

Netzwerktechnik

Prof. Dr.-Ing. Torsten Finke

FOM

1. März 2011(Rev.: 2e1707b65328)

Überblick

Ankündigungen

Netzwerkgrundlagen

Netzprotokolle

Name Service

Zeit

Datenübertragung und Fernwirkung

E-Mail

World Wide Web

Sicherheit





Netzwerktechnik – Folien

<http://www.dr-torsten-finke.de/lehre/algorithmen/bachelor/Script-Netzwerk.pdf>

Klausur

- ▶ Formalia
 - ▶ Dauer 60 Minuten
 - ▶ keine Hilfsmittel
 - ▶ Formvorschriften
- ▶ Inhalt
 - ▶ Inhalt komplett relevant
 - ▶ kein Repetitorium
 - ▶ Auswahlklausur
 - ▶ Schwerpunkt auf Verständnis

Netzwerktechnik – Literatur

-  HUNT, CRAIG: *TCP/IP Netzwerk- Administration*.
O'Reilly, 2002.
ISBN: 3897211793.
-  TANENBAUM, ANDREW S.: *Computer Networks*.
Prentice Hall, 4. Auflage, 2002.
ISBN: 978-0130384881.
-  TANENBAUM, ANDREW S.: *Modern Operating Systems*.
Prentice Hall, 2008.
ISBN: 978-0138134594.
-  TANENBAUM, ANDREW S. und JAMES GOODMAN: *Computerarchitektur. Strukturen, Konzepte, Grundlagen*.
Pearson Studium, 2. Auflage, 2001.
ISBN: 3827370167.

Netzwerke – Geschichte, Beispiel Internet

- 1969: ARPANET (Advanced Research Project Agency) an vier Standorten;
- 1974: Entwicklung des TCP/IP Protokolls;
- 1983: Abtrennung des MILNET;
- 1984: Aufbau des NSFNET (National Science Foundation);
- 1990: Kommerzialisierung des NSFNET per ANSNET (Advanced Networks and Services – IBM u.a.);
- 1998: Gründung der ICANN (Internet Corporation for Assigned Names and Numbers) zur Verwaltung des Internets.

Netzwerke – Nutzung, Beispiel Internet

- ▶ Mail;
- ▶ News;
- ▶ Dateitransfer (FTP);
- ▶ Remote Login (Telnet);
- ▶ WWW (World Wide Web – Teil des Internet).

Netzwerke – Konzepte: Grundstruktur

- ▶ LAN – Local Area Network;
- ▶ WAN – Wide Area Network;
- ▶ Medienebunden;
- ▶ Drahtlos;
- ▶ Verbund.

Netzwerke – Konzepte: Software

- ▶ Schichtendesign:
 - ▶ Aufgabengliederung;
 - ▶ Transparenz;
 - ▶ Flexibilität.
 - ▶ Verbindungsaufbau;
 - ▶ Flußrichtungen;
 - ▶ Flußsteuerung;
 - ▶ Prioritäten;
 - ▶ Fehlerbehandlung;
- ▶ Protokoll definiert Kommunikation zwischen gleichen Schichten;
- ▶ Schnittstelle definiert Kommunikation zwischen zwei benachbarten Schichten;
- ▶ Dienste:
 - ▶ jede Schicht bietet der oberen Schicht Dienste an;
 - ▶ verbindungsorientiert/verbindungslos;
 - ▶ zuverlässig/unzuverlässig.

Netzwerke – Schichtenmodelle: OSI

- ▶ OSI (Open Systems Interconnection) – ISO-Standard, 1983
- ▶ 1. Bitübertragung;
- ▶ 2. Sicherung (Rahmenübertragung);
- ▶ 3. Vermittlung (von Host zu Host);
- ▶ 4. Transport (von Programm zu Programm)
- ▶ 5. Sitzung (z.B. Checkpoints in Datenströmen);
- ▶ 6. Darstellungsschicht (ASCII, Unicode, EBCDIC);
- ▶ 7. Verarbeitung (z.B. Kodierung von Textformaten).

Netzwerke – Schichtenmodelle: TCP/IP

1. Bitübertragung und Sicherung (z.B. per LAN);
2. Internet
 - ▶ IP – Internet Protocol);
 - ▶ ICMP – Internet Control Message Protocol;
3. Transport
 - ▶ TCP (Transport Control Protocol);
 - ▶ UDP (User Datagram Protocol);
4. Verarbeitung
 - ▶ Telnet,
 - ▶ FTP,
 - ▶ SMTP,
 - ▶ DNS,
 - ▶ WWW.

Protokolle im Überblick – TCP/IP-Modell

Schicht	Protokolle							
Verarbeitung	FTP	HTTP	SMTP	DNS	Telnet	NNTP	SSH	POP3
Transport	TCP				UDP			
Vermittlung	IP				ICMP			
Sicherung	ARP		RARP	SLIP	PPP	IEEE 802		

- ▶ Verarbeitung erfolgt durch Prozesse(Applikationen);
- ▶ alle unteren Schichten werden softwaremäßig durch den Betriebssystemkern realisiert.

