundante Einrichtungen dienen somit primär der Sicherstellung der Verfügbarkeit von Informationen und informationsverarbeitender Systeme. Je nach Anforderungen an die Verfügbarkeit kann man eine Unterscheidung nach folgenden Kriterien vornehmen (vgl. [Rob94, S. 502f]):

Hot Site: Beschreibt eine Einrichtung, die unmittelbar nach Ausfall einer Komponente die Verarbeitung aufnehmen kann; die Verfügbarkeit wird dadurch nicht oder nur sehr unwesentlich gestört.

Warm Site: Beschreibt eine Einrichtung, die im Prinzip vorhanden ist, aber noch geeignet konfiguriert oder anderweitig vorbereitet werden muss. Die Verfügbarkeit kann kurzfristig nicht gewährleistet sein und der Ersatzbetrieb kann u.U. mit reduziertem Durchsatz erfolgen.

Cold Site: Beschreibt eine Einrichtung, die zwar vorhanden ist, aber vollkommen neu auf die entsprechende Situation vorbereitet werden muss; die Verfügbarkeit ist auf jeden Fall geraume Zeit nicht gewährleistet.

Die Begriffe stammen eigentlich aus dem Rechenzentrumsbetrieb, gelten aber für die Bereitstellung von Internet–Diensten adäquat. Hier kann entweder die Netzinfrastruktur redundant gehalten werden (z.B. Router, Internet–Zugänge, Netzwerke) oder die Server für Internet–Dienste können redundant gehalten werden (z.B. WWW Server oder Mail Server).

8.5.5 Auditwerkzeuge

Ursprünglich wurde der englische Begriff Audit im Rechnungswesen für die Rechnungsprüfung verwendet. In der Informatik hat sich der Begriff sowohl im Sinne der Revision als

auch des Controllings durchgesetzt. Unter Auditwerkzeugen versteht man daher Systeme zur nachträglichen und periodischen Überprüfung der Systemsicherheit als auch Systeme zur laufenden Überwachung des Systemzustands.

Daneben lassen sich Auditwerkzeuge nach ihrer Funktionalität unterscheiden. Man unterscheidet Angriffssimulatoren, Überwachungsprogramme und Programme zur Überprüfung der Systemsicherheit (vgl. Abbildung 8.13).

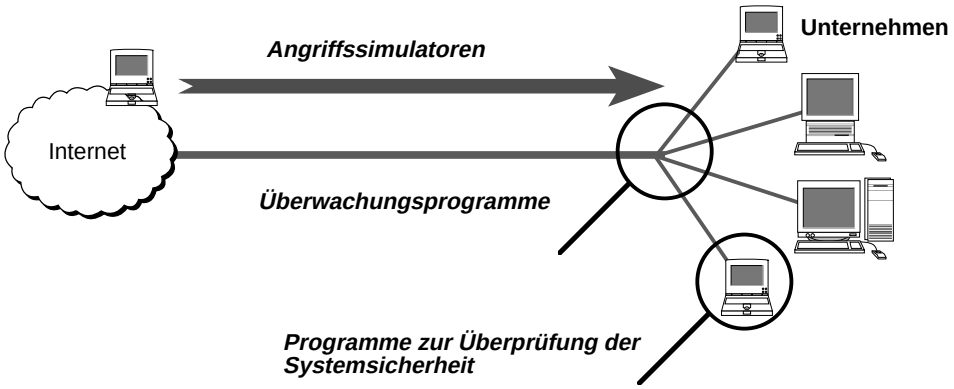

Abbildung 8.13: Typen von Auditwerkzeugen

Angriffssimulatoren sind Programme bzw. Skripts mit denen aus dem Internet (oder allgemein aus einem Netzanschluss) heraus Angriffe auf beliebige andere Systeme gestartet werden. Ein sehr bekannter und kontrovers diskutierter Vertreter eines solchen Werkzeugs ist das Programm SATAN (Security Administrator Tool for Analyzing Networks) und sein Nachfolger SAINT (Security Administrators Integrated Network Tool, siehe http://www.wwdsi.com/saint/).

Programme zur Überprüfung der Systemsicherheit werden lokal auf dem System eingesetzt, das geprüft werden soll. Typische Vertreter solcher Programme sind Systeme zur Überprüfung der Passwortsicherheit.

Überwachungsprogramme schliesslich sind Systeme, die den Datenverkehr im Netz selbst überwachen und überprüfen. Diese Programme sind häufig auch als Bestandteile von Firewallsystemen realisiert.

Kapitel 9

Realisierung/Implementierung der Massnahmen

9.1 Einleitung

Im letzten Kapitel wurden Sicherheitsmassnahmen aufgezählt, die es einem Unternehmen ermöglichen, den Risiken bei der Internetnutzung zu begegnen. Dabei wurden sowohl technische wie auch organisatorische / personelle Massnahmen berücksichtigt.

Zur Realisierung der technischen Sicherheitsmassnahmen können diese entweder selbst auf die Bedürfnisse zugeschnitten und implementiert werden, oder es kann auf existierende Produkte bzw. Produktklassen zurückgegriffen werden. Grosse Unternehmen mit komplexen Wünschen an die Internet–Dienste und grossem Volumenaufkommen wählen häufig den erstgenannten Weg. Ein bekanntes Beispiel ist AT&T, die die Entwicklung ihrer massgeschneiderten Firewall–Software in mehreren Artikeln publiziert haben [Che90, Bel92].

Kleine und mittlere Unternehmen greifen dagegen in der Regel auf existierende Standardlösungen zurück. Sie haben normalerweise weder den Bedarf an spezialisierten Eigenentwicklungen, noch qualifiziertes Personal, das diese spezialisierte Aufgabe bewältigen könnte. Das Internet bietet mit seinen standardisierten Protokollen und Diensten die Möglichkeit, auf universell einsetzbare Sicherheitsprodukte zurückzugreifen und diese durch geeignete Konfiguration an die eigenen Bedürfnisse anpassen zu können.

In diesem Kapitel werden den Sicherheitsmechanismen aus dem letzten Kapitel gängige Produkte bzw. Produktklassen gegenübergestellt, die zur Realisierung der genannten Anforderungen geeignet sind. gibt einen Überblick über die nachfolgend besprochenen Realisierungsmassnahmen und zeigt in der Zelle an, in welchem Abschnitt sie besprochen wird. Ein Überblick über die Zuordnung zwischen Produkten / Produktklassen und Massnahmen gibt die Abbildung 9.1. In den Zellen ist angegeben, in welchem Abschnitt dieses Kapitels die Realisierung näher erläutert wird.

Realisierungsmassnahme / Gegenmassnahme	Auswahl von Netzkomponenten	Einsatz von Firewall		Einsatz von Kryptographie			
	geeignete Netzkomponente	Packetfilter	Application Gateways	PGP	S/MIME	SSH	SSL
Schulung							
Sicherheitsbewusstsein							
Handlungsanleitungen							
Netztopologie	9.3						
situative Rechtevergabe							
Zuständig- und Verantwortlichkeiten							
kryptographische Methoden				9.6.2	9.6.3	9.6.5	9.6.6
Firewalls		9.4.2	9.4.2				
redundnate Einrichtungen	9.3						
Auditwerkzeuge							

Abbildung 9.1: Massnahmen und dazugehörige Produkte resp. Produktklassen

9.2 Offene vs. proprietäre Lösungen

Die auf dem Markt erhältlichen Sicherheits–Softwareprodukte lassen sich in zwei Klassen unterteilen:

- Produkte mit offener Spezifikation, häufig auch mit offengelegtem Quellcode
- Proprietäre Produkte, bei denen lediglich das Anwendungsgebiet benannt, nicht aber die Funktionsweise bekannt ist.

Diesen beiden Klassen liegen unterschiedliche Sichtweisen von Sicherheit zugrunde. Bei den proprietären Produkten geht der Hersteller davon aus, dass es zusätzliche Sicherheit bietet, wenn den Benutzern und auch den potentiellen Angreifern die genaue Funktionsweise nicht bekannt ist. Diese auf den ersten Blick möglicherweise einleuchtende Logik hat schwerwiegende Konsequenzen:

Erstens lässt sich die Funktionsweise einer Software vor den „richtigen“ Leuten erfahrungsgemäss ohnehin nicht verbergen. Wer es darauf abgesehen hat, die Interna herauszufinden, dem gelingt das bei Einsatz von genügend Energie auch; sei es durch Disassemblieren oder Tracen des ausgelieferten Programmes, durch Methoden des „social engineering“ oder durch einfach nur durch genaues Hinschauen. Das letzteres häufig sehr einfach ist, wurde beispielsweise jüngst beim „T–Online–Hack“ [Luc98] bewiesen, als zwei 16–jährige Realschüler mit sehr geringen Programmierkenntnissen eine Methode fanden, der Windows–Zugangssoftware zum Internet–Anschluss von T–Online die Passwörter der Benutzer zu entlocken.

Zweitens schafft das Nichtvorhandensein einer Spezifikation häufig den trügerische Eindruck von Sicherheit beim Benutzer. Meldungen wie „Daten werden jetzt verschlüsselt“ erzeugen ein Gefühl von Sicherheit, vor allem, wenn der Benutzer nicht weiss, wie der dazu nötige Schlüssel gewonnen wurde und folgedessen dabei auch nichts falsch gemacht haben kann.

Drittens wird die Bewertung des Sicherheitssystemes einer Firma durch externe Beratern erschwert, wenn Sicherheitssoftware ohne genaue Spezifikation zum Einsatz kommt. Viele Produkte, die für sich alleine bestimmten Anforderungen genügen tun dies im Zusammenspiel nicht mehr. Beispielsweise kann eine Software, die zum Zwecke der Angriffserkennung den Datenfluss auf einem Netzwerkstrang protokolliert und auswertet, dazu beitragen, dass Schlüssel oder Passwörter von Authentifikationssystemen im Klartext in Protokolldateien abgelegt werden – nämlich dann, wenn eine Authentifikationssoftware das vom Benutzer eingegebene Passwort über die Leitung schickt, ohne dass der Netzwerkadministrator davon Kenntnis hat.

Viertens hat das Offenlegen der Quelltextes eines Softwarepaketes den unschätzbaren Vorteil, dass weltweit Tausende von Benutzern Fehler in der Software aufspüren und beheben können. Wie sich in zahlreichen Projekten freier Softwareentwicklung gezeigt hat, führt dies zu einer deutlich schnelleren Erkennung und Behebung von Fehlern, als das bei einer geschlossenen Entwickler- und Testergruppe möglich wäre. Dies gilt ebenso für Sicherheitsprobleme. Beispielsweise dauerte es nach der Entdeckung des F00F–Hardwarefehlers in Intels Pentium–Prozessoren, bei dem beliebige Benutzer ohne Administratorrechte die CPU anhalten konnten, nur wenige Stunden, bis für das freie Betriebssystem Linux ein Workaround vorhanden war, der das Problem beseitigt. Bei kommerziellen Betriebssystemen dauerte es erheblich länger, für manche gibt es sogar bis heute keine Möglichkeit, sich vor den Auswirkungen des Fehlers zu schützen.

Sicherheit durch Geheimhaltung der Spezifikation wird häufig als „Security by obscurity“ bezeichnet und sowohl von bekannten Kryptologen wie auch von Firewall–Experten als nicht sinnvoll bezeichnet.

Bei real existierenden Softwareprodukten fallen proprietäre Lösungen häufig mit kommerziellen Produkten zusammen, und offengelegte Lösungen mit frei entwickelter Software. Dies kann daran liegen, dass manche Firmen in der Spezifikation von Sicherheitsmechanismen ihr Geschäftsgeheimnis sehen, das sie nicht preisgeben wollen. Dass es auch anders geht, wurde beispielsweise von Netscape mit der Offenlegung des SSL–Protokolles (siehe Abschnitt 9.7.2) bewiesen. Der Feedback der Öffentlichkeit zu den Sicherheitsschwächen in den ersten beiden Versionen des Protokolles führte zur Entwicklung von SSL 3.0, das allgemein als sehr sicher gilt. Einen ähnlichen Weg ging die finnische Firma Datafellows, die die Secure Shell (SSH; siehe Abschnitt 9.7.1) inklusive Quelltext im Internet zur Verfügung steht.

Ein Ziel des SINUS–Projektes ist es, zu untersuchen, inwieweit frei erhältliche Sicherheitssoftware mit offengelegtem Quellcode dazu geeignet ist, einen adäquaten Schutz für ein firmeninternes Netzwerk zu bieten (vgl. Kapitel 1. Aus den oben genannten Gründen lehnen

wir „Security by obscurity“ ab. In diesem Kapitel werden daher ausschliesslich Produkte mit offengelegtem Quelltext oder zumindest bekannter Spezifikation betrachtet. Das heisst nicht, dass diese Produkte alle Freeware sind; im Gegenteil sind die meisten kommerziell, und zumindest die kommerzielle Nutzung ist an gewisse Einschränkungen (meist Zahlung von Lizenzgebühren) gebunden.

9.3 Auswahl von Netzwerkkomponenten unter Sicherheitsaspekten

Die gängigste für Local Area Networks (LAN) eingesetzte Kommunikationstechnologie ist das Ethernet. Es basiert auf der Grundidee eines abgeschirmten Leiters, an den die einzelnen Rechner hochohmig angeschlossen sind. Jeder Rechner kann Daten senden, indem er kurze, hochfrequente Signale in das Kabel einspeist. Die Signale breiten sich in beide Richtungen aus, bis sie das Kabelende erreicht haben und dort an den Abschlusswiderständen entsorgt werden. Damit ein Rechner gezielt einem anderen Daten schicken kann, gibt es Adressierungsmechanismen; physikalisch gesehen kommen die Daten aber an jedem Rechner auf dem Netzwerkstrang vorbei. Das heisst, dass auch jeder Rechner alle Daten lesen könnte, auch wenn sie gar nicht für ihn bestimmt sind.

An einen Ethernet–Strang können, je nach verwendeter Ausprägung, bis zu mehrere hundert Rechner angeschlossen werden. Aus Sicherheitsgründen kann es jedoch zweckmässig sein, mehrere kleine Netzwerkstränge einzurichten, so dass der Ausbreitungsradius der auf dem Netz gesendeten Daten limitiert wird. Die Aufteilung der Teilnetze geschieht am besten nach organisatorischen Gesichtspunkten, wie in Abschnitt 8.3.1 beschrieben.

Um mehrere Teilnetze miteinander zu verbinden, gibt es unterschiedliche Klassen von Komponenten, die im Folgenden kurz beschrieben werden:

Repeater: Ein Repeater ist das einfachste Gerät zum Verbinden zweier Netzwerkstränge. Er gibt die physikalischen Signale der einen Leitung an die andere weiter und umgekehrt, arbeitet also auf Schicht 1 des OSI–Referenzmodelles. Repeater werden im Allgemeinen dann eingesetzt, wenn ein Netzwerkstrang aufgetrennt werden muss, weil die maximale Länge (je nach Technologie zwischen 180 und 500 Meter) oder die maximale Anzahl von Rechnern überschritten ist. Aus dem Blickwinkel der Sicherheit betrachtet bringt ein Repeater allerdings keinen Gewinn: es sind weiterhin alle Daten von allen Rechnern aus lesbar. Repeater sind also ungeeignet, wenn das Netz aus Sicherheitsgründen aufgetrennt werden soll.

Bridges: Eine Bridge erbringt eine ähnliche Aufgabe wie ein Repeater, nämlich das Weiterleiten eines Datenpaketes von einem Netzwerkstrang auf einen anderen. Im Unterschied zum Repeater arbeitet eine Bridge allerdings auf Schicht 2 des OSI–Referenzmodelles. Das heisst, die Daten werden aus den elektrischen Signalen rekonstruiert, zwischengespeichert und vor der Weitergabe wieder in elektrische Signale umgewandelt. Dadurch ist eine Bridge in der Lage, die Adressierungsinformationen

des Ethernet auszuwerten. Daten werden von einer Bridge nur dann weitertransportiert, wenn der Sender auf der einen und der Empfänger auf der anderen Seite im Netz lokalisiert ist. Damit ist es möglich, dass zwei Rechner auf einem Netzwerkstrang miteinander kommunizieren, ohne das ein dritter Rechner, der an einem über eine Bridge verbundenen Netzwerkstrang angeschlossen ist, diese Kommunikation abhören kann. Allerdings ist dieser Schutz nur bedingt als sicher zu bezeichnen: Die Adressierung in Ethernets basiert weitgehend auf Vertrauen zwischen den einzelnen Rechnern im Netz. Es ist einem bösartigen Rechner ohne weiteres möglich, Pakete mit gefälschter Adressierungsinformation zu verschicken, um die Bridge zu täuschen. Auf diese Weise kann er veranlassen, dass die Bridge bestimmte Daten trotzdem weiterleitet, auch wenn sich Sender und (regulärer) Empfänger auf der selben Seite der Bridge befinden. Als Abtrennung eines Netzes mit hochsensitiven Daten von einem relativ unsicheren Netz ist eine Bridge also nicht geeignet.

Switches: Ein Switch ist im Prinzip nichts anderes als eine Bridge mit mehr als zwei Anschlüssen. Er empfängt die Daten, die auf einem der angeschlossenen Netzwerkstränge transportiert werden, und entscheidet, ob sich der Sender ebenfalls auf diesem Strang oder auf einem anderen angeschlossenen Netzwerkstrang befindet. Ist letzteres der Fall, dann leitet er das Paket an den entsprechenden Strang weiter. Ähnlich wie bei einer Bridge basieren auch bei den meisten Switches die bekannten Adressierungsinformationen auf den Angaben der einzelnen Rechner im Netz; Fälschungen sind ohne weiteres möglich und veranlassen den Switch, das Paket an den Netzwerkstrang des Angreifers weiterzuleiten. Als sicherheitsrelevantes Trennungsinstrument sind diese Switches also ebenfalls nicht geeignet. Lediglich bei einigen wenigen (teuren) Produkten lassen sich die Adressierungsinformationen vom Administrator statisch setzen und sind im Betrieb dann nicht mehr veränderbar; in diesem Fall kann eine sichere Netztrennung erreicht werden.

Router: Ein Router ist für das Weiterleiten von Daten auf der Schicht 3 des OSI–Referenzmodelles (entsprechend Schicht II des TCP/IP–Referenzmodelles, siehe Abschnitt 3.2) zuständig. Dadurch kann er Netzwerke mit unterschiedlicher physikalischer Funktionsweise und unterschiedlicher Adressierungsschemata miteinander koppeln. Seine Adressierungsinformation bezieht er aus der Routing–Tabelle. Diese kann entweder von Hand fest gesetzt werden, oder durch ein dynamisches Routingprotokoll den Verhältnissen angepasst werden. Im ersten Fall lassen sich mit Routern sehr sichere Trennungen zwischen zwei Netzwerksträngen erreichen. Da viele im Handel erhältlichen Router ausserdem Filterungsmöglichkeiten anhand von IP–Nummern und Portnummern bieten, handelt es sich im Prinzip um einfache Paketfilter, wenn auch ohne spezielle Firewall–Funktionen wie Protokollierung, Alarmierung und dynamische Regeln. Werden Routingprotokolle eingesetzt, so können die Routingtabellen extern modifiziert werden und somit der Trennungseffekt eines Routers aufgehoben werden. Moderne Routingprotokolle bieten zwar die Möglichkeit, die übertragenen Informationen kryptographisch zu authentifizieren. Damit lässt sich anhand von Schlüsseln und Zertifikaten genau bestimmen, welchen benachbarten Rou-

tern vertraut werden darf. Allerdings ist die Konfiguration dieser Funktion sehr kompliziert. Für kleine und mittlere Unternehmen ist der Einsatz von Routingprotokollen innerhalb der unternehmensinternen Netze nicht sinnvoll; es sollten stets statische Routingtabellen zum Einsatz kommen.

Ein Repeater ist als nicht dazu geeignet, die Vertraulichkeit der auf einem Netzwerkstrang transportierten Daten gegenüber einem anderen Strang zu schützen. Eine Bridge oder ein Switch sind dazu bedingt geeignet. Bei hohen Sicherheitsanforderungen solle ein Router oder ein Firewall (Paketfilter oder Application Gateway) zum Einsatz kommen.

9.4 Einsatz von Firewalls

9.4.1 Grundlegendes

In Abschnitt 8.5 wurden die grundlegenden Eigenschaften der Firewall–Typen Paketfilter und Application Gateways beschrieben. In Kapitel 12 wird auf die Funktionsweise von Firewalls noch genauer eingegangen. Innerhalb der beiden Produktklassen gibt es nochmals spezielle Unterschiede.

9.4.2 Paketfilter

Paketfilter filtern den Datenverkehr auf IP–Ebene (Schicht II des TCP/IP–Schichtenmodelles). Dazu werten sie die Kontrollinformationen von Internet- und Transportschicht aus und entscheiden anhand eines Regelsatzes, ob das Paket weitergeleitet werden soll oder nicht. Bei den Paketfilterprodukten werden prinzipiell zwei Produktklassen unterschieden:

- zustandslose (reine) Paketfilter, sowie
- zustandsorientierte Paketfilter.

Die Unterscheidung ist nur bei zustandsorientierten Transportprotokollen von Bedeutung. Im TCP/IP–Schichtenmodell ist dies das Transmisson Control Protocol (TCP). Zustandslose Paketfilter entscheiden bei jedem einzelnen einkommenden IP–Paket über die Weiterleitung, unabhängig von eventuellen anderen, bisher gefilterten Segmenten der selben TCP–Verbindung. Zustandsorientierte Filter dagegen verfolgen den Zustand jeder TCP–Verbindung. Dadurch erlauben sie eine bessere Kontrolle über die zu filternden Daten. Die technischen Unterschiede sind in Abschnitt 12.2.5 genauer beschrieben. An dieser Stelle sei nur gesagt, dass ein zustandsorientierter Paketfilter mehrere Vorteile besitzt: zum einen kann er bestimmte Angriffe auf die Zustandsmaschine von TCP erkennen, was ein zustandsloser Paketfilter nicht kann. Zum anderen ist die Konfiguration eines zustandsorientierten Paketfilters einfacher und besser lesbar. Das verringert die Wahrscheinlichkeit

für Konfigurationsfehler, die sich negativ auf die Sicherheit auswirken können. Für Anwendungen mit hohen Sicherheitsanforderungen sind daher zustandsorientierte Paketfilter vorzuziehen. Produktbeispiele sind der SINUS–Firewall, der im Rahmen des SINUS–Projekts entwickelt wurde (siehe Abschnitt 12.5) und mit Einschränkungen auch der BSD Paketfilter (Abschnitt 12.6.2).

Für weniger sensitive Anwendungen kann auch ein zustandsloser Paketfilter verwendet werden. Diese Funktionalität ist in den meisten kommerziellen Routern zu finden, zum Beispiel von Cisco, Bay Networks oder Livingstone. Diese Geräte haben gegenüber Unix–Systemen mit Firewall den Vorteil des geringeren Wartungsaufwandes, der höheren Stabilität und vor allem des höheren Paketdurchsatzes.

9.4.3 Kopplung der Firewalltypen

Die beiden genannten Firewalltypen haben unterschiedliche Vor- und Nachteile. Aus diesem Grund ist es in der Praxis häufig nötig bzw. sinnvoll, beiden Typen miteinander zu koppeln. Abbildung 9.2 zeigt eine generische Anordnung für eine solche Kopplung.

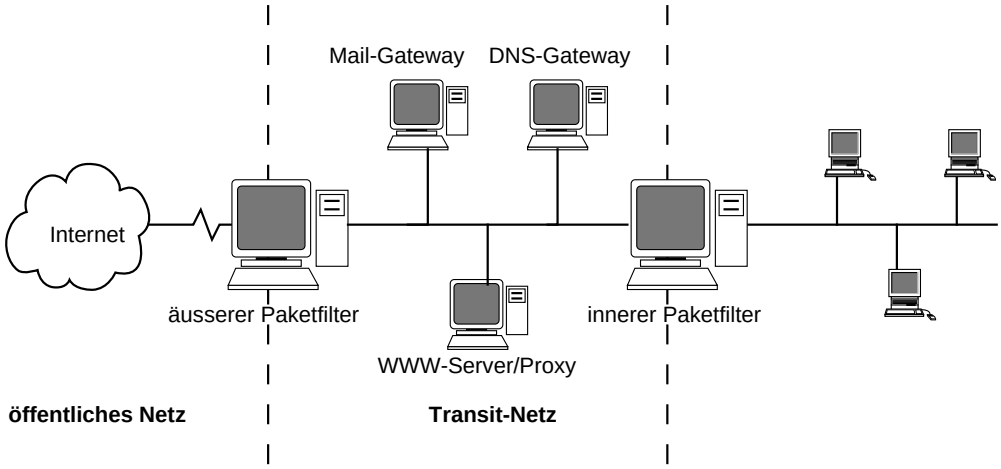

Abbildung 9.2: Kopplung von Paketfilter und Application Gateways

Die gezeigte Anordnung unterteilt das Netz in drei Zonen. Zwischen dem Internet, auf das eine unbegrenzte Anzahl Leute Zugriff hat und das per se als unsicher angenommen werden muss, und dem sicheren internen Netz existiert eine Zwischenzone, hier als Transitnetz bezeichnet. Die Komponenten des Transitnetz stehen nur berechtigten Benutzern für genau definierte Dienste zur Verfügung. Um dies zur erzwingen, wird unter anderem der äussere (linke) Paketfilter benutzt, der gewisse Zugriffsrestriktionen garantieren soll. Dennoch können von ihnen gewisse Bedrohungen ausgehen, weshalb das interne Netz mit einem weiteren Paketfilter vor dem Transitnetz geschützt wird. Die beiden Paketfilter werden stets so konfiguriert, dass sie nie dieselben IP–Datagramme passieren lassen. Datenströme, die vom Internet in das innere Netz oder umgekehrt gelangen sollen, müssen stets einen der Application Gateways passieren.

Für die mittlere Zone werden unterschiedliche Begriffe verwendet. Manche Autoren nennen

sie „Demilitarisierte Zone“ (DMZ), andere „Semitrusted Area“ oder „Transit–Netz“. Uns erscheint der letzte Begriff am passendsten.

Die beiden Paketfilter haben eine unterschiedliche Funktion und müssen daher auch unterschiedlich ausgestattet sein. Grundsätzlich kann angenommen werden, dass für die Rechner der Proxies Betriebssysteme wie Unix zum Einsatz kommen, die man prinzipiell auch selber schützen könnte. Daher sind die Filterungsfunktionen des äusseren Paketfilters von geringerer Wichtigkeit. Statt dessen sollte er gute Fähigkeiten zur Angriffserkennung besitzen, da Angriffe auf das firmeneigene Netz am besten bereits am Eingangsrouter erkannt werden sollen. Schliesslich sollte der äussere Paketfilter gewisse Geschwindigkeitsanforderungen erfüllen, speziell wenn, wie in der Abbildung, von aussen erreichbare Dienste wie ein WWW–Server über ihn erreichbar sind.

Die Proxies und Server sind in der Regel Dienste, die wie normale Programme im User–Space des entsprechenden Betriebssystems ablaufen. Daher sind sie gewissen Bedrohungen ausgesetzt, die sich durch Unzulänglichkeiten der verbreiteten Betriebs- und Anwendungssysteme ergeben. Es muss davon ausgegangen werden, dass bei Bekanntwerden eines Programmierfehlers in der Proxy–Software ein Angreifer die Möglichkeit besitzt, den Rechner beliebige Kommandos ausführen zu lassen. Im ungünstigsten Fall kann dies unter der Benutzerkennung root geschehen. Als root ist es jedoch auch möglich, alle Pakete auf dem lokalen Netzwerk abzuhören oder Pakete mit beliebig gefälschter Absenderkennung zu verschicken. Daher wird im Falle der Kompromittierung der Proxy–Software ein Rechner im Transitnetz zu einer Quelle beliebiger Angriffe auf das innere Netz. Dem inneren Paketfilter kommt daher eine besondere sicherheitsrelevante Bedeutung zu. In vielen praktisch relevanten Fällen ist es glücklicherweise nicht notwendig, dass TCP–Verbindungen vom Transitnetz in das interne Netz aufgebaut werden müssen. Zugriffe von aussen, etwa auf das WWW oder den DNS–Server, werden bereits im Transitnetz abgewickelt, und für Zugriffe von innen auf das Internet werden die dazu benötigten TCP–Verbindungen von innen zu dem Gateway aufgebaut. Der eingesetzte Paketfilter sollte das honorieren und keine TCP–Verbindungen vom Transitnetz in das innere Netz zulassen.

Die in Abbildung 9.2 dargestellte Anordnung ist zwar als sehr sicher anzusehen, hat allerdings den Nachteil eines hohen Hardwarebedarfs. Bei den beiden folgenden Anordnungen werden Hardwarekomponenten eingespart, ohne dass es zu wesentlichen Sicherheitseinbussen kommt.

Wie oben beschrieben, lassen sich beim Einsatz vernünftiger Betriebssysteme im inneren Netz diese auch selber gegen Angriffe auf unbenutzte Ports und Dienste sichern. Gegen Angriffe auf die benutzten, von aussen erreichbaren Dienste ist ein Paketfilter ohnehin nutzlos. Daher besteht die Möglichkeit, den Paketfilter an dieser Stelle wegzulassen (siehe Abbildung 9.3). Die Lösung spart einen Rechner ein, hat jedoch auch Nachteile: zum einen fehlt die Instanz, die für die Angriffserkennung zuständig wäre. Mechanismen zur Angriffserkennung müssen nun auf jedem der Proxy–Rechner einzeln durchgeführt werden. Zum anderen ist die oben genannte Forderung, dass die beiden Paketfilter niemals die selben IP–Datagramme weiterleiten, nicht mehr zu erzwingen. Der innere Paketfilter muss nun so konfiguriert werden, dass er die Rechner des Internet und des Transitnetzes

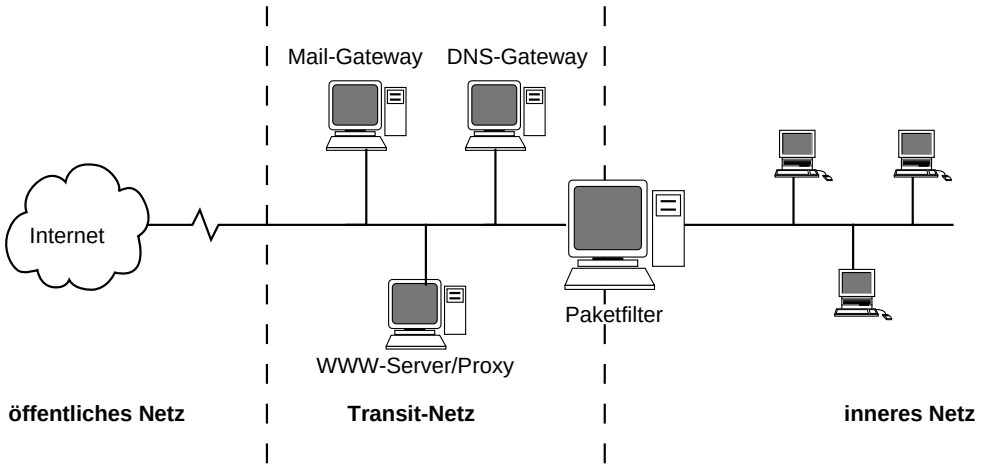

Abbildung 9.3: Einsparung des äusseren Paketfilters

anhand der IP–Nummern unterscheiden kann. Das kann zwar in ungünstigen Situationen die Möglichkeiten für Spoofing–Attacken eröffnen; in den meisten Fällen, wo ohnehin keine TCP–Verbindungen von aussen nach innen aufgebaut zu werden brauchen, spielt das aber keine Rolle.

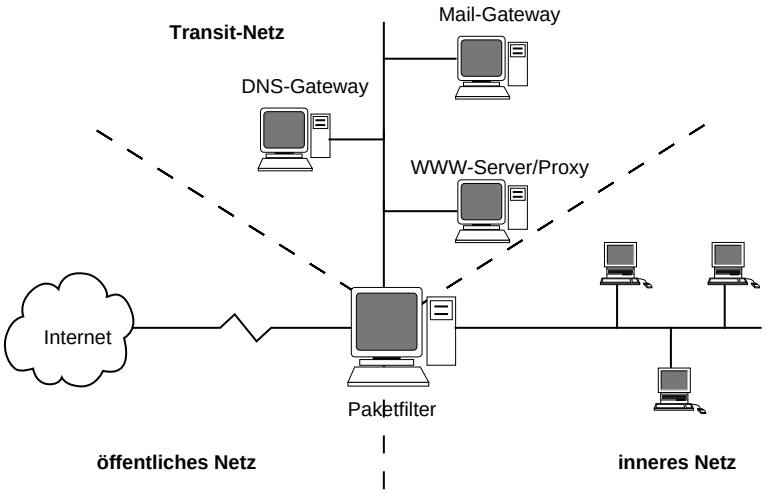

Abbildung 9.4: Ein Paketfilter mit drei Netzen

Eine andere Möglichkeit, einen Rechner einzusparen und trotzdem nicht auf die doppelte Paketfilterung zu verzichten, ist in Abbildung 9.4 skizziert. Hier kommt ein Paketfilter mit drei Netzwerkanschlüssen zum Einsatz. Jeder Datentransport zwischen Internet, innerem Netz und Transitnetz passiert den Paketfilter. Die Anordnung hat keinen prinzipiellen Nachteil gegenüber der zuerst vorgestellten Lösung mit den zwei Paketfiltern. Problematisch ist allerdings, dass die Konfiguration des Paketfilters komplizierter als in den beiden obigen Fällen ist. Der Filter muss so konfiguriert sein, dass er keine Pakete direkt vom Internet in das interne Netz oder umgekehrt passieren lässt. Somit sind alle Anwendungen im inneren Netz gezwungen, Internet–Zugriffe über die Gateways abzuwickeln. Von manchen Autoren wird ausserdem noch ein anderen Nachteil erwähnt: sollte es einem Angreifer gelungen sein, Administratorrechte auf der Paketfilter–Maschine erlangt zu haben, dann kann

er von dort direkt das interne Netz belauschen oder angreifen. Dieser Nachteil ist aber eher akademischer Natur. Da ein Paketfilter in der Regel im Kernel eines Betriebssystems abläuft, sind hier die gängigen Angriffsmethoden wie Buffer Overflow Attacken wirkungslos. Da im Kernel keine Kommandos ausgeführt werden können, kann mit gezielt fehlerhaften Datenpaketen schlimmstenfalls die Maschine zum Absturz gebracht werden. Dies ist jedoch auch in den anderen vorgestellten Anordnungen möglich. Allerdings ist es bei der Lösung mit den drei Netzwerkkarten besonders wichtig, darauf zu achten, dass auf der Paketfilter–Maschine wirklich nur die Filtersoftware und keine Anwendungsdienste laufen.

9.4.4 Auswahl der Firewall–Komponenten

Bei Realisierung der in Abbildung 9.2 dargestellten Anordnung werden an den äusseren Paketfilter keine allzu hohen Sicherheitsanforderungen gestellt. Es kann ein beliebiger Paketfilter oder Router mit Paketfilter–Funktionalität zum Einsatz kommen. Da die Unternehmung meist ohnehin einen Router für die Verbindung ihres LAN mit der Internet–Anbindung (z.B. eine ISDN–Leitung) benötigt, bietet es sich an, beide Funktionen in einem Gerät zu koppeln. Andererseits kann es aus organisatorischen Gründen sinnvoll sein, zwei getrennte Geräte einzusetzen: nämlich, wenn für das Management der Netzwerkkomponenten eine andere Abteilung zuständig ist als für die sicherheitsrelevanten Komponenten; im Extremfall kann sogar eine der beiden Aufgaben von einem externen Outsourcer wahrgenommen werden.

Der innere Paketfilter muss hohen Sicherheitsanforderungen genügen. Er muss das interne Netz vor Angriffen schützen, selbst wenn die Gateways im Transitnetz kompromittiert wurden. Hier sollte eine spezielle Paketfilter–Software zum Einsatz kommen, wie sie in Kapitel 12.2 besprochen wird. Da die Software häufig nur Verbindungen von innen nach aussen zulassen soll, ist ein zustandsorientierter Paketfilter sinnvoll.

Für die Gatways besteht je nach Anwendungsdienst die Wahl zwischen allgemeinen Gateways mit vielen Funktionen oder speziellen Werkzeugen, die auf besonders hohe Sicherheit optimiert sind, dafür aber meist weniger Funktionen bieten. Für die Dienste der wichtigsten Internet–Nutzungsszenarien soll hier eine grobe Orientierungshilfe gegeben werden:

Einkommende E–Mail. Um E–Mail zu empfangen, benötigt die Unternehmung einen SMTP–Server, der zweckmässigerweise in ihrem Transitnetz realisiert wird. Er muss von aussen über SMTP ansprechbar sein, damit andere Mailserver E–Mail abliefern können. Daher ist er speziellen Angriffen ausgesetzt, die durch gezielte Verletzung des SMTP oder des E–Mail–Nachrichtenformates unter Ausnutzung bekannter Sicherheitsschwächen verbreiteter Programme getätigt werden können. Eine weitere Gefahr einkommender E–Mail besteht in der Möglichkeit, gefährliche Dokumenttypen mit Viren oder anderen Schadprogrammen verschicken zu können. Ausser in Spezialfällen empfehlen wir daher die Benutzung eines speziellen, auf Sicherheit hin optimierten SMTP–Gateways für den Empfang von E–Mail aus dem Internet. Ein solches Gateway sollte in der Lage sein, einkommende E–Mail zwischenzuspeichern und vor der Weitergabe auf Viren zu überprüfen.

