17. Sicherheit

Inhalt

- Grundlagen
  - Angriffsarten und Gegenmaßnahmen
  - Kryptographische Verfahren
- Countermeasures

17.2

17.1 Grundlagen

"Arsenal" der Angreifer

- Wer greift an?
  - Primär reguläre Benutzer
  - "Einbrecher" (Paßwort ermittelt)
- Bedrohungen
  - Unerlaubtes Mithören
  - Fälschen von Nachrichten und Programmen
  - Unerlaubte Nutzung von Ressourcen
  - Vandalismus
- Angriffsformen
  - Nachrichtenverkehr abhören (Eavesdropping)
  - Vorspiegelung falscher Tatsachen (Masquerading)
  - Nachrichten im Verlauf der Übertragung fälschen (Message Tampering)
    - Store-and-Forward-Netze
    - Man in the Middle
  - Nachrichten speichern und später wieder einspielen (Replay)
    - Verschlüsseln allein hilft nicht

17.4

... und der Verteidiger

- Vertrauenswürdige Basis (Trusted Base)
- Sicherheitsanforderungen an Client/Server-Systeme
  - Kommunikationskanäle sichern
    - auch Speicherbereiche auf einem Rechner
    - verhindert Abhören
  - Client und Server müssen sich wechselseitig mißtrauen
    - Authentifizierung
    - Server: Ist Client rechtmäßiger Vertreter seines Benutzers?
    - Client: Ist Server authentisch?
  - Nachrichten mit integriertem Verfallsdatum
    - verhindert u.a. Replay

17.5

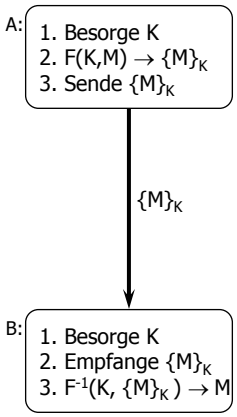
Konzepte und Techniken

- Verschlüsselungstechniken (Kryptographie)
  - Verschlüsseln von Nachrichten
  - Grundlage für Authentifizierung
    - Wer eine Nachricht mit einem bestimmten Schlüssel erfolgreich entschlüsselt ist authentisch
  - Digitale Signaturen (authentische Unterschriften)
- Echtheit beglaubigen (Authentifizierung)
  - Ein Geheimnis, daß nur Einer kennen kann, identifiziert ihn
  - Authentifizierungsdienst
- Zugriff kontrollieren (Schutz)
  - Objekt-basiert
  - Subjekt-basiert

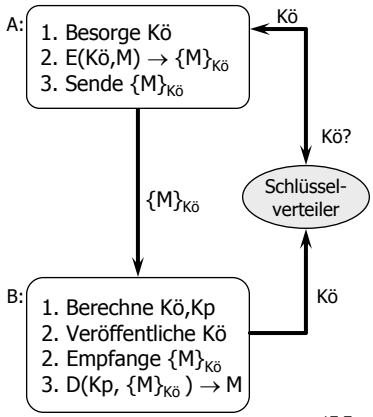
17.6

Kryptographie

- Geheime Schlüssel



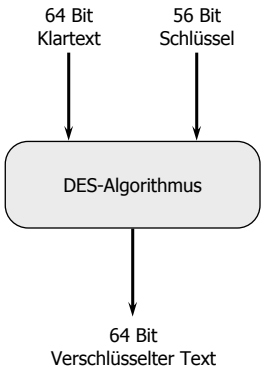
- Öffentliche Schlüssel



17.7

Geheime Schlüsselverfahren

- Data Encryption Standard (DES)
- $F=F^{-1}$
- 16 schlüsselabhängige Runden (Rounds)
  - Bit-Rotationen
- 3 schlüsselunabhängige Transpositionen
- spezielle DES-Chips
  - Faktor 1000 schneller als öffentliche Schlüsselverfahren
- siehe z.B. Tanenbaum, *Computer Networks*, 2. Auflage, 1996



17.8

Öffentliche Schlüsselverfahren

P = 9484758362143462738472734648389448928374783928347920048238479237282717262623627223487283482347238428349823472983883
Q = 3458374593875240294203723472987439845730958098538752987429874203482-48304573984573952929649276428764954938573983331 1