Sicherungsschicht

- ▶ Hauptaufgaben:
 - ▶ Rahmenweise Datenübertragung;
 - ▶ Fehlererkennung;
 - ▶ Flusskontrolle.
- ▶ Beispiele:
 - ▶ Point to Point:
 - ▶ SLIP (keine Fehlerbehandlung, nur IP, keine dyn. IP-Adressen, keine Authentifikation);
 - ▶ PPP – Byte-Rahmen:

Flag	Adresse	Steuerung	Protokoll	Nutzdaten	Prüfsumme	Flag
01111110	11111111	00000011	1 oder 2	variabel		01111110

- ▶ Broadcast:
 - ▶ Ethernet (IEEE 802.3 – einfach, schnell, nicht echtzeitfähig);
 - ▶ Token-Ring (IEEE 802.5).

Vermittlungsschicht – Adressen/Protokolle

- ▶ IP-Adressen:
 - ▶ 32 Bit (Host-/Netzmaske);
 - ▶ Klassenteilung – reserviert:
 - ▶ A: 10.0.0.0/255.0.0.0;
 - ▶ B: 172.16.0.0–172.31.255.255/255.255.0.0;
 - ▶ C: 192.168.0.0–192.168.255.255/255.255.255.0.
 - ▶ CIDR (Classless Inter Domain Routing).
- ▶ Steuerung:
 - ▶ ICMP (Internet Control Message Protocol);
 - ▶ ARP (Address Resolution Protocol);
 - ▶ RARP (Reverse Address Resolution Protocol).

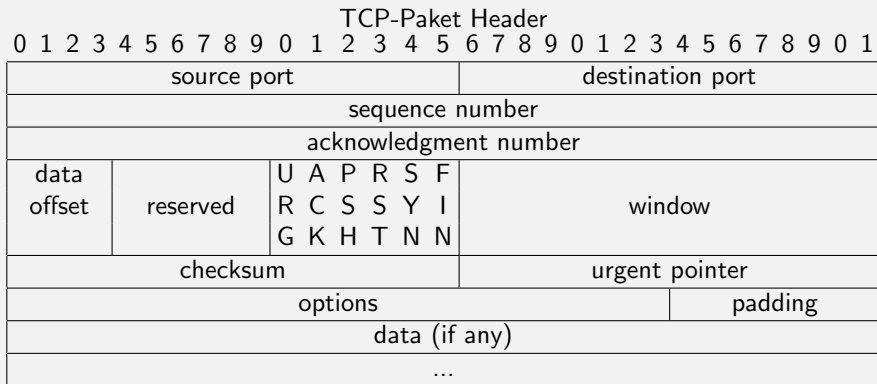
Vermittlungsschicht – IP

IP-Paket Header																															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
IPV				IHL				ToS				datagram size																			
ID								flags				fragment offset																			
TTL				Prot				header checksum																							
source IP address																destination IP address															
options																...															
data																...															

Vermittlungsschicht – ICMP

ICMP-Paket Header																					
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type				Code				checksum													
data depending on type and code																					
...																					
Type	Code	Bedeutung (RFC792)																			
0		Echo Reply																			
3		Destination Unreachable																			
	0	Net Unreachable																			
	1	Host Unreachable																			
	2	Protocol Unreachable																			
	3	Port Unreachable																			
4		Source Quench																			
5		Redirect																			
8		Echo Request																			

Transport – TCP (verbindungsorientiert/zuverlässig)



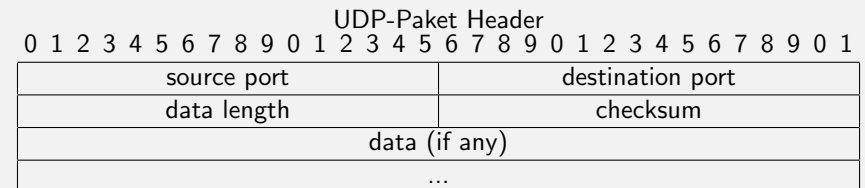
Transport – TCP-Verbindungsverwaltung

- ▶ Dreiwegen-Handshake für Verbindungsaufbau:
 1. Request der Verbindung (SYN);
 2. Bestätigung von der Gegenstelle (SYN+ACK);
 3. Transport (ACK);
- ▶ Verbindungsabbau:
 - ▶ Request für Abbau (FIN);
 - ▶ Bestätigung von der Gegenstelle (FIN oder FIN+ACK im Duplex-Betrieb);

TCP-Stack - Zustandsverwaltung

- CLOSED**: keine Verbindung;
- LISTEN**: Server wartet auf Request;
- SYN RCVD**: Verbindungsrequest beim Server eingetroffen;
- SYN SENT**: Client hat Verbindung begonnen zu öffnen;
- ESTABLISHED**: normale Verbindung;
- FIN WAIT 1**: Client hat Abbaurequest gesendet;
- FIN WAIT 2**: Bestätigung des Servers zum Abbau;
- TIMED WAIT**: Client wartet, bis keine Pakete mehr eintreffen;
- CLOSING**: beide Seiten wollen abbauen;
- CLOSE WAIT**: Client hat Abbau begonnen;
- LAST ACK**: Server wartet bis keine Pakete mehr eintreffen;

Transport – UDP (verbindungslos/unzuverlässig)



Verarbeitungsschicht – Domain Name Service(DNS)