Ausgehende E–Mail. Die Gefahr, angegriffen zu werden, ist hier deutlich kleiner, da das eigene SMTP–Gateway bestimmt, wann und zu wem es eine Verbindung aufbaut. Prinzipiell können Implementierungsfehler auch bei dieser Verbindungsaufbaurichtung ausgenutzt werden; ein Programmfehler, der einen solchen Angriff ermöglicht, ist aber noch nie in den einschlägigen Quellen bekannt geworden. Da jedoch die meisten Application Gateways für einkommende Mail auch zum Versenden von Mail benutzt werden können, bietet es sich an, auch hier spezielle Sicherheitssoftware zu verwenden. Ausnahmen sind möglich, etwa wenn im Transitnetz grosse Mailinglisten mit hoher Teilnehmerzahl und viel Traffic installiert sind; eine kosten- und netzlastsparende Versendemöglichkeit für solche Listen bieten spezielle MTAs, nicht jedoch die gängige Firewall–Software.

Zugriff von aussen auf den firmeneigenen WWW–Server. Mehr als 50 Prozent der WWW–Server im Internet benutzen für diesen Zweck den frei verfügbaren WWW–Server Apache. Er läuft sehr stabil, ist schnell, wird stetig weiterentwickelt und beherrscht eine grosse Palette von Zusatzfunktionen. Wenn Sicherheitslücken im Apache gefunden werden, werden diese in der Regel nach dem Bekanntwerden schnell behoben. Ausser für ganz spezielle Anwendungen gibt es keinen Grund, ein anderes Produkt als WWW–Server einzusetzen.

Zugriff von innen auf externe WWW–Server. Die Sicherheitsgefahren in diesem Szenario liegen weniger in der Verletzung von Protokollen, sondern mehr in der Ausnutzung prinzipieller Sicherheitsgefahren von WWW–Technologien wie Java–Applets, Javascript–Programmen oder WWW–Dokumenten mit Viren oder anderen Schadprogrammen. Um diese herauszufiltern, wird ein WWW–Gateway benötigt, dass mit den entsprechenden Filterungsfunktionen ausgestattet ist.

9.5 Einsatz kryptographischer Massnahmen

Für Anwendungen, bei denen sicherheitsrelevante Daten über unsichere Leitungen verschickt werden müssen, können kryptographische Massnahmen zu Einsatz kommen. Die grundlegenden kryptographischen Hilfsmittel wurden in Abschnitt 8.4.1 erläutert. Kapitel 11 aus dem dritten Teil dieses Berichtes gibt eine ausführlichere Einführung in das Gebiet der Kryptographie.

In diesem Abschnitt soll der Einsatz kryptographischer Massnahmen für konkrete Anwendungsszenarien betrachtet werden. Dazu werden Produkte und Protokolle vorgestellt, die sich für die Implementierung der Sicherheitsmassnahmen eignen. Alle diese Produkte basieren auf dem vorgestellten Konzept der hybriden Verschlüsselungssysteme.

9.5.1 E–Mail–Sicherheit

Wie in Abschnitt 6.4 erwähnt, werden E–Mails von den gängigen Mail Transport Agents in Internet im Klartext übertragen. Daher sind sie gegen unberechtigte Einsichtnahme

dritter und gegen absichtliche Veränderungen nicht geschützt. Ferner lässt sich die Identität des Autors einer E–Mail leicht fälschen. Zur Sicherung der Vertraulichkeit kann Verschlüsselung, zur Sicherheit der Integrität und Authentizität digitale Signaturen eingesetzt werden. Es gibt verschiedene Standards zur E–Mail–Sicherung; allen gemein ist, dass sie sowohl Verschlüsselung als digitale Signaturen bieten.

9.6 PGP

Pretty Good Privacy (PGP) ist eine Implementierung eines hybriden Kryptosystems. Es erlaubt das Verschlüsseln und/oder Signieren von Dateien und bietet eine komfortable Schlüsselverwaltung.

Intern basiert PGP auf dem Public–Key–Kryptosystem RSA, der Blockchiffre IDEA und der kryptographischen Hashfunktion MD5 (s. dazu Kapitel 11). Jeder Teilnehmer kann sich ein Schlüsselpaar selbst erzeugen und seinen öffentlichen Schlüssel verbreiten; eine Vertrauensinstanz ist nicht notwendig.

PGP ist prinzipiell für die Verschlüsselung und Signatur beliebiger Dateien geeignet. Sein Hauptanwendungsgebiet ist jedoch die Sicherung von E–Mail. Das hängt unter anderem damit zusammen, dass PGP das erste Programm dieser Art war und die Benutzung symmetrischer Verschlüsselungsprogrammen für E–Mail besonders unpraktisch ist.

PGP zeichnet sich durch interessante Zusatzfunktionen aus. So kann eine verschlüsselte Datei in ein Format gebracht werden, das nur aus reinen ASCII–Zeichen besteht und damit ohne weitere Konvertierung direkt als Mail verschickbar ist. Beim Verschlüsseln können mehrere öffentliche Schlüssel benutzt werden, so dass ein Chiffrat für mehrere Empfänger entschlüsselbar ist.

Wie bei allen Public–Key–Systemen ist ein Schwachpunkt der Austausch des öffentlichen Schlüssels. Wer eine Nachricht mit einem öffentlichen Schlüssel chiffriert, muss man sicherstellen, dass dies wirklich der Schlüssel des gewünschten Empfängers ist. Ansonsten könnte ein Angreifer einen selbstgenerierten Schlüssel unter dem Namen eines anderen Netzteilnehmers veröffentlichen. Er besässe dann den zugehörigen privaten Schlüssel und könnte Chiffrate an den öffentlichen Schlüssel dechiffrieren.

Zur Überprüfung der Authentizität eines Schlüssels bietet PGP einen Mechanismus an: Der Fingerprint ist eine kryptographische Hashsumme über den öffentlichen Schlüssel. Jeder Teilnehmer kann ihn selbst berechnen und auf einem sicheren Kanal (am besten persönlich; für normale Sicherheitsanforderungen genügt wohl auch ein Telefonat) mit dem Besitzer vergleichen. Schlüssel, bei denen man sicher ist, dass sie dem vorgeblichen Besitzer gehören, kann man mit dem eigenen privaten Schlüssel signieren und signiert weitergeben. Auf diese Weise entsteht das „Web of Trust“: Wenn ein Teilnehmer A den öffentlichen Schlüssel von B signiert hat (also sicher ist, dass der zugehörige private Schlüssel dem Teilnehmer B gehört) und der Schlüssel von C eine Signatur von B trägt (angefertigt mit dem privaten Schlüssel von B), dann hat A damit einen Beweis der Authentizität des Schlüssels von C. Voraussetzung ist natürlich, dass B den Schlüssel wirklich vor dem Signieren überprüft hat. PGP

findet solche Vertrauensketten automatisch bzw. gibt eine Warnung aus, wenn ein Schlüssel benutzt wird, zu dem keine Vertrauenskette besteht.

Da PGP die erste brauchbare Implementierung eines Public Key Kryptosystems war, die es auch unbedarften Nutzern ermöglichte, ihre E–Mails zu verschlüsseln, löste es heftige Streitigkeiten um rechtliche Fragestellungen aus. Genauere Angaben dazu finden sich in Abschnitt 11.4.1 im dritten Teil dieses Berichts.

9.6.1 S/MIME

Ebenso wie PGP ist S/MIME ein Protokoll zur Sicherung von E–Mails mittels Verschlüsselung und digitaler Signaturen. Anders als bei PGP wurde bei S/MIME versucht, das zugrundeliegende Protokoll von vorne herein zu standardisieren. Geplant war die Anmeldung als IETF–Standard. Dieses Vorhaben ist jedoch gescheitert.

Auch S/MIME bedient sich eines hybriden Ansatzes zur Verschlüsselung. Im Gegensatz zu PGP beherrscht S/MIME keine Vertrauensketten, sondern lediglich eine Spezialform, nämlich hierarchische Vertrauensbeziehungen. Das bedeutet, dass der öffnetliche Schlüssel eines Benutzers von einer Zertifizierungsstelle signiert sein kann, und der öffentliche Schlüssel dieser Stelle wiederum von einer übergeordneten Zertifizierungsinstanz. Die Wurzel eines solchen Baumes, häufig „Root–CA“ genannt, genauer gesagt deren öffentlicher Schlüssel, muss dem Benutzer bekannt sein. Ein Signieren dieses Schlüssels ist nicht notwendig.

S/MIME wird von den Mail–Clients der bekannten WWW–Browser von Netscape und Microsoft unterstützt. Sie haben von Hause aus die öffentlichen Schlüssel bekannter kommerzieller Zertifizierungsstellen eingebaut. Daher besteht grundsätzlich die Gefahr, dass bereits beim Herunterladen der genannten Browser von unsicheren Web- oder FTP–Servern die Schlüssel gefälscht sind. Diese Gefahr kann jedoch als gering eingeschätzt werden, da dies innerhalb kurzer Zeit bemerkt und publik gemacht würde. Dennoch bieten seriöse Zertifizierungsunternehmen zusätzliche Wege zur Überprüfung der Echtheit ihres Schlüssels an. Darüber hinaus bieten die Browser die Möglichkeit, weitere Schlüssel in die Liste mitaufzunehmen. Auch hier sollten diese Schlüssel vorher auf Authentizität überprüft werden.

S/MIME ist darauf ausgelegt, fehlende Glieder in der Vertrauensbeziehung online aus dem Netz zu laden. Der Benutzer braucht daher, anders als bei PGP, kein eigenes Verzeichnis öffentlicher Schlüssel seiner Kommunikationspartner zu führen. S/MIME bedient sich des X.509–Standards für Schlüssel und Zertifikate, ein Bestandteil des X.500–Verzeichnisdienstes, eine Art elektronisches Telefonbuch iom Internet.

Die Benutzung von S/MIME ist derzeit noch mit Problemen verbunden. S/MIME ist sehr wenig verbreitet, daher haben bisher kaum Benutzer öffentliche Schlüssel in diesem Format. Im Gegensatz dazu hat PGP eine recht grosse Verbreitung, zumindest im privaten und universitären Bereich. Der Zugriff auf X.500–Verzeichnisdienste ist problematisch, wenn der Mailrechner während der Erstellung einer Mail keinen Internetzugriff hat. Dies ist, zumindest bei Privatleuten und kleinen Firmen mit Wählleitungen, die Regel. Das grösste Pro-

blem sind jedoch die Mail–Programme. Die bisher verfügbaren wurden alle in den USA programmiert und unterliegen den Exportrestriktionen für kryptographische Software. Daher haben symetrische Verschlüsselungsverfahren darin nur 40 Bit, asymetrische 512 Bit Schlüssellänge. Beides ist zu gering, um bei der heute verfügbaren Rechenleistung nicht gebrochen zu werden.

9.7 Sicherung von Transaktionen

Neben Offline–Anwendungen wie E–Mail gibt es auch bei Online–Anwendungen, also solchen mit ständiger Verbindung zu einem Kommunikationspartner, Bedarf an kryptographischen Mechanismen. Auch hierzu existieren mehrere Lösungen. Allen gemeinsam ist, dass sie ihre Dienste unterhalb der Anwendungsschicht im TCP / IP–Schichtenmodell erbringen. Die Anwendung kann also Verbindung (im Sinne von TCP- oder rsh–Verbindungen) zur Gegenseite aufbauen, ohne sich um die darunterliegenden Sicherheitsmassnahmen kümmern zu müssen.

9.7.1 SSH

Die Secure Shell (SSH) ist als ein sicherer Ersatz für Telnet, rlogin, rsh und rcp gedacht. Sie wurde 1995 von Tatu Ylonen aus Finnland entwickelt und wird frei im Internet vertrieben. Mittels kryptographischer Techniken ermöglicht sie eine weitgehende Resistenz gegen die bisher vorgestellten Angriffstechniken.

Verschlüsselung. Alle beteiligten Rechner besitzen RSA–Schlüsselpaare. Am Anfang der Verbindung wird ein Sitzungsschlüssel generiert und mit RSA verschlüsselt ausgetauscht. Die gesamte Kommunikation geht mit diesem Sitzungsschlüssel chiffriert über die Leitung, wobei IDEA, DES oder andere symmetrische Algorithmen verwendet werden. Ein Angreifer kann durch reines Abhören keine Informationen gewinnen, die ihn zum Einloggen in das System befähigen. Der RSA–Schlüssel, mit dem die symmetrischen Schlüssel ausgetauscht werden, wird jede Stunde neu erzeugt und niemals in einer Datei abgespeichert. Das verhindert, dass aufgezeichnete verschlüsselte Datenströme dechiffriert werden können, wenn ein Angreifer zu einem späteren Zeitpunkt Root–Rechte auf dem Serverrechner bekommt und die geheimen Schlüssel auslesen kann.

Authentifizierung. Alle beteiligten Rechner besitzen RSA–Schlüssel und authentifizieren sich gegenseitig beim Verbindungsaufbau. Zunächst werden die öffentlichen Schlüssel ausgetauscht und mit den bereits bekannten verglichen. Wenn die korrekten öffentlichen Schlüssel bereits vorher bekannt sind, können auf diese Weise aktive Angriffe, bei der ein Angreifer den Schlüssel gegen seinen eigenen austauscht, erkannt werden. Lediglich beim allerersten Verbindungsaufbau besteht die Möglichkeit eines aktiven Angriffes, wenn die öffentlichen Schlüssel noch nicht vorher ausgetauscht wurden. Anschliessend findet eine Authentifizierung über RSA statt.

X11–Verbindungen. Bei einem Login mit SSH auf einem entfernten Rechner können X–Clients gestartet werden, ohne dass diese eine TCP–Verbindung auf den X–Server öffnen müssen. Dadurch entfallen die in Abschnitt 4.7 geschilderten Sicherheitsprobleme.

Einloggen ohne Passwort (wie bei den r–Tools) ist ebenfalls möglich. Allerdings verlässt sich SSH nicht auf IP–Nummern, Rechnernamen und Benutzernamen, sondern nur auf RSA–Authentifizierung.

TCP–Forwarding. Beliebige TCP–Verbindungen können über eine SSH–Verbindung übertragen werden (Tunneling) und werden somit automatisch gegen Abhören und Spoofing–Attacken geschützt. Die Bedienung ist allerdings umständlich und nicht für jeden Dienst geeignet. Ausserdem wird vorausgesetzt, dass die Kommunikationspartner einen Account auf dem Zielsystem besitzen. Dies ist nicht bei allen

SSH–Server gibt es nur für Unix, frei erhältliche Clients für Unix und OS/2, kommerzielle Clients für MS Windows.

SSH bietet keinen Mechanismus zum kryptographisch authentischen Austausch öffentlicher Schlüssel. Der Benutzer/Systemadministrator muss sich selbst darum kümmern. SSH bemerkt allerdings, wenn sich ein bereits bekannter Host–Schlüssel geändert hat, und warnt den Benutzer. vor einer möglichen Man–in–the–Middle–Attacke.

9.7.2 SSL

Secure Socket Layer (SSL) ist ein von der Firma Netscape entwickeltes Protokoll, das zwischen den Schichten III und IV im TCP/IP–Schichtenmodell arbeitet. Es bietet transparente Verschlüsselung und Authentizitätsprüfung mittels hybriden kryptographischen Verfahren. SSL wird vor allem für die sichere Übertragung von Dokumenten im WWW eingesetzt, aber auch andere Anwendungsszenarien sind möglich.

WWW–Server mit SSL–Fähigkeit sind weltweit frei verfügbar. Der bekannteste dürfte der Apache (http://www.apache.org/) sein, der mittels der ebenfalls frei verfügbaren Bibliothek SSLeay SSL–fähig wird. WWW–Browsern mit SSL–Fähigkeit gibt es zwar auch. Jedoch kommen die beiden verbreitetsten, von Netscape und Microsoft, aus den USA und unterliegen den Exportrestriktionen für kryptographische Software. Somit sind ausserhalb der USA für Verschlüsselung nur 40 Bit symmetrisch bzw. 512 Bit asymetrisch möglich. Dies mag für einfache Warenbestellungen mit Kreditkartennummer ausreichend sein; für wichtige Finanztransaktionen ist es das definitiv nicht. Bei Anwendungen, wo es lediglich auf die Authentifizierung ankommt, spielen die Exportrestriktionen hingegen keine Rolle: die Version 3 von SSL benutzt dafür einen eigenen Schlüssel von 128 Bit Länge, unabhängig von der Länge des Verschlüsselungs–Keys.

Es gibt mehrere Ansätze zur Beseitigung des Exportproblems:

Aus Australien kommt ein Patch namens Fortify für die Browser von Netscape. Sie verwandeln einen Export–Browser in einen, der starke Verschlüsselung beherrscht. Von Seiten des Exportgesetze her dürfte das legal sein. Unklar ist jedoch die Frage, ob solchermassen veränderte Software noch unter die erworbenen Lizenzen von Netscape fällt oder ei-

ne Raubkopie darstellt. Ausserdem ist es den meisten Anwendern nicht zuzumuten, ihren Browser zu patchen. Daher ist Fortify für den kommerziellen Einsatz eher ungeeignet, auch wenn die Software fürt Privatleute eine interessante Variante ist.

Viele Banken in Deutschland und auch in der Schweiz setzen für das Internet–Banking eine Verschlüsselungssoftware ein, die als Java–Applet läuft. Sie ermöglicht eine 1024 / 128–Bit–Verschlüsselung unabhängig von den Verschlüsselungseigenschaften des darunterliegenden SSL. Die Lösung ist jedoch sehr aufwendig und teuer und kommt daher für kleine und mittlere Unternehmen im Normalfall nicht in Frage.

Einsatz eines ausserhalb der USA programmierten Browsers: Auf dem Browsermarkt gibt es einige interessante Entwicklungen. So unterstützt zum Beispiel eine Version von NCSA Mosaic starke Verschlüsselung mit SSL auf Basis der freien Bibliothek SSLeay. Mosaic ist in anderen Bereichen dem Netscape Navigator unterlegen, beispielsweise durch das Fehlen von Javascript und von Frames. Die Entwicklung schreitet jedoch schnell voran, und wo bei graphischen Goodies Abstriche zugunsten der Sicherheit gemacht werden kann, ist NCSA Mosaic eine interessante Alternative.

Kapitel 10

RSD–XPS: Ein Expertensystem für die Internet–Sicherheitskonzeption

Im Rahmen des SINUS–Projektes wurde das prototypische Expertensystem RSD–XPS entwickelt. RSD–XPS soll den Anwender bei der Umsetzung des Rapid Secure Development (RSD) unterstützen. RSD ist ein Leitfaden für die sichere Implementierung von Online–Diensten, bei dem die Sicherheit von Anfang an in den Gestaltungsprozess integriert wird. Auf diese Weise können nicht nur die Sicherheitsmassnahmen benutzerfreundlicher gestaltet, sondern auch die Dienste selbst schneller bereit gestellt werden. Die Sicherheitskonzeption wurde vor allem für kleinere bis mittlere Unternehmen entwickelt, die grösstenteils bis heute noch nicht über das nötige Wissen im Bereich der Internet–Technologie verfügen, um sie sicher einzusetzen. Das Expertensystem RDS–XPS wurde entwickelt, um dem Benutzer ein Softwarehilfsmittel zur Verfügung zu stellen, welches ihn Schritt für Schritt durch die RSD–Sicherheitskonzeption führt. Das System bietet dem Benutzer neben einer effizienten Durchführung der Sicherheitskonzeption und damit einer schnelleren Anpassung auch die Möglichkeit, sein Wissen über die Internet–Technologie und ihre sichere Nutzung zu erweitern und zu vertiefen.

Ziel dieses Kapitels ist es das RSD–XPS vorzustellen. Zu diesem Zweck werden zu Beginn kurz verschiedene Expertensystemtechniken eingeführt und die eingesetzten Techniken vorgestellt. Im Anschluss daran wird gezeigt, inwieweit die Sicherheitskonzeption in das System integriert wurde.

10.1 Expertensystemtechniken

Der vorliegende Abschnitt soll einen kurzen Einblick in das RSD–XPS von der Seite der Expertensystemtechniken geben. Neben den verschiedenen Möglichkeiten der Wissensrepräsentation werden die Komponenten des Expertensystems und das Vorgehen für die Entwicklung des Expertensystemen angesprochen. Neben den hier aufgeführten Techniken und Methoden gibt es noch zahlreiche andere, deren Nennung allerdings den Rahmen dieses Berichtes sprengen würde.

10.1.1 Klassifikation des Expertensystems

Die Klassifikation von Expertensystemen ermöglicht es von den Erfahrungen ähnlicher Problemstellungen zu profitieren. Es gibt verschiedene Kriterien für die Klassifizierung. Weit verbreitet ist die Klassifikation von Expertensystemen nach Art des zu lösenden Problems. Puppe teilt die Anwendungen in folgende Problemlösungen ein [Pup90]:

Diagnostik/Klassifikation: Die Diagnostik hat sich aus der Medizin heraus entwickelt und wird heute vor allem in technischen Systemen eingesetzt. Hierbei wird die Lösung aus einer vorgegebenen Menge von Alternativen ausgewählt. Die Abduktion, d.h. das Schliessen aufgrund von Beobachtungen auf Systemzustände, ist ein typisches Vorgehen für ein Diagnostik–Expertensystem.

Konstruktion: Bei der Konstruktion wird die Lösung aus kleinen Bausteinen zusammengesetzt. Zu dem Problemlösungstyp Konstruktion zählen die Planung, die Konfiguration und die Zuordnung [MFPW95].

Simulation: Die Simulation dient zur Vorhersage eines Systemverhaltens. Ausgehend von einem Anfangszustand und bekannten Schritten werden mögliche Folgezustände gesucht. Expertensysteme zur Simulation dienen z.B. zum Testen oder Überprüfen einer Verdachtsdiagnose.

Anhand des Problemlösungstyps kann auf die geeigneten Wissensrepräsentationsformalismen geschlossen werden (Abbildung 10.1).

Wissensrepräsentation \ Problemlösungstyp	Diagnostik	Konstruktion	Simulation
Logik	○	○	○
Produktionsregeln	●	●	●
Objekte	●	●	●
Einschränkungen	○	●	●
probabilistisches Schliessen	●	○	○
nicht-monotones Schliessen	○	○	○
temporales Schliessen	○	○	●

● geeignet ○ evtl. geeignet

Abbildung 10.1: Zuordnung von Problemlösungsklassen zu Wissensrepräsentationsformalismen nach Puppe

Klassifikation des RSD–XPS

Beim RSD–Verfahren werden, ausgehend von Anforderungen, Massnahmen für die sichere Nutzung von Online–Diensten vorgeschlagen. Da die Lösungen in diesem Fall aus einer Menge von Elementen ausgesucht werden, ist das RSD–XPS primär zum Problemlösungstyp Diagnostik bzw. Klassifikation zuzuordnen.

10.1.2 Wissensrepräsentation

Aufgabe der Wissensrepräsentation ist die Entwicklung, Bereitstellung und Verarbeitung der Formalismen zur Abbildung von Expertenwissen auf maschinenverarbeitbare Operationen.

Formalismen zur Wissensrepräsentation

Formalismen zur Wissensrepräsentation sind Werkzeuge zur Modellierung des Expertenwissens. Sie sollen die Strukturierung und Wartbarkeit des menschlichen Wissens [DG97] unterstützen. Es gibt eine Vielfalt von Formalismen. Daher werden an dieser Stelle nur die vorgestellt, die im Prototypen Verwendung gefunden haben.

Logik: Die Logik ist der am weitesten verbreitete und flexibelste Formalismus. Im Speziellen wird die Prädikatenlogik erster Ordnung (PL1) angewandt. Die Darstellung von Wissen mit Hilfe der PL1 ist deklarativ, d.h. das Wissen wird in Form von Aussagen, die auch für sich allein genommen gültig sind, gespeichert. Durch die Loslösung des Wissens von der Ausführungsreihenfolge wird es leichter verständlich und wartbar. Die PL1 ist sehr ausdrucksstark, aber gerade deswegen ist es oftmals nicht möglich effiziente Verfahren für ihre Verarbeitung anzugeben. Sie ist auch nicht für die Bearbeitung von Schlussfolgerungen geeignet, die auf unsicherer Information aufbauen.

Produktionsregeln: Auch Produktionsregeln sind ein weit verbreitetes Verfahren zur Wissensdarstellung. Eine Regel besteht aus einer Vorbedingung und einer Aktion. Die Aktion ist entweder eine Implikation oder Deduktion, durch die ein Wahrheitsgehalt einer Festellung hergeleitet, oder eine Handlung, durch die ein Zustand verändert wird. Für die Regelausführung gibt es prinzipiell zwei Alternativen, die Vorwärtsverkettung und die Rückwärtsverkettung. Nachteil der regelbasierten Wissensrepräsentation ist die Unstrukturiertheit der Regeln und der schwer verständliche Kontrollfluss bei deren Abarbeitung.

Objekte: Durch die Wissensrepräsentation mittels Objekten wird eine eine bessere Strukturierung der Wissensbasis erzielt. Diese Art der Wissensrepräsentation ist dazu geeignet, Objekte zu beschreiben und in Beziehung zu setzen. Die Aussagen über ein Objekt werden in sogenannten Klassen oder Frames zusammengefasst. Frames werden eine Menge von Eigenschaftstypen, sogenannten Slots zugeordnet. Ein Slot ermöglicht die systematische Speicherung von Default–Wissen über die Eigenschaften oder Attribute, die ein Objekt besitzt. Slots können an Prozeduren gebunden sein, welche aktiviert werden, sobald der Wert des zugehörigen Slots modifiziert wird. Die Vererbung von Eigenschaften ist eines der wichtigsten Konzepte der objektorientierten Wissensrepräsentation. Sie ermöglicht eine ökonomische Datenhaltung, da die Eigenschaften jeweils nur an einem Ort gespeichert werden müssen.

Wissensrepräsentation im RSD–XPS

Der grösste Teil des Wissens wird im RSD–XPS mit Objekten und Produktionsregeln dargestellt. Wie die Modellierung des Wissens mit Hilfe dieser Formalismen genau aussieht, wird im folgenden Abschnitt beschrieben.

10.1.3 Wissensmodellierung

Die Wissensmodellierung befasst sich hauptsächlich mit der inhaltlichen Seite der Wissensrepräsentation, indem sie das Wissen kategorisiert und die Beziehungen zwischen den einzelnen Kategorien festlegt. Die Wissensrepräsentationsformalismen stellen die Hilfsmittel für die Modellierung bereit. Die Wissensmodellierung umfasst die Wissensakquisition und die Wissensoperationalisierung. Hauptaufgabe der Wissensakquisition ist die Erhebung und Formalisierung des Expertenwissens. Die Wissensmodellierung stellt den Übergang vom konzeptuellen Modell in ein maschinenhandhabbares Wissensmodell dar.

Wissensmodellierung im RSD–XPS

Die Szenarien Dienste, Gefahren, Gegenmassnahmen und Realisierungsmassnahmen stellen reale Objekte dar. Aus diesem Grund wurde für die Darstellung dieses Wissens Objekte gewählt. Die Gemeinsamkeiten dieser Objekte wurden jeweils in einem Frame zusammengefasst. Das verbindende Element dieser Objekte sind die jeweiligen Tabellen aus dem RSD–Verfahren. Diese Tabellen verbinden Szenarien mit Internet–Diensten, Internet–Dienste mit Gefahren usw. Bei der sicheren Klassifikation gibt es zwei Haupttechniken für die Wissensrepräsentation: Entscheidungsbäume und -tabellen. Da das Wissen schon in Tabellenform vorliegt, wurde diese Repräsentationsform direkt in das System übernommen. Bei der Umsetzung der Entscheidungstabellen werden vorwärtsverkettete Regeln verwendet, wobei die Regeln Implikationen darstellen. Eine Ausnahme bildet die Verbindung von den Gegenmassnahmen zu den Realisierungsmassnahmen. Da sich die Umsetzung einer Gegenmassnahme schneller ändern kann als die Gefahren und Konzepte der Gegenmassnahmen, unterliegt der letzte Teil einer ständigen Änderung. Um diese Anpassung schnell und unkompliziert bewerkstelligen zu können, sucht sich die Gegenmassnahme ihre entsprechenden Realisierungsmassnahmen.

Für jede Regel, die durchgeführt werden konnte, wird eine Klausel in der Prolog–Datenbasis abgelegt. Diese Klausel gilt als bewiesen, ist also ein Faktum. Diese Faktenbasis stellt das fallspezifische Wissen und die Zwischenresultate dar. Sie wird unter anderem für die Erklärungskomponente und für das nicht–monotone Schliessen verwendet.

Das Problemlösungsverfahren wird in einem übergeordneten Frame gespeichert. Dieser Frame weiss, welche Regeln in welche Phase ausgeführt werden sollen. Der Frame wird Sicherheitskonzeption genannt und besitzt fünf Instanzen. Diese entsprechen den fünf Phasen des RSD–Verfahrens. Abbildung 10.2 zeigt die Konzeption dieses Frames.

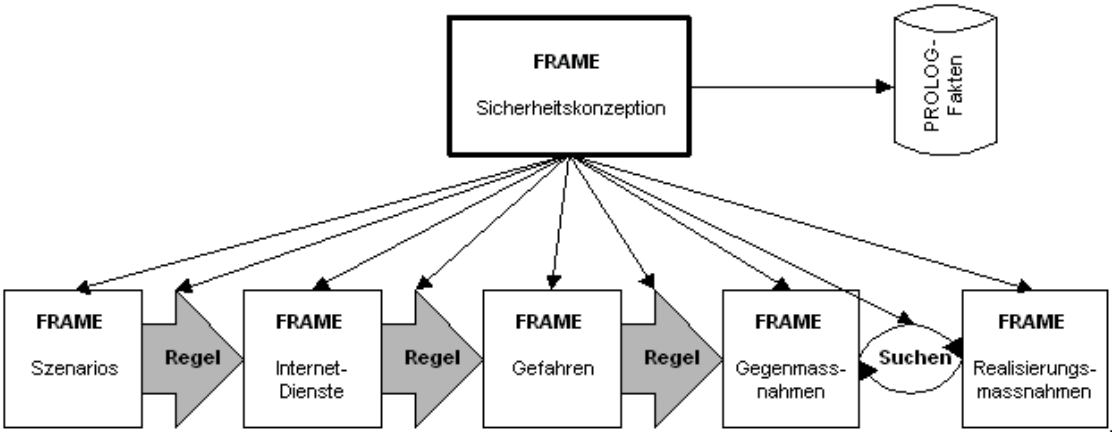

Abbildung 10.2: Konzepte des Expertensystems

10.1.4 Softwarewerkzeuge

Für die Realisierung des Expertensystems existieren eine Anzahl von Softwarewerkzeugen, die in die drei Klassen KI–Programmiersprachen, Expertensystem–Shells und Expertensystemwerkzeuge eingeteilt werden können [MFPW95].

KI–Programmiersprachen: Für die Entwicklung von Expertensystemen wurden eine Anzahl von speziellen Programmiersprachen entworfen. LISP und PROLOG sind die bekanntesten Vertreter dieser Sprache.

Expertensystem–Shells: Shells sind Werkzeuge, die auf eine bestimmte Klasse von Anwendungen zugeschnitten sind. Die Art des Wissens und das Problemlösungsverfahren sind fest definiert, die Shell muss nur noch mit dem bereichsspezifischen Wissen gefüllt werden. Gewisse Shells enthalten aber auch schon Basiswissen.

Expertensystemwerkzeuge: Expertensystemwerkzeuge bieten verschieden Wissensrepräsentationsformalismen und elementare Inferenzverfahren an. Sogenannte Hybride Werkzeuge bieten mehrere Repräsentationsformalismen gleichzeitig an, die untereinander integriert sind. Dadurch werden sie relativ mächtig.

Softwarewerkzeuge für das RSD–XPS

Für die Implementierung des Expertensystems wurde das EMA–XPS der Universität Wuppertal verwendet. Das hybride EMA–XPS ist ein frei verfügbares grafisches Expertensystemwerkzeug. Es ergänzt die KI–Werkbank Babylon 2.3 der Gesellschaft für Mathematik und Datenverarbeitung (GMD) um eine grafische Oberfläche auf der Basis von X–Windows und Motif. Babylon 2.3 wurde vollständig in CommonLISP programmiert. EMA–XPS hat den Vorteil, dass es auf allen UNIX–Systemen lauffähig ist. Ausserdem ist es ein sehr mächtiges Tool, das viele Problemtypen unterstützt.

10.2 Unterstützung der Sicherheitskonzeption durch RSD–XPS

Eine Aufgabe des Expertensystems besteht darin, den Benutzer bei der RSD–Sicherheitskonzeption zu unterstützen, d.h. ausgehend von den Unternehmungsaufgaben die Internet–Technologie sicher einzusetzen. Daher ist es notwendig, dass der Problemlösungsprozess des Systems analog zu dem Verfahren in der Sicherheitskonzeption aufgebaut ist. Für die Akzeptanz des Systems ist es wichtig, dass das System alle Entscheidungen begründen kann.

Durch den Einsatz des Expertensystems, welches nach einem strikt vorgegebenen Problemlösungsprozess vorgeht, können mögliche Flüchtigkeitsfehler vermieden und somit die Konzeption effizienter gestaltet werden. Daneben kann das System dem Benutzer dabei helfen, seine Kenntnisse über die Sicherheitskonzeption und die Internet–Technologie zu erweitern und zu vertiefen. Für diesen Zweck muss das System die in der Wissensbasis verwaltete Information in geeigneter Form bereitstellen.

Basierend auf dem RSD–Ansatz führt das Expertensystem den Benutzer durch die fünf Phasen

1. Nutzungskonzeption
2. Dienstauswahl
3. Risikoanalyse
4. Gegenmassnahme
5. Realisierung

Anhand dieser Phasen wird im folgenden gezeigt, wie das System die RSD-Konzeption unterstützt. Die einzelnen Phasen werden dabei nochmals kurz erläutert und die sich aus ihnen ergebenden Systemanforderungen vorgestellt. Eine genaue Beschreibung der einzelnen Phasen des RSD–Verfahrens findet sich in Teil II.

Die Oberfläche des RSD–XPS ist in drei Teile gegliedert (s. Abbildung 10.3): einen oberen Teil, die Schrittwahl (1), in der die verschiedenen Schritte des RSD–Verfahrens ausgewählt werden können; einen linken Teil, den Funktionsteil (2), in dem die Knöpfe eingeblendet werden, die für den ausgewählten Schritt benötigt werden, und der die eigentlichen Funktionen innerhalb eines Schrittes darstellt; schliesslich einen rechten Teil, den Ausgabeteil (3), in dem die Ausgabe des Expertensystems dargestellt wird. Es verhält sich wie ein Textfenster, in dem nach Belieben in alle Richtungen gerollt werden kann.

10.2.1 Nutzungskonzeption

Ziel der Nutzungskonzeption ist eine genaue Definition des Einsatzgebietes und damit verbunden eine Vorselektion der benötigten Internet–Dienste. Auf diese Weise kann sichergestellt werden, dass wirklich nur die für die vorgesehene Nutzung notwendigen Dienste eingesetzt werden und damit unnötige Risiken vermieden werden können.