- RSA von Rivest, Shamir und Adelman (1978)
- Ansatz: Faktorisierung sehr großer Zahlen aufwendig
- Falltürfunktion f(x)=y
  - Umkehrfunktion schwierig zu berechnen
- Berechnung von Kö und Kp:
  - Wähle Primzahlen P und Q (beide größer als 10100)
  - N := P*Q
    Z := (P-1)*(Q-1)
  - d := Zahl relativ prim zu Z
  - e := e*d = 1 mod Z
  - Kö := <e,N> und Kp := <d,N>
- Beispiel
  - P = 13
    Q = 17
  - N := 13*17 = 221
    Z := 12*16 = 192
  - d := 5
  - e*5 = 1 mod 192

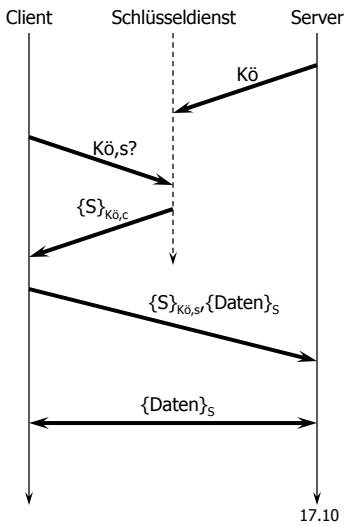
    1 mod 192 = 193, 385, 577, ...

    e := 385/5 = 77
  - Kodiert werden k Bits (2k < N)
  - E(e,N,M) = Me mod N
    D(d,N,C) = Cd mod N

17.9

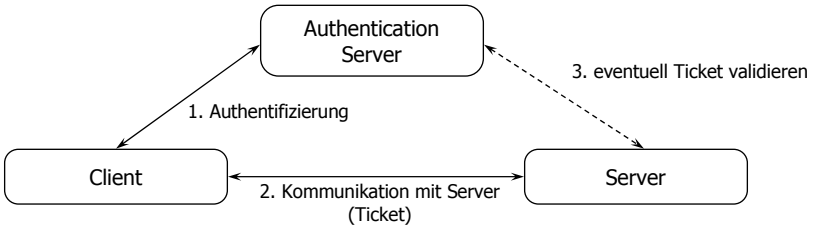
Kombinierte Techniken

- Geheime Schlüssel
  - Austausch des gemeinsamen Schlüssels aufwendig
  - Verschlüsselung großer Datenmengen effizient möglich (Hardware-Unterstützung)
- Öffentliche Schlüssel
  - Bekanntgabe und Weitergabe des öffentlichen Schlüssels unkritisch
  - Verschlüsselung zeitaufwendig
- Kombination
  - Mit Hilfe öffentlicher Schlüssel wird ein geheimer Sitzungsschlüssel (Session Key) ausgetauscht
  - Verschlüsselung der Daten mit Session Key



17.10

Authentifizierung



- Authentifizierungsdienst besonders schützen
  - Keine Benutzerprozesse
  - Firewall
  - Besonderer physischer Schutz
- Needham und Schröder, 1978
  - Authentifizierung mit geheimen Schlüsseln
  - Authentifizierung mit öffentlichen Schlüsseln

17.11

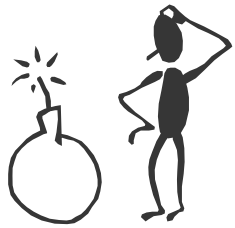
Authentifizierung mit geheimen Schlüsseln

- $C \rightarrow AS$: C, S, N_C
  - Client C möchte mit Server S kommunizieren
  - N_C = Frischedatum
- $AS \rightarrow C$: $\{N_C, S, KSession, \{KSession, C\}_{KS}\}_{KC}$
  - Authentifizierungsdienst
    - Sitzungsschlüssel KSession
    - Ticket $\{KSession, C\}_{KS}$
- $C \rightarrow S$: $\{KSession, C\}_{KS}$
- $S \rightarrow C$: $\{N_S\}_{Ksession}$
  - Ist Client authentisch oder wiederholt er nur eine Nachricht
- $C \rightarrow S$: $\{N_S + 1\}_{Ksession}$