- ▶ Einfachste Namensauflösung: Datei hosts
- ▶ DNS: RFC 1034, 1035 (www.ietf.org/rfc)
- ▶ Hierarchischer Internet Namensraum – Wurzelpunkt: „.“;
- ▶ verteilte Datenbank: Name – Adresse;
- ▶ FQDN (full qualified domain name):
 - ▶ TLD (top level domain) – Verwaltung durch ICANN;
 - ▶ Domain (z.B. Institution) – Verwaltung durch z.B. DENIC (deutsches Network Information Center);
 - ▶ Subnetz (lokale Verwaltung oder ISP (Internet Service Provider));
 - ▶ Sub-Sub-...-Netz, Host (eigene Verwaltung);
- ▶ Forwarding/Delegation

DNS-Angaben

- ▶ Root-Nameserver
(ftp://FTP.RS.INTERNIC.NET/domain/named.root)


```

      .                3600000   IN   NS    A.ROOT-SERVERS.NET.
      A.ROOT-SERVERS.NET. 3600000   A    198.41.0.4
      ...
      .                3600000   NS   K.ROOT-SERVERS.NET.
      K.ROOT-SERVERS.NET. 3600000   A    193.0.14.129
      ;
      
```
- ▶ Forwarder;
- ▶ Zonentabellen (paarweise);
- ▶ Zeitverhalten;

DNS – Zonen Records

- SOA:** Zonenparameter;
 - A:** verweist auf IP-Adresse;
- PTR:** verweist auf Namen (Problematik: Reverse Name Service Lookup);
- NS:** Nameserver;
- CNAME:** Alias (verweist auf canonischen Namen);
- MX:** Mail-Exchanger (mit Priorität);
- SRV:** Service Hosts (mit Priorität und Port)
- HINFO:** Host-Beschreibung.

DNS – Zonentabelle (Beispiel BIND)

```

Zone: dom.de
@      IN  SOA    ns root.ns(
        2004042202      ; serial
        28800           ; refresh, seconds
        7200            ; retry, seconds
        3600000         ; expire, seconds
        86400 )         ; minimum, seconds
        IN  NS     ns
        IN  MX     10 mail      ; Primary Mail Exchanger

alf    IN  A      192.124.4.100      ;
        IN  HINFO  PA-RISC ObenBSD ; HW/OS
bert   IN  A      192.124.4.101      ;
conny  IN  A      192.124.4.102      ;
www    IN  CNAME  alf              ;
pub    IN  SRV    0 1 21 bert        ; 21 - FTP
  
```

DNS – Inverse Zonentabelle

Zone: 4.124.192.in-addr.arpa (192.124.4.0/24)

```

@   IN  SOA  ns.dom.de. root.ns.dom.de.(
      2004042202      ; serial
      28800           ; refresh, seconds
      7200            ; retry, seconds
      3600000         ; expire, seconds
      86400           ; minimum, seconds
      IN  NS  ns.dom.de.
      IN  MX  10 mail.dom.de.      ; Primary Mail Exchanger

100 IN  PTR  alf.dom.de.          ;
101 IN  PTR  bert.dom.de.        ;
102 IN  PTR  conny.dom.de.       ;

```

DNS – Betrieb

- ▶ Clients:
 - ▶ Resolver(Library) greift auf DNS zu;
 - ▶ direkter Zugriff auf Nameserver: nslookup, host, dig


```

$ nslookup
Default Server: ns.home.de
Address: 192.168.1.1
> set type=mx
> site.de
Server: ns.home.de
Address: 192.168.1.1
Non-authoritative answer:
site.de preference = 10, mail exchanger = mail.site.de

```
- ▶ Redundanz – Master/Slave;
- ▶ Beschränkung: Client-Zugriff, Zonentransfer;
- ▶ Caching only;
- ▶ Dynamisches DNS (Kombination mit DHCP – Dynamic Host Control Protocol);
- ▶ Sicherheit: DOS

Netzwerke und Zeit

- ▶ Zeit sollte auf allen Rechnern im Netz bekannt sein
 - ▶ möglichst kontinuierlich,
 - ▶ möglichst synchron,
 - ▶ möglichst präzise.
- ▶ Zeitübertragungs-Protokolle
 - Daytime (RFC-867):** TCP/UDP-Port 13, ASCII-String, Format nicht festgelegt;
 - Time (RFC-868):** TCP/UDP-Port 37, binär, 32-Bit Zähler für Sekunden seit 1900-01-01 00:00.00 UTC;
 - NTP/SNTP (RFC-1305/2030):** UDP-Port 123.

Zeit im Netzwerk – NTP

- ▶ Quellen:
 - ▶ Lokale Zeitgeber (unpräzise);
 - ▶ Funkuhren (DCF-77), GPS-Geber;
 - ▶ Zeitserver (ntp1.ptb.de, pool.ntp.org)
- ▶ Einrichten


```

Un!x: /etc/ntp.conf: server ntp1.ptb.de; Kommando:
      ntpdate, ntpd;
W2k: net time /setsntp:ntp1.ptb.de (Registry); Kommando:
      net start w32time

```

Verarbeitungsschicht – Datenübertragung: FTP

- ▶ File Transfer Protocol (RFC 959)
- ▶ Zugriff auf entferntes Dateisystem;
- ▶ betriebssystemübergreifend:
 - ▶ Abbildung Dateisystemeigenschaften
 - ▶ Dateikonventionen des Betriebssystems, z.B. Zeilenende in Textdatei:
 - ▶ Modus: ASCII/BINARY;
 - ▶ DOS, Windows, VMS: CR LF;
 - ▶ Unix: LF;
 - ▶ Apple: CR.
- ▶ Zugriff per Authentifikation am Server;
- ▶ Transfer per get und put bezogen auf Client-Standort;
- ▶ Sicherheit:
 - ▶ Kommunikation komplett unverschlüsselt (password sniffing);
 - ▶ Serverseitig Beschränkung auf Teildateisystem;
 - ▶ Logging.