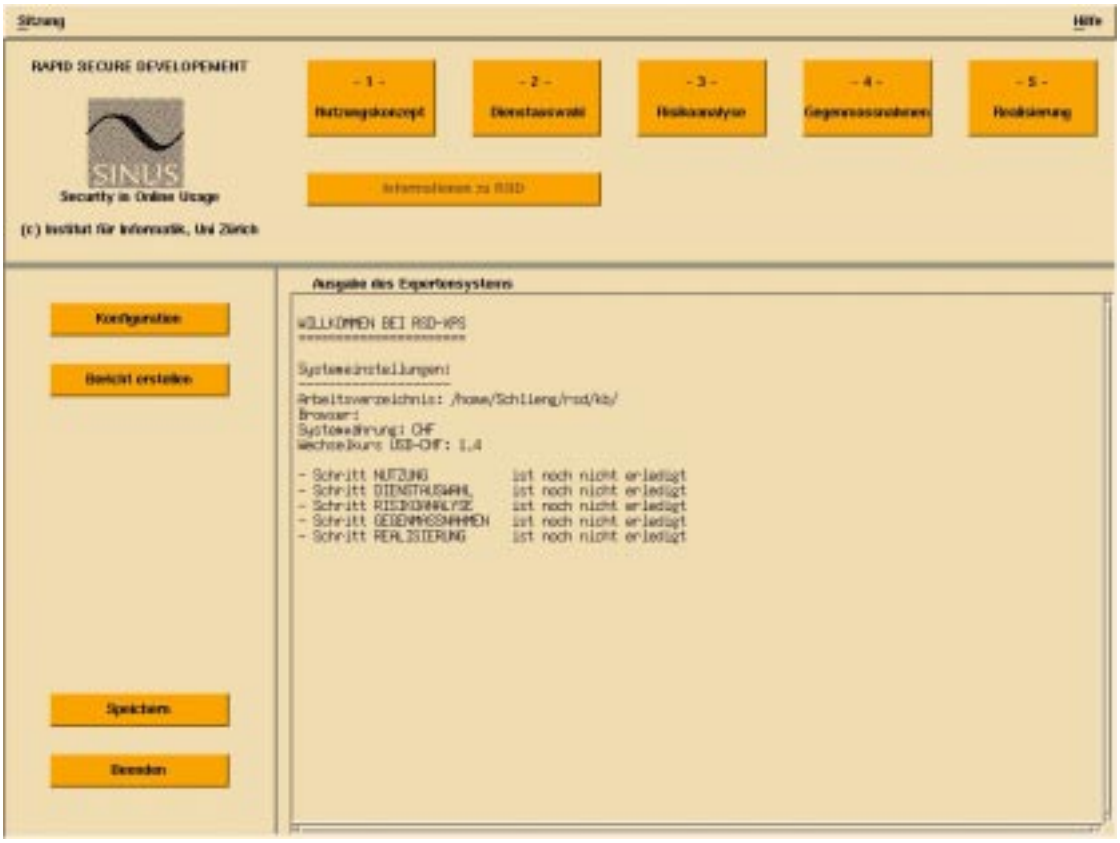

Abbildung 10.3: Die RSD–XPS Oberfläche nach dem Start

Zielgruppe / Nutzungsmöglichkeit	intern	extern	öffentlich	individuell
Informationsbeschaffung	●	●	●	●
Informationsbereitstellung	●	●	●	●
elektronischer Handel	●	●	●	
Kooperation	●	●		

Abbildung 10.4: Nutzungsszenarien

Daher muss der Benutzer in dieser Phase der Konzeption ein Nutzungsszenario auswählen. Dabei setzt sich ein Nutzungsszenario in der Regel aus Nutzungsmöglichkeit und der Zielgruppe zusammen. Für die Auswahl des Szenarios werden dem Benutzer verschiedene vordefinierte Szenarien präsentiert. Abbildung 10.4 zeigt die in der Wissensbasis gespeicherten Nutzungsszenarien. Der Benutzer hat aber auch die Möglichkeit eigene Nutzungsszenarien zu definieren.

Abbildung 10.5 zeigt eine Ausgabe des Expertensystems. Es wurden die beiden Szenarien „Elektronischer Handel extern“ und „Elektronischer Handel öffentlich“ ausgewählt.

```
Zusammenfassung:
----------------
Szenarios, die mit einem '+' versehen sind, wurden ausgewählt.
Szenarios, die mit einem '-' versehen sind, wurden nicht
ausgewählt.

+ 'Elektronischer Handel extern'
- 'Elektronischer Handel intern'
+ 'Elektronischer Handel oeffentlich'
- 'Gruppenunterstützung extern'
- 'Gruppenunterstützung intern'
- 'Informationsbereitstellung extern'
- 'Informationsbereitstellung individuell'
- 'Informationsbereitstellung intern'
- 'Informationsbereitstellung oeffentlich'
- 'Informationsbeschaffung extern'
- 'Informationsbeschaffung individuell'
- 'Informationsbeschaffung intern'
- 'Informationsbeschaffung oeffentlich'
```

Abbildung 10.5: Zusammenfassung des Nutzungskonzeptes

10.2.2 Dienstauswahl

In der Phase der Dienstauswahl werden die Internet–Dienste ausgewählt, die für die Anforderungen benötigt werden. Aufgrund der gewählten Nutzungsszenarien werden dem Benutzer eine Auswahl der in der Wissensbasis als geeignet gespeicherte Dienste präsentiert. Soll beispielsweise gemäss der Nutzungskonzeption das Internet für die Zusammenarbeit eingesetzt werden, bietet sich unter anderem der Einsatz von E-Mail für die Kommunikation, des WWW zur gemeinsamen Nutzung von Daten, von X11 für den Einsatz von verteilten Anwendungen mit grafischer Benutzeroberfläche und von NFS für den gemeinsamen Zugriff auf Daten an. In Abbildung 10.6 sind die Nutzungsszenarien mit den dafür geeigneten Diensten ersichtlich. Eine vertiefende Erläuterung der verschieden Internet–Dienste findet sich in Abschnitt 6 oder in der Literatur, z.B. [Kro95].

Aufgrund der ausgewählten Nutzungsszenarien (im Beispiel aus Abschnitt 10.2.1 sind dies „Elektronischer Handel extern“ und „Elektronischer Handel öffentlich“) wählt das RSD–XPS automatisch geeignete Dienste aus. Dieser Sachverhalt wird in Abbildung 10.7 dargestellt.

Nutzungsszenario \ Internet-Dienst		E-Mail		FTP		Telnet		Berkeley r-Tools		Usenet		WWW							Gopher		Archie		WAIS		IRC		talk		lp, lpr		X-11		SSH		NFS		AFS	
												Client		<->		Server																					
		Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	HTML	Plug-Ins	JavaScript	Java	Active-X	CGI		Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server
Informationsbeschaffung	intern	●		●		○		○		○		●	○		●	○					○		○		○		○				●		○		●		●	
	extern	●		●		○		○		○		●	○		●	○					○		○		○		○				●		○		●		●	
	öffentlich	●		●		○				○		●			●						●		●		○								○				○	
	individuell	●		●		○						●			●				○								○						○				○	
Informationsbereitstellung	intern		●		●		○				○	●	○		●	○	●			○		○		○		○		○			●		○		●		●
	extern		●		●		○				○	●	○		●	○	●			○		○		○		○		○			●		○		●		●
	öffentlich		●		●		○				○	●			●		●			○		○		○									○				○
	individuell		●		●		○					●			●		●			○							○							○				○
elektronischer Handel	intern	●	●									●	○		●	○	●														●	●					
	extern	●	●									●	○		●	○	●														●	●					
	öffentlich	●	●									●			●		●																				
Zusammenarbeit	intern	●	●			○	○			○	○	●	○		●	○	●		○	○					○	○	○	○	○	○	●	●	○	○	●	●	●	●
	extern	●	●			○	○			○	○	●	○		●	○	●		○	○					○	○	○	○	○	○	●	●	○	○	●	●	●	●

● geeignet ○ nur für spezielle Zwecke, bzw. nicht überall verfügbar

Abbildung 10.6: Nutzungsszenarien und Internet Anwendungen

Schritt 2: Dienstauswahl
==========

Anwendungsdienst E-Mail Client wurde ausgewählt. +
Anwendungsdienst E-Mail Server wurde ausgewählt. +
Anwendungsdienst HTML wurde ausgewählt. +
Anwendungsdienst Java-Applets wurde ausgewählt. +
Anwendungsdienst CGI wurde ausgewählt. +
Anwendungsdienst X-Windows Client wurde ausgewählt. +
Anwendungsdienst X-Windows Server wurde ausgewählt. +
administrativer Dienst ICMP wurde ausgewählt. +
administrativer Dienst DNS Client wurde ausgewählt. +

Abbildung 10.7: Start der Dienstauswahl

Durch Löschen oder Hinzufügen von einzelnen Diensten kann der Benutzer nun die Auswahl seinen speziellen Bedürfnissen anpassen. Nachdem er einen oder mehrere Dienste ausgewählt hat führt das System eine Risikoanalyse durch. Der Benutzer hat auch nach Beendigung dieser Phase jederzeit die Möglichkeit Änderungen an seiner bisher getroffenen Auswahl vorzunehmen.

10.2.3 Risikoanalyse

In der Phase der Risikoanalyse werden die Risiken, welche mit den ausgewählten Diensten verbunden sind, betrachtet. Die Fragestellung ist dabei, welche Risiken vermieden oder zumindest durch entsprechende Gegenmassnahmen vermindert werden können. Laut der DIN VDE Norm 31000 setzt sich das Risiko aus der zu erwartenden Häufigkeit eines gefährdenden Ereignisses und aus dem zu erwartenden Schadensausmasses bei Ereigniseintritt zusammen. Demzufolge muss für die Feststellung der Risiken die konkrete Gefahr, die Wahrscheinlichkeit , das diese Gefahr eintritt, und der Wert der durch Güter, die durch diese Gefahr bedroht sind, bekannt sein. Die potentiellen Gefahren für die Informationsinfrastruktur werden in Zusammenhang mit ihren Schwachstellen betrachtet. Auf diese Weise ist es möglich, die konkreten Risiken festzustellen. In Abbildung 10.8 wir dieser Zusammenhang nochmals grafisch dargestellt.

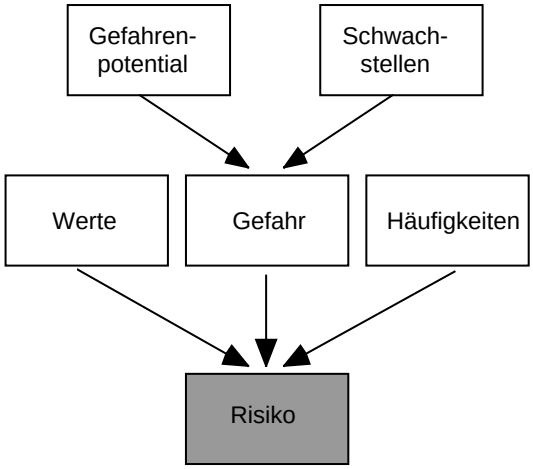

Abbildung 10.8: Aspekte des Begriffes Risiko

Als Rahmenmodell für die Risikonanalyse des RDS–XPS dient das Modell von [SB92]. Zur Bestimmung der Risiken sind vier Schritte durchzuführen:

Wertanalyse: Jede Komponente der Informationsinfrastruktur wird bewertet. Diese Bewertung kann auf einer ordinalen oder monetär auf einer kardinalen Skala erfolgen.

Bedrohungsanalyse: Die potentielle Bedrohung für die Informationsinfrastruktur und die Art und Weise, wie sie sich bemerkbar macht, werden festgestellt.

Schwachstellenanalyse: Die Schwachstellen im bestehenden Sicherungssystem und den benutzten Internet–Diensten werden festgestellt und analysiert. Die Wirkung dieser Schwachstellen auf die Informationsinfrastruktur wird eingeschätzt.

Risikobewertung: Für jede einzelne Gefahr wird ihre Eintrittswahrscheinlichkeit geschätzt. Dazu müssen empirisch ermittelte Eintrittswahrscheilichkeiten zur Verfügung stehen. Zudem werden Aussagen über die Schadenshöhe gemacht. Dieser Schritt ist der Kern des Risikomanagement–Modells, der in den vorangehenden Schritten erfassten Informationen zusammenfasst und systematisiert.

Dieses Modell bietet die Grundstruktur für eine idealisierte Risikoanalyse. Das RSD–XPS stützt sich konkret auf das kardinale Konzept. Bei dem kardinalen Konzept liegt, wie bei den meisten anderen realen Konzepten, der Schwerpunkt auf einem Teilaspekt der idealisierten Struktur. Durch die kardinale Bewertung der Risiken wird zudem eine Kosten–Nutzen–Analyse für die Sicherheitsmassnahmen möglich. Die folgenden Abschnitte erläutern die einzelnen Schritte der Risikoanalyse gemäss dem RSD–XPS zugrunde liegenden kardinalen Bewertungskonzept.

10.2.3.1 Wertanalyse

Ziel der Wertanalyse ist die Bewertung jeder Komponente der Informationsstruktur. Die Wertanalyse erfolgt in drei Schritten [Hei96]. Zuerst wird eine Empfindlichkeitsanalyse für die Anwendungssysteme durchgeführt. Jedes Internet–System wird einem Empfindlichkeitsniveau zugeordnet. Diese Zuordnung orientiert sich an den empfindlichsten Methoden und Daten des jeweiligen Systems. Gemäss dieser Einordnung werden die Systeme dann gruppiert und zur Gesamtheit der Internet–Anwendungssysteme aggregiert. Nach Abschluss der Empfindlichkeitsanalyse wird untersucht, welche Auswirkung die partielle oder totale Nichtverfügbarkeit der Informationsinfrastruktur über verschiedene Perioden hinweg auf die Funktionsfähigkeit der Unternehmung hat. Die Auswirkungen können in Kategorien wie etwa „völlig funktionsunfähig“, „teilweise funktionsfähig“, etc. unterteilt werden, wobei auch die Dauer einer Nichtverfügbarkeit miteinbezogen werden muss. Zuletzt wird der Vermögenswert der Informationsinfrastruktur anhand der Anschaffungs- oder Wiederbeschaffungskosten ermittelt.

Die Sensitivität der Internet–Systeme, der Vermögenswert der Infrastruktur und die Konsequenzen einer partiellen oder totalen Nichtverfügbarkeit geben einen Anhaltspunkt für die Schätzung des totalen Wertes der mit dem Internet verbundenen materiellen und immateriellen Güter.

10.2.3.2 Bedrohungsanalyse

In der Bedrohungsanalyse sollen die Gefahrenpotentiale, welche auf die Schwachstellen einwirken, aufgezeigt werden. Dabei können drei Grundbedrohungen unterschieden werden [PW93]:

- Verlust der Vertraulichkeit
- Verlust der Integrität

- Verlust der Verfügbarkeit

Zu diesen Bedrohungen wird oftmals noch

- Verlust der Verbindlichkeit
- Verlust der Authentizität

hinzugezählt [PW93, BHHS95]. Abbildung 10.9 zeigt die möglichen Gefahren in Verbindung mit den fünf Grundbedrohungen.

Grundbedrohung / Gefahrenpotential	Verlust der Integrität	Verlust der Vertraulichkeit	Verlust der Verfügbarkeit	Verlust der Verbindlichkeit	Verlust der Authentizität
Stören der Verfügbarkeit von Netzkomponenten			●		
Verkehrsflussanalyse		●			
Abhören der IP-Pakete		●			
Wiederholung oder Verzögerung einer Information	●				●
Modifikation der Daten/Programme während der Übermittlung	●				
Leugnung der Sendung				●	
Leugnung des Empfangs				●	
gefälschte Authentizität					●
Spionage im internen System		●			
Modifikation der Daten/Programme im internen System	●		●		
System-Anomalien	●		●		
Urheberrechtsverletzung	●				●

Abbildung 10.9: Gefahrenpotentiale und Grundbedrohungen

10.2.3.3 Schwachstellenanalyse

Im Rahmen der Schwachstellenanalyse wird die Verwundbarkeit der Internet–Dienste analysiert. Bei der Schwachstellenanalyse des RSD–XPS werden nur die technischen, nicht die personellen oder organisatorischen Schwachstellen berücksichtigt. Dabei wird zwischen Schwachstellen in den Internet–Protokollen und Schwachstellen in den Protokollen der Anwendungsschicht unterschieden. Probleme in Protokollen der Anwendungsschicht wirken sich nur auf die spezifischen Anwendungen aus, wogegen sich die Schwachstellen in den Protokollen unterhalb der Anwendungsschicht auf alle darauf aufsetzenden Anwendungen auswirken [CB94, CZ96].

Die genauen Wechselwirkungen zwischen Gefahrenpotentialen und den Schwachstellen der einzelnen Internet–Dienste ergeben die konkreten Gefahren. Gelingt es beispielsweise einem Angreifer, durch die Schwachstelle bei NFS, dem schlechten IP–Nummern–basierten Authentifizierungsmechanismus, in ein System einzudringen, drohen Gefahren wie etwa

Spionage, Modifikation der Daten, Urheberrechtsverletzungen o.ä. und damit verbundene Systemanomalien. Die Internet–Dienste und die zugehörigen Gefahrenpotentiale sind in Abbildung 10.10 dargestellt.

10.2.3.4 Risikobewertung

Nachdem die Art der Gefahren bekannt ist, stellt sich die Frage nach deren Eintrittshäufigkeit und nach dem Schaden, der dabei verursacht wird. Diese Fragen zu beantworten, ist Aufgabe der Risikoanalyse. Im ersten Schritt wird versucht Aussagen über Eintrittshäufigkeiten von Gefahren zu machen. Dann wird aus den Häufigkeiten und dem in der Wertanlyse ermittelten Wert der bedrohten Güter das resultierende Risiko geschätzt.

Eintrittshäufigkeiten

Bisher gibt es noch wenige Studien auf diesem Gebiet. Hinzu kommt, dass eine Vielzahl der Angriffe erst gar nicht registriert wird. Aus diesem Grund ist es bisher nicht möglich genaue Zahlen für die Eintrittshäufigkeiten zu ermitteln. Die Risikoanalyse des RSD–XPS stützt sich auf drei verschiedene Studien, die eine Abschätzung für die Eintrittshäufigkeiten bieten:

Cohen (korrigiert): Die Studie von Cohen [Coh95] stützt sich auf Zahlen, die von repräsentativen Stellen, sogenannten Sites, ermittelt wurden. Cohen schätzt die Eintrittshäufigkeit von Internet Angriffen auf 900 Millionen pro Jahr. Dabei geht er davon aus, dass jede Stelle pro Tag einmal angegriffen wird und multipliziert dies mit der Zahl der Hosts im Internet, welche zum Zeitpunkt der Untersuchung 2,5 Millionen war. Mit der Begründung, dass pro Tag ein Angriff pro Stelle und nicht pro Host erfolgt, korrigiert Howard [How97] diese Schätzung auf 44 Millionen Angriffe pro Jahr. Die Anzahl Stellen entspricht am ehesten der Anzahl der Domänen, welche zu diesem Zeitpunkt rund 120000 betrug. Bei dieser Studie sollte allerdings nicht ausser acht gelassen werden, dass sich die Angaben auf sehr bekannte Stellen, die entsprechend attraktiv für Angreifer sind, beziehen. Die Zahlen dürften aus diesem Grund nicht auf den Durchschnitt der Internet–Sites übertragbar sein.

DISA: Die DISA–Studie ist eine Verwundbarkeitsstudie. Hier wurde versucht, durch künstliche Attacken auf Internet–Sites Rückschlüsse auf die tatsächlichen Eintrittshäufigkeiten zu ziehen. Das DoD (Department of Defence) der Vereinigten Staaten von Amerika hat solche Studien durchgeführt. In dem Zeitraum von 1992 bis 1995 versuchten Mitarbeiter der DISA in verschiedene Hosts des DoDs einzubrechen. 65% der 38000 durchgeführten Angriffe waren erfolgreich. Von diesen 65% wurden nur 4% bemerkt, von denen wiederum nur 27% gemeldet wurden. Demzufolge wurden also nur 0.7% der Angriffe gemeldet. Damit ergibt sich, bei einer Zahl von 500 gemeldeten Angriffen im Jahr 1995, eine geschätzte Angriffszahl von 70000 für das DoD. Davon ausgehend, dass das DoD 10% der Hosts im Internet repräsentiert, ergibt sich damit eine Gesamtzahl von 700000 Angriffen im Internet [How97].

Internet-Dienst \ Gefahrenpotential		Stören der Verfügbarkeit von Netzkomponenten	Verkehrsflussanalyse	Abhören der IP-Pakete	Wiederholung oder Verzögerung einer Information	Modifikation der Daten / Programme während der Übermittlung	Leugnung des Sendens	Leugnung des Empfanges	gefälschte Authentizität	Spionage im internen System	Modifikation der Daten / Programme im internen System	System-Anomalien	Urheberrechtsverletzung
E-Mail	Client	●	●	●	●	●	●	●	●		●	●	●
	Server	●	●	●	●	●			●	●	●	●	
FTP	Client	●	●	●		●			●			●	
	Server	●	●	●		●			●	●	●	●	
Telnet	Client	●	●	●		●			●			●	●
	Server	●	●	●		●			●	●	●	●	
Berkeley	Client	●	●	●		●			●			●	
r-Tools	Server	●	●	●		●			●	●	●	●	
Usenet	Client	●				●	●	●	●				
	Server	●				●			●				
WWW	HTML	●	●	●		●							
	Plug-Ins	●	●	●	●	●	●	●	●	●	●	●	
	JavaScript	●	●	●		●				●	●	●	
	Java-Applets	●	●	●	●	●	●	●	●	●	●		
	Active-X	●	●	●	●	●	●	●	●	●	●	●	●
	CGI	●	●	●	●	●	●	●	●	●	●	●	
Gopher	Client	●	●	●		●						●	
	Server	●	●	●		●				●	●	●	
Archie	Client	●				●							
	Server	●				●							
WAIS	Client	●				●							
	Server	●				●							
IRC	Client	●				●	●	●	●	●	●		
	Server	●				●							
talk	Client	●				●	●	●	●				
	Server	●				●	●	●	●				
lp, lpr	Client	●	●	●	●	●			●				
	Server	●	●	●	●	●			●				
X-11	Client	●	●	●	●	●			●				●
	Server	●	●	●	●	●			●	●	●		
SSH	Client	●	●									●	●
	Server	●	●									●	
NFS	Client	●	●	●	●	●			●			●	●
	Server	●	●	●	●	●			●	●	●	●	
AFS	Client	●										●	
	Server	●	●									●	
ICMP, IGMP		●	●	●	●	●			●				
ARP, RARP		●	●	●	●	●			●				
RIP	Client	●	●	●	●	●			●				
	Server	●	●	●	●	●			●				
SNMP	Client	●	●	●	●	●			●				
	Server	●	●	●	●	●			●		●		
DNS	Client	●	●	●		●			●				
	Server	●	●	●		●			●	●	●		
NTP	Client	●	●	●	●	●			●				
	Server	●	●	●	●	●			●				
finger	Client	●											
	Server	●								●			
NIS	Client	●	●	●		●							
	Server	●	●	●		●				●			
TFTP	Client	●	●	●		●						●	
	Server	●	●	●		●				●	●	●	
BOOTP	Client	●	●	●		●			●			●	
	Server	●	●	●									
ping		●				●			●	●			
traceroute		●				●			●	●			
RPC,	Client	●	●	●		●							
PMAP	Server	●	●	●		●							

Abbildung 10.10: Internet–Dienste und Gefahrenpotentiale

AFIWC: Die AFIWC–Studie (Air Force Informaiton Warface Center) von 1995 ist, wie die DISA–Studie, eine Verwundbarkeitsstudie. Hier wurden 13% aller Angriffe gemeldet. Daraus ergibt sich eine geschätzte Anzahl von 40000 Angriffen pro Jahr im Internet [How97].

Die Untersuchungen zeigen gravierende Unterschiede. So lassen sich die vergleichsweise hohen Zahlen in der Cohen–Studie etwa damit erklären, dass sie auf den Angaben von sehr bekannten Stellen basieren, welche für Angreifer äusserst attraktiv sind, und damit nicht den Durchschnitt der Internet–Sites repräsentiert.

Für die Bestimmung der Eintrittshäufigkeiten der einzelnen Gefahren werden empirische Daten benötigt. Die umfangreichste Datenbank über Ereignisse, bei denen Sicherheitsvorkehrungen verletzt wurden, wird von dem Computer Emergency Response Team Coordination Center (CERT/CC) der Carnegie Mellon University verwaltet. In seiner Studie analysiert Howard die Vorfälle aus dem Zeitraum von 1989 bis 1995 [How97]. Die Untersuchung spricht von 4299 Vorfällen im diesem Untersuchungszeitraum. Leider sind nicht zu allen Vorfällen die von den Angreifern verwendeten Schwachstellen bekannt. Dort wo eine Aufschlüsselung nach den Schwachstellen stattgefunden hat, entspricht nicht jede Eintragung zwingend einem Vorfall, da eventuell ein einzelner Vorfall durch Ausnutzung mehrerer Schwachstellen ermöglicht wurde. Trotzdem gibt diese Aufstellung nützliche Hinweise über besonders häufig verwendete Schwachstellen. Da die Untersuchung nur bis ins Jahr 1995 reicht, wurden neuere WWW–Dienste (CGI, Java–Applets, Javascript, ActiveX) nicht berücksichtigt. Dabei sind es gerade diese Technologien, die heute die grösste Verwendung finden und daher unbedingt in der Statistik miteinbezogen werden müssen.

Um diese Zahlen zu schätzen, wurden die CERT–Advisories, welche das CERT in regelmässigen Abständen veröffentlicht, zwischen Januar 1996 und November 1997 analysiert. Aus dieser Analyse geht hervor, dass die WWW–Technologien vermehrt Angriffen ausgesetzt sind. 8 der 51 veröffentlichten Advisories handeln von diesen Technologien (2 von Java, 5 von CGI–Skripten, 1 von Javascript), d.h. 16% aller wichtigen Schwachstellen wurden in neueren Technologien gefunden. Auch in Zukunft ist damit zu rechnen, dass diese Technologien vermehrt als Angriffsziel dienen. Die Resultate der Analyse der CERT–Advisories wurden mit der Arbeit von Howard kombiniert, um ein aktuelles Bild der verwendeten Schwachstellen zu erhalten. Dabei wurde die relative Zahl (16%) der WWW–Schwachstellen den Advisories entnommen und die restlichen 84% gemäss der Ergebnisse von Howard auf die Schwachstellen verteilt. Die relativen Häufigkeiten sowie die Schätzung der einzelnen Eintrittshäufigkeiten pro Host und Jahr sind in Abbildung 10.12 ersichtlich. Zur Berechnung der Eintrittshäufigkeiten wurde die totale Anzahl der Angriffe pro Jahr prozentual auf die Schwachstellen verteilt und durch die Anzahl der Hosts im Juli 1995, da die Schätzung aus dem Jahre 1995 stammen, geteilt.

Die grössten Schwachstellen im Internet sind die benutzten Passwörter. Etwa ein Fünftel aller Angriffe konnte wegen schwacher Passwörter realisiert werden. So wurden in 5% der Fälle ein Angriff durch Standard- oder gar fehlende Passwörter ermöglicht. Dagegen wurde nur ein Angriff bekannt, bei dem das Passwort abgehört wurde. Leider geht aus der Studie

nicht hervor, wie der Angreifer in den restlichen Fällen (95%) an die Passwörter gelangen konnte.

Die hohe Zahl der E–Mail–Angriffe ist auf die weite Verbreitung von Sendmail zurückzuführen. Über 50% der Angriffe könne auf eine fehlerhafte Sendmail–Installation zurückgeführt werden.

Die berechneten Häufigkeitstabellen haben den Nachteil, dass nicht für alle möglichen Gefahrenpotentiale empirische Häufigkeiten vorliegen. Die CERT–Statistik enthält beispielsweise keine Zahlen über Lauschangriffe oder über Virenbefall. Aus diesem Grund wurde eine erweiterte Schätzung erstellt. Dabei blieb der Rahmen der Schätzung bestehen, d.h. die totalen Häufigkeiten wurden aus der vorangehenden Analyse übernommen. Der restliche Anteil wurde auf die Gefahren gemässAbbildung 10.10 verteilt. Dabei wurde folgendermassen vorgegangen:

Die einzelnen Gefahren pro Dienst wurden auf einer ordinalen Skala bewertet. Ein ähnliches Verfahren schlägt Howard vor [How97]. Die Skala ist dadurch definiert, dass die Eintrittshäufigkeit von Abstufung zu Abstufung um jeweils den Faktor 10 auseinander liegen.

Die einzelnen Häufigkeiten pro Abstufung wurden ermittelt, indem die Resteintrittshäufigkeiten auf die Skala verteilt wurde. Die Summe aller Ereignisse inklusive der Angriffe durch Passwörter und der restlichen Angriffe muss die totale Angriffshäufigkeit der verwendeten Studie (Cohen, DISA, AFIWC) ergeben. Abbildung 10.12 zeigt die Bewertung der einzelnen Gefahren, die für die drei Schätzungen ermittelten Werte der Skala sind inAbbildung 10.11 angegeben.

Studie / Skalenelement	Cohen korrigiert		DISA korrigiert		AFIWC	
	Anzahl Ereignisse pro Jahr	1 Angriff pro Anzahl Jahre	Anzahl Ereignisse pro Jahr	1 Angriff pro Anzahl Jahre	Anzahl Ereignisse pro Jahr	1 Angriff pro Anzahl Jahre
sehr häufig	1.50E-01	6.7	2.39E-03	418	1.37E-04	7299
häufig	1.50E-02	67	2.39E-04	4180	1.37E-05	72990
mittel	1.50E-03	670	2.39E-05	41800	1.37E-06	729900
selten	1.50E-04	6700	2.39E-06	418000	1.37E-07	7299000
unwahrscheinlich	1.50E-05	67000	2.39E-07	4180000	1.37E-08	72990000

Abbildung 10.11: Resultate der erweiterten Schätzungen

Es darf nun allerdings nicht der Eindruck entstehen, dass die Risiken des Internet genau bekannt sind. Die Häufigkeitstabellen basieren auf einer Anzahl Hypothesen und Unsicherheiten, die kritisch betrachtet werden müssen. So wurde zwar der Versuch gemacht, die Risiken der Internet–Dienste zu quantifizieren, aber bisher existiert keine öffentlich zugängliche Untersuchung über die Häufigkeiten der einzelnen Gefahren, so dass sie aus verschiedenen Untersuchungen zusammengetragen werden mussten. Dabei war es notwendig verschiedene Erhebungssysteme miteinander zu kombinieren.

Die genaue Zahl der Missbräuche im Internet ist unklar. Hier werden drei verschiedene Schätzungen verwendet, deren Resultate jeweils um 1–2 Grössenordnungen variieren. Diese Streubreite zeigt, wie schwierig es ist, realistische Werte zu finden. Sie bedeutet jedoch nicht, dass es sich nicht um sinnvolle Schätzungen handelt. Sie basieren auf unterschiedlichen Erhebungsmethoden und Untersuchungsobjekten, die für sich genommen durchaus

repräsentativ sein können. So kann bei einem Host, der ein attraktives Angriffsziel darstellt, durchaus mit den Zahlen von Cohen gearbeitet werde. Für kleinere, weniger attraktive Hosts bieten sich dagegen die Resultate der AFIWC–Studie an. Die DISA–Studie schliesslich repräsentiert die Mittelwerte und trifft wahrscheinlich auf die Mehrzahl der Internet–Hosts zu.

Die Verteilung der totalen Anzahl der Angriffe auf die verschiedenen Dienste und deren Gefahren wurde auf der Basis der Angaben in [How97] gemacht. Diese Angaben sind jedoch ungenau und lassen Raum für Interpretationen. Für eine vertiefte Analyse müsste direkt die Datenbank des CERT verwendet werden.

Von den Angaben in [How97] aus wurden die Häufigkeiten für das Internet hochgerechnet. Die zugrundeliegende Hypothese dabei war, dass die Häufigkeitsverteilung der registrierten Angriffe mit der Häufigkeitsverteilung aller Angriffe übereinstimmt. Um neuere Web–Technologien zu berücksichtigen wurden die CERT–Advisories hinzugezogen. Allerdings lassen sie keinen genauen Schluss auf die Anzahl der Angriffe zu, sondern geben nur eine Indikation dafür, welche Dienste wie stark missbraucht werden können. Dieses Verfahren gelangte aus Mangel an detaillierteren Informationen zur Anwendung.

Die Angaben des CERT enthüllen nicht alle Gefahren. Vielfach werden Angriffe erst gar nicht registriert, wie etwa Lauschangriffe, oder sie werden vom CERT nicht erfasst, wie etwa die Virenverbreitung, so dass für diese Gefahren keine Häufigkeiten vorliegen. Aus diesem Grund wurde die Schätzung erweitert. Allerdings beruht diese erweiterten Schätzung nicht auf empirischen Daten, sondern auf einer Einteilung in eine ordinale Skala. Die Resultate wurden mit der ersten Schätzung verglichen und liegen meistens in derselben Grössenordnung. Dabei gehen zwar die Feinheiten der ersten Schätzung verloren, aber die Häufigkeit ist genauer ersichtlich.

Mit der Eintrittshäufigkeit und den Resultaten aus der Wertanalyse kann nun das konkrete Risiko berechnet werden. Das Risiko ergibt sich aus:

Risiko = Vermögenswert in CHF * Häufigkeit des Schadenseintritts.

Wie schon gesagt, gibt die Risikozahl nur eine Indikation auf das zu erwartende Risiko. Für eine bessere Entscheidungsgrundlage kann das Dienstportfolio mit den errechneten Risikokennzahlen in Risikoklassen eingeteilt werden [Kra89]. Dazu werden die Dienste in die Risikoklassenmatrix (Abbildung 10.13) eingetragen.

Dabei kristallisieren sich die vier Klassen seltene Fälle (A), Problemfälle (B), unkritische Fälle (C) und Routinefälle (D) heraus. Risiken der Kategorie A sind unwahrscheinlich und könne kaum durch notwendige Massnahmen gesenkt werden. Eine Kosten-/Nutzenanalyse der Gegenmassnahmen und eine genaue Analyse deren Effektivität und Effizienz geben Aufschluss über einzusetzende Schutzmassnahmen. In der Kategorie B sind die Problemfälle, deren Risiko dringlichst durch Schutzmassnahmen zu senken ist, zusammengefasst. Die Risiken der Kategorie C sind dagegen die unkritischen Fälle, bei denen die entsprechenden Schutzmassnahmen einer kritischen Kosten-/Nutzenanalyse zu unterziehen sind. Als Routinefälle gelten die Risiken der Kategorie D. Sie sollten mit entsprechenden Massnahmen gesenkt werden.

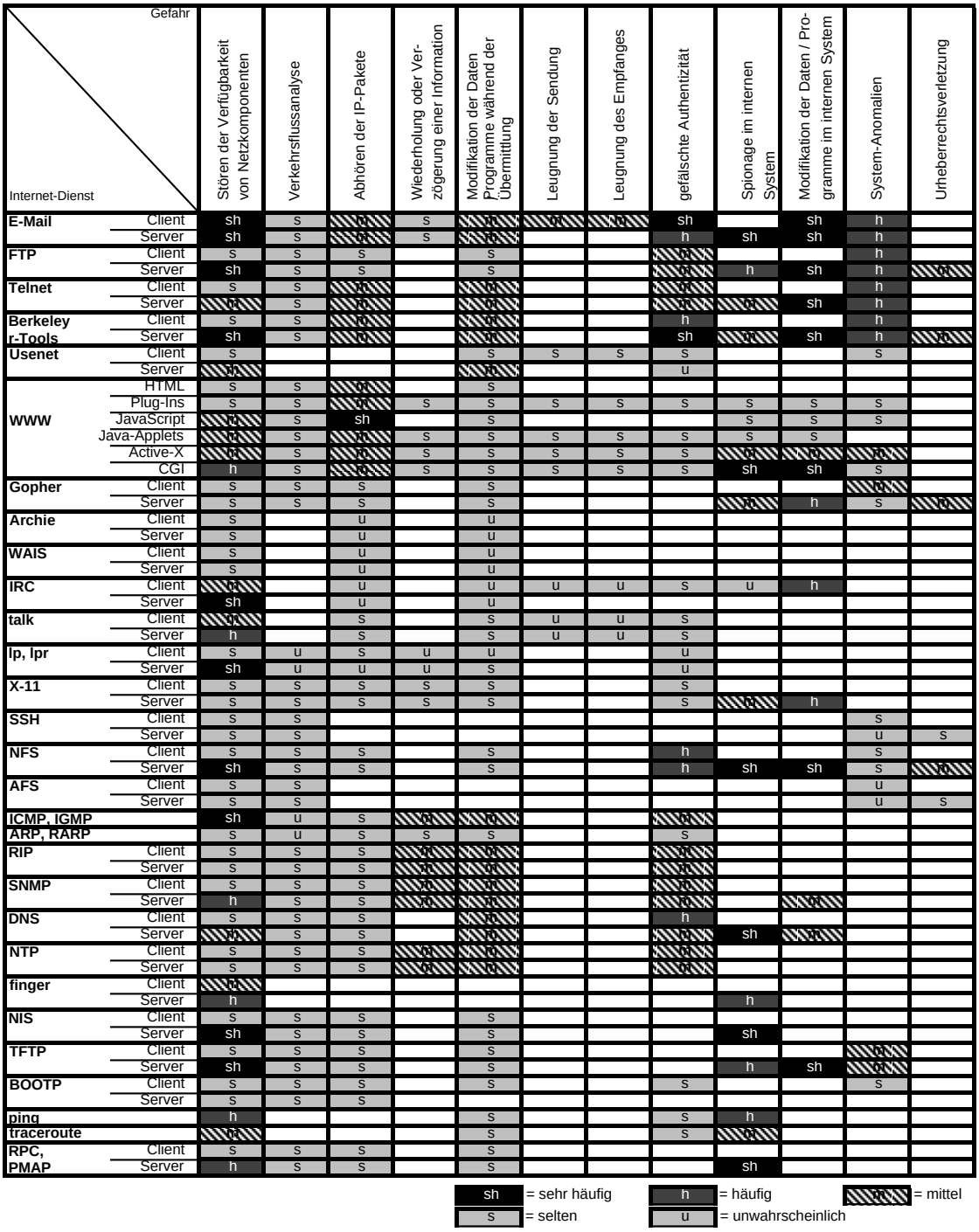

Abbildung 10.12: Erweiterte Schätzung der Eintrittshäufigkeiten anhand der ordinalen Skala

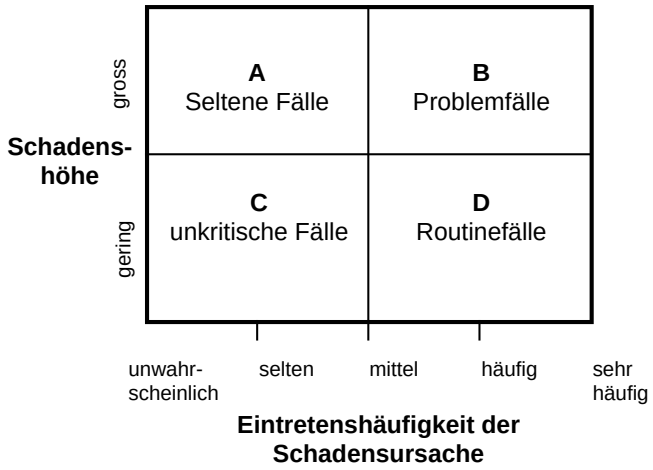

Abbildung 10.13: Beziehung zwischen Eintrittshäufigkeit und Schadenshöhe nach Krallmann

Abbildung 10.14 zeigt einen Ausschnitt aus der Bildschirmausgabe der Risikoanalyse.