17.12

Schwachstelle

- C → S: {KSession, C }$_{KS}$
  - Woher weiß der Server, daß empfangenes Ticket noch frisch ist?
- Man will vermeiden, daß Tickets eine beliebig lange Lebensdauer haben
  - Server steht Client nur zeitlich begrenzt zur Verfügung
  - Client wird das Nutzungsrecht zu einem späteren Zeitpunkt entzogen
- Lösung
  - Zeitstempel oder Verfallsdatum in Ticket integrieren:
    {Ksession, C, Verfallsdatum}$_{KS}$



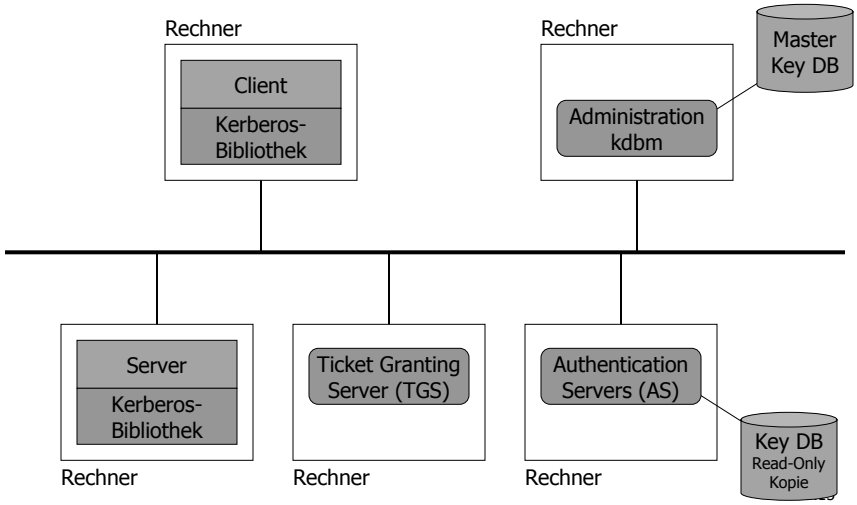
17.13

Kerberos (Version 4)

- Authentifizierungsprotokoll (Steiner et al., 1988)
- basiert auf dem Verfahren von Needham und Schröder
  - Geheime Schlüssel
  - Erweiterung um Zeitstempel
- Integriert u.a. in
  - UNIX (MIT)
  - AFS
  - OSF/DCE
- Kerberos-Ticket: {Ticket(C,S)}$_{PK(S)}$:= {C,S,t1,t2,Ksession}$_{PK(S)}$
  - Nur gültig im Zeitintervall [t1,t2]
- Authentication Server (AS)
- Ticket-Granting Server (TGS)

17.14

Komponenten von Kerberos

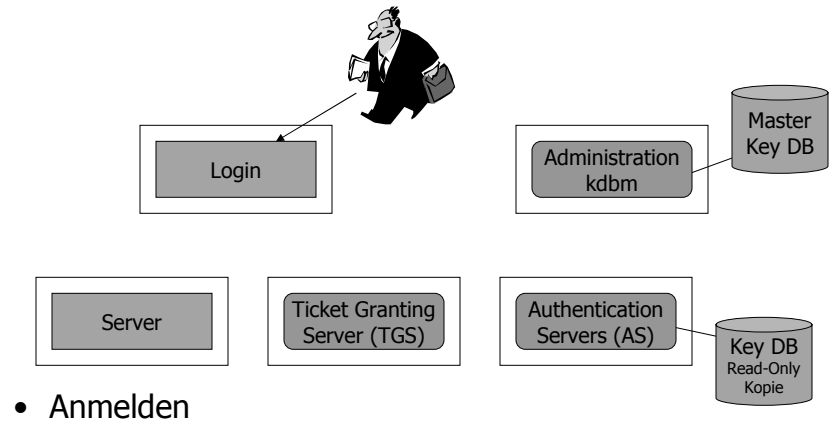


Kerberos-Namen

- 3 Komponenten
  - Benutzername (name)
  - Instanz (instance)
    - Verschiedene Sicherheitsstufen für einen Benutzer
    - Administratorrechte: root oder admin
    - Default: NULL
  - Realm
    - Schutz- und Authentifizierungsdomäne
- Beispiel
  - sturm@informatik.uni-trier.de
  - grutter.admin@informatik.uni-trier.de