Verarbeitungsschicht – Datenübertragung: FTP

- ▶ Kommunikation über ein Verbindungs-Paar:
 - ▶ Steuerung über Server-Port 21;
 - ▶ Datenfluss über Server-Port 20.
- ▶ Alternative Betriebsmodi:
 - ▶ Active FTP:
 1. Client teilt Server auf Port 21 seinen Datenport (> 1024) mit (z.B. PORT 1234);
 2. Server quittiert über Port 21;
 3. Server initiiert Datenfluss von Port 20 an Client-Datenport;
 4. Client bestätigt paketweise gegenüber Server Port 20.
 - ▶ Passive FTP:
 1. Client fordert am Server Port 21 passive mode an (PASV);
 2. Server sendet von Port 21 die Nummer des Server-Daten-Ports;
 3. Client fordert vom eigenen Datenport Server-Daten am Server-Daten-Port an;
 4. Server liefert Daten vom Daten-Port.
- ▶ Berücksichtigung bei Übertragung über Firewalls hinweg (passive Mode bevorzugen!)

Verarbeitungsschicht – Fernwirkung: Telnet

- ▶ Telnet Protokoll nach RFC 854;
- ▶ Hauptverwendung: Terminalemulation – Server bietet login und anschließende Text-Terminal-Sitzung an.
- ▶ Universalclient,
 - ▶ Zugriff über well known ports (z.B. in /etc/services);
 - ▶ Beispiele
 - ▶ telnet server.any.com www, anschließend GET index.html liefert Klartextseite vom Webserver;
 - ▶ telnet how.late.is.it daytime liefert Uhrzeit;
- ▶ Gegebenenfalls am Client Echo-Option aktivieren;
- ▶ Option zur Session-Unterbrechung z.B. per Shortcut;
- ▶ Sicherheit
 - ▶ Kommunikation komplett unverschlüsselt (password sniffing);
 - ▶ Risiko des Session-Hijacking.

Verarbeitungsschicht – Email

- ▶ Hauptzweck: Datenübermittlung
- ▶ Normen:
 - ▶ RFC 821(Übertragungsprotokoll)
 - ▶ RFC 822(Nachrichtenformat)
 - ▶ X.400
- ▶ Teilaufgaben:
 - ▶ Erstellung – Mail User Agent (MUA)
 - ▶ Transfer – Mail Transport Agent (MTA)
 - ▶ Berichterstellung
 - ▶ Auslieferung/Verwertung(Mailboxen) – Mail Delivery Agent (MDA)
 - ▶ Anzeige – MUA
- ▶ Email Grobaufbau (komplett unverschlüsselter ASCII-Text):
 - ▶ Envelope
 - ▶ Header
 - ▶ Body

Email – Mailformat

```

From fortune_caroline@stade.fr  Fri May 21 19:30:04 2004
Return-Path: <fortune_caroline@stade.fr>
Received: from localhost (localhost [127.0.0.1])
    by jana.igh-essen.de (8.11.2/8.11.2/SuSE Linux 8.11.1-0.5) with ESMTMP id i4LHU4e
    for <fi@localhost>; Fri, 21 May 2004 19:30:04 +0200
Delivered-To: igh-webmaster@igh-essen.com
Received: from igh-essen.com
    by localhost with POP3 (fetchmail-5.6.0)
    for fi@localhost (multi-drop); Fri, 21 May 2004 19:30:04 +0200 (MEST)
Received: (qmail 20681 invoked by uid 634); 21 May 2004 17:15:42 -0000
...
From: "caroline" <fortune_caroline@stade.fr>
Subject: THE PRESIDENT/DIRECTOR.
X-Mailer: CommuniGate Pro WebUser Interface v.4.1.8
Date: Fri, 21 May 2004 19:14:21 +0200
Message-ID: <web-1395956@mail.stade.fr>
MIME-Version: 1.0
Content-Type: text/plain; charset="ISO-8859-1"; format="flowed"
Content-Transfer-Encoding: 8bit
X-Fetchmail-Warning: no recipient addresses matched declared local names

```

Email – Benutzeragenten (MUA)

- ▶ Aufgaben
 - ▶ Erstellen/Beantworten von Mails
 - ▶ Verwaltung von Mailadressen (kollege@firma.de)
 - ▶ Mails anzeigen/weiterleiten
 - ▶ Mails in Mailboxen einsortieren
- ▶ Funktionsmerkmale (Auswahl)
 - ▶ Texterfassung/Editor
 - ▶ Mails suchen/sortieren nach gegebenen Kriterien
 - ▶ Anhänge
 - ▶ Verschlüsselung
 - ▶ anlegen von Ausgangskopien
- ▶ Beispiele:
 - ▶ mutt(ASCII-basiert)
 - ▶ mozilla(browser-basiert)
 - ▶ Eudora
 - ▶ Outlook