10.2.3.5 Gegenmassnahmen

Der nächste Schritt in der RSD–Sicherheitskonzeption ist, gegen die eruierten Risiken Gegenmassnahmen zu ergreifen. Es gibt grundsätzlich vier Möglichkeiten, Risiken einzuschränken [SB92]:

Risikovermeidung: Der Internet–Dienst wird nicht eingesetzt oder durch einen gleichwertigen, jedoch sicheren Dienst ersetzt.

Schutzmassnahmen: Es werden Vorkehrungen zur Sicherung des eingesetzten Dienstes getroffen. Das Risiko wird dadurch reduziert.

Schadensbegrenzung: Ist trotz Schutzmassnahmen ein Schaden eingetreten, so sollte er in Grenzen gehalten werden. Dazu gehört eine effektive Datensicherung, Ermitteln des Angreifers, Feststellen der benutzten Schwachstelle, Verbessern der Schutzmassnahmen und Information potentieller neuer Opfer usw.

Überwälzen: Die Risiken können auch versichert werden. Dies ist vor allem bei grossen Risiken der Fall, welche selten auftreten und gegen welche keine ausreichenden Gegenmassnahmen ergriffen werden können. Die Risiken, welche im Zusammenhang mit dem Internet stehen, dürften wohl schwerlich zu zahlbaren Konditionen versicherbar sein.

Nach Anwendung dieser vier Massnahmen bleibt ein Restrisiko übrig, welches nicht weiter eingeschränkt, sondern akzeptiert werden muss.

Ein gewisser Teil des Risikos wurde vermieden, indem durch das Nutzungsszenario unnötige Dienste erkannt und nicht eingesetzt wurden. Die ausgewählten Internet–Dienste müssen jedoch genau konfiguriert werden, damit auch wirklich alle überflüssigen Dienste

Schritt 3: Risikoanalyse
==========

Aktiviere Gefahren, Gegenmassnahmen und Realisierungsmassnahmen.
Bitte warten.....
Risikoklasse 'mittel erweitert (DISA)' gewählt.

Details:

DIENST: Wert des/der Szenarios Gefahr	Häufigkeit 1 Ereignis pro Jahr und Host	jährliches in Anzahl Jahren	Risiko in CHF
'CGI': 2`000`000			
* 'Abhören der IP-Pakete'	2.4E-5	41`828	48
* 'gefälschte Authentizität'	2.4E-6	418`277	5
* 'Leugnung des Empfangs'	2.4E-6	418`277	5
* 'Leugnung der Sendung'	2.4E-6	418`277	5
* 'Modifikation der Daten / Programme im internen System'	2.4E-3	418	4`782
* 'Modifikation der Daten / Programme während der Übermittlung'	2.4E-6	418`277	5
* 'Spionage im internen System'	2.4E-3	418	4`782
* 'Stören der Verfügbarkeit von Netzkomponenten'	2.4E-4	4`183	478
* 'System-Anomalien'	2.4E-6	418`277	5
* 'Verkehrsflussanalyse'	2.4E-6	418`277	5
* 'Wiederholung oder Verzögerung einer Information'	2.4E-6	418`277	5

Totales Risiko CGI:			10`122

(...)

TOTALES RISIKO pro Jahr und Host:			46`510 CHF
			==========

Abbildung 10.14: Ausschnitt aus der Risikoanalyse

deaktiviert sind. Im folgenden werden Schutzmassnahmen aufgezeigt und Gefahren zugeordnet. Durch diese Zuordnung wird es ermöglicht, zu jedem Dienst die nötige konzeptionelle Gegenmassnahme zu definieren.

Computersysteme müssen mit Grundfunktionen für die Sicherheit ausgestattet werden, um den in der Risikoanalyse aufgezeigten Gefahren entgegenzuwirken. Diese Grundfunktionen werden als Sicherheitsdienste bezeichnet [Woj91]. In der ISO / OSI–Sicherheitsarchitektur [Stu94] werden fünf Technologie–unabhängige Sicherheitsdienste unterschieden (siehe auch Kapitel 2.4:

- Integrität soll vor unbefugter Modifizierung oder Löschung von Daten und Programmen schützen.
- Vertraulichkeit gewährleistet den Schutz von Daten vor unbefugter Aufdeckung.
- Zugriffskontrolle soll gegen die unbefugte Nutzung von Ressourcen schützen.
- Unwiderrufbarkeit hat zu gewährleisten, dass keiner der Kommunikationspartner seine Teilnahme an der Kommunikation bestreiten kann.
- Authentizität ist die Echtheitsprüfung der Identität eines Kommunikationspartners.

[Woj91] zählt zu den Sicherheitsdiensten noch

- Auditing (Prüfung) stellt einen ex–post Sicherheitsdienst dar und wurde als separater Dienst in ISO definiert. Die wichtigste Aufgabe des Auditing ist die Protokollierung aller Ereignisse.

Die Konkretisierung der Sicherheitsdienste in einer technischen Implementierung wird Sicherheitsmechanismus genannt.

Bei Sicherheitsüberlegungen wird generell zwischen sicheren und unsicheren Systemteilen unterschieden. Sicher sind jene Teile, welchen vertraut wird bzw. welche unter persönlicher Kontrolle stehen. Aus diesem Grund gilt oft das interne Netz eines Unternehmens als sicher. Durch den Anschluss an das interne Netz wird das System Teil eines unsicheren Systems. Damit ergeben sich zwei grundsätzliche Problemkreise [CM95]:

1. Wie ist die Kommunikation zwischen unsicherem und sicherem Teilnetz möglich, ohne die sicheren Systeme zu gefährden?
2. Wie können sichere Teilnetze miteinander über das unsichere Internet kommunizieren?

Kapitel 8 gibt einen Überblick über die Sicherheitsmechanismen, die realisiert werden können. Eine Gegenüberstellung der Gefahren und den ihnen zugehörigen Gefahren stellt Abbildung 10.15 dar. Mit der konkreten Umsetzung dieser Gegenmassnahmen innerhalb einer Organisation beschäftigt sich die Phase der Realisierung.

Abbildung 10.16 zeigt einen Ausschnitt der Gegenmassnahmen für den E–Mail–Dienst.

Gegenmassnahme / Bedrohung	Abschottung von Netz-segmenten durch Firewalls	Einsatz digitaler Signaturen	Einsatz von Verschlüsselung	Steganographie	Redundante Einrichtungen	Protokollierung	Antiviren-Programme	Verbot der Ausführung von Diensten und Dienstteilen
Stören der Verfügbarkeit von Netzkomponenten	●				●			
Verkehrsflussanalyse	●				●			
Abhören der IP-Pakete	●		●					
Wiederholung oder Verzögerung einer Information	●	●						
Modifikation der Daten/Programme während der Übermittlung	●	●						
Leugnung der Sendung		●						
Leugnung des Empfangs						●		
gefälschte Authentizität	●	●						
Spionage im internen System	●		●					●
Modifikation der Daten/Programme im internen System	●	●						●
System-Anomalien	●						●	●
Urheberrechtsverletzung				●				

Abbildung 10.15: Gefahren und Gegenmassnahmen

'E-MAIL CLIENT' wird geschützt durch folgende Massnahmen:
* 'Abschottung von Netzsegmenten durch Firewalls'
* 'Einsatz digitaler Signaturen'
* 'Protokollierung'
* 'Redundante Einrichtungen schaffen'
* 'Verbot der Ausführung von Diensten und Dienstteilen'
* 'Einsatz von Verschlüsselung'
* 'Einsatz von Anti-Viren-Software'
...

Abbildung 10.16: Gegenmassnahmen nach Internet–Dienst geordnet (Ausschnitt: E–Mail–Client)

10.2.3.6 Realisierung

Die Realisierung beschäftigt sich mit der Umsetzung von Gegenmassnahmen. Die konzeptionellen Schutzmechanismen, die in dem vorherigen Kapitel kurz vorgestellt wurden, werden durch eine oder mehrere Realisierungsmechanismen verwirklicht. Diese Realisierungsmechanismen lassen sich in die Klassen der technischen und der organisatorisch–personellen Massnahmen einteilen. Technische Massnahmen basieren auf dem Einsatz von Hard- und Software und deren Konfiguration, wohingegen sich die organisatorisch–personellen Massnahmen mit der Umsetzung der Schutzmechanismen innerhalb einer Organisation sowohl aus organisatorischer als auch aus personeller Sicht beschäftigen. Abbildung 10.17 zeigt die Realisierungsmassnahmen und bringt sie in Verbindung mit den Gegenmassnahmen.

Realisierungsmassnahme / Gegenmassnahme	technische Massnahmen											organisatorisch-personelle Massnahmen			
	Firewall-Produkt	Signatur-Software	Verschlüsselungs-Software	Steganographie-Software	Antivirenprogramm	Server	Netzwerkkomponenten	Protokoll-Programm	Deinstallation	Konfiguration	bauliche Massnahmen	Wartung	Kontrolle	Sicherheitsvorschriften	Schulung
Abschottung von Netzsegmenten durch Firewalls	●									●		●	●		
Einsatz digitaler Signaturen		●								●		●			●
Einsatz von Verschlüsselung			●							●		●			
Steganographie				●						●		●			
Antiviren-Programme					●					●		●			●
Redundante Einrichtungen						●	●			●	●	●			
Protokollierung								●		●		●	●		
Verbot der Ausführung von Diensten und Dienstteilen									●					●	●

Abbildung 10.17: Gegenmassnahmen und Realisierungsmassnahmen

Neben der Beschaffung und Installation der einzelnen Produkte gehört auch die Wartung und Überwachung zur Realisierung.

Das System macht zur jeder Gegenmassnahme Angaben über die Realisierungsmöglichkeiten. Für die Kosten / Nutzen–Analyse können die Kosten für eine Realisierungsmassnahme vom Benutzer eingegeben werden. Bei der Kosten / Nutzen–Analyse wird zu jeder Realisierungsmassnahme deren Kosten und die Summe der Risiken angegeben, die die Massnahme vermindert.

Realisierungsmöglichkeiten zeigen konkrete Schritte für die Umsetzung der Gegenmassnahmen auf. Die Kosten- / Nutzenanalyse soll einen Anhaltspunkt geben, welche Massnahmen realisiert werden sollen.

10.3 Implementierung

Im vorliegenden Kapitel werden verschiedene implementierungsspezifische Details aufgezeigt. Vorgestellt werden die Objekte, die Verbindung zwischen den Objekten und zuletzt die Darstellung des fallspezifischen Wissens.

10.3.1 Objekte

Die statische Struktur des Expertensystems stellen Objekte dar. Abbildung 10.18 stellt alle Frames (Klassen) und deren Fächer (Attribute) in einer Übersichtsgrafik dar.

Der Frame Sicherheitskonzeption vererbt seine Eigenschaften an die ihm untergeordneten Frames, welche die einzelnen Phasen in dem RSD–Verfahren repräsentieren. Jeder Schritt im Verfahren hat genaue Kenntnisse über die weiteren Objekte, die in dieser Phase gebraucht werden. Dies wird durch eine Assoziation zum entsprechenden Frame dargestellt. So kennt z.B. die Nutzungskonzeption alle möglichen Szenarien, die Dienstauswahl kennt alle Internet–Dienste usw.

Das Problemlösungsverfahren wird durch die fünf Schritte repräsentiert. Jeder Schritt weiss, welche Bedingungen erfüllt sein müssen, damit er aufgerufen werden darf, und welchen Endzustand er haben muss, damit der Schritt als beendet gelten darf. Dieses Wissen wird in Form von LISP–Funktionen gespeichert, die jeweils am Anfang und am Ende eines Schrittes aufgerufen werden. Darüber hinaus kennt das System die Regeln, die ausgeführt werden müssen, um den Endzustand jedes einzelnen Schrittes zu erreichen.

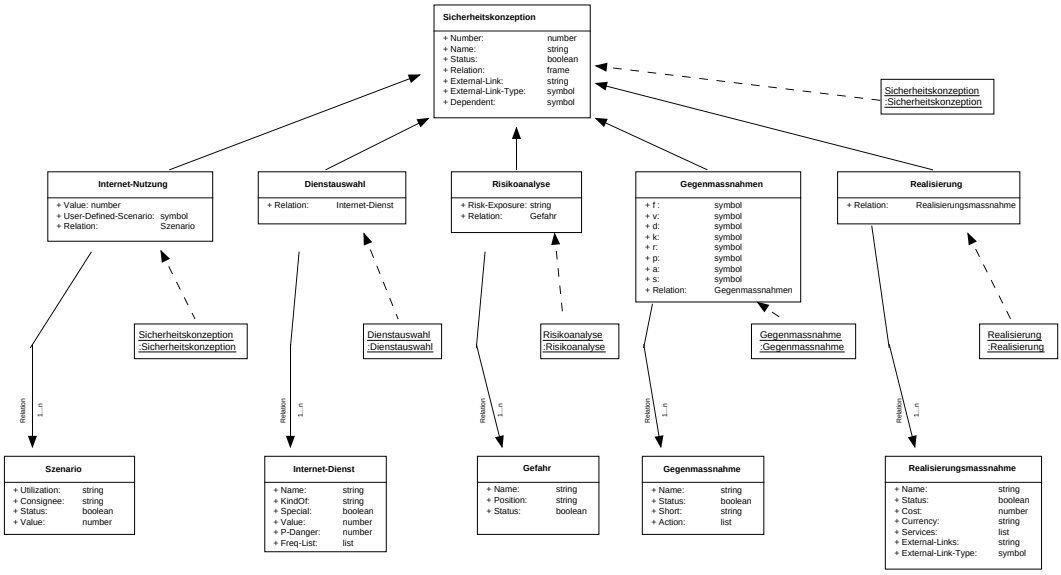

Abbildung 10.18: Frames des Expertensystems (Darstellungsform: Unified Modelling Language UML)

Alle Objekte sind zu Beginn der Konsultation instanziiert. Wird ein Objekt vom Benutzer oder durch Regeln aktiviert, so erhält das Fach Status den Wert „wahr“. Die Darstellung

im Expertensystem entspricht also jener innerhalb einer objektorientierten Datenbank: Alle bekannten Objekte sind gespeichert, je nach Bedarf oder Situation werden sie referenziert.

Die einzelnen Frames haben je nach Bedarf eine grössere oder kleinere Anzahl Methoden. Diese Methoden können fünf Aufgaben erfüllen:

Benutzerinteraktion: Viele Methoden dienen nicht zur Repräsentation des Wissens, sondern zur Benutzerein- und -ausgabe. Sie wurden trotzdem im entsprechenden Frame gekapselt, damit die nötige Struktur und Übersicht innerhalb des Programmes erhalten wird.

Aktive Werte: Eine gewisse Anzahl Methoden sind Fächern als aktive Werte zugewiesen.

Lese- und Schreibzugriffe: Eine gewisse Anzahl Methoden dienen zum Lese- und Schreibzugriff auf die Fächer.

Abfrage der Prolog–Faktenbasis: Die Objekte erhalten einen Grossteil ihres Wissens aus der Prolog–Faktenbasis. In dieser steht z.B. warum der Internet–Dienst aktiviert wurde, welches die Gefahren des Dienstes sind usw. Um an dieses Wissen zu gelangen, verfügt das Objekt über die entsprechenden Abfragemethoden.

Wissensrepräsentation: Einige Methoden dienen zur Repräsentation von Wissen. So werden z.B. die Vorbedingungen eines Schrittes, die erfüllt sein müssen, bevor er ausgeführt werden kann, in LISP–Funktionen gespeichert.

Die Objekte und Frames stellen die statische Struktur des Expertenwissens dar. Im nächsten Kapitel wird nun die dynamische Struktur – die Verbindungen zwischen den Objekten – diskutiert.

10.3.2 Verbindungen

Die Verbindungen zwischen den einzelnen Objekten werden mit Produktionsregeln und einem Suchprozess erstellt. Zuerst werden die Verbindungen vom Nutzungskonzept über die Dienstauswahl bis zur Risikoanalyse und den Gegenmassnahmen dargestellt, anschliessend der Suchprozess, bei dem die Gegenmassnahme ihre Realisierungsmöglichkeiten sucht.

10.3.2.1 Produktionsregeln

Die Produktionsregeln repräsentieren das dynamische Wissen. Mit Hilfe dieser Regeln werden anhand der Anforderungen die benötigten Dienste ausgesucht, deren Gefahren und entsprechende Gegenmassnahmen aktiviert. Die Regeln wurden zur besseren Übersicht und Verwaltung in drei Regelmengen aufgeteilt. Alle Regeln werden vorwärtsverkettet interpretiert.

Anwendungsdienste aktivieren

Ausgehend von den in der Nutzungskonzeption definierten Anforderungen an die Online–Dienste werden die benötigten Anwendungsdienste ausgesucht. Für jedes Nutzungsszenario gibt es eine Regel, die beim Aufruf der Dienstauswahl ausgeführt wird.

abhängige Dienste aktivieren

Nach der Auswahl der Anwendungsdienste wird ein weiteres Regelpaket ausgeführt. Die darin enthaltenen Regeln aktivieren die abhängigen Dienste. Es gibt zwei Arten abhängiger Dienste. Die erste Art muss immer ausgewählt sein. ICMP ist z.B. solch ein abhängiger Dienst, da ICMP fest mit dem IP Protokoll verankert ist. Die zweite Art sind jene Dienste, die in Abhängigkeit von einem Anwendungsdienst aktiviert werden müssen. Der RPC–Dienst ist ein Beispiel eines solchen Dienstes. Sind AFS, NFS oder NIS aktiviert, wird auch der RPC–Dienst benötigt.

Gefahren und Gegenmassnahmen aktivieren

Die letzte Regelmenge aktiviert die Gefahren und Gegenmassnahmen. Die Gegenmassnahme kann nicht unabhängig vom betreffenden Internet–Dienst ausgesucht werden, da eine Massnahme nicht für jeden Dienst umsetzbar ist.

10.3.2.2 Suchprozess

Die letzte Phase des Lösungsprozesses stellt die Verbindung zwischen den Gegenmassnahmen und deren Realisierungsmöglichkeiten dar. Dieser Übergang wird mit Hilfe eines Suchprozesses vorgenommen. Dabei wird folgendermassen vorgegangen:

Jedes Objekt der Gegenmassnahmen kennt seine Anforderungen an das Realisierungsobjekt, und jede Realisierungsmassnahme weiss, welche Dienste sie schützen kann. Die Gegenmassnahme kontaktiert nacheinander alle ihren Anforderungen entsprechenden Realisierungen. Trifft die Gegenmassnahme auf eine Realisierung, die einen ausgewählten Dienst schützen kann, so wird diese Massnahme aktiviert.

10.3.3 Fallspezifisches Wissen

Die Zwischen- und Endresultate sowie das fallspezifische Wissen werden in Form von Prolog–Fakten gespeichert. Jedes Objekt, das durch die Anwendung von Regeln oder durch den Suchprozess aktiviert wird, speichert seine Aktivierung und das Wissen, durch welche Instanz es aktiviert wurde, in der Prolog–Faktenbasis.

Babylon verwendet für die Formulierung von Prolog–Klauseln die funktionale Schreibweise, bei der der Ausdruck voll geklammert wird und das erste Argument das Prädikat oder die Funktion bezeichnet. Nachfolgendes Beispiel sagt aus, dass das Szenario “Informationsbeschaffung öffentlich“ durch den Benutzer aktiviert wurde.

```
((szenario benutzer informationsbeschaffung_oeffentlich))
```

Die Prolog–Faktenbasis füllt sich so während der Konsultation mit wahren Klauseln auf. Einzelne Objekte können nun wiederum diese Fakten herbeiziehen, um beispielsweise eine Begründung für ihre Auswahl zu erhalten. Mit den Fakten wird also auch die Erklärungskomponente realisiert.

Die Prolog–Faktenbasis wird jedoch nicht nur für die Erklärung einer Auswahl herbeigezogen. Die Anfrage kann auch in die andere Richtung gehen, indem z.B. ein Szenario nach seinen Internet–Diensten gefragt wird. Die Darstellung der Zwischen- und Endresultate wird also auch anhand dieser Fakten gemacht.

10.3.3.1 Rückkopplungen

Da das Expertensystem Rückkopplungen unterstützt, müssen nachträgliche Änderungen der Benutzereingaben besonders behandelt werden. So erlaubt das System z.B., dass der Benutzer nach der Anzeige der Realisierungsmassnahmen die Dienstauswahl nochmals aufrufen und seine dortigen Angaben revidieren kann. Somit können Fakten hinfällig werden, die zur Auswahl des Realisierungsportfolios geführt haben. Um das System immer in einem gültigen Zustand zu halten, wird in dem Expertensystem Prototypen Backtracking verwendet. Werden in einem Schritt Änderungen gemacht, die eine Revision der daraus folgenden Resultate bedingen, werden alle Fakten ab dieser Änderung gelöscht. Die neuen Fakten werden beim Fortschreiten in der Sicherheitskonzeption dann wieder neu aufgebaut.

Dieses Kapitel hat einen Einblick in die Implementierung gegeben. Die wichtigsten Elemente des Expertensystems sind die Objekte, die Regeln und die Prolog–Fakten. Da das Expertensystem unzählige Objekte und Regeln beinhaltet, konnte nur ein kleiner Ausschnitt der Implementierung gezeigt werden.

10.4 Zusammenfassung

Das Rapid Secure Development (RSD) ist ein konzeptionelles Vorgehen, das zum Ziel hat, ausgehend von einem Nutzungskonzept eine sichere Implementierung der Internet–Dienste zu erreichen. Das komplexe Thema der Internet–Sicherheit kann durch geeignete Abstraktion in die fünf Schritte Nutzungskonzeption, Dienstauswahl, Risikoanalyse, Gegenmassnahmen und Realisierung aufgeteilt werden. Das hier vorgestellt Expertensystem orientiert sich an den fünf Schritten der RSD–Verfahrens und verbindet wirkungsvoll das konzeptionelle Vorgehen und die Realisierungsdetails.

Dank der Verbundenheit mit dem RSD wird mit Hilfe des Expertensystems neben einer Lösung für die Internet–Sicherheitsproblematik auch die Konzeption an sich kommuniziert. Das Expertensystem stellt also eine weitere Ergänzung des entwickelten RSD–Verfahrens dar. Es ist ein zusätzliches Wissensmedium, das den Wissenstransfer von der Forschung zu

den Unternehmungen unterstützt. Die Darstellung des Verfahrens in Form eines Software–Hilfsmittels bietet dem Anwender dabei den Vorteil einer geringeren Einarbeitungszeit und einer kleineren Fehlerrate. Um mit der Sicherheitskonzeption beginnen zu können, muss der Anwender nur Kenntnisse über die grundlegenden Schritte der Sicherheitskonzeption besitzen. Alles weitere benötigte Wissen wird ihm während der Benutzung mittels der eingebundenen HTML–Dokumente vermittelt. Durch das programmatische Vorgehen können Flüchtigkeitsfehler und Ungenauigkeiten vermieden bzw. verhindert werden. Zudem lädt das System dazu ein, verschiedene Szenarien durchzuspielen und dabei die Risikoentwicklung sowie vorgeschlagene Sicherheitsmassnahmen zu betrachten.

Einen grossen Teil des Expertensystems nimmt die Risikoanalyse ein. Eine ausdrucksstarke Bewertung der Risiken ist unbedingt notwendig, um aus den Risiken notwendige Sicherungsmassnahmen ableiten zu können. Über die Häufigkeit der einzelnen Gefahren gibt es praktisch noch keine Erfahrungen. Aus diesem Grund wurden verschiedene Schätzungen angestellt. Trotzdem bleibt zu hoffen, dass auf diesem Gebiet weitere Untersuchungen durchgeführt werden, so dass besser abgestützte Angaben über die empirischen Häufigkeiten der einzelnen Gefahren gemacht werden können.

Bei der Entwicklung des Prototyps wurde darauf geachtet, dass weitere Realisierungsmassnahmen ohne grossen Aufwand implementiert werden können. Ebenso ist die Integration weiterer Dienste, Gefahren und Gegenmassnahmen einfach zu bewerkstelligen. Dies ist eine notwendige Voraussetzung dafür, dass das RSD–XPS mit der raschen Entwicklung des Internet Schritt halten kann.

Teil III

Sicherheitsmassnahmen im Detail

Kapitel 11

Kryptographische Massnahmen

11.1 Einleitung

Um eine Zugriffskontrolle für Rechner, Dienste oder Daten zu realisieren, die bestimmten Benutzern mehr Privilegien als anderen einräumen, oder zur Sicherung von Daten beim Transport über unsichere Netze sind kryptographische Methoden unerlässlich. Das folgende Kapitel soll einen Überblick über gängige Techniken der Kryptographie geben, die Grenzen aufzeigen und den aktuellen Stand der Forschung in diesem Gebiet umreissen.

11.2 Klassische Verschlüsselungssysteme

11.2.1 Definition

Eine (klassische) Verschlüsselungsfunktion oder Chiffre ist abstrakt gesprochen eine Funktion $f_k : M \to C$. Dabei ist M die Menge aller Klartexte, C die Menge aller Chiffrate, und k ein Element aus K, der Menge aller Schlüssel. Die Entschlüsselung wird mit einer Funktion $f_k^{-1} : C \to M$ durchgeführt. An eine Verschlüsselungsfunktion werden folgende Anforderungen gestellt:

- Bei Kenntnis des Schlüssels k soll das Chiffrat $c = f_k(m)$ eines Klartextes m effizient berechenbar sein.
- Bei Kenntnis des Schlüssels k soll der Klartext $m = f_k^{-1}(c)$ aus einem Chiffrat c zurückgewonnen werden können.

Der Absender und der Empfänger einer Nachricht müssen über einen gemeinsamen Schlüssel k verfügen, den sie vorher über einen sicheren Kanal (am besten persönlich) ausgetauscht haben.

Ein Angreifer, der den Schlüssel k nicht kennt, soll keine Möglichkeit haben, aus der Kenntnis des Chiffrates c auf den Klartext m oder den Schlüssel k zu schliessen. Genauer:

Known Cipertext Attack: Bei Kenntnis des Chiffrates c soll es keinen Weg geben, den Klartext m zu berechnen ohne den Schlüssel k zu kennen, d. h. die Funktion $m = f_k^{-1}(c)$ soll bei unbekanntem k nicht berechenbar sein.

Known Plaintext Attack: Bei Kenntnis des Chiffrates c und des Klartextes m soll es nicht möglich sein, den Schlüssel k zu berechnen, d.h. es soll keine berechenbare Funktion $k = g(m, c)$ geben, so dass $c = f_{g(m,c)}(m)$ gilt.

Chosen Plaintext Attack: Bei freier Wahl eines Klartextes m und Kenntnis des dazugehörigen Chiffrates $c = f_k(m)$ soll es nicht möglich sein, daraus den Schlüssel k zu berechnen, d.h. es soll keine berechenbare Funktion $k = g(m, c)$ geben, so dass $c = f_{g(m,c)}(m)$ gilt.

Die einzige bekannte Chiffre, die beweisbar sicher ist, ist der One–Time–Pad: Dabei ergibt sich das Chiffrat aus der bitweisen XOR–Verknüpfung von Klartext und Schlüssel. Der Schlüssel besitzt also die gleiche Länge wie das Chiffrat und wird nur einmal verwendet. Nimmt man ideale stochastische Eigenschaften der Schlüssel–Bitfolge an, so ist zu einem Chiffrat jede beliebige Bitfolge als dazugehöriger Klartext gleich wahrscheinlich. Somit ist die erste Forderung erfüllt. Die zweite und dritte Forderung sind, so wie sie oben formuliert sind, zwar nicht erfüllt, denn sind Klartext und Chiffrat bekannt, kann man den Schlüssel sofort errechnen. Da beim One–Time–Pad aber jeder Schlüssel nur einmal verwendet wird, sind Known Plaintext Attack und Chosen Plaintext Attack wirkungslos, denn sie liefern keine Infomationen für das Entschlüsseln zukünftiger Chiffrate.

11.2.2 Die Caesarchiffre

Die Caesarchiffre ist ein Beispiel für eine sehr einfache Chiffre, anhand der man sich obige Forderungen klarmachen kann. Sie geht auf Julius Caesar zurück, der damit Lageberichte in Schlachten verschlüsselt haben soll [Bau94, S. 67]. Das Klartext- und Chiffretext–Alphabet besteht jeweils aus den 26 Buchstaben A,B,C,...,Z, denen die Zahlenwerte 0,1,2,...,25 zugeordnet werden. Der Schlüssel k ist eine Zahl zwischen 1 und 25. Beim Verschlüsseln werden die Buchstaben einfach um k Stellen im Alphabet verschoben. Ist beispielsweise $k = 3$, so wird der Text ARBEITSTAGUNG zu DUEHLWVWDJXQJ chiffriert. Mathematisch ausgedrückt ist

$$c = f_k(m) = (m + k) \bmod 26$$

die Verschlüsselungsfunktion der Caesarchiffre.

Die drei oben genannten Kriterien werden von der Caesarchiffre alle nicht erfüllt.

- Ist ein Chiffrat bekannt, so benötigt man lediglich 25 Versuche, um den Klartext herauszubekommen. Dies kann nicht als schwierig bezeichnet werden. Wenn man weiss, dass der Klartext in deutscher / englischer / ... Sprache abgefasst ist, und kennt man Häufigkeitstabellen der Buchstaben in den jeweiligen Sprachen, dann genügen sogar erheblich weniger Versuche.

- Sind Klartext und Chiffrat bekannt, erhält man sofort den Schlüssel, indem man einfach die Zahlenwerte der Buchstaben voneinander subtrahiert.
- Darf der Klartext frei gewählt werden, ergibt die Eingabe eines A (Zahlenwert 0) sofort den Schlüssel.

Varianten der Caesarchiffre sind die Vigineré–Chiffre, bei der verschiedene Schlüsselwerte zyklisch zum Einsatz kommen, und die Permutationschiffre, bei der die Klartextbuchstaben nicht durch eine Verschiebung im Alphabet, sondern durch eine Permutation auf die Buchstaben des Chiffrates abgebildet werden. Auch diese Varianten halten zumindest keiner chosen–plaintext–Attacke stand: Die Eingabe von AAAAA... bei der Vigineré–Chiffre bzw. ABCD...Z bei der Permutationschiffre liefert sofort den Schlüssel.

11.2.3 Klassifizierung von Chiffren

Klassische Chiffren werden in der Literatur in Strom- und Blockchiffren unterteilt. Bei Stromchiffren ergibt sich ein Zeichen des Chiffrates aus jeweils einem Zeichen des Klartextes und des Schlüssels. Bei Blockchiffren erfolgt die Chiffrierung in Blöcken von fest vorgegebener Länge. Innerhalb eines Chiffratblockes hängt jedes Zeichen von allen Zeichen des dazugehörigen Klartextblockes ab. Strenggenommen gibt es keine Unterscheidung zwischen Strom- und Blockchiffren. Denn durch geeignete Definition des Begriffes ÒZeichenÓ lässt sich jede Blockchiffre auch als Stromchiffre darstellen. Beispielsweise ist die oben beschriebene Caesarchiffre eine Stromchiffre, denn jeder Buchstabe des Chiffrates ergibt sich aus genau einem Buchstaben des Klartextes. Definiert man als Zeichen jedoch nicht einen Buchstaben, sondern ein Bit der (ASCII-) Repräsentation des Buchstabens, so handelt es sich um eine Blockchiffre mit Blockgrösse 8 Bit. In der Literatur hat sich die Unterscheidung jedoch durchgesetzt, und sowohl die Caesar–Chiffre als auch der One–Time–Pad werden zu den Stromchiffren gezählt.

Abgesehen vom One–Time–Pad, das u.a. zur Verschlüsselung des roten Telefons zwischen Moskau und Washington verwendet wird, ist heute keine ernsthafte Verwendung von Stromchiffren mehr öffentlich bekannt. Statt dessen kommen Blockchiffren zum Einsatz, wenn klassische Kryptographie benötigt wird.

11.2.4 Der Data Encryption Standard (DES)

Die wohl bekannteste Blockchiffre ist der Data Encryption Standard (DES), der 1977 von der NSA entworfen wurde und bis heute im Einsatz ist [X3.81]. Beim DES werden Klartextblöcke zu je 64 Bit auf 64 Bit lange Chiffretextblöcke abgebildet. Die Schlüssellänge beträgt 56 Bit. Die Sicherheit des DES ist umstritten. Sowohl die Geheimniskrämerei der NSA bezüglich der Entwurfskriterien für den DES als auch mehrere veröffentlichte Angriffsversuche geben Anlass zu der Vermutung, dass der DES knackbar ist. Es ist jedoch bis heute kein effizienterer Angriff als das vollständige Absuchen aller Schlüssel bekannt.

Der Hauptkritikpunkt am DES ist jedoch die geringe Schlüssellänge von 56 Bit. In [DH77] wird beschrieben, wie mit Investitionskosten von einer Million US–Dollar eine Maschine gebaut werden kann, die eine DES–Verschlüsselung in einem Tag knackt. Berücksichtigt man den schnellen technischen Fortschritt, so kann man davon ausgehen, dass in wenigen Jahren eine solche Maschine erheblich billiger und schneller sein wird und somit eine Schlüssellänge von 56 Bit keinen ernsthaften Schutz mehr bieten kann. Ein Vorschlag zur Lösung dieses Problems ist der Triple–DES (3DES), bei dem drei DES–Verschlüsselungen hintereinander ausgeführt werden. Aus einem Klartext m wird beim 3DES das Chiffrat c mithilfe zweier 56–Bit–Schlüssel k und l berechnet:

$$c = DES_k(DES_{l-1}(DES_k(m)))$$

Der Vorteil des 3DES besteht darin, dass zur Verschlüsselung weiterhin die alten DES–Hardwareimplementierungen verwendet werden können, die schnell arbeiten und billig erhältlich sind. Die Vermutung, dass die Menge der DES–Verschlüsselungen bezüglich Komposition abgeschlossen ist und somit der 3DES die gleiche kryptographische Komplexität wie ein einfacher DES besitzt, konnte erst 1992 widerlegt werden [CW92].

11.2.5 Der International Data Encryption Algorithmus (IDEA)

Eine neuere Blockchiffre, bei der die Entwurfskriterien öffentlich bekannt sind, ist der International Data Encryption Algorithm (IDEA[1]). Die Blockgrösse von Klartext und Chiffrat beträgt wie beim DES 64 Bit, die Schlüssellänge jedoch 128 Bit. Eine IDEA–Verschlüsselung besteht aus neun Runden. In jeder Runde werden die Daten in 4 Wörter zu je 16 Bit aufgeteilt. Diese Wörter werden mittels dreier Verknüpfungsoperationen miteinander und mit 16 Bit langen Teilschlüsseln verknüpft. Zu den wesentlichen Stärken des IDEA gehört, dass beim Entwurf auf die Beweisbarkeit seiner kryptographischen Eigenschaften Wert gelegt wurde. Das Chiffrat soll in einer möglichst schwer vorhersagbaren Weise vom Klartext abhängen (Konfusion), und jedes Bit der Ausgabe soll von jedem Bit der Eingabe abhängen (Diffusion):

- Der Zusammenhang zwischen Chiffrat und Klartext ist sehr komplex. Dazu wurden die drei Verknüpfungen so gewählt, dass sie möglichst wenig mathematische Struktur haben. Zum Beispiel gelten für zwei verschiedene Verknüpfungen keine Assoziativ- und Distributivgesetze, es sind also für 16–Bit–Wörter a, b und c zwei verschiedene Verknüpfungen $\diamond$ und $\star$

  $$(a \diamond b) \star c = a \diamond (b \star c) \qquad \text{oder} \qquad (a \diamond b) \star c = (a \star c) \diamond (b \star c)$$

  keine allgemeingültigen Formeln.