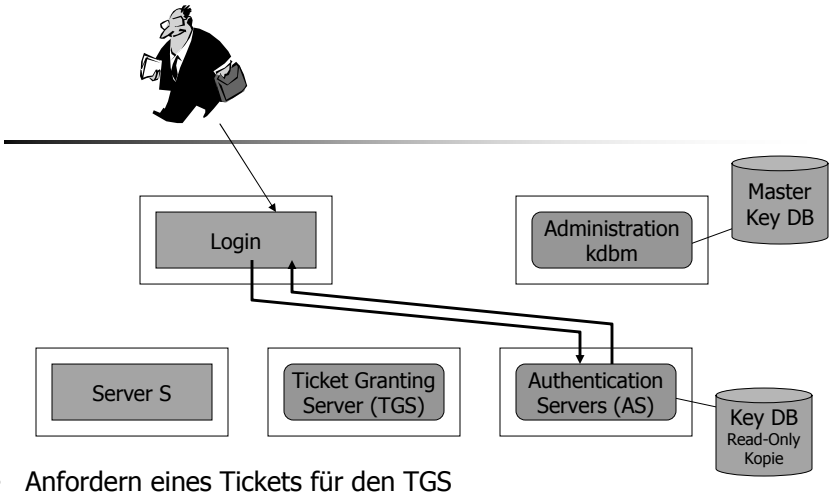
17.16

1. Authentifizierung



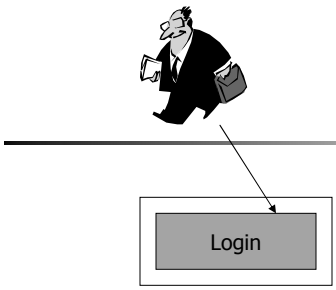
- Anmelden
  - Eingabe des Benutzernamens

17.17



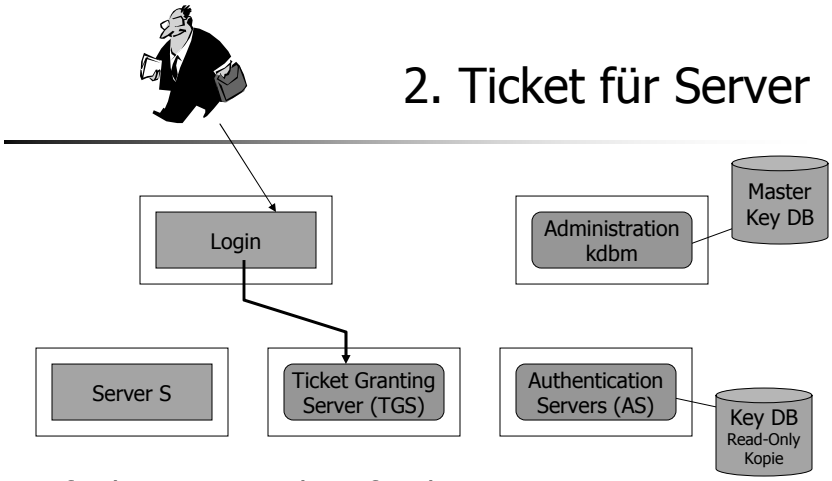
- Anfordern eines Tickets für den TGS
  - Client, TGS
- Antwort von AS
  - $\{ Session_{Client,TGS}, \{T_{TGS}\}K_{TGS} \}K_{client}$
  - T_{TGS} = (TGS, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,TGS}$)

17.18



- Client empfängt
  - $\{ Session_{Client,TGS}, \{T_{TGS}\}K_{TGS} \}K_{client}$
- Eingabe des Passwords
- Nur der wirkliche „User" kann entschlüssen
  - Session-Key zwischen Client und TGS
  - Ticket für TGS
- Beachte: Password wurde nicht ausgetauscht
- Ticket längere Zeit nutzbar (Zeitstempel)

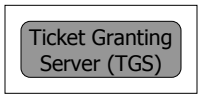
17.19



- Anfordern eines Tickets für den Server S
  - $S, \{T_{TGS}\}K_{TGS}, \{A_{Client}\}Session_{Client,TGS}$
- Authentizitätsnachweis A_{client}
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)