Email – Nachrichtenformate

- ▶ Header/Envelope nach RFC 822
 - ▶ To: Empfänger
 - ▶ From: Absender
 - ▶ CC: Kopie
 - ▶ BCC: Blindkopie
 - ▶ Subject: Betreff
 - ▶ Message-ID: Eindeutige Kennung der Nachricht
 - ▶ Received: Stempel vom MTA (Envelope)
 - ▶ X-: anwendungsspezifische Header
- ▶ MIME (Multipurpose Internet Mail Extension)
 - ▶ Hauptzweck: Verwendung von Sonderzeichen und Anhängen
 - ▶ MIME Header (Auswahl):
 - ▶ Content-Type
 - ▶ Content-Transfer-Encoding

Email – MIME

- ▶ Content-Type (Beispiele nach RFC 2045):
 - ▶ Text/plain: ASCII-Text
 - ▶ Image/jpeg: Raster-Bild
 - ▶ Video/mpeg: Laufbild
 - ▶ Application/postscript: Druckbares Dokument
 - ▶ Message/rfc822: Nachricht nach RFC 822
 - ▶ Multipart/alternative: gleiche Nachricht in verschiedenen Formaten (möglichst vermeiden!)
 - ▶ Multipart/mixed: unabhängige Teile
- ▶ Content-Transfer-Encoding:
 - ▶ in Emails werden nur ASCII-Text-Zeichen akzeptiert
 - ▶ Base64 Encoding:
 - ▶ 3 byte in 4 x 6 bit zerlegen
 - ▶ 6-Bit Code-Zuordnung: 0-„A“, 1-„B“ ... 51-„z“, 52-„0“ ... 62-„+“, 63-„/“
 - ▶ ggf. Stopfen mit folgenden „=“
 - ▶ Quoted Printable: ASCII über 127: Gleichheitszeichen plus Hex-Code, z.B. „ß“ entspricht =DF

Email – Transport(MTA)

- ▶ Hauptaufgaben:
 - ▶ Daten weiterleiten (ggf. verzögert)
 - ▶ Verlust verhindern/Probleme berichten
- ▶ Schwierigkeiten:
 - ▶ Schleifen aufbrechen
 - ▶ Missbrauch vermeiden (Spam-Relay)
 - ▶ komplexe Regelwerke (z.B. localhost, aliases)
- ▶ Typische Systeme:
 - ▶ sendmail
 - ▶ postfix
 - ▶ qmail
 - ▶ exchange

Email-Transport – SMTP(Simple Mail Transport Protocol)

- ▶ Server lauscht an TCP-Port 25
- ▶ Klartext-ASCII Protokoll:


```
$ telnet mail.firma.de 25
HELO my.host.de      -> 250 mail.firma.de Hello my.host.de,
                    pleased to meet you

MAIL FROM: me@my.host.de -> 250 Sender ok
RCPT TO: you@firma.de   -> 250 Recipient ok
DATA                   -> Enter mail, end with "."
                    on a line by itself

From: me
To: you
Subject: party

hello You
...
.
QUIT                 -> 250 Message accepted for delivery
```

Email – Endzustellung

- ▶ Empfänger nicht erreichbar (z.B. nur temporär am Netz): Transport bis zur Mailbox beim ISP
- ▶ POP3 (Post Office Protocol) Aktives Abholen über TCP-Port 110:


```
$ telnet pop3.isp.de 110
                    +OK ready

USER ich          -> +OK Password required
PASS geheim      -> +OK ich has 12 visible messages
LIST             -> +OK 120 visible messages (5482949 octets)
                    1 3173

RETR 1           -> ...
DELE 1           -> ...

QUIT
```
- ▶ POP3 arbeitet in Klartext-ASCII (Sniffing Risiko!)
- ▶ Server-basierte Alternative: IMAP(Internet Message Access Protocol)
 - ▶ zentrale Mailboxen,
 - ▶ auch mobil verfügbar,
 - ▶ aber komplexe Struktur

Email – Zustellung/Auslieferung

- ▶ automatisches Einsortieren in Mailboxen
- ▶ Weiterleiten
- ▶ Verarbeiten
 - ▶ automatisch beantworten(vacation)
 - ▶ einem Prozess zuführen
- ▶ Filtern/Analysieren
 - ▶ Viren, Würmer, Trojaner
 - ▶ Spam (erkennen, vermeiden, rückverfolgen)
- ▶ typisches Werkzeug: procmail

Verarbeitungsschicht – World Wide Web

- ▶ Beginn 1989 am CERN (Tim Berners Lee)
- ▶ Idee: Nutzung von Dokumenten und Diensten in verteilten Systemen
- ▶ Grafische Browser seit 1993 (Mosaic, Marc Andreessen)
- ▶ 1994: Gründung W3C (MIT und CERN): www.w3c.org
- ▶ Hypertext
 - ▶ Inhalt
 - ▶ Format, Verknüpfung(URL), Funktion
 - ▶ universell: ASCII

Architektur

- ▶ Client
 - ▶ Browser (Netscape, InternetExplorer, Opera, Lynx, ...)
 - ▶ Anfrage auf Zielsystem (DNS-Anfrage)
 - ▶ Anforderung an Webserver
 - ▶ Präsentation (MIME – Plugins, Hilfsanwendungen)
- ▶ Server
 - ▶ lauscht an Port 80(443)
 - ▶ meist multithreaded
 - ▶ Authentifikation/Authorisierung
 - ▶ beschafft/generiert Dokument
 - ▶ liefert MIME-Type und Dokument
- ▶ Logische Verbindung: URL (Uniform Ressource Locator)
 - ▶ Aufbau: `service://server/ressource`
 - ▶ Ressource: optionale Details (Argumente, Positionen)