- Es gelten keine Isomorphiebeziehungen zwischen den Verknüpfungen, es gibt also beweisbar keine Funktion f, die

  $$f(a \diamond b) = f(a) \star f(b)$$

[1] siehe http://www.ascom.ch/systec/idea.html

für zwei unterschiedliche Verknüpfungen $\diamond$ und $\star$ erfüllt.

- Die Anordnung dieser Operationen sind so gewählt, dass die daraus entstehenden Ausdrücke nicht vereinfachbar sind. Insbesondere gilt für die vier Verknüpfungen im mittleren Block jeder Runde, dass es beweisbar keine Verknüpfungen geben kann, die das Ergebnis mit nur drei Rechenschritten errechnen können.
- In jeder Runde hängen alle 64 Ausgabebits von allen 64 Eingabebits der Rundenfunktion ab. Kleine Aenderungen bei der Eingabe wirken sich also stark in der Ausgabe aus.

Aufgrund seiner übersichtlichen Struktur ist der IDEA sehr einfach in Software zu implementieren. Die Ausführungsgeschwindigkeit ist jedoch geringer als bei den anderen gängigen Blockchiffren.

Der wesentliche Nachteil des IDEA ist die Tatsache, dass der Algorithmus patentiert ist. Das Patent wird von der Schweizer Firma Ascom gehalten, die IDEA in Zusammenarbeit mit der ETH Zürich entworfen hat. Die Lizenzbedingungen erlauben die unentgeltliche private Nutzung des Algorithmus; bei gewerblicher Nutzung werden jedoch Lizenzgebühren fällig [Sch96, S. 325].

11.3 Public Key Kryptographie

Public Key Kryptosysteme (PKK), auch asymmetrische Kryptosysteme genannt, stellen eine jüngere Entwicklung der Kryptographie dar. Unter diesem Begriff werden mehrere Grundfunktionen zusammengefasst:

- **Verschlüsselung.** Im Unterschied zu den klassischen (symmetrischen) Systemen werden Ver- und Entschlüsselung mit unterschiedlichen Schlüsseln durchgeführt. Jeder Teilnehmer in einem Public–Key–Verschlüsselungssystem besitzt zwei Schlüssel, einen geheimen und einen öffentlichen. Letzterer wird allgemein bekannt gemacht; er ermöglicht es, Dokumente zu verschlüsseln, die ausschliesslich mit dem privaten Schlüssel wieder entschlüsselt werden können.

  Anschaulich kann man sich ein Public–Key–Verschlüsselungssystem wie einen privaten Briefkasten vorstellen. Jeder kann Dokumente in einen Briefkasten einwerfen. Nur der Hausbewohner besitzt jedoch den Schlüssel, kann den Briefkasten öffnen und die Dokumente lesen.

- **Signaturen.** Der Begriff der digitalen Signatur ist erst mit der Erfindung der Public Key Kryptographie aufgekommen, obwohl sich Signaturschemata auch mit symmetrischer Kryptographie realisieren lassen. Bei einem Public–Key–Signatursystem wird die Signatur zu einem Dokument mit dem geheimen Schlüssel eines Teilnehmers erstellt und ist mit dem öffentlichen Schlüssel von allen Teilnehmern überprüfbar. Sie stellt einen Beweis dafür dar, dass das Dokument mit dem zugehörigen privaten

Schlüssel signiert und seitdem nicht mehr verändert wurde. Die digitale Signatur ist das Analogon zur herkömmlichen Unterschrift unter ein Dokument.

- **Blinde Signaturen.** Eine blinde Signatur ist ein Verfahren zur Signierung eines Dokumentes mit einem privaten Schlüssel, ohne dass der Signierende Einblick in das Dokument nehmen kann. Dennoch können andere, denen das Dokument offen vorliegt, die Signatur überprüfen.

  Den Einsatz eines blinden Signaturschemas kann man anschaulich mit einer Wahl erklären. Der Wähler füllt seinen Stimmzettel im Geheimen aus und steckt ihn in den Wahlumschlag. Diesen wirft er unter den Augen der Aufsichtspersonals in die Wahlurne. Das Personal bestätigt die Stimmabgabe in Form eines Eintrages in die Wahlunterlagen, ohne den Inhalt des Stimmzettels zu kennen.

In einem Public–Key–Kryptosystem besitzt jeder Teilnehmer zwei Schlüssel, einen privaten und einen öffentlichen. In der Literatur wird der öffentliche Schlüssel meist mit e (für encryption), der private mit d (für decryption) bezeichnet. Ein Public–Key–Kryptosystem besteht also aus zwei Funktionen f_e und g_d mit den Eigenschaften

$$c = f_e(m) \qquad \text{und} \qquad m = g_d(c)$$

sowie einer Schlüsselerzeugungsfunktion

$$(d, e) = \text{kgen}(random)$$

Die Arbeitsweise der Funktionen f und g ist allen Teilnehmern bekannt. Zwischen den beiden Schlüsseln besteht ein mathematischer Zusammenhang.

Jeder Teilnehmer in einem Public–Key–Verschlüsselungssystem besitzt also ein Schlüsselpaar (e, d). e wird als öffentlicher Schlüssel bezeichnet; er ist jedem anderen Teilnehmer des Systems bekannt. d ist dagegen der private Schlüssel und muss geheim gehalten werden. Der Vorteil eines Public–Key–Kryptosystems besteht darin, dass jeder Teilnehmer eine verschlüsselte Nachricht generieren kann, die nur von einem bestimmten anderen Teilnehmer entschlüsselt werden kann.

Für die Funktionen f und g muss also gelten

1. Für jeden Klartext m muss $g_d(f_e(m)) = m$ sein, d.h. g_d ist die Umkehrfunktion von f_e.
2. Ein Angreifer, der eine verschlüsselte Nachricht in die Hände bekommt, darf trotz Kenntnis von c und e in der Beziehung $c = fe(m)$ nicht den Klartext m zurückgewinnen können.
3. Ein Angreifer, der eine Nachricht mit dem öffentlichen Schlüssel eines anderen Teilnehmers verschlüsselt, darf trotz Kenntnis von m und c in der Beziehung $m = g_d(c)$ nicht den privaten Schlüssel d errechnen können.

Aus (1) folgt, dass zwischen den Schlüsseln e und d mathematisch gesehen eine Beziehung bestehen muss. (2) besagt, dass es nicht möglich sein darf, die Funktion f_e umzukehren, obwohl eine Umkehrfunktion (nämlich g_d) existiert. (3) heisst mathematisch ausgedrückt, dass die Gleichung $m = g_d(c)$ nicht nach d auflösbar sein darf.

Es ist leicht einzusehen, dass zumindest die Forderung (2) nicht erfüllbar ist. Denn wenn eine Umkehrfunktion zu f_e existiert, kann man auch einen Algorithmus angeben, der sie berechnet: Nämlich das Durchprobieren aller Klartexte m, solange bis ein m mit $c = fe(m)$ gefunden ist. In den real existierenden Public–Key–Kryptosystemen ist daher in (2) (und ebenso in (3)) die Formulierung dahingehend abzuschwächen, dass es keine Möglichkeit geben darf, innerhalb vernünftiger Zeit die gesuchte Grösse zu ermitteln. Auch diese Forderung ist (bisher) noch nicht mit konkreten Funktionen erfüllbar. Denn bisher ist noch kein schwieriges mathematisches Problem bekannt, das beweisbar eine Lösung hat, es aber beweisbar keinen effizienten Algorithmus zur Lösungsfindung gibt. Um Public Key Kryptosysteme realisieren zu können, bedient man sich daher mathematischer Funktionen, für die trotz intensiver Forschungsarbeit noch kein effizienter Algorithmus zur Berechnung der Umkehrfunktion gefunden wurde.

Ohne auf die Details konkreter Realisierungen von Public Key Kryptosystemen einzugehen, seien hier kurz einige Beispiele für Funktionen genannt, mit denen die genannten Bedingungen zu erfüllen versucht werden:

- **Das Faktorisierungsproblem.** Seien p und q zwei grosse Primzahlen (mit über 100 Dezimalstellen). Dann ist es leicht, die Funktion $n = h(p, q) := p * q$ zu berechnen, d.h. die beiden Primzahlen miteinander zu multiplizieren. Umgekehrt ist es jedoch sehr schwer, ein bekanntes Produkt zweier grosser Primzahlen zu faktorisieren, d.h. p und q aus gegebenem n zu berechnen. Ein guter Überblick über das Faktorisierungsproblem findet sich in [Bre89].

- **Der diskrete Logarithmus.** Sei p eine grosse Primzahl und $(\mathrm{K}[p], +, *)$ der endliche Körper mit p Elementen, dargestellt durch die natürlichen Zahlen $\{0, 1, \ldots, p-1\}$ und die Verknüpfungsoperationen modulo p. Dann ist es leicht, zu festem $a > 1$ und gegebenem x ein y mit $y = ax \bmod p$ zu berechnen. Umgekehrt ist es dagegen schwer, aus einem bekannten y das x zurückzugewinnen, d.h. den Logarithmus in einem Primkörper zur Basis a zu berechnen. Ein Überblick über den Stand der Forschung zum diskreten Logarithmus–Problem findet man in [McC90].

- **Das diskrete Wurzel–Problem.** Sei n eine grosse zusammengesetzte Zahl. Dann ist es leicht, zu einem festen $a > 1$ und gegebenem x ein y mit $y = xa \bmod n$ zu berechnen. Ist dagegen y bekannt, so ist es schwer, x zu berechnen. Falls n ein Produkt zweier Primzahlen ist, dann ist die Berechnung der diskreten Wurzel modulo n vom Aufwand her äquivalent zur Faktorisierung von n [Rab79].

„leicht“ bedeutet dabei jeweils, dass es einen effizienten Algorithmus für die Berechnung gibt; „schwer“ dagegen heisst, dass bisher kein Algorithmus gefunden wurde, der das

Problem in vernünftiger Zeit löst. Sollte sich dies einmal ändern, sind die Public–Key–Verfahren, die auf diesen Problemen beruhen, gebrochen.

11.3.1 RSA

Das wohl bekannteste Public Key Kryptosystem ist das nach seinen Erfindern Rivest, Shamir und Adleman benannte RSA [RSA78]. Es basiert auf allen drei der oben genannten Einwegfunktionen; wird auch nur eines der beschriebenen Probleme gelöst, so ist RSA gebrochen. Dies ist jedoch in den letzten 21 Jahren nicht geschehen (oder zumindest ist nichts davon öffentlich bekannt).

Mit RSA können zwei der oben genannten Verfahren durchgeführt werden, nämlich Public–Key–Verschlüsselung und digitale Signaturen.

Die Schlüsselpaare können von den Benutzern selbst erzeugt werden, es ist keine Schlüsselvergabeinstanz notwendig. Der öffentliche Schlüssel besteht bei RSA aus zwei Zahlen, eine in der Grössenordnung von 400 bis 800 Dezimalstellen, die andere im allgemeinen 1–2 Stellen. Der private Schlüssel enthält eine Zahl von etwa gleicher Länge wie die grosse Zahl des öffentlichen Schlüssels.

RSA–Berechnungen bestehen aus Langzahlarithmetik mit Zahlen in der Grössenordnung der Schlüssellänge. Daher sind sie erheblich langsamer als Chiffrier- und Dechiffrieroperationen bei Blockchiffren. Eine SPARC II braucht bei einem 1024–Bit–Schlüssel etwa eine Sekunde für die geheimen Berechnungen (Dechiffrierung und Signaturerstellung), die öffentlichen Operationen (Chiffrierung und Signaturprüfung) sind etwa um Faktor 10 schneller; dies ist eine Besonderheit bei RSA und hängt damit zusammen, dass die Zahlenwerte des öffentlichen Schlüssels speziell auf Geschwindigkeit optimiert werden können. Hardwareimplementierungen erreichen etwa die 5- bis 10–fache Geschwindigkeit [Sch96, S. 469].

Das RSA–Verfahren ist in den USA patentiert, in anderen Ländern ist dieses Patent jedoch ungültig. Das USA–Patent läuft im September 2000 aus [Sch96, S. 474].

11.3.2 Das El–Gamal Signaturverfahren

Ein Verfahren, mit dem auf Basis des diskreten–Logarithmus–Problems digitale Signaturen erstellt werden können, ist das El–Gamal–Verfahren [ElG84, ElG85]. Varianten des ElGamal–Verfahrens erlauben komplexere Anwendungen wie Identitätsbeweise, Passwort–Authentifikation oder Schlüsselaustausch.

Die Komplexität der Berechnungen liegt bei gleicher Schlüssellänge etwas höher als bei RSA. Das El–Gamal–Verfahren basiert jedoch ausschliesslich auf dem diskreten–Logarithmus–Problem. Für dieses sind weniger effiziente Algorithmen bekannt als für das Faktorisierungsproblem, so dass ein El–Gamal–Kryptosystem bei gleicher Sicherheit mit kleinerer Schlüssellänge auskommt. Typische Schlüssellängen sind 60–200 Dezimalstellen lang. Ein öffentlicher Schlüssel besteht aus zwei Zahlen dieser Grössenordnung.

Das El–Gamal–Verfahren ist nicht patentiert. Jedoch behauptet die Firma Public Key Partners (PKP) in den USA, El–Gamal falle unter das Patent des Diffie–Hellmann–Kryptosystems [HDM80]. Da jedoch auch dieses Patent 1997 ausgelaufen ist, wird die Streitfrage wohl nie geklärt werden [Sch96, S. 479].

11.3.3 Kryptographische Hashfunktionen

Ein wesentliches Hilfsmittel zur Konzeption kryptographischer Systeme stellen kryptographische Hashfunktionen (auch Einwegfunktionen genannt) dar.

Eine herkömmliche Hashfunktion h bildet eine Menge von Texten T auf eine Menge von Hashsummen S ab, wobei S wesentlich kleiner als T ist. Es gibt Kollisionen, also unterschiedliche Texte t_1, t_2 aus T mit $h(t_1) = h(t_2)$. Die Funktion h ist so gewählt, dass alle Hashwerte möglichst gleich häufig auftreten sollen. An eine kryptographische Hashfunktion werden zusätzlich noch folgende Forderungen gestellt:

1. Es soll unmöglich sein, zu einer gegebenen Hashsumme s einen Text t zu finden, so dass $s = h(t)$ gilt.
2. Es soll unmöglich sein, zwei unterschiedliche Texte t_1 und t_2 zu finden, so dass $h(t_1) = h(t_2)$ gilt.

Es ist leicht einzusehen, dass Forderung (2) nicht erfüllbar ist. Denn da Kollisionen existieren, lassen sie sich auch durch systematisches Probieren aller Kombinationen finden. Eine ähnliche Überlegung gilt für (1). Für praktische Realisierungen von kryptographischen Hashfunktionen ist daher das „unmöglich“ durch „praktisch unmöglich“ zu ersetzen: Die bekannten Realisierungen haben eine Hashsummenlänge von 128 Bit oder mehr, und erfordern daher das Durchprobieren von etwa 2^{128} Texten zum Verletzen der ersten, sowie etwa 2^{64} Texten zum Verletzen der zweiten Forderung. Natürlich ist nicht garantiert, dass zu einer konkreten Einwegfunktion einmal ein besserer Algorithmus gefunden wird, der die Umkehrfunktion effizienter berechnet; somit muss es in (1) und (2) strenggenommen sogar „mit den bekannten Algorithmen nicht durchführbar“ heissen.

11.3.4 Der Message Digest 5 (MD5)

Der Message Digest 5 (MD5) von Rivest [Riv92] ist eine Realisierung einer kryptographischen Hashfunktion. Seine Eingabe kann eine Bitfolge beliebiger Länge sein, die Ausgabe besteht aus 128 Bit. Beim MD5 werden jeweils 128 Bit der Eingabe genommen und von einer Funktion mit Gedächtnis weiterverarbeitet. Das Ergebnis dieser Verarbeitung dient als Gedächtnis für die nächsten 128 Bit der Eingabe. Auf diese Weise hängt die Ausgabe gleichermassen von der gesamten Eingabe ab. Am Anfang ist das Gedächtnis mit einem fest gewählten Initialisierungswert gefüllt.

MD5 ist die verbesserte Version von MD4, bei dem einige Regelmässigkeiten in der Berechnung der Hashsumme aufgedeckt wurden, die für die Kryptoanalyse möglicherweise relevant sein können. Der MD5 galt lange Zeit als sehr sicher. Erst 1996 wurde eine Schwachstelle entdeckt, durch die bei einem veränderten Initialisierungswert der Aufwand zum Auffinden einer Kollision stark verkleinert werden kann [Dob96]. Ob dieses Ergebnis Auswirkungen auf die Sicherheit von MD5 hat, ist noch unbekannt.

11.4 Kryptographische Produkte und Standards

11.4.1 Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) ist eine Implementierung eines Public–Key–Kryptosystems. Sie erlaubt das Verschlüsseln und/oder Signieren von Dateien und bietet eine komfortable Schlüsselverwaltung.

Intern basiert PGP auf RSA, IDEA und MD5. Jeder Teilnehmer kann sich ein Schlüsselpaar selbst erzeugen und seinen öffentlichen Schlüssel verbreiten; eine Vertrauensinstanz ist nicht notwendig.

Bei einer Verschlüsselung mit PGP wird zunächst ein zufälliger Session–Key generiert und die Datei mit diesem Key IDEA–verschlüsselt. Anschliessend wird der Session–Key mit dem öffentlichen RSA–Schlüssel des Empfängers IDEA–verschlüsselt und beide Chiffrate zusammen in eine Datei gepackt. Diese Datei kann man über ein unsicheres Netz an den Empfänger schicken, beispielsweise per E–Mail. Nur der richtige Empfänger kann mit seinem privaten RSA–Schlüssel die RSA–Chiffrierung rückgängig machen und den Session–Key wiedergewinnen. Damit kann schliesslich der Text dechiffriert werden.

Zur Berechnung einer Signatur wird zunächst die MD5–Hashsumme des zu signierenden Textes berechnet und mit dem privaten RSA–Schlüssel des Erstellers signiert. Klartext und Signatur werden zusammen veröffentlicht. Ein Leser, der den öffentlichen Schlüssel des Signierenden besitzt, kann die Signaturoperation rückgängig machen und erhält somit die MD5–Hashsumme zurück. Wenn sie mit einer selbstberechneten Hashsumme über den empfangenen Text übereinstimmt, war die Signatur korrekt; ansonsten ist entweder Text/Signatur unterwegs verfälscht worden, oder der Signaturschlüssel war nicht das Gegenstück zu dem bekannten öffentlichen Schlüssel.

Der Schwachpunkt bei dieser Art von Verschlüsselung ist allerdings der Austausch des öffentlichen Schlüssels. Wer eine Nachricht mit einem öffentlichen Schlüssel chiffriert, muss sicherstellen, dass dies wirklich der Schlüssel des gewünschten Empfängers ist. Ansonsten könnte ein Angreifer einen selbstgenerierten Schlüssel unter dem Namen eines anderen Netzteilnehmers veröffentlichen. Er besässe dann den zugehörigen privaten Schlüssel und könnte Chiffrate an den öffentlichen Schlüssel dechiffrieren.

Zur Überprüfung der Authentizität eines Schlüssels bietet PGP einen Mechanismus an: Der Fingerprint ist eine kryptographische Hashsumme über den öffentlichen Schlüssel. Jeder Teilnehmer kann ihn selbst berechnen und auf einem sicheren Kanal (am besten persönlich;

für normale Sicherheitsanforderungen genügt wohl auch ein Telefonat) mit dem Besitzer vergleichen.

Schlüssel, bei denen man sicher ist, dass sie dem vorgeblichen Besitzer gehören, kann man mit dem eigenen privaten Schlüssel signieren und signiert weitergeben. Auf diese Weise entsteht das Web of Trust: Wenn ein Teilnehmer A den öffentlichen Schlüssel von B signiert hat (also sicher ist, dass der zugehörige private Schlüssel dem Teilnehmer B gehört) und der Schlüssel von C eine Signatur von B trägt (angefertigt mit dem privaten Schlüssel von B), dann hat A damit einen Beweis der Authentizität des Schlüssels von C. Voraussetzung ist natürlich, dass B den Schlüssel wirklich vor dem Signieren überprüft hat. PGP findet solche Vertrauensketten automatisch bzw. gibt eine Warnung aus, wenn ein Schlüssel benutzt wird, zu dem keine Vertrauenskette besteht.

11.5 Kryptographische Protokolle

Basierend auf den Grundtechniken der klassischen Kryptograhie, der Public–Key–Kryptographie und der Einwegfunktionen lassen sich Protokolle generieren, mit denen sehr komplexe Anforderungen realisiert werden können. Beispiele dafür sind:

Coin Flipping: Von zwei Personen, die nur über Telekommunikationsleitungen miteinander kommunizieren können, wirft einer eine Münze (d.h. trifft eine Zufallsentscheidung) und beweist dem anderen das Ergebnis.

Key Escrow: Ein Geheimnis, z.B. ein Schlüssel, wird auf mehrere Personen / Instanzen verteilt. Um es zu rekonstruieren, ist die Zusammenarbeit aller oder eines bestimmten Teils der Geheimnisträger nötig.

Vertragsabschluss: Mehrere Parteien einigen sich auf ein Dokument / Vertrag / Zahlenwert. Die Einigung tritt gleichzeitig bei allen Teilnehmern in Kraft, und zwar genau dann, wenn alle Parteien den Vertrag unterschrieben haben.

Auktionen: Mehrere Personen geben verdeckt Angebote ab. Am Ende wissen alle, welcher Teilnehmer das höchste Gebot abgegeben hat; die übrigen Gebote sind jedoch keinem anderen Teilnehmer bekannt. Die aktuellen Protokolle setzen jedoch voraus, dass sich mehr als zwei Drittel der Teilnehmer eines Auktionssystems protokollkonform verhalten.

Digital Cash: Die Teilnehmer eines digitalen Zahlungssystems besitzen Münzen verschiedener Werte, die von einer Zentralstelle (der elektronischen Bank) ausgegeben wurden. Die Teilnehmer können mit diesen Münzen handeln, die Echtheit prüfen und sie bei der Bank einzahlen. Es ist nicht möglich, Münzen zu duplizieren. Die Bank kann den Weg der einzelnen Münzen nicht zurückverfolgen, d. h. der Handel mit diesen Münzen hinterlässt keine Datenspur.

Diese Techniken stecken jedoch grösstenteils noch in den Kinderschuhen. Zum einen erfordern sie teilweise einen sehr hohen Rechen- und Kommunikationsaufwand und sind deshalb nur in kleinen Teilnehmerkreisen durchführbar. Zum anderen befinden sich Entwurf und Kryptoanalyse komplexerer Protokolle in einem ständigen Wettlauf, und es werden noch häufig Fehler und Sicherheitslücken in bekannten Protokollen gefunden.

11.6 Aktuelle Forschungsthemen

Es gibt zahlreiche aktuelle Forschungsaktivitäten in der Kryptologie. Sie lassen sich im wesentlichen einteilen in

- Untersuchung der Sicherheit bekannter Algorithmen und Protokolle und Auffinden von Schwachstellen in bestehenden Systemen;
- Entwicklung neuer kryptographischer Verfahren, die trotz der ständig wachsenden Geschwindigkeit von Rechnern und Prozessoren auf absehbare Zeit nicht gebrochen werden können;
- Weiterentwicklung kryptographischer Protokolle für komplexe Anwendungen wie digitale Zahlungsmittel, Vertragsabschluss oder elektronische Wahlen.

Kapitel 12

Firewalls

12.1 Einleitung

Rechneranlagen, wie sie heute von Firmen zur Datenverarbeitung eingesetzt werden, besitzen ein enormes Ausmass an Komplexität. Viele Sicherheitslücken und Angriffsvarianten basieren auf Fehlern im Betriebssystem, in der Anwendungs- oder in der Netzwerksoftware. Wegen der grossen Komplexität dieser Softwareprodukte ist es nicht möglich, diese vollständig fehlerfrei zu halten. Bei Betriebssystemsoftware sind Programmierfehler häufig besonders sicherheitskritisch, da sie meist sehr hohe Privilegien hat, also auf viele Daten zugreifen kann.

Abhilfe gehen diese Angriffsvarianten ist im Allgemeinen nur möglich, indem das fehlerhafte Programm oder das Betriebssystem korrigiert wird. Bei kommerziellen Betriebssystemen ist man dabei meist auf den Hersteller angewiesen, der möglichst schnell einen Patch liefern sollte. Bei einem System, dessen Sourcecode frei verfügbar ist, wie Linux oder einigen BSD–Derivaten, ist dagegen häufig der Patch bereits verfügbar, wenn eine Sicherheitslücke bekannt wird. Bei komplexeren Problemen ist es zumindest möglich, die entsprechende Funktion solange ausser Kraft zu setzen, bis eine verbesserte Systemversion erhältlich ist. Dies setzt natürlich gewisse Kenntnisse von Programmierung und Betriebssystemarchitektur voraus.

Sicherheitslücken dieser Art werden in Form von Sicherheitsbulletins veröffentlicht, die von einer zentralen Stelle, dem Computer Emergency Response Team (CERT), verbreitet werden. Wenn eine solche Lücke entdeckt und an das CERT gemeldet wird, versucht dieses zunächst, die dadurch ermöglichten Angriffe nachzuvollziehen, die betroffenen Systeme zusammenzustellen und die Hersteller zu konsultieren. Daher vergeht häufig einige Zeit, bis eine Lücke grossflächig bekannt wird und Massnahmen zur Beseitigung getroffen werden können. Bis der Systemverantwortliche diese Informationen erhalten, verstanden und umgesetzt hat, können andere die Lücken ausnutzen und in Rechnersysteme einbrechen oder Anlagen lahmlegen. Auch ist nicht garantiert, dass eine Lücke überhaupt an das CERT oder eine andere Organisation gemeldet wird; dies hängt stark von der Motivation des Entdeckers ab.

Aus dem Gesagten folgt, dass es letzten Endes nicht möglich ist, Angriffe auszuschliessen, die auf Fehlern in der Systemsoftware vernetzter Rechner basieren. Geht man jedoch von der Annahme aus, dass die Rechner im lokalen Netzwerk im Wesentlichen von vertrauenswürdigen Personen genutzt werden und die eigentliche Gefahr nur durch Netzwerkzugang von aussen droht, dann ergibt sich ein anderer Ansatzpunkt für die Vermeidung der in diesem Dokument beschriebenen Schwachstellen: Die Sicherung des Übergangspunktes zwischen dem lokalen Netz und dem globalen Internet, oder allgemeiner, den Punkt, an dem das eigene, vertrauenswürdige Netz an ein anderes, weniger vertrauenswürdiges Netz angeschlossen ist.

Hinter dem Begriff Firewall verbirgt sich ein Sammelsurium verschiedener Konzepte und Techniken. Gemeinsam ist allen, dass sie versuchen, an der Schnittstelle (mindestens) zweier, unterschiedlich vertrauenswürdiger Netze einen kontrollierten Übergang zu ermöglichen. Der zentrale Gedanke dabei ist, gerade diejenigen Daten passieren zu lassen, die für den ordnungsgemässen Betrieb der benötigten Dienste erforderlich sind. Nicht benötigte Dienste und Protokolle werden gesperrt. Denn je weniger Protokolle an dem Übergangspunkt zugelassen werden, desto weniger können Sicherheitsmängel der Dienstimplementierungen zu Angriffen auf das interne Netz genutzt werden. Insbesondere sollen diejenigen Daten gesperrt werden, die dazu geeignet sind, bekannte Sicherheitslücken in der Systemsoftware auszunutzen.

Allen Firewall–Systemen gemeinsam ist also das Filtern der Daten anhand definierter Merkmale. Die Regeln für die Filterung werden in Form eines regelbasierten Systems formuliert.

Im einfachsten Fall besteht ein Firewall aus einem Rechner, der an zwei physische Netze angeschlossen ist, und auf dem eine spezielle Software läuft, die für die Filterung zuständig ist. Der Rechner muss so konfiguriert sein, dass er keine Daten direkt vom einen zum anderen Netzwerkinterface transportiert, die nicht das Filterungsprogramm durchlaufen haben.

Die verschiedenen Firewall–Konzepte unterscheiden sich u.A.

- in der Ebene oder den Ebenen des zugrundeliegenden Netzwerkprotokolles, auf der die Daten untersucht werden,
- in der „Intelligenz“ des regelverarbeitenden Systems,
- in den Kriterien, die zur Entscheidungsfindung herangezogen werden,
- in der Transparenz, also der Sicht, in der der Benutzer bzw. die Applikation die Firewall sieht,
- im Grad der Fehlertoleranz und der Ausfallsicherheit und
- im Angebot an zusätzlichen Funktionen, z.B. Protokollierung des Netzwerkverkehrs oder Alarmierung.

Für viele Anwendungen genügt es nicht, einen Firewall mit einem einzelnen Rechner oder Programm zu realisieren. Grund dafür kann z.B. sein, dass der Firewall Daten für unterschiedliche Netzwerkdienste transportieren und filtern muss, die sich gegenseitig in ihren

Sicherheitseigenschaften beeinflussen. In diesem Fall wird ein Firewall aus mehreren Komponenten (Hard- und / oder Software) zusammengesetzt.

Im folgenden sind einige gängige Konzepte anhand von Beispielen aufgeführt. Dabei wird nur noch auf das für uns wichtige TCP / IP Protokoll eingegangen; die Konzepte sind auch auf andere Protokolle wie das Bitnet oder Novell IPX übertragbar, aber die Details bei der Realisierung unterscheiden sich natürlich.

12.2 Paketfilter

Paketfilter verbinden zwei Netzwerke und filtern Daten auf der Vermittlungs- und Verbindungsschicht, also den Schichten 3 und 4 der ISO / OSI Schichtenhierarchie bzw. den Schichten II und III des TCP / IP–Protokolls. Ein Paketfilter ist üblicherweise ein Rechner, der mit mindestens zwei Netzwerkkarten ausgestattet und an verschiedene physikalische Netze angeschlossen ist. Er erbringt eine Routing–Funktionalität zwischen dem inneren und dem äusseren Netz. Allerdings werden nicht alle, sondern nur bestimmte IP–Datagramme weitergeleitet. Daher werden Paketfilter häufig auch Screening Router genannt.

12.2.1 Filterung nach IP–Nummern

Der einfachste Paketfilter benötigt für die Filterung nur zwei Angaben, nämlich Ursprungs- und Ziel–IP–Nummer der einkommenden IP–Pakete. Anhand einer Tabelle entscheidet er, welche Kombinationen durchgelassen und welche ausgefiltert werden. Diese beiden Informationen sind im Header jedes einzelnen IP–Paketes enthalten.

Mit solch einfachen Paketfiltern lässt sich nur eine sehr begrenzte Sicherheit erreichen. Denn die Filterregeln erlauben lediglich, den Zugriff auf einen Rechner aus einem anderen Netz heraus vollständig zu erlauben oder zu verbieten. Eine differenzierte Filterung nach Diensten ist nicht möglich.

Viele kommerziell erhältliche Router beherrschen Filterung nach IP–Nummern. Sie ist einfach zu konfigurieren und kann als Schutzmechanismus für Netze mit geringen Sicherheitsanforderungen oder in Kombination mit anderen Mechanismen ausreichend sein. Ein häufiges Anwendungsbeispiel ist die Abschottung eines Netzes für Studentenarbeitsräume. Aus einem solchen Netz heraus ist oft nur der Zugriff auf Rechner innerhalb des Campus gestattet; auf das restliche Internet darf nicht zugegriffen werden.

12.2.2 Beispiel für Filterung nach IP–Nummern

Eine Firma benutze das Class–C–Netz 193.75.101.0 und bietet ihren Mitarbeitern Zugriff auf das World Wide Web sowie Erreichbarkeit per E–Mail. Der Anschluss des internen Netzes zur Aussenwelt sei durch einen Paketfilter realisiert, der im inneren Netz die IP–Nummer 193.75.101.1 trägt. Die Rechner .2, .3 und .4 seien DNS–Server, Mail–Server und WWW–Proxy, laufen unter UNIX, und werden von den Systemadministratoren verwaltet und auf

Sicherheitsaspekte überprüft. Die übrigen IP–Nummern werden von Mac's und PC's benutzt, die in den Dienstzimmern der Mitarbeiter stehen. Je nach Experimentierfreudigkeit der Benutzer haben die Administratoren kaum die Möglichkeit, auf die dort installierte Software und damit auf die Netzwerksicherheit dieser Rechner Einfluss zu nehmen. Das Setup ist in 12.1 dargestellt.

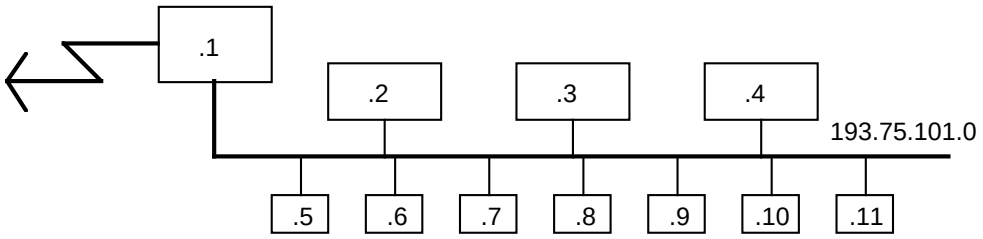

Abbildung 12.1: Einsatz eines einfachen Paketfilters

Eine mögliche Zuordnung von Ursprungs- und Ziel–IP–Nummern zu Erlaubnis- oder Verbotsentscheidungen zeigt Abbildung 12.2.

Ursprung	Ziel	Erlaubnis
193.75.101.1–4	*	+
	193.75.101.1–4	+
193.75.101.5–255	*	–
	193.75.101.5–255	–

Abbildung 12.2: Konfiguration eines einfachen Paketfilters

Die Konfiguration erlaubt also dem Firewall–Rechner und den Servern uneingeschränkten Zugriff auf alle Rechner im Internet. Die übrigen Rechner sind dagegen abgeschottet. Ihre Kommunikation mit der Aussenwelt kann nur mit den Servern erfolgen: Ausgehende E–Mails werden an die Server geschickt, die sie wiederum weiterverteilen; einkommende Mail wird vom Mail–Server entgegengenommen und kann von den Benutzermaschinen abgeholt werden. WWW–Verbindungen sind nur über den Proxy–Server möglich. Der Server nimmt die Anfragen der Clients entgegen, setzt sie um in Anfragen an die Server in der weiten Welt, nimmt das Ergebnis (meist eine HTML–Seite) entgegen und liefert diese an den Client.

Der Sicherheitsgewinn besteht im Wesentlichen darin, dass keine IP–Pakete direkt von aussen an die Benutzermaschinen gelangen können. Hat zum Beispiel ein Benutzer versehentlich einen FTP–Server auf seinem Mac gestartet und kein Passwort gesetzt, könnten andernfalls sämtliche Dokumente auf der lokalen Platte von aussen abgerufen werden. Dies wird durch den Firewall verhindert.

Diese einfache Form der Paketfilterung hat jedoch erhebliche Nachteile: Die Entscheidung für Erlaubnis oder Verbot einer Kommunikation zwischen zwei Rechnern hängt ausschliesslich von den IP–Nummern der Rechner ab. Es ist nicht möglich, nur bestimmte Protokolle der Transportschicht (wie z.B. TCP, UDP), oder gar bestimmte Dienste, die auf der Transportschicht aufbauen, zu erlauben.

Beispielsweise gibt es in obigem Szenario Schwierigkeiten, wenn die Firma ihren Mitarbeitern den Zugriff auf einen News–Server ausserhalb des lokalen Netzes ermöglichen möchte. Regeln, die eine TCP–Verbindung der Clients mit dem NNTP–Port eines Rechners ausserhalb des Firmennetzes erlauben, sonst aber alle anderen Formen der Kommunikation zwischen diesen Rechnern verbieten, sind mit obigem Firewall nicht zu machen. Es bleibt die Möglichkeit, entweder einen eigenen News–Server auf einem der (als sicher angenommenen) Unix–Rechner zu betreiben, oder ein News–Gateway analog zum WWW–Proxy zu betreiben. Beides sind ansonsten überflüssige Massnahmen, solange das Datenaufkommen gering ist.

12.2.3 Filterung nach Diensten

Wegen dieser Einschränkungen sind Paketfilter, die nur nach IP–Nummern filtern, nicht mehr sehr verbreitet. Die gängigen Paketfilter–Produkte verwenden die IP–Nummern und die Protokollnummer aus dem IP–Header, sowie bei TCP–Segmenten und UDP–Paketen die Portnummern aus dem Header TCP-/UDP–Header.