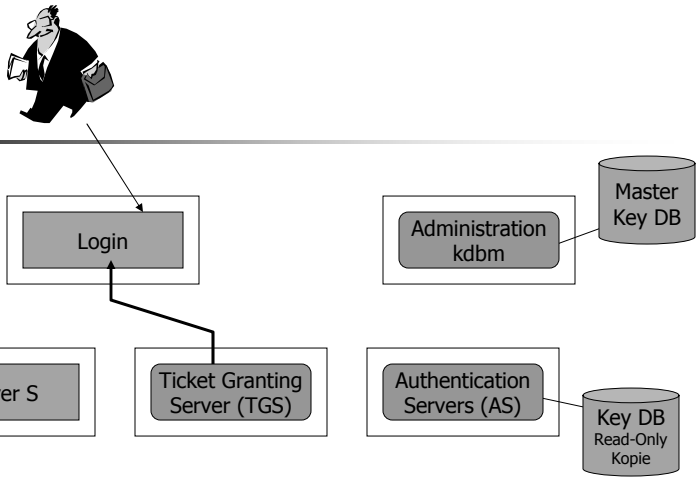
17.20

- TGS empfängt
  - S, $\{T_{TGS}\}K_{TGS}$, $\{A_{Client}\}Session_{Client,TGS}$



- Ticket öffnen
  - T_{TGS} = (TGS, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,TGS}$)
- Authentizität überprüfen
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)
- Zeitstempel darf nur wenige Minuten alt sein
  - Synchronisierte Uhren

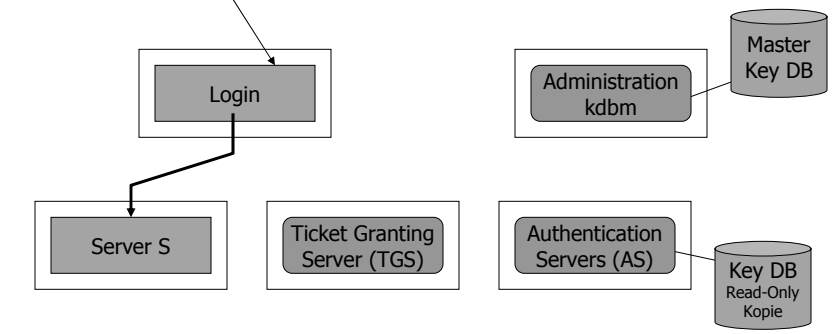
17.21



- Annahme: Client darf Server benutzen
- TGS sendet Ticket für Server S an Client
  - $\{ \{T_{Client,S}\}K_S, Session_{Client,S} \}Session_{Client,TGS}$

17.22

3. Auftrag an Server

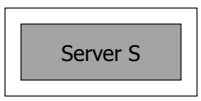


- Client wendet sich an Server S
  - $\{ A_{Client} \}Session_{Client,S}, \{ T_{Client,S} \}K_S$
- Erneut Übertragung eines Authentizitätsnachweises
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)

17.23

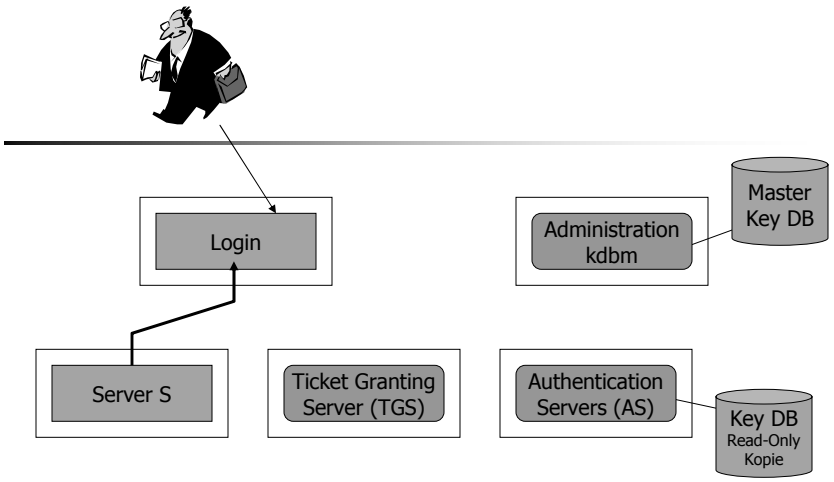
3. Auftrag an Server

- S empfängt
  - $\{ A_{Client} \}Session_{Client,S}, \{ T_{Client,S} \}K_S$



- Ticket öffnen
  - $T_{client,S}$ = (S, Client, Client-Addresse, Timestamp, TTL, $Session_{Client,S}$)
- Authentizität überprüfen
  - A_{Client} = (Client, Client-Adresse, Zeitstempel)
- Zeitstempel überprüfen
- Services ggf. ausführen