URL – Uniform Ressource Locator

- ▶ Allgemeine Beschreibung von Dienst oder Dokument im Netz
- ▶ Dienste:
 - ▶ http: HyperText Transport Protocol (<http://www.site.de>)
 - ▶ ftp: File Transfer Protocol (<ftp://ftp.server.de/pub/file>)
 - ▶ file: lokale Datei (<file:///home/user/text>)
 - ▶ telnet: Fernwirkung (<telnet://my.server.de>)
 - ▶ mailto: Email (<mailto:friend@any.where>)
- ▶ Dokumentenbezug mit Position
(<http://www.de/pfad/text#label>)
- ▶ Argumente für dynamische Dokumente
(<http://www.de/cgi-bin/prog?var=wert&dies=das>)

Web-Server – Funktionen

- ▶ Main-Server/Sub-Server
- ▶ URL auflösen
- ▶ Client identifizieren
- ▶ Zugriff prüfen
- ▶ Dokument aus Dateisystem beschaffen/von Hilfsprozess erstellen lassen
- ▶ Ausliefern (Angabe von Typ, Gültigkeitsdauer, Umleitung, ...)
- ▶ Logging
- ▶ Zustandslosigkeit behandeln(Cookies)
- ▶ Beispiele:
 - ▶ Apache
 - ▶ Internet Information Server

Hypertext – HTML (Hypertext Markup Language)

- ▶ Formatierung, Gestaltung
- ▶ Dokumentenverknüpfung (unidirektional)
- ▶ Interaktivität (Formulare)
- ▶ Syntax aufbauend auf Tags (<tag>text</tag>), Style Sheets
- ▶ Beschränkung auf ASCII (benannte Sonderzeichen – häl;tte – hätte)
- ▶ Beispiel

```
<html>
<head><title>Beispiel</title></head>
<body>
<h1>Hello World</h1>
<form action="/cgi-bin/prog">
...
</form>
<a href="http://www.home.de">Startseite</a>

</body>
</html>
```

- ▶ Beachten: Sprachstandard, Features, Browservielfalt

Dynamische Dokumente – Serverseitig

- ▶ CGI (Common Gateway Interface):
 - ▶ Server liefert Argumente an Programm
 - ▶ Programm verarbeitet Argumente, ermittelt Resultate
 - ▶ optionaler Zugriff auf Datenbank
 - ▶ Programm erstellt Dokument (z.B. in HTML) und liefert an Webserver
 - ▶ Webserver leitet Dokument an Client
- ▶ Eingebettete Scripte
 - ▶ Spracherweiterung HTML (z.B. <?php echo hallo?>)
 - ▶ Server aktiviert Interpretermodul
 - ▶ performant, aber riskant für den jeweiligen Server-Thread
- ▶ Vorteil: Rechenleistung auf Server, Kontrolle
- ▶ Nachteil: schlechte Interaktivität
- ▶ Beachten: Absicherung

Dynamische Dokumente – Clientseitig

- ▶ Einbettung von Script-Code in HTML

```
<head>
<script language=javascript>
function hello() {
    x = ...
}
</script>
```

- ▶ Virtuelle Maschine im Browser (z.B. Java Virtual Machine – JVM), Funktion kommt als (binäres) Applet
- ▶ Plugins/Hilfsprogramme
- ▶ Vorteil: komfortabel, flexibel, hoch interaktiv
- ▶ Nachteil: UNSICHER!, (meist) langsam

HyperText Transport Protocol – HTTP

- ▶ HTTP 1.0 definiert: Frage, Antwort, Schluss
- ▶ HTTP 1.1 erlaubt persistente Verbindungen
- ▶ HTTPS – sichere (verschlüsselte) Übertragung
- ▶ Requests:
 - ▶ GET: Dokumentenanforderung
 - ▶ HEAD: Anforderung nur des Headers (z.B. zur Prüfung von Zeitmarken)
 - ▶ PUT: Dokument hochladen zum Server
 - ▶ POST: Nachrichtenstrom zum Server
 - ▶ DELETE: Dokument auf Server löschen (sofern erlaubt)
- ▶ Replies numerisch (z.B. 404 – Seite nicht gefunden)

HTTP – Nachrichten-Header (optional)

- ▶ User-Agent: Client nennt Browser-Typ
- ▶ Accept: Client nennt MIME-Types, die er verarbeitet
- ▶ Host: Client nennt seinen DNS-Namen
- ▶ Cookie: Client sendet erhaltenen Cookie
- ▶ Date: Zeit (bidirectional)
- ▶ Server: Webserver Identifikation
- ▶ Content-Type: Server nennt MIME-Typ der Seite
- ▶ Set-Cookie: Server sendet Cookie