Damit wird eine Filterung möglich, mit der sich genau angeben lässt, welche Rechner welche anderen Dienste auf welchen anderen Rechnern ansprechen dürfen, sofern die Verbindung durch den Firewall geht.

Das Problem mit dem Zugriff auf den NNTP–Dienst aus obigem Beispiel ist mit einem solchen Paketfilter einfach zu lösen. In den Regelsatz aus Abbildung 12.2 werden zwei weitere Regeln eingefügt (mittlere zwei Zeilen in Abbildung 12.3).

Ursprung	Ziel	Erlaubnis
193.75.101.1–4	*	+
	193.75.101.1–4	+
193.75.101.5–255	* Port 119	+
Port 119	193.75.101.5–255	+
193.75.101.5–255	*	–
	193.75.101.5–255	–

Abbildung 12.3: Beispiel für Filterung nach IP–Nummern, Protokollen und Diensten

Dabei wird ein Paketfilter vorausgesetzt, der den Regelsatz von oben nach unten abarbeitet, bis er eine zutreffende Regel gefunden hat, und diese ausführt. Die Rechner mit den IP–Nummern 193.75.101.5–255 dürfen also Verbindungen auf den Port 119 eines Rechners im äusseren Netz öffnen (mittlere zwei Zeilen), jedoch nicht auf andere Portnummern (untere zwei Zeilen).

Mit Hilfe eines Filters, der auf Basis von IP–Nummern und Protokoll-/Portnummern filtert, lässt sich ein besserer Schutz für die Server–Rechner als im obigen Beispiel realisieren. Die Server sollten ja nur die Dienste E–Mail und WWW bereitstellen. Daher brauchen sie auch nur auf den dafür zuständigen Portnummern von aussen zu erreichen sein. Abbildung 12.4 zeigt die verfeinerte Konfiguration.

Ursprung	Ziel	Erlaubnis
193.75.101.3 Port 25 / tcp	* Port 25 / tcp 193.75.101.3	+ +
193.75.101.4 Ports 1024..65535 / tcp alle Ports / tcp	* alle Ports / tcp 193.75.101.4 Ports 1024..65535 / tcp	+ +
193.75.101.5–255 Port 119 / tcp	* Port 119 / tcp 193.75.101.5–255	+ +
193.75.101.5–255	* 193.75.101.5–255	– – – –

Abbildung 12.4: Verfeinerte Paketfilter–Konfiguration

Die ersten zwei Regeln erlauben es dem E–Mail–Server (193.75.101.3), auf den SMTP–Port 25 eines Rechners im äusseren Netz zuzugreifen. Eingehende Pakete werden nur erlaubt, wenn die Absender–Portnummer 25 ist; das deutet auf die Antwortpakete einer TCP–Verbindung auf den SMTP–Port hin.

Analog dazu könnte man den Zugang des WWW–Proxies auf den Port HTTP–Port 80 des äusseren Netzwerkes begrenzen. Dies erweist sich jedoch im Praxisbetrieb nicht als sinnvoll. Zu viele HTTP–Server benutzen eine andere Portnummer, z.B. 8000 oder 8080. Ausserdem binden HTML–Dokumente häufig auch Ressourcen ein, die über andere Protokolle zu erreichen sind, z.B. über FTP. Daher wird in der Paketfilter–Konfiguration in Abbildung 12.4 dem WWW–Proxy der TCP–Zugriff auf alle Ports im Netz gestattet. Auf seiten des Proxy–Rechners (193.75.101.3) werden jedoch nur die nichtprivilegierten Ports ($>=$ 1024) zugelassen, denn ein WWW–Proxy läuft üblicherweise nicht unter root–Rechten, kann also niemals TCP–Verbindungen von privilegierten Ports aus öffnen.

12.2.4 Filterung nach Verbindungsaufbaurichtung

Die bisher vorgestellten Filterungsmechanismen haben einen Nachteil: Mit ihnen ist es nicht möglich, die Richtung festzulegen, aus der eine TCP–Verbindung aufgebaut werden darf. Beispielsweise ist es die Intention der ersten beiden Regeln aus 12.4, den Aufbau einer TCP–Verbindung von einem beliebigem Port des Mail–Servers (193.75.101.3) auf den Port 25 eines beliebigen Rechners im Internet zuzulassen. Auf diesem Port arbeitet normalerweise ein SMTP–Server. Die oberste Zeile erlaubt den Datentransport vom Client zum SMTP–Server, die zweite Zeile die umgekehrte Richtung vom Server zum Client. Eine TCP–Verbindung kann nur zustande kommen, wenn beide Datenübertragungsrichtungen möglich sind, auch wenn der “logische“ Datenfluss (aus Sicht der höheren Protokollebenen) ausschliesslich in Richtung vom Client zum Server geht.

Das Problem besteht nun darin, dass die Regeln implizit auch eine TCP–Verbindung von Port 25 jedes beliebigen Rechners im Netz auf einen beliebigen Port des Mail–Server–Rechners zulassen. Es ist zwar sehr unwahrscheinlich, dass für eine “normale“ TCP–Verbindung der Ausgangsport 25 gewählt wird. Jedoch kann ein Angreifer mit Superuser–

Rechten auf seinem eigenen Rechner den Ausgangsport selber frei wählen. Sobald auf dem Mail–Server–Rechner weitere TCP–Dienste laufen, sind diese von aussen angreifbar.

Eine Abhilfe gegen dieses Problem ist möglich, indem der Paketfilter bei der Filterung von TCP–Paketen die Verbindungsaufbaurichtung mitberücksichtigt. Dazu wird neben den bereits genannten Parametern IP-, Protokoll- und Portnummern noch eine weitere Statusinformation aus den Kontrolldaten von TCP und IP für die Filterung berücksichtigt: die TCP–Flags. Ein TCP–Segment, das den Aufbau einer Verbindung initiiert, hat stets das SYN–Flag (Synchronise) gesetzt und das ACK–Flag (Acknowledge) gelöscht. Alle weiteren Segmente haben das ACK–Flag gesetzt. Ein Paketfilter, der nach Verbindungsaufbaurichtung filtert, darf also in der Richtung vom Client zum Server alle Pakete durchlassen, während er in der umgekehrten Richtung nur solche Pakete passieren lassen darf, bei denen das ACK–Flag in den TCP–Flags gesetzt ist. Abbildung 12.5 zeigt die verbesserte Konfiguration aus dem begleitenden Beispiel.

Ursprung	Ziel	TCP–Flags	Erl.
193.75.101.3 Port 25 / tcp	* Port 25 / tcp 193.75.101.3	alle ACK	+ +
193.75.101.4 Ports 1024..65535 / tcp alle Ports / tcp	* alle Ports / tcp 193.75.101.4 Ports 1024..65535 / tcp	alle ACK	+ +
193.75.101.5–255 Port 119 / tcp	* Port 119 / tcp 193.75.101.5–255	alle ACK	+ +
193.75.101.1–255	* 193.75.101.1–255	alle alle	−− −−

Abbildung 12.5: Filterung nach Verbindungsaufbaurichtung

Die Regeln sind so zu verstehen, dass die dritte Spalte (TCP–Flags) zum Bedingungsteil der Regel gehört. Das heisst, dass die zweite Regel nur ausgeführt wird, wenn das ACK–Flag gesetzt ist (und damit das Paket keinen neuen Verbindungsaufbau einleitet). Damit wird die oben beschriebene Angriffsmöglichkeit mittels Verwendung der Absenderportnummer 25 wirkungslos.

12.2.5 Zustandsorientierte Paketfilterung

In den bisherigen Beispielen benötigt jedes Erlauben einer TCP–Verbindung zwei Regeln, nämlich eine für jede Richtung. Die beiden Regeln sind auseinander ableitbar, also im Prinzip redundant. Bräuchte man nur eine der beiden Regeln anzugeben, würde das die Grösse der Konfigurationsdatei verringern und die Übersichtlichkeit erhöhen.

Die bisher beschriebenen, zustandslosen Paketfilter filtern zudem jedes einzelne Paket. Das bedeutet, für jedes einkommende Paket wird der Regelsatz von oben nach unten durchgelesen, bis eine passende Regel gefunden wird. Dann kommt die Anweisung aus dem Ausführungsteil zur Anwendung, das Paket wird also weitergeleitet oder gesperrt.

Bei einem zustandsorientierten Paketfilter werden nicht einzelne Pakete, sondern TCP–Verbindungen betrachtet. Der Firewall speichert den Zustand jeder TCP–Verbindung, die zwischen dem inneren und dem äusseren Netz gerade aktiv ist. Die eigentliche Filterung findet nur noch beim Aufbau einer TCP–Verbindung statt. Wenn ein TCP–Segment mit gesetztem SYN–Flag und gelöschtem ACK–Flag eintrifft, wird anhand des Regelsatzes überprüft, ob diese Verbindung zugelassen ist. Im Regelsatz steht nur eine Regel pro erlaubter Verbindungsmöglichkeit, die die erlaubte Verbindungsaufbaurichtung angibt. Treffen Segmente mit anderen Flags ein, so wird in der internen Liste nachgeschaut, ob die entsprechende Verbindung (in einer der beiden möglichen Richtungen) bereits existiert. Wenn ja, wird das Segment weitergeleitet, ansonsten wird es gelöscht.

Dieser Ansatz besitzt zwei wesentliche Vorteile gegenüber der zustandslosen Filterung. Erstens braucht in der Konfigurationsdatei nur noch eine Regel pro Verbindungsmöglichkeit angegeben zu werden. Das verkürzt die Datei und macht sie lesbarer. Zweitens ist die Geschwindigkeit normalerweise höher als bei einem zustandslosen Paketfilter. Denn das Heraussuchen einer bestehenden TCP–Verbindung lässt sich effizienter implementieren, als das Suchen einer passenden Regel in der Konfiguration.

In Anlehnung an die Konfigurationssprache des SINUS–Firewall (siehe Abschnitt 12.5) kann die Paketfilter–Konfiguration aus obigem Beispiel so aussehen wie in Abbildung 12.6.

```
accept tcp from 193.75.101.3 to port 25;
accept tcp from 193.75.101.4 port 1024..65535;
accept tcp from 193.75.101.5..255 to port 119;
block all;
```

Abbildung 12.6: Zustandsorientierte Paketfilterung

Jedes Regelpaar aus den früheren Beispielen schrumpft also zu einer Regel zusammen. Bei der hier gewählten Konfigurationssprache steht der Ausführungsteil der Regel am Anfang (accept / block), der Bedingungsteil steht dahinter und besteht aus Protokoll (tcp / udp / all), Quell- und Ziel–IP–Nummern und -portnummern. Nicht benötigte Angaben werden weggelassen; die Regel in der ersten Zeile gilt also für beliebige Quell–Portnummern und Ziel–IP–Nummern. Dies macht die Konfigurationsdateien kompakt und gut lesbar.

Das Konzept der zustandsorientierten Paketfilterung ist nur für TCP–Verbindungen sinnvoll anwendbar. Firewalls dieser Art arbeiten bei UDP–Paketen, ICMP–Paketen etc. genauso wie gewöhnliche, zustandslose Paketfilter.

12.2.6 Paketfilter mit dynamischen Regeln

Die bisher betrachteten Paketfilter–Mechanismen arbeiten mit statischen Regelsätzen. Die Regeln werden bei der Einrichtung des Firewalls vom Administrator festgelegt oder von einem Programm (z.B. einem graphischen Topologieeditor) erzeugt. Im Betrieb werden die Regelsätze nicht verändert.

Manchmal ist es nötig, bestimmten Rechnern ihre Verbindungsrechte zeitweilig zu entziehen, wenn bestimmte zusätzliche Bedingungen auftreten. Beispielsweise ist es sinnvoll,

Rechnern, die zu viele Daten übertragen oder zu viele Verbindungen innerhalb einer kurzen Zeitspanne aufbauen, für eine Weile auszusperren. Dazu dienen dynamische Regeln, die zu den bisherigen (statischen) hinzugefügt werden und nach einiger Zeit wieder gelöscht werden können, beispielsweise bei Eintreten anderer Bedingungen, durch Eingriff des Systemadministrators oder automatisch nach Ablauf einer festgesetzten Zeitspanne.

Ein Beispiel für den Einsatz dynamischer Regeln ist folgendes Einsatzszenario: Ein Rechner, der als SMTP–Server konfiguriert ist, soll vom ganzen Internet aus per TCP auf Port 25 erreichbar sein. Sollte es jedoch vorkommen, dass ein einzelner Rechner im Internet mehr als 10 SMTP–Verbindungen innerhalb einer Minute aufbaut, dann kann davon ausgegangen werden, dass versucht wird, kommerzielle Massenmailings abzusetzen, oder den Mail–Server durch eine Denial–of–Service–Attacke lahmzulegen. In diesem Fall soll dem verdächtigen Rechner der Zugriff für die nächsten 15 Minuten verweigert werden.

12.2.7 Weitere Funktionen

Filterung nach Datenmenge

Eine verbreitete Angriffsmöglichkeit auf E–Mail–Server besteht darin, sehr grosse Mails einzuliefern, um die Benutzer zu belästigen oder die Platte zu überfüllen. Dem kann entgegengewirkt werden, wenn ein Paketfilter zum Einsatz kommt, der pro TCP–Verbindung nur eine maximale Datenmenge zulässt. Ist die Menge überschritten, wird beiden Verbindungspartnern ein TCP Reset geschickt und somit die Verbindung abgebrochen.

Filterung nach Benutzern

Bei einem solchen Paketfilter können im Bedingungsteil einer Regel neben den TCP / IP Verbindungsdaten auch Angaben über die Benutzerkennungen gemacht werden, für die der Ausführungsteil der Regel gelten soll. Beispielsweise kann eine Regel, die den Benutzern Müller und Meier den Zugriff auf das World Wide Web (genauer: den HTTP–Port) erlauben soll, lauten:

```
accept tcp from 193.75.101.0 users mueller, meier to port 80;
```

Informationen über die Benutzernamen sind weder in den Headern der Schichten II und III, noch in den Nutzdaten der HTTP–Pakete enthalten. Die Informationen über Benutzer können über das ident–Protokoll gewonnen werden [Joh93, Wei97a]. Dazu muss auf dem Rechner, auf dem der HTTP–Client läuft, ein ident–Server (identd) installiert sein, der auf Anfrage den Login–Namen des Benutzers, der eine TCP–Verbindung geöffnet hat, mitteilt. Nach aussen (d.h. an den WWW–Server) sollen diese Informationen natürlich nicht weitergegeben werden.

Filterung der TCP-/UDP–Nutzdaten spezieller Dienste

Bei manchen Diensten reicht eine Filterung nach IP- und Portnummern nicht aus, um einen sinnvollen Betrieb zu ermöglichen. Ein bekanntes Beispiel ist FTP (siehe Abschnitt 7.4). Das Problem wird gelöst, indem der Paketfilter die Daten, die auf dem FTP–Kommandokanal übertragen werden, mitliest. Taucht ein PORT–Kommando auf, interpretiert er die übertragene Portnummer und erlaubt es dem Server, einmalig die angegebene FTP–Datenverbindung zu öffnen. Eine andere Anwendung der Nutzdatenfilterung ist es, bestimmte Inhalte zu sperren, etwa Java–Applets oder den Zugriff auf CGI–Skripten. Hier kann man beliebig komplizierte Heuristiken anwenden, um unerwünschte Nutzdaten innerhalb der Pakete auszusortieren. Jedoch sind durch den Aufwand an CPU–Zeit Grenzen gesetzt.

Ersetzung von IP–Nummern

Die meisten Paketfilter akzeptieren oder blockieren IP–Pakete, verändern sie aber nicht. Daher trägt jedes nach aussen weitergeleitete Datagramm die IP–Nummer des absendenden Rechners im inneren Netz. Dies wird häufig als Sicherheitsrisiko angesehen; es ermöglicht Rückschlüsse über die interne Struktur des Firmennetzes und erlaubt eine Verfolgung der Aktivitäten einzelner Rechner. Um dies zu vermeiden, bieten einige Produkte die Möglichkeit, IP–Nummern des inneren Netzes durch eine feste IP–Nummer (meist die des Paketfilters oder des Routers) zu ersetzen. Dabei muss im allgemeinen auch die dazugehörige Portnummer ersetzt werden, um doppelte Sockets zu vermeiden. Bei einkommenden Paketen an diese Sockets werden IP- und Portnummer wieder zurückgesetzt, bevor sie an den Rechner weitergeleitet werden, für den sie bestimmt sind. Somit ist nur eine IP–Nummer des gesamten Unternehmensnetzes im Internet bekannt.

Audit und Alarmierung

Treten bestimmte, unerwünschte Dinge ein, so ist es sinnvoll, dies dem Systemadministrator mitzuteilen. Dies sollte je nach Schwere des Vorfalles mit unterschiedlichen Massnahmen geschehen, z.B. durch einen Eintrag im Logfile, eine E–Mail an den Administrator oder sogar dem Auslösen akustischer oder optischer Signale.

Protokollierung

Ein Paketfilter sollte die Möglichkeit bieten, Informationen über angenommene und abgewiesene Verbindungsanforderungen sowie sonstige Betriebsdaten des Firewalls dauerhaft zu speichern. Dabei stellt sich die Frage nach der Fälschbarkeit, wenn diese Daten als Beweismittel herhalten sollen. Jüngere Forschungsergebnisse der Kryptographie bieten die Erstellung eines ewigen Logfiles, bei dem in regelmässigen Zeitabständen Hashsummen über

das File bei einer Vertrauensinstanz (z.B. einem Notar) hinterlegt werden [And96], ein anerkannter juristischer Beweis ist dies jedoch nicht. Eine andere, teurere Möglichkeit ist, die Logfiles auf einer einmal beschreibbaren CD zu speichern und diese zu hinterlegen.

Grenzen von Paketfiltern

Paketfilter arbeiten auf den Schichten II und III des TCP / IP–Schichtenmodelles. Sie werten die Kontrollinformationen aus den entsprechenden Protokollheadern (IP, TCP, UDP, ICMP etc.) aus und entscheiden auf Basis eines Regelsatzes, ob ein IP–Datagramm weitergeleitet oder gesperrt werden soll.

Obwohl der Inhalt eines TCP–Datenstromes in den Nutzdaten der TCP–Segmenten vorliegt und damit jedes einzelne Byte der Verbindung durch den Paketfilter geht, ist der Paketfilter prinzipiell nicht in der Lage, den Datenstrom eindeutig zusammenzusetzen. Dies liegt an den Fehlertoleranzeigenschaften von TCP / IP. Der Empfänger eines TCP–Segmentes bestätigt dessen Empfang. Die Position innerhalb des Datenstroms ergibt sich aus der Sequenznummer. Bleibt die Bestätigung aus, schickt der Sender das Segment nach einiger Zeit erneut weg. Nichts hindert einen bösartigen Sender daran, mehrere Pakete mit derselben Sequenznummer, aber unterschiedlichem Inhalt zu verschicken. Der Paketfilter sieht zwar die Bestätigung, kann aber nicht erkennen, für welches der Pakete sie gilt. Wird diese Technik bei mehreren aufeinanderfolgenden Segmenten eingesetzt, und haben die gleich numerierten Segmente sogar noch unterschiedliche Längen, dann entsteht beim Versuch, die in Frage kommenden Datenströme zu rekonstruieren, schnell eine hohe kombinatorische Komplexität.

Daher ist es nicht möglich, mit Paketfiltern deterministisch den Inhalt einer TCP–Verbindung auszuwerten. Einfache Mustervergleichsalgorithmen können zwar durchaus angewendet werden, beispielsweise zum Suchen bestimmter „verbotener“ Begriffe innerhalb eines Datenstromes mit ASCII–Text. Im Falle mehrerer, gleich numerierter Segmente kann hier eine vorsichtige Strategie zum Einsatz kommen, bei der der Verbindunsgsabbruch ausgelöst wird, wenn nur einer der möglichen rekonstruierbaren Datenströme das verbotene Schlüsselwort enthält. Für das syntaktische oder semantische Parsen einer komplexeren Sprache ist diese Technik jedoch ungeeignet.

Eine weitere Grenze von Paketfiltern ist in [Cha92] beschrieben. Durch Fehler beim Erstellen der Paketfilter–Konfiguration kann es schnell passieren, dass wesentlich mehr Verbindungsmöglichkeiten geöffnet werden, als eigentlich beabsichtigt sind. Beispielsweise erlaubt die Paketfilter–Regel

```
accept tcp from outside port 1024..65535 to 193.75.101.4 port 80;
```

den Zugriff von aussen auf den HTTP–Port des WWW–Server–Rechners 193.75.101.4. Wird dagegen versehentlich die Angabe “port 80“ am Ende weggelassen, dann erlaubt diese Regel plötzlich sämtliche TCP–Verbindungen auf den genannten Rechner. Fehler dieser Art bleiben oft lange Zeit unbemerkt, da sie die Funktionalität der Netzanbindung nicht behindern. Sie können aber gefährliche Sicherheitslücken öffnen.

12.3 Application Gateways

Die meisten kommerziellen Firewall–Produkte fallen in die Klasse der Application Gateways. Das Grundprinzip besteht darin, dass zwischen dem äusseren und dem inneren Netzwerk keine IP–Verbindung besteht. Statt dessen existieren Rechner, die mit beiden Netzen verbunden sind und für die gewünschten Dienste Gateways bereitstellen.

Ein Zugriff von innen auf einen äusseren Rechner kann nicht direkt gemacht werden. Statt dessen wird eine Anfrage an das Gateway gemacht, das seinerseits den Zugriff auf den äusseren Rechner durchführt und das Ergebnis an den inneren weiterleitet. Das Prinzip ist ähnlich wie beim IP–Routing über einen Paketfilter. Der Unterschied besteht aber in der Ebene, auf der die Weiterleitung passieren soll, und der dadurch verfügbaren Informationen für die Filterregeln.

Der grosse Vorteil von Application Gateways ist ihre Sicht auf die übertragenen Daten. Im Gegensatz zu Paketfiltern sehen Application Gateways den Datenstrom, der über einer TCP–Verbindung übertragen wird. Die Rekonstruktionsschwierigkeiten, die in Abschnitt 12.2 beschrieben wurden, haben Application Gateways nicht. Das kommt daher, dass sie gegenüber den beiden miteinander kommunizierenden Anwendungen jeweils als das andere Ende auftreten.

Im einfachsten Fall wird ein Application Gateway (auch Proxy genannt) zwischen zwei miteinander kommunizierende Anwendungen geschaltet. Anstatt einer TCP–Verbindung zwischen Client und Server werden zwei TCP–Verbindungen benutzt, eine zwischen Client und Proxy, die andere zwischen Proxy und Server. Dies klingt allerdings einfacher, als es ist:

Die Schwierigkeit besteht darin, dem Client mitzuteilen, dass er keine TCP–Verbindung direkt an den Server öffnen darf/kann. Manche Dienste besitzen die Möglichkeit, die Verbindungen umzuleiten, andere aber nicht. Das Problem wird bei vielen Firewall–Toolkits dadurch gelöst, dass die Bibliotheken, die für den Aufbau von Netzwerkverbindungen zuständig sind, so modifiziert werden, dass sie die Zugriffe umleiten. Das bedeutet allerdings, dass auf jedem Arbeitsplatzrechner im inneren Netzwerk, der Internet–Dienste über Proxies nutzen darf, neue Bibliotheken installiert werden. Andere Lösungen bestehen im Austausch des Clients oder sogar in der Ersetzung des Protokolls durch ein anderes, fortschrittlicheres Protokoll.

Das reine Aufsplitten der TCP–Verbindungen und Umleiten des Datenstroms über den Proxy bringt vom Aspekt der Sicherheit her noch nicht viel. Um sinnvolle Netznutzung von Angriffen zu unterscheiden, muss der Proxy den ein- und ausgehenden Datenstrom analysieren, gegebenenfalls die Weiterleitung der Daten verzögern oder unterbinden oder sogar die übertragenen Daten verändern. Diese Aktionen sind von Dienst zu Dienst unterschiedlich, was bedeutet, dass es für jeden Dienst eigene Application Gateways geben muss.

Beispielsweise muss ein Proxy für einkommende E–Mail sich nach aussen hin so verhalten wie der MTA der Organisation. Er muss also insbesondere alle Optionen des SMTP–Protokolles kennen, wie sie in RFC 821 beschrieben sind. Dennoch soll der Proxy nicht zu

komplex werden; die üblichen Mailrouting- und -verarbeitungsfunktionen braucht er nicht zu besitzen.

Die Installation einer auf Application Gateways basierenden Firewall–Lösung ist aufwendiger als der Einsatz von Paketfiltern. Jedoch wird der erreichbare Sicherheitsgrad in den meisten Fällen höher eingestuft. Die Gefahr, durch Fehlkonfiguration die Nutzung gefährlicher Internet–Dienste versehentlich zu ermöglichen, ist geringer als bei Paketfiltern. Durch die Trennung von innerem und äusserem Netz auf IP–Ebene wird die Topologie des inneren Netzes nach aussen gut versteckt. Andererseits sind Application Gateways komplexere Programme als Paketfilter. Daher sind sie potentiell eher das Opfer von Programmierfehlern, die die Sicherheit gefährden können.

12.3.1 Application Gateways für E–Mail

Ein klassisches Beispiel für einen Application Gateway ist der E–Mail–Gateway. Mail wird im Internet meist über das Simple Mail Transfer Protocol (SMTP) transportiert. Innerhalb der Organisation nehmen Mail Transport Agents (MTA) die Mails entgegen und stellen sie den Mail User Agents (MUA) zum Abholen zur Verfügung (siehe Abschnitt 6.4).

Die gängigen MTAs sind sehr flexibel konfigurierbar. Sie lassen sich problemlos so konfigurieren, dass sie Mail nicht direkt an den MTA des Empfängers, sondern an ein Gateway zustellen sollen.

Für einkommende Mail von aussen kann anstelle des eigentlichen MTA ein Application Gateway für E–Mail nach aussen bekannt gemacht werden. Dies geschieht über den MX–Record (siehe Abschnitt 6.4). Das Gateway nimmt einkommende Mail entgegen und leitet sie nach einigen Überprüfungen an den internen MTA weiter. Somit kommt der MTA nie direkt mit anderen Hosts im Internet in Berührung. Für ausgehende E–Mail kann auf ähnliche Weise ein Zwischenschritt eingefügt werden.

Sinnvolle Überprüfungen, die das Mail–Gateway vornehmen sollte, beinhalten:

- Überprüfung der Syntax von RFC 821/822, insbesondere Verwendung unerlaubter Steuerzeichen in Header–Feldern.
- Überprüfung, ob die E–Mail überhaupt für das innere Netz bestimmt ist. Viele MTAs nehmen beliebige Mails aus beliebigen Quellen entgegen und stellen sie an beliebige Ziele zu. Damit ermöglichen sie den Versendern unverlangter kommerzieller Massenmails, ihre Botschaften ohne Hinweis auf den Absender zuzustellen.
- Verhindern von Mailbomben, indem die Anzahl/Grösse der Mails pro absendendem Rechner und Zeiteinheit limitiert wird; zurückweisen weiterer Mails, wenn das Limit überschritten ist.
- Überprüfung des Inhaltes einer Mail auf gefährliche Datentypen, z.B. Word–Dokumente mit Makro–Viren, ausführbare Programme oder fehlerhaft komprimierte Archive.

12.3.2 WWW–Proxies

HTTP, das Standardprotokoll für die Übertragung von Hypertext–Dateien im WWW, besitzt von Hause aus die Möglichkeit, Proxy–Server zu verwenden. FTP, gopher und WAIS (zwei ältere Protokolle für verteilte Informationssysteme) sind im Umweg über eine HTTP–Verbindung ebenfalls Proxy–tauglich.

Neben der Filterung auf Protokollverletzungen und gefährliche / unerwünschte Inhalte erfüllen WWW–Proxies noch eine weitere, sinnvolle Aufgabe: Sie können mit einer Cache–Funktion ausgestattet sein, d.h. einmal geladene WWW–Dokumente zwischenspeichern und bei einem weiteren Abruf sofort zur Verfügung stellen. Dadurch wird unnötige Netzbelastung vermieden und die Geschwindigkeit beim WWW–Zugriff erhöht.

Um ein wirksames Mittel zur Erhöhung der Sicherheit zu sein, sollte ein HTTP–Proxy mindestens folgende Filterungen vornehmen:

Überprüfung der korrekten Syntax beim HTTP, insbesondere Verwendung illegaler Steuerzeichen in Header–Feldern und überlange Header–Zeilen.

Überprüfung des Dokumenteninhaltes auf Schadhaftigkeit, z.B. Word–Dokumente mit Viren, Active–X–Controls, Java–Applets und Javascript–Programme. Inhalte dieser Art sollten je nach angestrebtem Sicherheitsgrad entweder ganz verboten oder nur von bestimmten Quellen erlaubt werden können. Die Benutzer sollten gewarnt und die potentiell schadhaften Daten protokolliert werden.

Ausfilterung von Cookies, entweder komplett oder zumindest anhand einer Positivliste.

Ausfilterung sonstiger Detailinformationen über die Benutzer, z.B. Kennung des benutzten Browsers oder Betriebssystems.

12.3.3 Application Gateways für andere Dienste

Die meisten kommerziellen Firewall–Pakete enthalten zumindest Proxies für NNTP, DNS, Telnet und die Berkeley r–tools. Manche Pakete bieten darüber hinaus auch Unterstützung für IRC, NTP, lpd, NFS, NIS und andere Dienste. Die Bedienung ist teilweise sehr umständlich, da die genannten Protokolle nicht alle Proxy–tauglich sind. Manche Gateways sind nur mit speziell angepassten Clients verwendbar, was die Brauchbarkeit weiter einschränkt.

12.4 Bastion Hosts

Eine spezielle Variante der Proxies stellen die Bastion Hosts dar. Ein Bastion Host ist ein speziell gesicherter Rechner, auf dem man potentiell gefährliche Dienste ablaufen lassen kann, ohne dass von ihnen eine Gefahr ausgeht.

Ein verbreitetes Beispiel sind WWW–Applikationen mit Java–Applets. In vielen Installationen ist das Abrufen von ausführbaren Programmen von externen WWW–Servern aus

Sicherheitsgründen generell untersagt und wird vom WWW–Proxy verhindert. Für manche Anwendungen werden jedoch z.B. Java–Applets benötigt. Es bestände die Möglichkeit, diese Anwendungen an speziellen Internet–Arbeitsplätzen freizugeben. Diese Arbeitsplätze können an ein eigenes Netz angeschlossen werden, das nicht auf das hausinterne Netz zugreifen kann. Dies hätte jedoch zur Folge, dass man seinen Arbeitsplatz verlassen muss, um diese Dienste zu nutzen.

Eine Lösungsmöglichkeit ist ein Bastion Host. Dieser Rechner erfüllt eine ähnliche Aufgabe wie ein Internet–Arbeitsplatz. Er arbeitet in einem getrennten Netz, das nicht mit internen Netzen verbunden ist, in denen wichtige und sicherheitskritische Daten bearbeitet werden. Im Gegensatz zum Internet–Arbeitsplatz ist er jedoch vom Arbeitsplatz aus bedienbar. Dies kann über das X11–Protokoll realisiert werden, wofür allerdings ein spezielles X–Gateway nötig ist, da das X–Protokoll selbst ebenfalls mit Sicherheitsproblemen behaftet ist. Ein Beispiel für den Einsatz eines Bastion Host ist in Abbildung 12.7 dargestellt.

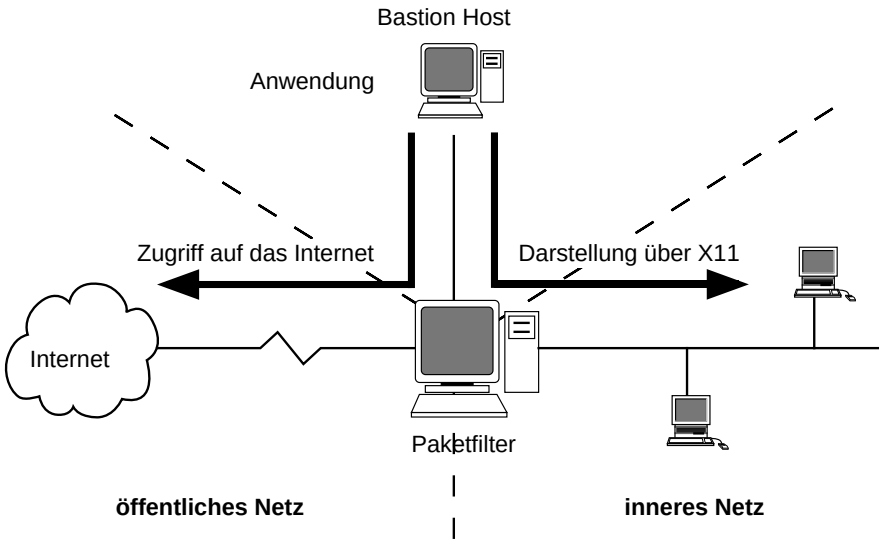

Abbildung 12.7: Einsatz eines Bastion Hosts

Auch für Konsolen–orientierte Dienste oder andere X11–Programme wie Telnet, IRC, MUD–Clients etc. können Bastion Hosts eingesetzt werden. Für Konsolen–Programme sind sie besonders einfach zu realisieren; man benötigt lediglich einen Unix–Rechner, auf dem man von den Arbeitsplatzrechnern der Mitarbeiter aus ein Login zulässt.

Telnet (und auch die r–Tools) sind nicht unmittelbar Proxy–tauglich. Um einem Benutzer des internen Netzwerkes die Möglichkeit zu geben, sich auf einem äusseren Rechner anzumelden, gibt es zwei Wege:

Auf dem Firewall–Rechner läuft ein spezieller telnetd, der Proxy–Funktionen übernimmt. Der Benutzer muss sich zunächst auf diesem Rechner anmelden; von dort kann er ein Login auf einen äusseren Rechner starten. Der Rechner, über den die Verbindung gehen muss, wird auch Bastion Host genannt. Diese Vorgehensweise hat den Nachteil, dass sich aus Benutzersicht die Bedienung ändert: Verbindungen über Bastion Hosts sind nicht transparent. Für einen menschlichen Benutzer ist das nicht weiter schlimm; ein automatischer

Loginvorgang aus einem anderen Programm heraus kann dagegen möglicherweise nicht mehr funktionieren. Zum Beispiel kann eine WWW–Seite eine Telnet–URL enthalten; wenn der Benutzer sie anwählt, wird er jedoch keine Verbindung bekommen, da direkte Telnet–Verbindungen über den Bastion Host nicht mehr funktionieren.

Um die Bedienung zu vereinfachen, kann die Client–Software im inneren Netz so verändert werden, dass sie Zugriffe an einen äusseren Rechner automatisch an den Bastion Host umleitet. Meist geschieht das durch Veränderung bzw. Einfügung der Programmbibliotheken, über die ein Anwendungsprogramm (in diesem Fall der Telnet–Client) mit der TCP/IP–Implementierung des Betriebssystems kommuniziert. Der Vorteil dieser Methode ist die Transparenz aus Sicht der Anwendung bzw. des Benutzers. Nachteilig ist aber, dass an allen Programmen, Bibliotheken und/oder Betriebssystemen der Rechner im inneren Netz Veränderungen vorgenommen werden müssen. Dies ist zeitaufwendig und eine zusätzliche Fehlerquelle. Bei Systemen, deren Quelltext nicht öffentlich verfügbar ist, ist es manchmal sogar überhaupt nicht möglich.

12.5 Der SINUS–Firewall

12.5.1 Einleitung

Auf dem Markt existiert eine Vielzahl unterschiedlicher Firewall–Lösungen. Jedoch entsprach keine davon den Bedürfnissen im SINUS–Projekt. Viele Produkte konzentrieren sich auf ganz spezielle Askepte der Filterung; um sie sinnvoll einzusetzen, müsste man sie baukastenartig mit anderen Produkten kombinieren. Die meisten kommerziellen Lösungen werden als “Black–Box“ vertrieben. Wenn der Quelltext nicht verfügbar und die Arbeitsweise nicht dokumentiert ist, dann kann das Verhalten in speziellen Situationen nur durch Experimente herausgefunden werden. Dies erschwert den Einsatz eines Produktes zum Studium neuer Techniken und schwächt das Vertrauen in die Sicherheit. Daher wurde im Rahmen des Projektes ein eigener Firewall vollständig neu entwickelt und implementiert.

Der SINUS–Firewall unterscheidet sich in mehreren Punkten von anderen Produkten. Im wesentlichen handelt es sich um einen Paketfilter, also einen Firewall, der die einzelnen TCP/IP–Pakete überprüft und anhand der Steuerinformationen von Transport- und Vermittlungsschicht entscheidet, ob ein Paket weitergeleitet werden darf oder nicht. In erster Linie sind dabei die IP–Nummern und Portnummern von Absender- und Ziel–Rechner relevant. Die meisten gängigen Dienste im Internet haben feste Portnummern, so dass sich mit diesen Informationen steuern lässt, von welchen Rechnern aus welche Dienste auf welchen Rechnern angesprochen werden können.