17.24



- Client will Authentizität des Servers
  - Challenge/Response
- Server sendet an Client
  - { Zeitstempel+1 }$Session_{Client,S}$

17.25

Ändern eines Passwords

- Client wendet sich AS
  - Eingabe des alten Passwords
  - AS sendet ein Ticket für kdbm an Client
  - Client sendet neues Password an kdbm
    - Verschlüsselt?
- Kdbm
  - Instanz = NULL
    - Nur eigene Passwords änderbar
  - Instanz != NULL
    - Durchsuchen einer Berechtigungsliste
    - Ggf. Password ändern
- Logging aller Vorgänge
- Analog Eintrag neuer Clients und Löschungen

17.26

Digitale Signaturen

- Wie kann man sicher sein, daß eine Nachricht von X tatsächlich von X kommt?
- Eine Lösung
  - Öffentliche Schlüssel
  - Einschalten eines vertrauenswürdigen Notars N:

    1. A $\rightarrow$ B: M, A, $\{M\}_{SK(A)}$
    2. B $\rightarrow$ N: A
    3. N $\rightarrow$ B: A, PK(A)

  - Braucht nicht gesamte Nachricht nochmal verschlüsseln (MD5, ...)
- Anwendungen
  - Email, z.B. Privacy Enhanced Mail (PEM)
  - WWW
  - Allgemeine digitale Signaturen, z.B. Pretty Good Privacy (PGP)
  - Sichere Capabilities

17.27

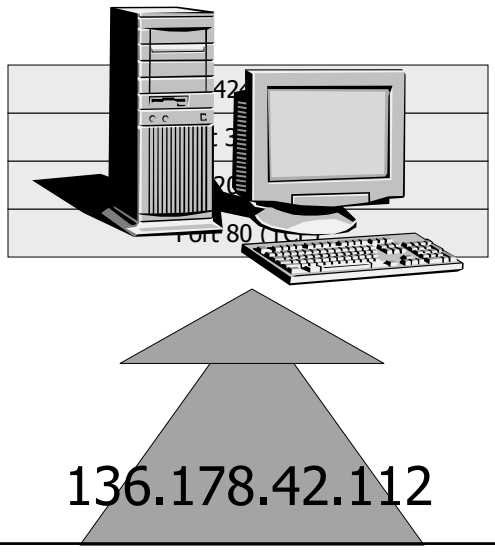
17.2 Countermeasures

Angriff!

- Einbruch
  - Unerlaubte Nutzung der vorhandenen Ressourcen
  - Unterschiedliche Konsequenzen (Harmlos bis Vandalismus)
- Denial of Service
  - Erlaubte Nutzung eines Systems verhindern
  - Überfluten mit Aufträgen
  - DDoS (Distributed DoS) besonders kritisch
  - Cyberwar
- Diebstahl von Information
  - Spionage

17.29

Locked on Target!



17.30

Attacke: Packet Interception

- Nachrichten beobachten: Sniffing
  - Mit root-Rechten auf Linux einfach
- Informationsgewinn
  - Gutartig: Netz beobachten
- Möglichkeiten stark netzabhängig
  - Ethernet-Bus lädt dazu ein
  - Ethernet-Switch: Nur notwendiger Verkehr auf Link
  - Ethernet-Switch mit Virtual LAN

17.31

Attacke: Port Scanning

- Suche nach geöffneten Ports
  - Nachricht an UDP-Port senden
  - Verbindung zu TCP-Port versuchen
    - Half Open Scan (SYN ohne ACK, SYN oder RST zurück)
    - Andere Flag-Kombinationen
    - Christmas Tree oder Null (u.U. Crash)
- Simple Attacken einfach erkennbar
- Vielfältige Möglichkeiten
  - Server Shutdown, Crash, Inkonsistenzen, …