Web-Caching – Proxy

- ▶ Idee: zwischenspeichern von Webseiten
- ▶ direkt im Browser
- ▶ Client-seitig
 - ▶ Proxy (viel Speicher) im LAN
 - ▶ lauscht an Port 3128, 8080, ...
 - ▶ schnelle Anbindung intern
 - ▶ Reduzierung des externen Traffics
 - ▶ Zugriffskontrolle (Authorisierung/Authentifikation)
 - ▶ Logging
 - ▶ Bandbreitenverwaltung
 - ▶ Namensauflösung
- ▶ Server-seitig (lauscht an Port 80)
 - ▶ Lastreduzierung
 - ▶ Lastverteilung
 - ▶ Schutz
- ▶ Beispiel: Squid

IT-Sicherheit – Begriffe

- ▶ Bedrohungen:
 - ▶ Aufdecken vertraulicher Informationen, Datenschutz
 - ▶ Datenmanipulation (Integrität, Echtheit, Urheberschaft)
 - ▶ Unterbrechen der Systemverfügbarkeit
 - ▶ Unfälle und höhere Gewalt
- ▶ Angreifer-Motive:
 - ▶ Neugierde, Ehrgeiz, Rache
 - ▶ Wirtschaftliche Vorteile
 - ▶ Spionage
- ▶ Sicherheitsrichtlinie
 - ▶ minimale Zugriffsrechte
 - ▶ Redundanz (Mehrschichtigkeit)
 - ▶ Auffinden von Schwachstellen
 - ▶ Grundkonzept: permissiv/restriktiv
 - ▶ Beteiligung
 - ▶ Einfachheit

Verschlüsselung – Kryptographie

- ▶ Grundprinzipien der Kryptographie:
 - ▶ Chiffre = Verschlüsselung(Nachricht, Schlüssel) – $C = E(N, E_k)$
Nachricht = Entschlüsselung(Chiffre, Umkehrschlüssel) –
 $N = D(C, E_d)$
 - ▶ Prinzip von Kerckhoff: nur Schlüssel geheim, aber Algorithmen öffentlich
 - ▶ Redundanz in Chiffren (Unmöglichkeit der zufälligen Erzeugung von Chiffren)
 - ▶ Aktualität (Vermeidung von Replay)
- ▶ Elementarverfahren:
 - ▶ Substitutionsschiffre(Cäsar)
 - ▶ One Time Pads (theoretisch sicher)

Symmetrische Kryptographie

- ▶ Schlüssel und Entschlüssel leicht auseinander ermittelbar
- ▶ Problem: Schlüssel muss gesichert übertragen werden
- ▶ Gängige Verfahren:
 - ▶ DES (Data Encryption Standard – IBM)
 - ▶ blockweise (64 bit) Verschlüsselung (56 Bit)
 - ▶ iteratives Vertauschen in 19 Schritten
 - ▶ verbessert als Triple DES

$$C = E(D(E(N, K_1), K_2), K_1)$$

$$N = D(E(D(C, K_1), K_2), K_1)$$
 - ▶ Twofish (stark), Blowfish (veraltet) (B. Schneier)
 - ▶ IDEA, RC5 (gut, patentiert)
 - ▶ Rijndahl, Serpent (sehr stark)

Public-Key Kryptographie

- ▶ Grundidee (Diffie, Hellman): jeder Teilnehmer verwendet ein Schlüsselpaar (G, P)
 - ▶ geheimer Schlüssel G , öffentlicher Schlüssel P
 - ▶ $N = G(P(N))$
 - ▶ Ableiten von G aus P schwierig (Primzahlrechnung)
 - ▶ G und P einfach gemeinsam berechenbar
 - ▶ RSA (Rivest, Shamir, Adleman):
 - ▶ wähle $p, q \gg 1$, prim (z.B. $11 \cdot 13 = 143$, real über 100 Stellen)
 - ▶ berechne $n = p \cdot q$, $z = (p - 1)(q - 1) = 120$
 - ▶ wähle $d > p, q$ teilerfremd zu z (z.B. $d = 17$ prim)
 - ▶ wähle e mit $(e \cdot d) \bmod z = 1$ (z.B. $e = 113$)
 - ▶ Verschlüsseln: $C = N^e \bmod n$
 - ▶ Entschlüsseln: $N = C^d \bmod n$
- ▶ deutlich langsamer/rechenaufwändiger als symmetrische Verfahren
- ▶ kombinierbar mit symmetrischen Verfahren

Public Key Kryptographie – Anwendung

- ▶ digitale Signaturen (MD5, SHA-1)
- ▶ Verwaltung öffentlicher Schlüssel
 - ▶ Public Key Infrastruktur
 - ▶ Zertifikate z.B. nach X.509

Benutzerauthentifikation

- ▶ Gegenstandsgebunden (Codekarte, Schlüssel)
- ▶ Identifikation(Name) und Geheimnis(Passwort)
 - ▶ nur gemeinsam akzeptieren
 - ▶ Passwortsicherheit:
 - ▶ ändern
 - ▶ gegen Erraten sichern (Wahrscheinlichkeit, zufällige Gleichheit)
 - ▶ niemals im Klartext speichern
 - ▶ Cracking
 - ▶ Challenge/Response
 - ▶ Einmalpasswörter
- ▶ Biometrie