12.5.2 Eigenschaften des SINUS–Firewalls

Im Gegensatz zu anderen Paketfiltern erfolgt beim SINUS–Firewall die Filterung von TCP–Verbindungen zustandsorientiert. Eine TCP–Verbindung ist durch die vier Parameter Absender–IP–Nummer, Ziel–IP–Nummer, Absender–Portnummer und Ziel–Portnummer

netzweit eindeutig charakterisiert. Nur das erste Paket jeder Verbindung passiert die Filterregeln. Wenn es angenommen wurde, wird die Verbindung in einer Datenbasis abgelegt. Treffen weitere Pakete derselben Verbindung ein, kann sofort entschieden werden, ob die Verbindung bereits existiert und die Pakete durchgelassen werden dürfen. Am Ende einer Verbindung werden die Datenbasis–Einträge entfernt. Diese zustandsorientierte Filterung bietet eine höhere Flexibilität beim Erkennen von TCP–spezifischen Angriffen und damit eine bessere Sicherheit. Beispielsweise können Angriffe auf die Sequenznummerierung von TCP erkannt und verhindert werden. Ein zusätzlicher Vorteil ist die höhere Geschwindigkeit. Die Datenbasis der offenen TCP–Verbindungen ist als Hash–Tabelle realisiert und hat erfahrungsgemäss selten mehr als 25–30 Einträge. Einen Eintrag darin aufzufinden, geht erheblich schneller als das Durchforsten des Regelsatzes, der häufig mehr als hundert einzelne Regeln enthält.

Eine weitere herausragende Eigenschaft des SINUS–Firewalls sind seine dynamischen Filterregeln. Unter einer dynamischen Regel versteht man eine Filterregel, die nicht ständig im Regelsatz des Firewalls enthalten ist, sondern nur beim Eintreten bestimmter Ereignisse aktiviert wird und nach einer festgelegten Zeitspanne wieder verschwindet. Ein Beispiel für den Einsatz dynamischer Regeln könnte das folgende Szenario sein: Ein Unternehmen betreibt einen WWW–Server und gewährt allen Rechnern auf der Welt Zugriff darauf. Sobald jedoch von einem einzelnen Rechner mehr als 30 Zugriffe innerhalb einer Minute abgesetzt werden, wird ein Angriff vermutet, bei dem der Server oder das Netzwerk durch übermässig starke Belastung für sinnvolle Benutzung unbrauchbar gemacht wird. In diesem Fall soll der Zugriff von diesem einen Rechner auf den WWW–Server gesperrt werden, während alle anderen Rechner weiterhin zugreifen können sollen. Nach 10 Minuten soll die Blockade wieder aufgehoben werden. Technisch gesprochen sollen also die Anzahl der Verbindungen innerhalb einer Minute gezählt werden; sollte diese Zahl 30 übersteigen, wird eine dynamische Regel aktiviert, die einer bestimmten IP–Nummer den Zugriff verbietet, und die Priorität gegenüber den bestehenden Regeln hat. Nach 10 Minuten soll die dynamische Regel automatisch wieder verschwinden.

12.5.3 Die graphische Management–Schnittstelle

Aus Gründen der Lastverteilung und der Fehlertoleranz ist es häufig sinnvoll, das firmeneigene Netzwerk mit mehreren Zugangspunkten an das Internet anzuschliessen. Zweckmässigerweise werden an allen diesen Zugangspunkten Firewalls zum Einsatz kommen. Je nach Anwendungen kann es auch innerhalb der Firma notwendig sein, dass Netz in mehrere Teilnetze von unterschiedlichem Sicherheitsgrad einzuteilen; auch diese Teilnetze werden mittels Firewalls miteinander verbunden. Bei grösseren Installationen mit sicherheitsrelevanten Anwendungen und vielen Firewalls kann jedoch die Komplexität der Firewall–Konfiguration schnell bis zur Unbeherrschbarkeit ansteigen. Daher ist es wichtig, Werkzeuge zur Konfiguration und Verwaltung mehrerer Firewalls einzusetzen. Der SINUS–Firewall verfügt über eine graphische Management–Schnittstelle, mit der der Administrator die Konfiguration sämtlicher Firewalls zentral verwalten kann. Das Tool bein-

haltet einen Editor, mit dem der Aufbau des Netzwerkes und die Position der Server und Firewalls graphisch eingegeben werden können (Abbildung 12.8). Es errechnet daraus einen Satz von Konfigurationsregeln für alle Firewalls. Diese Regeln können anschliessend von Hand verändert werden (Abbildung 12.9). Am Ende werden an jeden Firewall die von ihm benötigten Regeln übertragen.

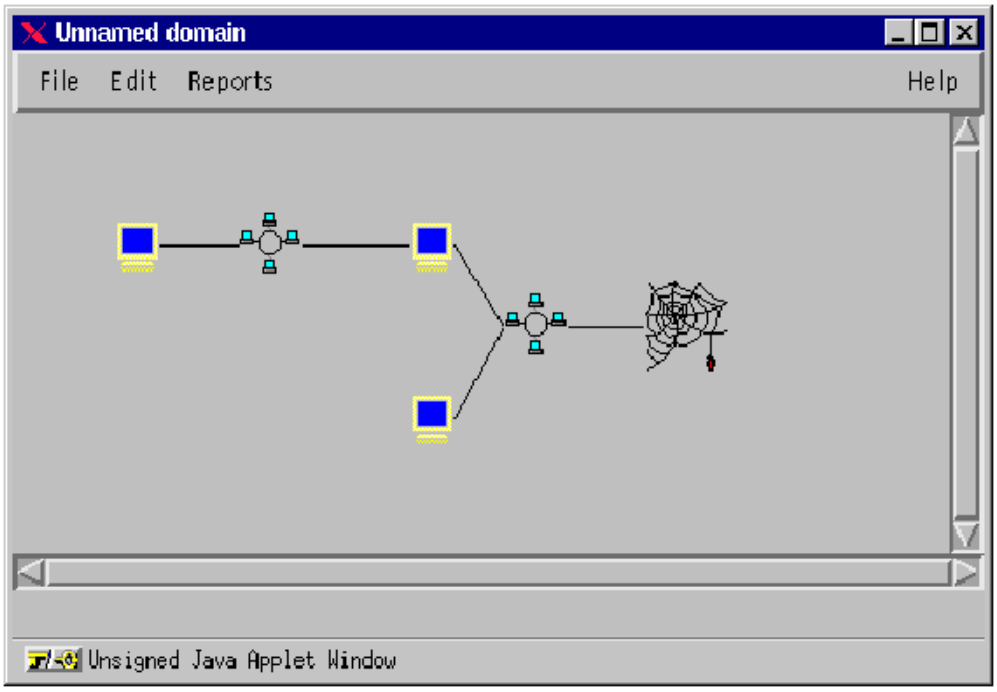

Abbildung 12.8: Eingabe des Netzwerkaufbaus

12.5.4 Architktur

Der Firewall läuft unter dem Betriebssystem Linux und besteht aus drei Komponenten (12.10). Die eigentliche Filterung findet in einem Kernel–Modul statt. Dies ist aus Effizienzgründen notwendig, da der Transport von IP–Paketen ebenfalls im Kernel erfolgt. Wenn für jedes einzelne Paket ein Datenaustausch zwischen Kernel- und Userspace erfolgen müsste, würde das die Filterung stark verlangsamen. Der Filter wird von einem Programm gesteuert, das im Userlevel läuft. Es liest die Konfigurationsdatei, überträgt die Konfiguration an den Filter, nimmt Meldungen des Filters entgegen und speichert diese in der Log–Datei. Die Kommunikation zwischen Filter und Steuerungsprogramm erfolgt über ein Unix Character Device. Die dritte Komponente ist die Management–Schnittstelle. Sie wurde in Java implementiert, ist also auf beliebigen Plattformen einsetzbar. Sie kommuniziert über TCP / IP mit dem Firewall.

Der Firewall–Prototyp wird beim SINUS–Projektpartner, der Telekurs Logistik AG, im produktiven Betrieb eingesetzt und laufend erweitert, um den Anforderungen gerecht zu werden.

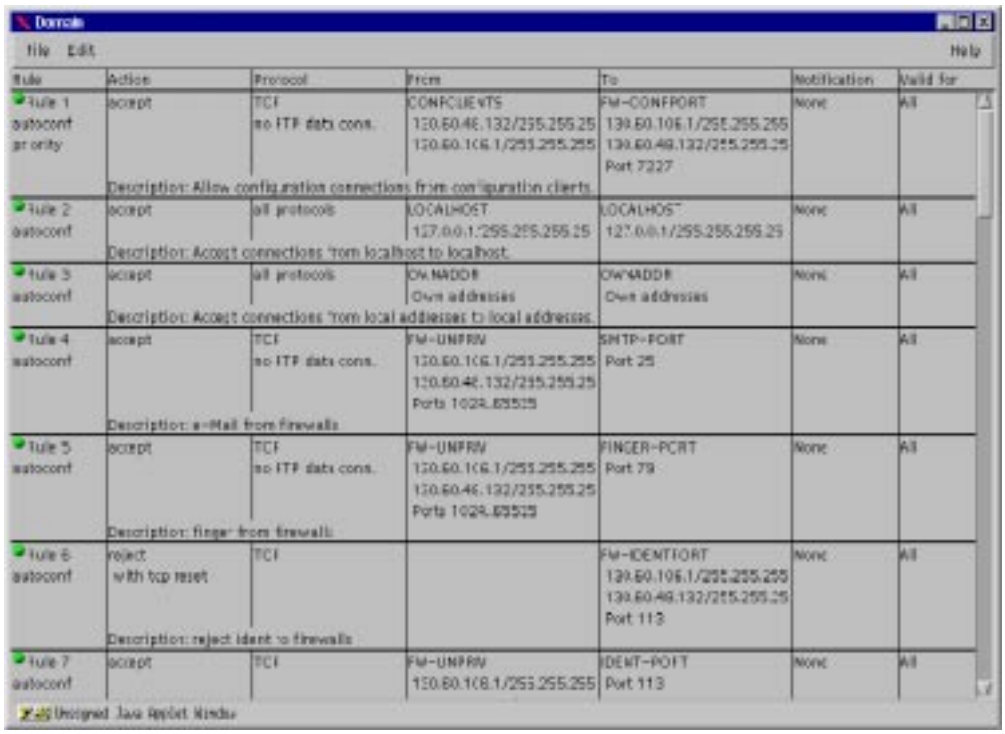

Abbildung 12.9: Der Regeleditor

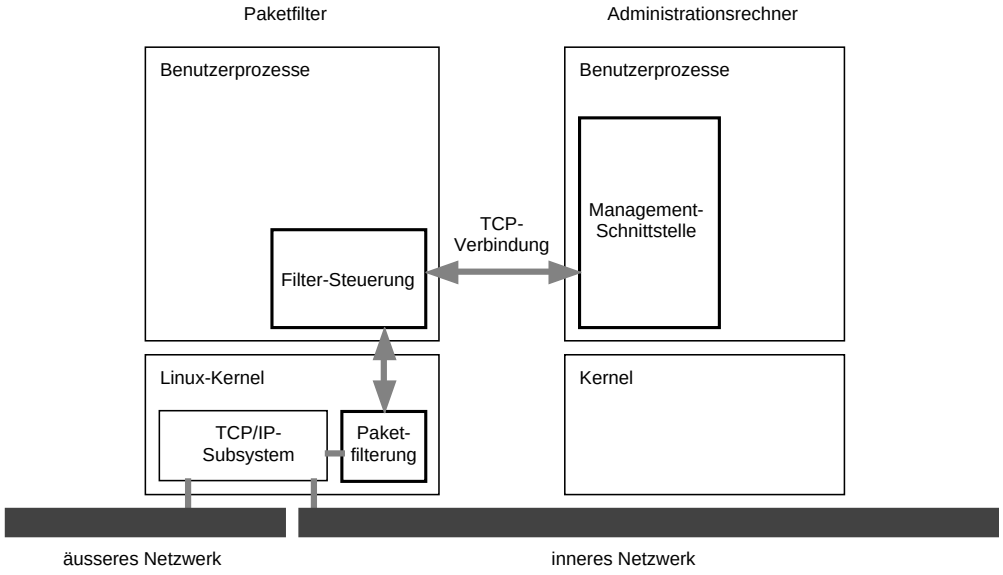

Abbildung 12.10: Architektur des SINUS–Firewall

12.6 Weitere Firewall–Produkte

Die auf dem Markt erhältlichen Firewall–Produkte lassen sich grob in frei erhältliche und kommerzielle Produkte unterteilen. Von manchen kommerziellen Produkten gibt es allerdings frei verfügbare Derivate mit eingeschränktem Funktionsumfang oder mit speziellen Lizenzbedingungen, etwa für ausschliesslich nichtkommerziellen Gebrauch.

In diesem Dokument sollen nur frei verfügbare Firewallsysteme vorgestellt werden. Der Grund dafür ist, dass die Autoren aus einem universitären Umfeld kommen, und daher nicht über die finanziellen Kapazitäten verfügen, die nötig wären, um die Entwicklungen im kommerziellen Bereich fundiert zu verfolgen. Mit den frei verfügbaren Produkten lässt sich das gesamte Spektrum an Anwendungen abdecken. Die im folgenden genannten Konzepte lassen sich auch auf kommerzielle Lösungen übertragen, auch wenn die Realisierung im konkreten Fall etwas anders aussieht.

12.6.1 Der Linux Kernel IP Filter

Die Kernel–Sourcen des frei verfügbaren Betriebssystems Linux enthält seit der Version 1.3 bereits von Hause aus einen IP–Paketfilter. Um den Filter in den Kernel einzubinden, muss der Kernel neu kompiliert werden, da die meisten Linux–Distributionen nur einen vorkompilierten Kernel ohne IP–Filter anbieten.

Der Paketfilter entspricht im wesentlichen dem in Abschnitt 12.2.3 vorgestellten Konzept der Filterung nach IP-, Protokoll- und Portnummern sowie TCP–Flags. Der Filter ist nicht verbindungsorientiert, es werden also getrennte Regeln für die beiden Datenübertragungsrichtungen einer TCP–Verbindung benötigt.

Der Filter unterscheidet zwischen drei Arten von Paketen: einkommende Pakete (in), ausgehende Pakete (out) und weitergeleitete Pakete (forward). Pakete, die von einem entfernten Rechner kommen und für die Linux–Maschine bestimmt sind, durchlaufen nur die in–Regeln. Pakete, die von der lokalen Maschine ausgehen, durchlaufen die out–Regeln. Pakete, bei denen weder Absender noch Empfänger lokal sind, durchlaufen in-, out- und forward–Regeln. Dieses Verhalten ermöglicht zwar differenzierte Filtermöglichkeiten und eine gute Kontrolle über die weitergeleiteten Pakete, erschwert aber die Konfiguration.

Mittels einer weiteren Kernel–Option, die beim Kompilieren aktiviert werden kann, und eines zusätzlichen Daemon–Programmes, kann ein Firewall auf Basis des Linux IP Filters die angenommenen Pakete loggen und beim Eintreten bestimmter Ereignisse geeignete Aktionen ausführen. Über diesen Weg sind auch dynamische Regeln realisierbar, allerdings sehr umständlich.

Zusammen mit dem Packet–Filter kann auch die IP–Masquerading Funktion von Linux benutzt werden. IP–Masquerading ist eine Variante der Network Address Translation (NAT). Es ermöglicht, dass beim Zugriff eines Rechners im inneren Netz auf das Internet die Absender–IP–Nummer auf die des Firewall–Rechners umgesetzt wird. Um die Eindeutigkeit der TCP/IP–Verbindung zu wahren, wird ggf. auf die Absender–Portnummer umge-

setzt. Nach aussen erscheinen also alle Zugriffe von der selben IP–Nummer, nämlich der des Firewall–Rechners. Der Masquerading–Code speichert die Zuordnung zwischen alter und neuer IP- und Portnummer und führt bei Antwortpaketen die umgekehrte Transformation durch.

IP–Masquerading hat mehrere Vorteile:

- Es wird nur eine einzige, offiziell geroutete IP–Nummer benötigt, und kein ganzes Netz.
- Die Zuordnung zwischen einer TCP / IP–Verbindung und dem Benutzer, der die Verbindung öffnet, wird erschwert. Dies kommt den Datenschutzanforderungen der Benutzer entgegen
- Da die Rechner des inneren Netzes keine weltweit geroutete IP–Nummer haben, können sie prinzipiell nicht direkt von aussen angesprochen werden (selbst wenn der Paketfilter überhaupt nicht benutzt wird).

Bei der Wahl der IP–Nummern für das innere Netz sollte man nach RFC 1918 vorgehen. Dort sind Nummernbereiche spezifiziert, die für den rein internen Gebrauch vorgesehen sind. Mit diesen Nummern besteht nicht die Gefahr, dass mit bestimmten Rechnern im Internet nicht kommuniziert werden kann, weil sie dieselben IP–Nummern benutzen wie das lokale Netz.

12.6.2 Der BSD Packet Filter

Ähnlich wie Linux ist auch das Betriebssystem NetBSD frei erhältlich. BSD heisst expandiert Berkeley Software Distribution und steht für Software, die an der University of California at Berkeley im Rahmen von Forschungsprojekten entwickelt wurde. Dort entstand in den 70er Jahren auch eine Weiterentwicklung des ursprünglich von AT&T entwickelten Betriebssystems UNIX unter dem Namen BSD UNIX. Um 1975 bekam BSD UNIX als erstes Betriebssystem eine Implementierung der TCP / IP–Kommunikationsprotokolle.

BSD UNIX war selbst nie wirklich frei erhältlich, da es auf Source–Code des Original–UNIX basierte, die dem Copyright von AT&T unterlagen. In den 80er Jahren ist die Weiterentwicklung des BSD UNIX weitgehend eingeschlafen. Erst Ende der 80er Jahre entstand mit NetBSD ein Projekt zur Weiterentwicklung und Entkommerzialisierung von BSD UNIX. Kurze Zeit später folgten zwei Abspaltungen des NetBSD–Projektes, FreeBSD und OpenBSD. FreeBSD ist eine eigenständige Entwicklung des BSD–Kernels für Intel i386–CPUs. Es ist speziell an die Bedürfnisse dieser CPU–Linie orientiert und daher etwas schneller als das plattformunabhängigere NetBSD. OpenBSD entspricht technisch stark dem NetBSD und unterscheidet sich von diesem durch das offenere Entwicklungskonzept, das viele Ähnlichkeiten mit Linux hat.

Für die freien BSD Unix–Derivate gibt es einen Paketfilter namens IP–Filter, der dem Linux–Paketfilter sehr ähnlich ist. Er war ursprünglich nicht Bestandteil des Kernels, sondern ein

separates Softwarepaket. Mittlerweile ist der IP–Filter Teil des NetBSD und des OpenBSD–Systems. Er arbeitet aber auch mit anderen BSD–ähnlichen Unix–Versionen zusammen, zum Beispiel FreeBSD, SunOS und BSD / OS.

Der IP–Filter erlaubt eine zustandslose Filterung von IP–Paketen nach den üblichen Filterkriterien IP-, Protokoll- und Portnummern, sowie TCP–Flags. Er ermöglicht zudem das Verfolgen der Zustände einer TCP–Verbindung. Diese Option muss in der Konfiguration einzeln angegeben werden. Man kann daher auch bestimmte Verbindungen zustandslos und andere zustandsorientiert filtern.

Der BSD IP–Filter beherrscht Network Address Translation, wobei im Gegensatz zu IP–Masquerading nicht zwingend das gesamte Netz auf eine einzelne IP–Nummer abgebildet wird, sondern auch auf ein anderes Netz abgebildet werden kann.

Weitere Eigenschaften des IP–Filters sind die Unterstützung von Transparent Proxies, das Zählen von Bytes, die eine Verbindung passiert haben, sowie das Erkennen und Behandeln defekter IP–Datagramme.

12.6.3 Das TIS Firewall Toolkit

Die amerikanische Firma Trusted Information Systems (TIS) bietet mit dem Firewall Toolkit (FWTK) eine Sammlung von Application Gateways und Tools zum Herunterladen an. Die Benutzung ist an einige Lizenzbedingungen geknüpft, aber für private, nichtkommerzielle Nutzung und Ausbildungszwecke ist das FWTK kostenlos. FWTK ist der kleine Bruder des Firewalls Gauntlet, der von TIS kommerziell vertrieben wird.

Das FWTK besteht im Wesentlichen aus drei Komponenten:

Application Gateways für gängige Internet–Dienste wie Telnet, FTP, SMTP und HTTP,

einem universellen Gateway, das für beliebige, auf TCP basierende Dienste verwendet werden kann,

Tools für netzorientierte und benutzerorientierte Zugriffskontrolle.

Beim Einsatz des FWTK wird üblicherweise ein Netzwerkaufbau verwendet, bei dem kein IP–Routing zwischen dem äusseren und dem inneren Netz möglich ist, zumindest nicht für die Dienste, die über die Application Gateways abgewickelt werden sollen. Dies wird entweder dadurch realisiert, dass der Firewall–Rechner mit zwei Netzwerkkarten ausgestattet ist und das Routing auf IP–Ebene explizit abgeschaltet wird, oder dass sich der Gateway–Rechner in einer sicheren Zwischenzone befindet, die auf beiden Seiten mit Paketfiltern geschützt ist.

Die Authentifikationstools werden über den Unix–Superserver inetd angesprochen. Sie entscheiden anhand einer zentralen Konfigurationsdatei, welche Gateways für welche Dienste, Benutzer und IP–Nummern aufgerufen werden sollen. Falls benutzerorientierte Authentifizierung zum Einsatz kommen soll, gibt es zudem eine Datenbank mit Benutzernamen, Passwörtern und Rechten. Diese Datenbank kann auf der Firewall–Maschine selbst oder von einer anderen Maschine im lokalen Netz administriert werden.

12.6.4 Spezielle Application Gateways

Für einzelne Internet–Dienste existiert eine Reihe spezialisierter Application Gateways. Die Verwendung einzelner, auf eine spezielle Aufgabe zugeschnittener Gateways erlaubt eine bessere Kontrolle über die zu überwachenden und zu filternden Eigenschaften des lokalen Netzes. Allerdings ist der Konfigurations- und Administrationsaufwand erheblich höher als bei Verwendung eines kompletten Paketes wie TIS.

SMTP–Gateways

smtpd ist ein simples SMTP–Gateway für einkommende Mails. Seine einzige Funktion besteht darin, einkommende Mails aus dem Internet per SMTP entgegenzunehmen und sie in regelmässigen Zeitabständen (z.B. alle 5 Minuten) an den Mail–Server der lokalen Organisation weiterzuleiten. Der Vorteil davon ist, dass Hosts aus dem Internet nie direkt mit dem lokalen MTA in SMTP–Verbindung treten. Das vereitelt einige Angriffe, die auf Buffer Overflows, Protokollverletzungen oder Überlastsituationen beruhen. Nachteilig ist allerdings die Verzögerungszeit, bevor die Mails an die Endbenutzer ausgeliefert werden. Auf Basis von smtpd lassen sich weitere Sicherheitsfunktionen implementieren, z.B. das Überprüfen eingehender E–Mails auf Attachments mit Viren.

qmail ist eigentlich kein Application Gateway, sondern ein vollwertiger MTA. Bei seinem Entwurf wurde allerdings spezieller Wert auf Sicherheitsaspekte gelegt. So besteht qmail, anders als z.B. sendmail, PP oder exim, nicht aus einer grossen Programmdatei, die alle Aufgaben erledigt. Das System ist vielmehr aufgeteilt in kleine, spezialisierte Programmteile. Jeder dieser Programmteile läuft genau mit den Privilegien, die er benötigt. Nur für einen geringen Teil der MTA–Funktionen wird das root–Recht benötigt, so dass die meisten dieser Programmteile ohne root–Rechte ablaufen können. Beim Versenden von Mails hat qmail Schwächen: Es erzeugt häufig unnötig hohe Netz- und Rechnerlast, weil es mehrere Mails stets parallel bearbeitet. qmail ist auch nicht in der Lage, Mails mit mehreren Adressaten innerhalb der gleichen Domain zu bündeln und gemeinsam zu verschicken. Für diese Aufgabe sind andere MTAs wie beispielsweise Sendmail oder PP besser geeignet. Daher verwenden viele Organisationen qmail nur als Gateway für einkommende E–Mails, während die interne Mail–Verarbeitung sowie ausgehende Mails weiterhin mit einem MTA durchgeführt wird.

WWW–Gateways

Ein bekannter Proxy für das World Wide Web ist squid. Er wurde hauptsächlich als Cache entworfen, um beim mehrfachen Zugriff auf gleiche Dokumente im WWW Netzkapazität einzusparen. Daneben besitzt er jedoch zahlreiche Funktionen zur Erhöhung der Sicherheit. Zu den wichtigsten Eigenschaften gehört die Möglichkeit, die Felder im HTTP–Header filtern zu können. Dadurch werden nicht alle Informationen, die der Browser bereitstellt, tatsächlich an den Server übermittelt. Dies, und die Tatsache, dass alle Zugriffe

aus der gleichen Domain von der selben IP–Nummer kommen, erschwert einem WWW–Serverbetreiber die Zuordnung zwischen Zugriffen und tatsächlichen Benutzern.

Junkbuster ist ein WWW–Gateway, mit dem man gezielt URLs und unerwünschte Inhalte in HTML–Dokumenten sperren kann. Beispielsweise lassen sich mit Junkbuster ganze WWW–Server oder bestimmte Verzeichnisse auf Servern sperren. Es können lästige oder sicherheitskritische Dinge wie Werbebanner, graphische Counter, Applets oder Cookies ausgeblendet werden. Auf diese Weise können Netzbandbreite eingespart und Sicherheitsrisiken des WWW vermieden werden.

Konzeption von Firewall–Lösungen

Wie bereits in der Einleitung erwähnt, werden Firewalls eingesetzt, um einen gesicherten Übergang zwischen zwei Netzen unterschiedlicher Vertrauenswürdigkeit miteinander zu koppeln. Bei der Planung des Firewall–Einsatzes sollte daher grundsätzlich die Frage im Vordergrund stehen, welche schätzenswerten Güter in einem Netz zur Verfügung stehen sollen, die gegen Übergriffe aus einem anderen Netz gesichert werden müssen.

Zunächst einmal ist das gesamte interne Netz gegenüber dem restlichen Internet schützenswert. Wie in Abschnitt 12.1 beschrieben wurde, gibt es Möglichkeiten, Rechner oder Netzwerkkomponenten unter Ausnutzung von Protokollschwächen oder Implementierungsfehlern lahmzulegen. Darüber hinaus können Konfigurations- oder Bedienungsfehler weitere Lücken öffnen. Angriffe dieser Art können nicht pauschal verhindert werden. Das Risiko kann dadurch minimiert werden, dass am zentralen Anschlusspunkt des internen Netzes an das Internet nur diejenigen Dienste und Protokolle zugelassen werden, die für den geplanten Einsatzzweck benötigt werden.

Von den beiden Strategien

- „Erlaube alle Verbindungen ausser denen, die explizit verboten sind“
- „Verbiete alle Verbindungen ausser den explizit erlaubten“

ist also im Normalfall die zweite vorzuziehen.

Die beiden oben vorgestellten Firewall–Typen, Paketfilter und Application Gateway, haben unterschiedliche Vor- und Nachteile. Daher ist es oft sinnvoll, am Verbindungspunkt zweier Netze beide Typen einzusetzen.

Application Gateways sind für Dienste vorzuziehen, die

- für den Betrieb mit Proxies geeignet sind (E–Mail, WWW, FTP, DNS)
- wegen Sicherheitsschwächen in den Applikationen eine Filterung der Nutzdaten erforderlich machen (E–Mail, WWW)
- von verschiedenen Benutzern im inneren Netz mit unterschiedlichen Privilegien genutzt werden können sollen und daher eine zusätzliche Authentisierungsstufe benötigen (Telnet, FTP)

- nur einen Teil ihrer üblichen Funktionalität nach aussen erlauben dürfen (z.B. nur Lese- und keine Schreibzugriffe auf NNTP–Server)
- neben Sicherheitsaspekten auch andere Vorteile (z.B. Lastverminderung oder Accounting) durch den Einsatz von Proxies erfahren (WWW, NNTP)
- die Architektur des inneren Netzes nach aussen nicht sichtbar sein soll (z.B. durch Verwendung eigener, im Internet unbekannter IP–Nummern)

Paketfilter sind von Vorteil, wenn

- der Zugriff auf bestimmte Dienste über den Anschlusspunkt hinweg gar nicht erlaubt werden soll, weil diese Dienste unnötig sind oder schwere Sicherheitslücken enthalten (z.B. X11, NTP, TFTP, bootp, lpd)
- einzelne Organisationen auf einfache Weise von der Nutzung eines Dienstes ausgeschlossen werden sollen (z.B. Massenmail–Versender oder WWW–Suchmaschinen)
- Zugriffe von aussen nur auf bestimmte Rechner im inneren Netz erfolgen dürfen (SMTP–Server, WWW–Server, DNS–Server)
- ein Verbindungsversuch an einen bestimmten Dienst eine Warnung oder einen Alarm auslösen soll (NTP, TFTP)
- Tools, mit denen Netzwerke nach Schwachstellen durchsucht werden können, erkannt werden sollen
- spezielle Optionen der Internet- und Transportschicht abgelehnt werden sollen (z.B. Source–Routing)

Beispielsweise kann der Zugriff auf den Mail–Server der Organisation so gestaltet werden, dass zunächst ein Paketfilter SMTP–Verbindungen auf alle anderen Rechner ausser den SMTP–Proxy abweist und der Proxy die Mails entgegennimmt, prüft und an den eigentlichen Mail–Server weiterleitet.

Analog dazu müssen ähnliche Überlegungen für die Gestaltung der Netzverbindungen im inneren Netz getroffen werden. Der Einsatz eines Firewalls ist immer dort notwendig, wo Netze unterschiedlicher Sicherheitsanforderungen innerhalb des Firmennetzes gekoppelt werden sollen, beispielsweise zwischen Netzen mit getrennter Administrationsverantwortung, Rechnerclustern, die sich gegenseitig über ihre IP–Nummern authentifizieren (müssen), konkurrierenden Abteilungen innerhalb der Organisation, oder auch zwischen einem Netz mit hohen Zuverlässigkeitsanforderungen und einem mit experimenteller Netzwerksoftware.

Weitere Hinweise zu Konzeption und Einsatz von Firewalls finden sich [CB94, CZ96].

Teil IV

Ein (fiktives) Beispiel

Kapitel 13

Fallbeispiel: Fiktives Beispiel

13.1 Einführung

Nachdem in den vorangegangenen Kapiteln die RSD–Sicherheitskonzeption vorgestellt wurde, wird nun das genaue Vorgehen anhand eines fiktiven Fallbeispiels illustriert werden. Banken sind prinzipiell stark daran interessiert, das Internet als Plattform für Finanztransaktionen zu verwenden. Eine Firma aus dem Finanzbereich schien für die RSD–Sicherheitskonzeption ein geeignetes Szenario, da in diesem Bereich hochsensible Daten vorhanden sind. Sicherheit geniesst in diesem Anwendungsbereich sehr hohe Priorität, da von ihr das Vertrauen der Kunden und auch das der Behörden abhängt. Demzufolge ist es beim Anschluss eines solchen Unternehmens wichtig, von Anfang an grossen Wert auf den Sicherheitsaspekt zu legen. Aus diesem Grund wurde für das Fallbeispiel eine fiktive Broker–Firma, die Stockbro (ÒStock Market BrokerÓ) gewählt. Anhand der fiktiven Firma Stockbro soll im folgenden Schritt für Schritt das RSD–Verfahren durchgeführt werden.

13.2 Ausgangslage

Die Broker–Firma Stockbro kann mit ihren ca. 50 Mitarbeitern zu den KMUs gezählt werden. Ihr Geschäftssitz liegt in Zürich. Stockbro verfügt darüber hinaus über Zweigstellen in Liechtenstein. Die Kunden haben die Möglichkeit, selbst auf ihr Portfolio Einfluss zu nehmen oder dieses von den Händlern betreuen zu lassen. Die derzeitige Transaktionsabwicklung geschieht mittels Telefon, Fax oder Post. Sämtliche Transaktionen werden von Zürich aus getätigt, die Zweigstellen in Liechtenstein dienen vor allem als Frontend zu den Kunden. Monatlich werden Anlagevorschläge mittels E–Mail an die Kunden verschickt.

Es werden verschiedene Softwaresysteme eingesetzt, die aber untereinander weitgehend noch nicht korreliert sind. Innerhalb der Firma Stockbro gibt es keine eigentliche Informatikabteilung. Für Fragen, die in das Informatikressort fallen, steht ein Mitarbeiter zur Verfügung, der allerdings noch andere Aufgaben zu erfüllen hat.

Mit der Realisierung des Internet–Anschlusses soll die Wettbewerbsposition des Unternehmens verbessert werden. Ein attraktives und aktuelleres Angebot für die Kunden und besse-

re Informations- und Kommunikationsmöglichkeiten für die Mitarbeiter sollten zu diesem Zweck realisiert werden. Darüber hinaus möchte das Unternehmen sich besser als Anbieter von hochqualitativen Finanzdienstleistungen profilieren.

Sicherheit ist dabei eine der Hauptanforderungen Der Hauptaugenmerk liegt auf der Vertraulichkeit und der Authentizität der Kundendaten.

13.3 Schritt 1: Nutzungskonzeption

Eines der Ziele ist die Schaffung einer Möglichkeit für die Kunden, ihr Portfolio über das Internet zu verwalten. Die Informationen sollen nur von den berechtigten Kunden abgefragt werden können. Zum einen würden dadurch die monatlichen Statements wegfallen, mittels denen die Kunden derzeit über aktuelle Angebote informiert werden, zum anderen könnten die Kunden sich jederzeit einen Überblick über ihr Portfolio verschaffen und damit schnell auf Veränderungen an der Börse reagieren.

Des Weiteren soll für die Leistungen des Unternehmens im Internet geworben werden. Um das Internet für die Mitarbeiter als Quelle schneller und aktueller Information nutzbar zu machen, ist es geplant, eine gesicherte Zugriffsmöglichkeit über das Internet bereitzustellen.

Es sollen demzufolge die Nutzungsszenarien

- Informationsbereitstellung
- Informationsbeschaffung
- elektronischer Handel

realisiert werden.

Nachdem geklärt wurde, welchen Nutzen das Unternehmen aus einer Anbindung an das Internet ziehen möchte, können in einem nächsten Schritt die Dienste bestimmt werden, die für die Realisierung notwendig sind.

13.4 Schritt 2: Dienstauswahl

Abhängig von den im ersten Schritt ausgewählten Nutzungsszenarien werden in der Dienstauswahl die dafür notwendigen Dienste ausgewählt. In Abbildung 10.6 ist die Verknüpfung zwischen Nutzungsszenarien mit den dafür geeigneten Internet–Diensten dargestellt. In dem vorliegenden Beispiel wurde das World Wide Web als Medium sowohl für die Informationsbereitstellung und -beschaffung als auch den elektronischen Handel ausgewählt. Folgende Internet–Dienste müssen realisiert werden:

HTML

HTML ist das grundlegende Übertragungsprotokoll im WWW. HTML–Dokumente bieten die Möglichkeit, neben passiver Information in Form von Text oder Grafik, auch aktive Inhalte miteinzubeziehen.

Java

Java von Sun Microsystems ist die bekannteste Programmiersprache für aktive Inhalte. Java ist dank der JVM (Java Virtual Machine) plattformunabhängig.

CGI

CGI (Common Gateway Interface) dient dazu, Parameter vom Web–Browser an den HTTP–Server zu übermitteln, diese Parameter durch ein CGI–Programm auszuwerten und in Abhängigkeit davon ein neues Dokument zu erzeugen, welches an den Client zurückgegeben wird.

DNS

DNS ist ein verteilter Namensdienst, der textuelle Rechnernamen in IP–Adressen abbildet und umgekehrt. Bezogen auf die Namensgebung fungiert ein Name–Server quasi als Schnittstelle zwischen den Domänen des Unternehmens und der Aussenwelt. Da das Unternehmen Stockbro mit zwei Domänen im Internet vertreten ist (Schweiz und Liechtenstein), müssen mindestens zwei Name–Server bereitgestellt werden, die im Hinblick auf Ausfallsicherheit nicht auf demselben Rechner installiert werden sollten.

Eine genauere Beschreibung der einzelnen Dienste befindet sich in Kapitel 6.

Nachdem für das Unternehmen Stockbro die Nutzungskonzeption und die Dienstauswahl durchgeführt wurden, müssen nun die mit den Internet–Diensten verbundenen Risiken analysiert werden.