Attacke: IP Spoofing

- Inkorrekte Senderadresse
  - Empfänger sendet Nachricht nicht an Angreifer
- Angreifer kann Antwort abfangen
  - Fortführen der Kommunikation
  - Hijacking Attack
- Angreifer braucht Antwort nicht
  - DoS gegen Empfänger
- Angreifer will, daß Dritter Antwort erhält
  - smurf Attack
  - Empfänger = Sender $\Rightarrow$ Probleme im Empfänger

17.33

Schutzmöglichkeiten

- Kein Schutz
- Security by Obscurity
- Rechnerschutz (Host Security)
  - Rechner schützt sich selbst
  - „Harden", Bastion Host
- Netzschutz (Network Security)
  - Spezielle Rechner/Netze schützen ein ganzes Netz
  - Firewalls

17.34

Security-Paradigmen

Default Deny Stance

Was nicht explizit erlaubt ist, ist verboten.

- Fail-Safe-Ansatz
- Schrittweise notwendiges zulassen
- Nutzer merken häufiger die Behinderung
- Neues muß explizit freigeschaltet werden.

Default Permit Stance

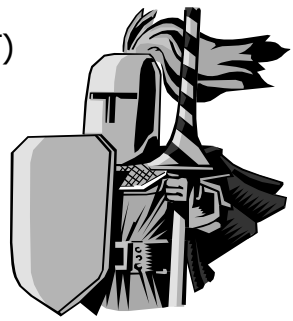
Was nicht explizit verboten ist, ist erlaubt.

- Nutzer beobachten seltener Behinderungen
- Neues funktioniert auf Anhieb

17.35

Schutztechniken

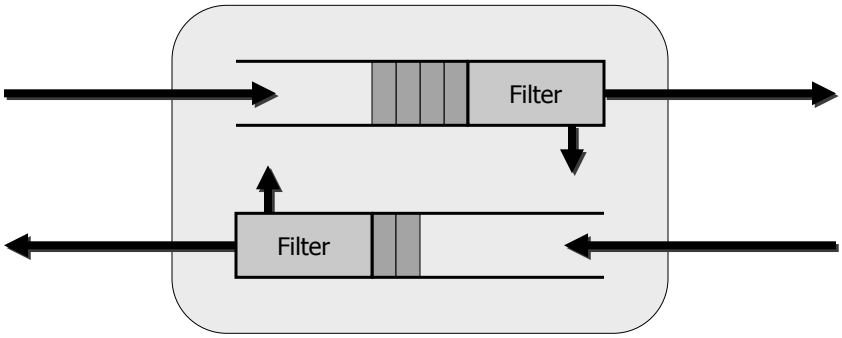
- Packet Filter
- Proxy Server
- Network Address Translation (NAT)
- Virtual Private Networks (VPN)



17.36

Packet Filter

- Selektives Weiterleiten von Paketen



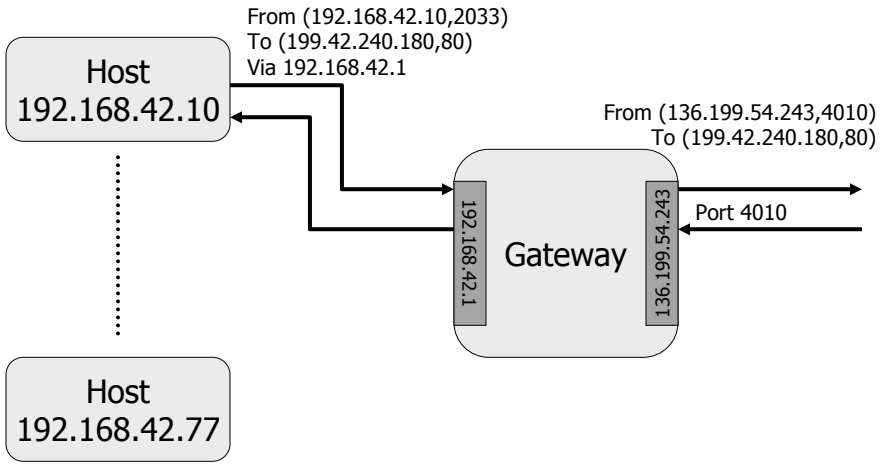
17.37

Filterinformationen

- Informationen in den Headern
  - IP Source Address
  - IP Destination Address
  - Protocol
  - TCP bzw. UDP Source Port
  - TCP bzw. UDP Destination Port
  - ICMP Message Type
  - Paketgröße
  - Flags
- Zusätzliche Informationen
  - Eingehendes Interface
  - Ausgehendes Interface
- Filterzustand
  - Antwort auf vorheriges Paket?
  - Anzahl bzw. Frequenz von Pakten mit bestimmten Eigenschaften
  - Identisches Paket (Replay)
  - Fragmentierung