Autorisierung – Berechtigungen

- ▶ zugriffsbeschränkende Kriterien
 - ▶ Identität (Anonymous, Anwender, Administrator)
 - ▶ Zugriffsart (lesen, schreiben, ausführen, löschen, entschlüsseln, ...)
 - ▶ Zugriffsobjekt (Datei, Betriebssystem, Hardwareressource, Dienst)
 - ▶ Ort, Zeit, Dauer
 - ▶ wer darf wann wo wie auf welches Objekt zugreifen?
- ▶ Zugriffsregulierung
 - ▶ objektgebunden (z.B. Access Control Lists – ACL)
 - ▶ aktionsgebunden (welcher Prozess darf wie auf welche Objekte zugreifen)
- ▶ Vertrauenswürdigkeit (Single Sign On)

Angriffsformen

- ▶ Spoofing (z.B.: IP, ARP, DNS)
- ▶ Sniffing
- ▶ Scanning (Dienste, Betriebssystem, Dateirechte)
- ▶ Hijacking
- ▶ Replay(Man in the middle)
- ▶ Denial of Service
- ▶ Böartiger Code: Viren, Trojaner, Würmer, Buffer Overflow, Mailbombe
- ▶ Verschleierung
 - ▶ Verdeckte Kanäle
 - ▶ Tunnel
 - ▶ Steganographie
- ▶ Logische Bomben
- ▶ Backdoor
- ▶ Generische Angriffe(falsche Parameter, undocumented features, ...)
- ▶ Ausnutzen menschlicher Schwächen (Unkenntnis, Naivität, Verführbarkeit, ...)

Elementarschutz

- ▶ physikalischer Schutz
 - ▶ bauliche Sicherung (Feuer, Wasser, Erschütterung, Einbruch)
 - ▶ Zugangssicherung (Hauptschalter, Medienzugang)
 - ▶ räumliche Trennung
- ▶ Backup/Recovery
- ▶ Redundanz/Hochverfügbarkeit
- ▶ Anwenderschulung und Sensibilisierung
- ▶ Testen und Prüfen

Abwehrmechanismen im Netz

- ▶ Callback
- ▶ Einsatz kryptografischer Methoden und Werkzeuge
 - ▶ SSH(putty) statt Telnet, SCP (WinSCP) statt FTP
 - ▶ SSL
- ▶ Virenschanner
- ▶ Sandboxing
- ▶ funktionale Trennung
 - ▶ separate Maschinen
 - ▶ virtuelle Maschinen
 - ▶ Proxies
 - ▶ Integritätscheck
- ▶ Honigtopf/Teergrube
- ▶ Einbruchserkennung (Intrusion Detection)
- ▶ Firewall

Firewall

- ▶ Aufgabe: sichern eines Teilnetzes nach außen
- ▶ Grundkonzepte
 - ▶ Paketfilter (auf verschiedenen Schichten)
 - ▶ Zustandsüberwachung (stateful inspection)
 - ▶ Application Gateway (Proxies)
 - ▶ minimal ausgestattet
 - ▶ überwacht
- ▶ kein Schutz gegen:
 - ▶ böswillige Insider (Personen/Prozesse)
 - ▶ verschlüsselte Inhalte
 - ▶ Übertragungen auf anderen Wegen (Datenträger, Modemstrecken)

Firewall – Implementierung

- ▶ Topologien
 - ▶ Router/Firewall
 - ▶ Bastion Host
 - ▶ DMZ (demilitarisierte Zone)
 - ▶ innere Firewall
 - ▶ Grenznetz
 - ▶ äußere Firewall
- ▶ Kombination mit z.B. NAT (Network Address Translation, Masquerading), DNS, Proxy

VPN (virtuelles privates Netzwerk)

- ▶ Nutzung gesicherter (verschlüsselter) logischer Tunnel durch WAN
- ▶ Sicherung von Funknetzen
- ▶ Anbindung mobiler Systeme (Road-Warrior)
- ▶ Realisierung auf oder oberhalb der Vermittlungsschicht (z.B. IPSec)

Kommunikationssicherheit – IPSec Konzepte

- ▶ optional in IPv4 (z.B. zwischen Vermittlung/Transport), obligatorisch in IPv6
- ▶ Sicherheitsmerkmale:
 - ▶ Authentizität des Senders
 - ▶ Integrität (Schutz gegen Veränderung während der Übertragung)
 - ▶ Replay-Verhinderung
 - ▶ Vertraulichkeit (Verschlüsselung, alternative Methoden)
- ▶ IPSec-Modi
 - ▶ Transport(Host-Host): IP-Paket = IP-Header + IPSec-Header + Nutzdaten
 - ▶ Tunnel(Netz-Netz): IP-Paket = (neuer) IP-Header + IPSec-Header + (IP-Paket)

Kommunikationssicherheit – IPSec Implementation

- ▶ Paket-Manipulation:
 - ▶ Authentication Header (AH, Protokoll 51): Authentizität, Integrität, Antireplay
 - ▶ Encapsulating Security Payload(ESP; Protokoll 50): zusätzlich Vertraulichkeit
 - ▶ IP Compression (IPCOMP): Kompression
- ▶ Verbindungsverwaltung (Dienstprozess)
 - ▶ ISAKMP (IN Security Association and Key Mgmt. Protocol): Verbindungsverwaltung
 - ▶ IKE (Internet Key Exchange Protocol): Verbindung aufbauen und aufrecht erhalten
- ▶ Beispiele
 - ▶ Kame/Racoon(Unix), isakmpd (Openbsd), FreeS/Wan (Linux)
 - ▶ MS-IPSec