13.5 Schritt 3: Risikoanalyse

Die Risikoanalyse erfolgt in 3 Schritten: Wertanalyse, Schwachstellenanalyse und Risikobewertung.

13.5.1 Wertanalyse

Ziel der Wertanalyse ist es, den materiellen Wert der bedrohten Güter festzulegen. Abbildung 13.1 zeigt für die bei Stockbro bereits eingesetzten Systeme als auch für die geplante WWW–Lösung eine Schadensabschätzung in SFr. Aus Gründen der Einfachheit wurde auf einen genauen Betrag verzichtet, sondern Klassen von 0, 1000, 100000 SFr. eingeführt.

Ausfalldauer / Anwendung	5 Minuten	3 Stunden	1 Tag
Büroanwendungen	0	1000	10000
Wirtschaftsdatenbank	1000	10000	100000
Finanzverwaltungssoftware	0	1000	10000
Adressverwaltung	0	1000	10000
E-Mail	0	1000	1000
Fax-Server	0	0	0
Buchhaltungsprogramm	0	1000	1000
Datei- und Druckerdienste	1000	10000	100000
Arbeitsplatz-PC	0	1000	1000
öffentlicher WWW-Server	0	0	1000
privater WWW-Server	0	1000	1000
Datenbank-Request-Server und Client	0	1000	1000

Abbildung 13.1: Möglicher Schaden bei bestehenden und bei neuen Anwendungen

13.5.2 Schwachstellenanalyse

Ziel der Schwachstellenanalyse ist eine Analyse der Schwachstellen von den zu realisierenden Internet–Dienste, in diesem Fall des WWW's. Abbildung 10.10 zeigt die Wechselwirkung zwischen Internet–Diensten und Gefahren. Da es sich bei Stockbro um ein Finanzunternehmen handelt, sind bei allen Daten die Anforderungen an die Authentizität, Geheimhaltung und Integrität sehr hoch. Es muss bei dem Anschluss an das Internet beachtet werden, dass grundsätzlich nicht nur die Internet–Anwendungen, sondern alle Anwendungen einer Bedrohung ausgesetzt werden.

Bei einem Webserver bestehen grundsätzlich die gleichen Risiken wie bei anderen Internet–Servern. Durch einen schlecht konfigurierten Server können Angreifer Zugang zu unerlaubten Bereichen erlangen. Es ist denkbar, dass die Kunden durch Fälschen von DNS–Informationen auf einen anderen Server umgeleitet werden können. Eine weitere Möglichkeit für einen Angriff ist das Verändern der auf dem Server hinterlegten HTML–Seiten. Auf diese Weise könnten Kunden mit falschen Informationen versorgt werden. Dateien, die via WWW übertragen werden, können "bösartig" sein. Zudem wird der Datenfluss nicht verschlüsselt, d.h. sensible Daten könnten möglicherweise abgehört werden. Java ist dank des 4–schichtigen Schutzkonzeptes relativ sicher. Das Prinzip von Java–Applets lautet, dass grundsätzlich kein Applet einen vertrauenswürdigen Code enthält. Aus diesem Grund laufen Applets nur in einem kontrollierten und abgeschlossenen Bereich (sandbox). Trotz dieses Sicherheitskonzepts gibt es eine Reihe von Sicherheitslücken. Dabei besteht die grösste Gefahr in Implementierungsfehlern der einzelnen Java–Plattformen. Durch derartige Fehler werden Angreifern diverse Angriffspunkte dargeboten. Inzwischen gibt es eine Java–Version, die es erlaubt, ein Java–Applet mit einer digitalen Signatur des Herstellers zu versehen. Mit einer gültigen Signatur und der Zustimmung des Benutzers kann ein signiertes Applet auch ausserhalb der sandbox ausgeführt werden. CGI–Skripte werden oft in Skriptsprachen wie Shell–Skript, Perl usw. erstellt. Sie werden schnell und einfach programmiert und enthalten daher oft Fehler, welche zu unerwünschten Nebeneffekten führen.

Durch geschicktes Ausnützen dieser Fehler kann unter Umständen Zugang zum System erhalten werden.

Von zentraler Bedeutung für die Firma Stockbro sind die Wirtschaftsdatenbank, die Finanzverwaltungsanwendung und das Buchhaltungsprogramm. Für diese Anwendungen besteht durch den Internet–Anschluss die Gefahr eines Denial–of–Service–Angriffs, indem der Rechner, auf dem die jeweilige Anwendung läuft, zum Absturz gebracht wird. Im Falle der Wirtschaftsdatenbank könnte beispielsweise das Einspielen von Falschinformationen zu grossem Schaden führen. Die Datei- und Druckerdienste sind ebenfalls von essentieller Bedeutung, da sie die Basis für viele andere Anwendungen bilden. Folglich ist der zu erwartende Schaden, vor allem bei längeren Ausfällen, relativ gross.

Einen weiteren Angriffspunkt stellen die Arbeitsstationen, PCs unter Windows 95, dar. Sollten direkt auf den Mitarbeiterrechnern WWW–Browser installiert werden, könnten mittels der dort ausgeführten Programme verschiedenste Angriffe gegen die Geräte selbst und auch gegen die auf dem Netzwerk installierten Systeme ausgeführt werden. Insbesondere die neu zu realisierenden Anwendungen verwenden alle TCP/IP und sind somit direkt vom Internet aus verwundbar.

Die sicherheitsrelevante Frage bei der Installation von DNS–Servern ist die, wieviel Information man nach aussen hin sichtbar machen möchte. Für die Handhabung des Internet–Verkehrs werden intern mehr Einträge benötigt, als eigentlich gegen aussen gezeigt werden müssen. Diese sichtbar zu halten, gibt einem potentiellen Angreifer zusätzliche Informationen.

Der Datenbank–Request–Client ist vor allem dahingehend verwundbar, dass er Anfragen an falsche Server sendet und dem Kunden somit falsche Informationen zur Verfügung gestellt werden könnten. Der Datenbank–Request–Server verwaltet alle Passwörter der Kunden und ihre Zugangsberechtigungen, die in einer Datei gespeichert werden. Der Server ist somit gegen alle Angreifer verwundbar, die über eine richtige Kombination von Passwort und Benutzername verfügen.

13.5.3 Risikobewertung

Da nun die Art der Gefahren bekannt sind, mit der die Firma Stockbro bei der Realisierung eines WWW–Zuganges konfrontiert wird, stellt sich die Frage nach der Eintretenshäufigkeit und nach dem Schaden, der dadurch verursacht wird.

Abbildung 13.2 zeigt den bei einem Angriff zu erwartenden Schaden kombiniert mit der Eintretenshäufigkeit für die Anwendungen, die bei der Firma Stockbro eingesetzt werden bzw. eingesetzt werden sollen. Schaden wird dabei als entgangener Gewinn betrachtet.

Da die Transaktionen, zumindest bisher, zunächst auf dem Papier festgehalten und erst später übertragen werden, würde ein kurzzeitiger Ausfall der Finanzverwaltung relativ wenig Schaden verursachen. Um einiges schwerer wäre dagegen ein Ausfall von einem Tag oder mehr zu verkraften. Mit dem Einsatz des WWWs ist ein Anstieg der durch den Ausfall

Anwendung \ Ausfalldauer	5 Minuten		3 Stunden		1 Tag	
	Schaden	Wahrschein-lichkeit	Schaden	Wahrschein-lichkeit	Schaden	Wahrschein-lichkeit
Büroanwendungen	0	mittel	1000	gering	10000	gering
Wirtschaftsdatenbank	1000	gering	10000	sehr gering	100000	sehr gering
Finanzverwaltungssoftware	0	mittel	1000	gering	10000	gering
Adressverwaltung	0	mittel	1000	gering	10000	gering
E-Mail	0	hoch	1000	mittel	1000	gering
Fax-Server	0	gering	0	gering	0	gering
Buchhaltungsprogramm	0	mittel	1000	gering	1000	gering
Datei- und Druckerdienste	1000	gering	10000	sehr gering	100000	sehr gering
Arbeitsplatz-PC	0	hoch	1000	mittel	1000	gering
öffentlicher WWW-Server	0	hoch	0	mittel	1000	mittel
privater WWW-Server	0	mittel	1000	mittel	1000	gering
Datenbank-Request-Server und Client	0	mittel	1000	mittel	1000	gering

Abbildung 13.2: Möglicher Schaden und Eintrittshäufigkeit

der Finanzverwaltungsanwendung zu erwartenden Schadenshöhe verbunden, da es notwendig sein wird, Änderungen in den Kunden–Portfolios möglichst ohne Zeitverzögerung auf den entsprechenden WWW–Seiten darzustellen.

Ein länger andauernder Ausfall der Wirtschaftsdatenbank hätte gravierende Folgen für das Unternehmen, da die Datenbank sozusagen Stockbro‘s “Pulsnehmer“ am Finanzmarkt ist. Dagegen ist ein kurzzeitiger Ausfall der Büroanwendungen vergleichsweise unwichtig, da in der Regel nur kleinere Dokumente bearbeitet werden.

Eine ähnliche Betrachtung kann auch für die Daten gemacht werden (Abbildung 13.3). Die gespeicherten Daten haben sehr unterschiedliche Sicherheits–Anforderungsprofile. Beispielsweise ist die Manipulation einer Web–Seite schlimmer als ihr Diebstahl. Manipuliert etwa ein Angreifer die Daten, indem er Inhalte so verändert, dass sie Stockbro unter Umständen in Konflikt mit dem Gesetz bringen, ensteht ein immenser Schaden.

Daten \ Angriff	Manipulation eines wichtigen Eintrags		Diebstahl aller gespeicherten Daten		Totalverlust der gepseicherten Daten	
	Schaden	Wahrschein-lichkeit	Schaden	Wahrschein-lichkeit	Schaden	Wahrschein-lichkeit
Portfoliodaten	1000000	gering	1000000	gering	10000000	gering
Kursdaten	10000	gering	0		10000	gering
Daten von Büroanwendungen	10000	gering	100000	gering	10000	mittel
E-Mails auf Mailservern	10000	hoch	10000	hoch	1000	mittel
Faxe auf Faxserver	100000	gering	100000	gering	100000	gering
WWW-Seiten	10000	mittel	1000	hoch	10000	gering

Abbildung 13.3: Risiken bei Angriffen auf die gespeicherten Daten

Das grösste Schadenspotential liegt bei den Portfolio–Daten, welche von der Finanzverwaltungsanwendung gespeichert werden. Die Manipulation eines Eintrags würde zu einem Integritäts- und Authentizitätsverlust und damit verbunden zu einem Publicity–Schaden führen. Genauso verheerend wäre der Diebstahl von Kundendaten, da er zum einen zum Verlust der Vertraulichkeit führen würde und zum anderen evtl. einen Konkurrenten in die Lage versetzen könnte, mit den gestohlenen Daten Kunden abzuwerben.

13.6 Schritt 4: Gegenmassnahmen

In vierten Schritt werden die konzeptionellen Massnahmen gegen die erruierten Risiken ermittelt. Abbildung 10.15 zeigt die Gefahren mit den entsprechenden Gegenmassnahmen.

Die Grundhaltung bei der Wahl der Gegenmassnahmen muss aufgrund der hohen Sicherheitsanforderungen, die der Anwendungsbereich Finanzmarkt mit sich bringt, das Konzept des default–deny–stance sein, d.h. alles, was nicht ausdrücklich erlaubt ist, ist verboten. Die konkrete Umsetzung dieser konzeptionellen Gegenmassnahmen auf technischer und organisatorischer Ebene geschieht in Schritt 5, der Realisierung.

13.7 Schritt 5: Realisierung

In dieser Phase werden die oben gewählten Gegenmassnahmen realisiert. Die Installation dieser Massnahmen erfolgt in verschiedenen Schritten. Als erstes müssen die Mitarbeiter über die geplanten Aktionen informiert werden. Zu diesem Zweck soll bei der Firma Stockbro eine kleine interne Hauszeitung ins Leben gerufen werden, die über die geplanten Installationen informiert. Sie kann auch dazu dienen, das Bewusstsein der Mitarbeiter gegenüber sicherheitsrelevanter Vorgänge zu schärfen. Bedeutend schwieriger ist die Information der Kunden, die den nächsten Schritt bildet, da das dort vorhandene Wissen und die Erreichbarkeit deutlich heterogener ist.

Nach der Information sowohl der Mitarbeiter als auch der Kunden kann mit der Installation des Systems begonnen werden. Sobald Teile des Gesamtsystems vorhanden sind, sollten sie potentiellen Kunden zum Testen vorgelegt werden. Haben alle Systeme ihren Funktionstest bestanden, können sie zu dem Gesamtsystem zusammengefügt werden. In einem firmeninternen Test müssen auch hier wieder Funktionstests durchgeführt werden. Danach müssen die Sicherheitsmechanismen getestet werden. Ist diese Testphase abgeschlossen, können erste interne Benutzer, d.h. das System ist immer noch nicht an das Internet angeschlossen, an das System gelassen werden. Sollte das System hier eine gewisse Stabilität zeigen, kann es bei einer kleinen Gruppe von Pilotbenutzern eingeführt werden.

In dieser Phase ist die Kommunikation zwischen Benutzer und Entwickler entscheidend; hier sind kleinere Änderungen noch möglich, die evtl. eine bedeutende Rolle für die Akkzeptanz des späteren Produkts haben. Durch systematische Kundenbefragungen können Kündenwünsche schon frühzeitig in die weitere Entwicklung eingearbeitet werden. Ist aus dem Pilotbetrieb ein erfolgreicher Systemeinsatz absehbar, können schrittweise Benutzer aufgenommen werden.

In dem Fall des Unternehmens Stockbro ergaben sich folgende Überlegungen:

Für die Netzwerkarchitektur wurde eine verteilte Lösung als am geeignetsten angesehen (siehe Abbildung 13.4). Hier werden die beiden Webserver–Anwendungen, öffentlich und privat, getrennt. Um eine effiziente Datenbankanbindung zu gewährleisten, sollte der private Webserver direkt bei Stockbro aufgestellt werden. Der öffentliche Webserver, der wesentlich weniger sicherheitskritisch ist, kann ohne weiteres bei einem Service–Provider untergebracht werden. Die Trennung dieser beiden Bereiche bietet zusätzliche Sicherheit, da der private Webserver so eingerichtet werden kann, dass überhaupt nur die Kunden, die ein Konto bei der Firma Stockbro unterhalten, eine Verbindung zu dem Server aufbauen

können. Zusätzlich wird durch die Trennung der Administrationsaufwand und damit die Wahrscheinlichkeit, dass hierbei Fehler unterlaufen, verringert.

Als wichtigste Entscheidung ist die Installation eines Firewalls zu nennen. Der Firewall trennt die unterschiedlich sicheren Netzwerke. Alle Regelverstösse müssen protokolliert werden. Um den hohen Anforderungen zu genügen, empfiehlt sich der Einsatz eines Firewalls, der aus mehreren Ebenen besteht.

Aufgrund der nicht unerheblichen Bedrohung durch die Arbeitsplatzrechner, sollte diesen Geräten der direkte Kontakt mit dem Internet untersagt sein. Zusätzlich muss die eigenmächtige Installation von Software von Seiten der Mitarbeiter strengstens untersagt sein. Des weiteren muss der WWW–Zugriff auf den eigentlichen Arbeitsplatzrechnern untersagt werden, da die Gefahr durch den ausführbaren Code zu gross ist. Attachments an E–Mails sollten auf ihre Gefahren hin untersucht werden. Die von der Firma Stockbro eingesetzten Anwendungen und die gespeicherten Daten müssen mit geeigneten Mitteln gegen Angriffe von aussen geschützt werden. Damit den Mitarbeitern troztdem das Internet als Informationsquelle offensteht, werdem sogenannte Internet–PC's aufgestellt, welche auf dem peripheren Netzwerk betrieben werden.

Es dürfen keine Kundendaten in unverschlüsselter oder nur schwach verschlüsselter Form über öffentliche Netzwerke versandt werden. Für den Zugriff auf Kundendaten ist eine Authentifizierung notwendig. Für diesen Zweck bietet sich die Verwendung eines One–Time–Passwortsystems an. Es ist vergleichsweise einfach zu implementieren, erfordert jedoch die Installation einer Client/Server–Software, welche im Falle des Servers auch konfiguriert werden muss. Der Nutzen, den ein Angreifer aus einem durch ÒZuschauenÓ erhaschten Passwort ziehen kann, ist gleich null, da dieses mit der einmaligen Verwendung durch den berechtigten Benutzer seine Gültigkeit verloren hat.

Für den Zugriff der Kunden auf ihre Portfolio–Daten empfiehlt sich der Einsatz von SSL (128–Bit–Verschlüsselung). Die führenden WWW–Browser sind amerikanische Produkte, die den amerikanischen Exportbeschränkungen unterliegen. Deshalb sollte eine Proxy–Software installiert werden, die zwischen Kunden und privatem Web–Server eine zusätzliche Verschlüsselung anbietet.

Da ein E–Mail–Server ein relativ komplexes und damit für Angriffe anfälliges Programm ist, sollte er nie direkt von einem Rechner aus dem Internet kontaktiert werden, wenn es eine E–Mail entgegenzunehmen gibt. Ebenso sollte der Server beim Versenden einer E–Mail keinen anderen Server kontaktieren. Vielmehr sollte ein auf dem peripheren Netzwerk installierter E–Mail–Proxy dazwischen geschaltet werden, d.h. sowohl der interne als auch die externen Mailserver kontaktieren den Mailproxy, welcher die E–Mails in die entsprechende Richtung leitet.

Es besteht die Möglichkeit, eine sogenannte hidden DNS–Installation vorzunehmen, bei der die Aufgaben der DNS–Server getrennt werden. Ein interner DNS–Server, der die Anfragen der internen Hosts beantwortet, befindet sich hinter dem Firewall. Dieser Server verfügt über sämtliche Informationen der Domänen von Stockbro. Vor dem Firewall befinden sich die externen DNS–Server, die nur die Information besitzen, die unbedingt für die Kommu-

nikation nach aussen notwendig ist.

Als externer Router wird ein PC unter Linux verwendet, da ein Computer wesentlich bessere Protokollierungsmöglichkeiten bietet als ein Hardware–Router. Dadurch kann sich Stockbro ein besseres Bild über die Nutzung ihrer Internet–Ressourcen durch Mitarbeiter und Aussenstehende machen. Als Firewall–Produkt unter Linux wird die SINUS–Firewall, beschrieben in Abschnitt 12.5, verwendet werden. Daneben soll eine WWW–Proxy–Software (z.B. squid) zum Einsatz kommen, um die Zugriffe auf das WWW zu kontrollieren, zu protokollieren und zusätzlich durch das Zwischenspeichern Bandbreite zu sparen. Bei der Konfiguration des SINUS–Firewalls muss dem Sicherheitprinzip der tiefsten Privilegien gefolgt werden, d.h. alles, was nicht erlaubt ist, ist verboten.

Der WWW–Proxy muss derart konfiguriert werden, dass er einzig und allein von den Internet–PC's der Firma Stockbro Verbindungsversuche annimmt.

Die letzte Barriere zwischen einem potentiellen Angreifer und dem firmeninternen Netzwerk stellt der interne Router dar. Er muss daher besonders schwierig zu überwinden sein. Aus diesem Grund empfiehlt sich der Einsatz eines Hardware–Routers, da er weniger Angriffsfläche bietet wie ein Computer.

Der WWW–Server hat die Aufgabe, den Kunden die benötigten Daten WWW–gerecht aufzubereiten. Bei der Installation muss besonders darauf geachtet werden, dass die CGI–Schnittstelle so gesichert wird, dass keine unerwünschten Programme ausgeführt werden können.

Um den steigenden Sicherheitsanforderungen und den sich ständig verändernden Umweltbedingungen Rechnung tragen zu können, müssen die Sicherheitselemente wie auch das Gesamtsystem hochgradig modular aufgebaut werden, so dass einzelne Komponenten ersetzt werden können, ohne das Gesamtsystem unbrauchbar zu machen.

13.8 Fazit

Die RSD–Sicherheitskonzeption erzwingt ein systematisches Vorgehen bei der Anbindung eines Unternehmens an das Internet. Diese Systematik verhindert, dass wesentliche Sicherheitsprobleme übersehen werden. Damit eignet sich die RSD–Methode besonders für hochsensible Anwendungsbereiche, wie sich am Beispiel der Firma Stockbro aus dem Finanzsektor gezeigt hat. Allerdings bietet das Verfahren keine vollautomatische sichere Internet–Nutzungskonzeption.

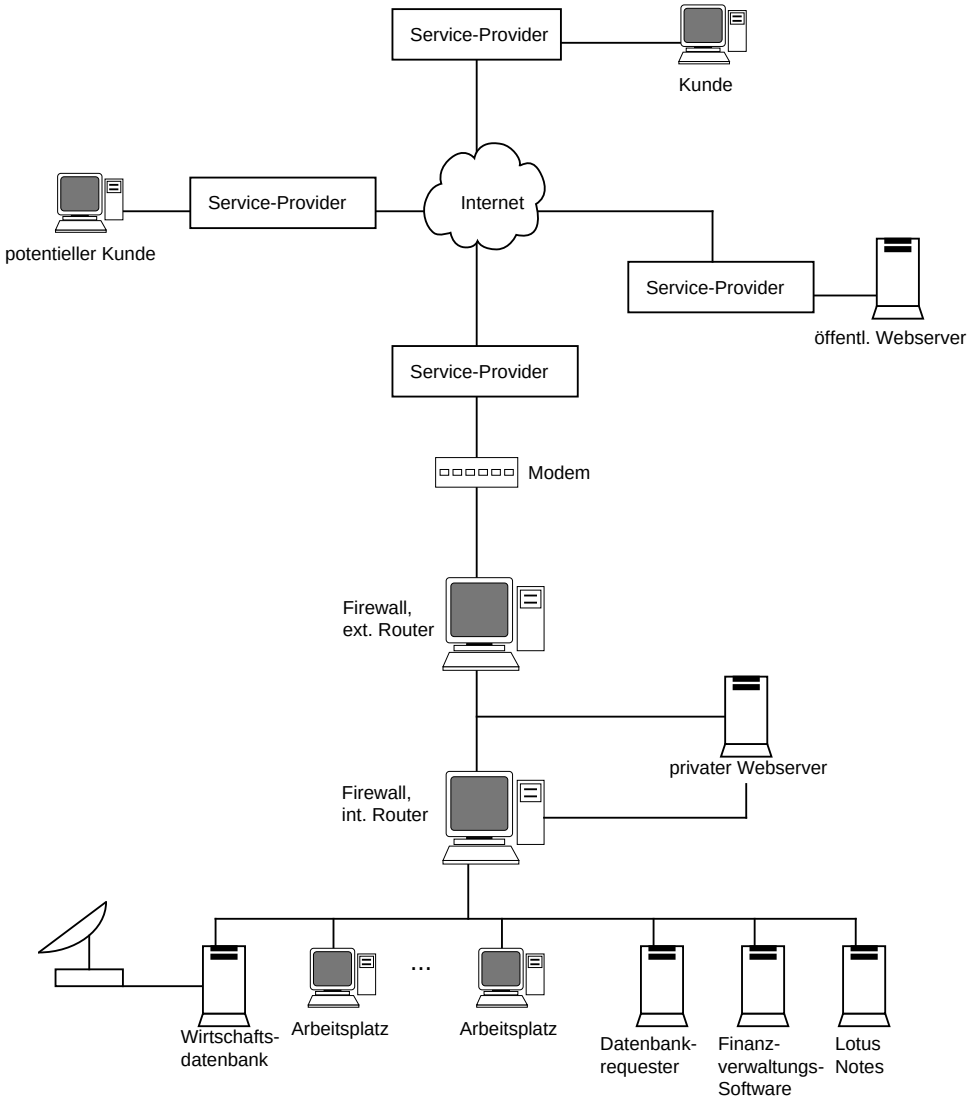

Abbildung 13.4: Netzarchitektur

Literaturverzeichnis

[Alt92] Steven Alter. *Information Systems - A Management Perspective*. Addison-Wesley Publishing, Reading, MA, 1992.

[And96] Ross Anderson. The Eternity Service. Technical report, Computer Laboratory, Universität Cambridge, 1996.

[Art93] Lowell Jay Arthur. *Improving Software Quality: An Insiders Guide to TQM*. John Wiley & Sons, Inc., New York, 1993.

[Bau94] Friedrich L. Bauer. *Kryptologie: Methoden und Maximen*. Springer Verlag, 2. edition, 1994.

[Bel89] Steven M. Bellovin. Security problems in the tcp/ip protocol suite. *Computer Communications Review*, 19(2):32–48, 1989.

[Bel92] Steven M. Bellovin. There Be Dragons. Technical report, AT&T Bell Laboratories, 1992.

[BFI95] BFI. Informatiksicherheit – Grundschutz von Informatiksystemen und -anwendungen. Weisung No. S02, Bundesamt für Informatik, 1995 1995.

[BHHS95] K. Bauknecht, R. Hauser, R. Holbein, and D. von Stockar. Sicherheit in der Informationstechnik. Skriptum, Institut für Informatik, Universität Zürich, SS 1995 1995.

[Bla94] Günther Blaschek. *Object-Oriented Programming with Prototypes*. Springer, 1994.

[Bor96] Marco Borer. Potentielle Bedrohungen und Schwachstellen für Unternehmen am Internet, Juli 1996. Arbeit entstand im Rahmen des SINUS-Projekts; Abgelegt in SINUS-Ordner.

[Bre89] D. Bressoud. *Factorization and Primality Testing*. Springer, 1989.

[BSI96] BSI. *IT Grundschutzhandbuch 1996. Massnahmeempfehlungen für den mittleren Schutzbedarf*, volume 3. Bundesanzeiger Verlag, Köln, 1996.

[CB94] William Cheswick and Steven M. Bellovin. *Firewalls and Internet Security. Repelling the Wily Hacker*. Addison Wesley ??, 1994.

[Cha92] D. Brent Chapman. Network (In)Security Through IP Packet Filtering. In *Proceedings of the Third USENIX Security Symposium*, 1992.

[Che90] Bill Cheswick. The Design of a Secure Internet Gateway. Technical report, AT&T Bell Laboratories, 1990.

[CM95] Christian Cachin and Ueli Maurer. Sicherheit im Internet: Illusion oder Realität. *Informatik/Informatique*, 2:18–23, 1995.

[Coh95] Frederick B. Cohen. *Protection and Security on the Information Superhighway*. John Wiley & Sons, Inc., New York, 1995.

[CW92] K. Campbell and M. Wiener. DES is not a group. *CRYPTO '92 Procedings*, pages 512–520, 1992.

[CZ96] D. Brent Chapman and Elizabeth D. Zwicky. *Einrichten von Internet Firewalls - Sicherheit im Internet gewährleisten*. O'Reilly International Thomson Verlag, 1996.

[Dev93] Mario Devargas. *Network Security*. NCC Blackwell Ltd, Oxford, UK, 1993.

[Dev95] Mario Devargas. *The Total Quality Management Approach to IT Security*. NCC Blackwell, Oxford, 1995.

[DG97] J. Dorn and G. Gottlob. Künstliche Intelligenz. In Peter Rechenberg and Gustav Pomberger, editors, *Informatik–Handbuch*, pages 819–838. Carl Hanser Verlag, 1997.

[DH77] W. Diffie and M. Hellmann. Exhaustive Cryptoanalysis of the NBS Data Encryption Standard. *IEEE Computer*, 10(6):74–78, 1977.

[DH96] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. *RFC 1883*, 1996.

[DL87] Tom DeMarco and Timothy Lister. *Wien wartet auf Dich! Der Faktor Mensch im DV-Management*. Hanser Verlag, 1987.

[Dob96] Hans Dobbertin. The Status of MD5 after a Recent Attack. *CryptoBytes, Technical Newsletter of RSA Data Security*, 2(2):1–6, 1996.

[Don97] Lutz Donnerhacke. Schleusen geöffnet. *Der Spiegel*, 08.02., 1997.

[ElG84] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *CRYPTO '84 Procedings*, pages 10–18, 1984.

[ElG85] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31:469–472, 1985.

[ETC96] Taher ElGamal, Jeff Treuhaft, and Frank Chen. Securing Communications on the Intranet and Over the Internet, July 1996 1996.

[FKK96] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL Protocol – Version 3.0, March 1996 1996.

[Gar95] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly & Associates, Inc., Sebastopol, CA, 1995.

[Han90] Jens Hanker. *Die strategische Bedeutung der Informatik für Organisationen*. Teubner Verlag, 1990.

[HDM80] M. Hellmann, W. Diffie, and R. Merkle. Public Key Cryptographic Apparatus and Method. *U.S. Patent #4,200,770*, 1980.

[Hei96] Lutz J. Heinrich. *Informationsmanagement- Planung, Überwachung und Steuerug der Informations-Infrastruktur*, volume 4. Oldenbourg Verlag, München, 1996.

[HI88] Sharam Hekmatpour and Darrel Ince. *Software Prototyping, Formal Methods and VDM*. Addison-Wesley Publishing, Workingham, UK, 1988.

[Hol96] Ralph Holbein. Secure Information Exchange in Organisations, 1996.

[How97] John D. Howard. An Analysis of Security Incidents on the Internet 1989-1995, 1997.

[Hun92] Craig Hunt. *TCP/IP Network Administration*. O'Reilly International Thomson Verlag, 1992.

[ISO96] ISO13335-1. Information Technology – Guidelines for the Management of IT Security, Part 1: Concepts and models for IT Security, 15. December 1996 1996.

[Jen97] Bruno Jenny. *Projektmanagement in der Wirtschaftsinformatik*. vdf Hochschulverlag, 2nd edition, 1997.

[Joh93] M. St. Johns. Identifaction Protocol. *RFC 1413*, 1993.

[Ker91] Heinrich Kersten. *Einführung in die Computersicherheit*. Oldenbourg Verlag, 1991.

[Kir96] Philipp Kirsch. Compound Documents in technischen Informationssystemen, August 1996 1996.

[Kra89] H. Krallmann. *EDV-Sciherungsmanagement - Integrierte Sicherungskonzepte für betriebliche Informations- und Kommunikationssysteme*. Schidt Verlag, Berlin, 1989.

[Kro95] Ed Krol. *Die Welt des Internet - Handbuch & Übersicht*. O'Reilly / International Thomson Verlag, Bonn, 1995.

[KS96] André Kaufmann and Pascal Sieber. Schweizer Firmen im Internet II - Eine empirische Untersuchung. Arbeitsbericht 87, Institut für Wirtschaftsinformatik, Universität Bern, Oktober 1996 1996.

[KTW97] Philipp Kirsch, Stephanie Teufel, and Harald Weidner. Sichere Nutzung von Online–Diensten. *Schweizer Bank*, (10):48f, Oktober 1997 1997.

[KW97] Philipp Kirsch and Harald Weidner. Online–Dienste im Internet. Eine kombinierte Anforderungs- und Risikoanalyse. In Günter Müller, Kai Rannenberg, Manfred Reitenspieß, and Helmut Stiegler, editors, *Verläßliche IT-Systeme. Zwischen Key Escrow und elektronischem Geld*, DuD Fachbeiträge, pages 269–280, Braunschweig, 1997. Vieweg Verlag.

[KWT97] Philipp Kirsch, Harald Weidner, and Stephanie Teufel. *SINUS – Security in Usage of Online Services*. PhD thesis, Chapman & Hall, 1997.

[Kya96] Othmar Kyas. *Sicherheit im Internet: Risikoanalyse, Strategien, Firewalls*. DATACOM Verlag, 1996.

[LL95] Stefan Lang and Peter Lockemann. *Datenbankeinsatz*. Springer Verlag, 1995.

[Luc98] Norbert Luckhardt. Nicht ganz dicht: Jugendliche knacken T–Online. *c't Magazin*, 7 / 98:62ff, 1998.

[Mau97] Uli Maurer. Sicherheit im Internet ist machbar, 22. April 1997 1997.

[Mau98] Philippe Maurer. Bedrohungen und Risiken im Internet, Februar 1998 1998.

[McC90] K. McCurley. The Discrete Logaritm Problem. Cryptography and Computational Number Theory. *Proceedings of the Symposium on Applied Mathematics Society*, pages 49–74, 1990.

[MFPW95] Josef Meyer-Fujara, Frank Puppe, and Ipke Wachsmuth. Expertensysteme und Wissensmodellierung. In Günther Görz, editor, *Einführung in die künstliche Intelligenz*, pages 705–753. Addison Wesley, 2 edition, 1995.

[NBM+97] Jay F. Nunamaker, Robert O. Briggs, Daniel D. Mittleman, Douglas R. Vogel, and Pierre A. Balthazard. Lessons Learnd from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings, Winter 1996-97 1997.

[OA90] Kazuo Ozeki and Tetsuichi Asaka. *Handbook of quality tools, the Japanese approach*. Productivity Press, Inc., Cambridge, MA, 1990.

[Plu97] Sendung Plusminus im Fernsehprogramm desNorddeutschen Rundfunks, 28.01. 1997.

[Pup90] Frank Puppe. *Problemlösungsmethoden in Expertensystemen*. Springer Verlag, 1990.

[PW93] Hartmut Pohl and Gerhard Weck. Stand und Zukunft in der Informationssicherheit. In Gerhard Weck Hartmud Pohl, editor, *Einführung in die Informationssicherheit*, pages 9–32. Oldenbourg Verlag, München, 1993.

[Rab79] M. Rabin. Digital Signatures and Public–Key–Functions as Intractable as Factorization. Technical report, MIT Lab. for Computer Science, Januar 1979.

[Rüh96] Edwin Rühli. *Unternehmungsführung und Unternehmungspolitik*, volume 1. Paul Haupt Verlag, 1996.

[Riv92] R. Rivest. The MD5 Message Digest Algorithm. *RFC 1321*, 1992.

[Rob94] Wendy Robson. *Strategic Management and Information Systems - An Integrated Approach*. Pitman Publishing, London, 1994.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SB92] Ingeborg Schaumüller-Bichel. *Sicherheits-Management - Risikobewältigung in informationstechnologischen Systemen*. BI Wissenschaftsverlag, 1992.

[Sch94] Josef Schreiber. *Beschaffung von Informatikmitteln: Pflichtenheft - Evaluation - Entscheidung*. Paul Haupt Verlag, 1994.

[Sch96] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, second edition, 1996.

[Sei97] Ulrich Seidel. *Das Recht des elektronischen Geschäftsverkehrs, technische Infrastruktur und Signaturgesetzgebung*. Vieweg Verlag, 1997.

[Stu94] Bruno Studer. *Sicherheit im Netzmanagement – Integration von Sicherheitsmechanismen in Netzmanagemet–Protokolle*. PhD thesis, Universität Zürich, 1994.

[Tan96] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[Tea96a] Computer Emergency Response Team. Denial of Service Attack via Ping. *CERT Advisory*, (96.26), 1996.

[Tea96b] Computer Emergency Response Team. TCP SYN Flooding and IP Spoofing Attacks. *CERT Advisory*, (96.21), 1996.

[Tea97] Computer Emergency Response Team. BIND – the Berkeley Internet Name Daemon. *CERT Advisory*, (97.22), 1997.

[Tea98] Computer Emergency Response Team. Multiple Vulnerabilities in BIND. *CERT Advisory*, (98.05), 1998.

[TSMB95] Stephanie Teufel, Christian Sauter, Thomas Mühlherr, and Kurt Bauknecht. *Computerunterstützung für die Gruppenarbeit*. Addison Wesley Verlag, 1995.

[TZL95] Stephanie Teufel, Urs Zurfluh, and Hannes Lubich. Projektantrag SINUS: Sichere Nutzung von Online–Diensten., 1995.

[WDR96] Sendung 'Ratgeber Computer' im Westdeutschen Rundfunk, 1996.

[Wei97a] Harald Weidner. Benutzeridentifizierung in Mehrbenutzersystemen. In Ch. Steigner, editor, *APS '97: Arbeitsplatz–Rechensysteme, Anwendungen, Architekturen und Netzwerke*. Fölbach Verlag, 1997.

[Wei97b] Harald Weidner. Sicherheit im internet - stand der technik. Institutsbericht 97.07, Institut für Informatik, Universität Zürich, März 1997 1997.

[Wey95] Markus Weyerhäuser. Taligent Commonpoint–Architektur: Frameworks - mehr als nur Objekte. *OBJEKTSpektrum*, 5:60–67, 1995.

[WGW90] John Ward, Pat Griffiths, and Paul Whitmore. *Strategic Planning for Information Systems*. John Wiley & Sons, Inc., New York, 1990.

[Woj91] Marek Wojcicki. *Sichere Netze: Analysen, Massnahmen, Koordination*. Carl Hanser Verlag, 1991.

[X3.81] ANSI X3.92. *American National Standard for Information System Data Link Encryption*. Americal National Standards Institute, 1981.

[Zur98] Urs E. Zurfluh. LAN–Strukturierung als Konsequenz der Internetkopplung. In *Sicherheit in Vernetzten Systemen*. DFN–CERT / DFN–PCA, 1998.