17.38

Network Address Translation (NAT)



17.39

Variationen

- Festes Host-Mapping
  - 1 Externe Adresse pro interne Adresse
  - Wechsel von ungültigen intern verwendeten IP-Adressen
- Dynamisches Host-Mapping
  - Wahl einer IP-Adresse aus einem Pool bei Bedarf
- Festes Host-Mapping, Dynamisches Port-Mapping
  - Extremfall: IP des Gateways für alle inneren Rechner
  - z.B. Internet Connection Sharing bei Windows
- Dynamische Host- und Port-Mapping

17.40

Vor- und Nachteile

Vorteile

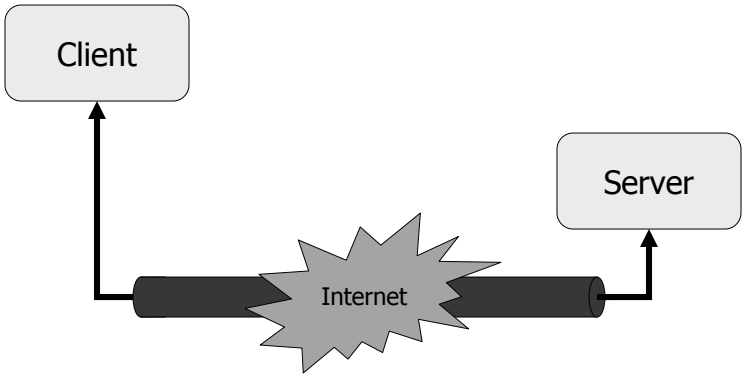
- Ungültige innere Adressen erzwingen Kontrolle über ausgehenden Verkehr
- Einschränkungen beim eingehenden Verkehr (nur für bekannte Ports)
- Innere Netzstruktur bleibt unsichtbar

Nachteile

- Zustand ist notwendig, insbesondere bei UDP
- Manche Protokolle betten IP-Adressen in den Nutzdaten ein (z.B. ftp)
- Kryptosysteme interpretieren NAT-Pakete ggf. als verändert
- Logging und Packet Filtering wird ggf. erschwert

17.41

Virtual Private Networks (VPN)



- Verschlüsselter Datentunnel

17.42

Vor- und Nachteile

Vorteile

- Gesamter Verkehr wird verschlüsselt
- Entfernte Nutzung von schwer abzusichernden Protokollen

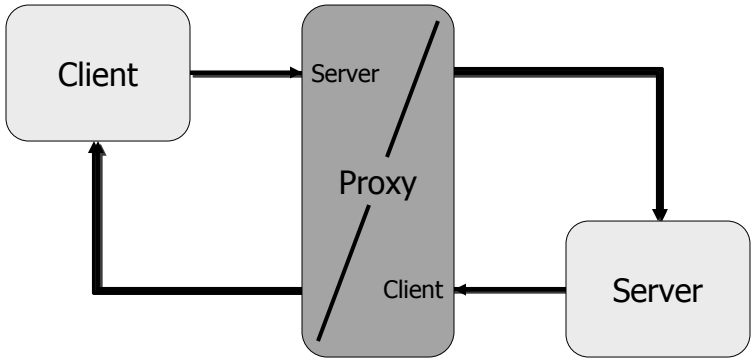
Nachteile

- Beteiligte Rechner sind am Netz und damit angreifbar
- Entfernter Teil wird integraler Bestandteil des Netzes

17.43

Proxy Server

- Empfang und Weiterleitung von Nachrichten



17.44

Vor- und Nachteile

Vorteile

- Proxies können sehr gut Loggen
- Leistungssteigerung durch Caching
- Intelligentes Filtern
  - Application-Level Proxy
- User-Level Authentifizierung
- Maximale Isolation gegen Fehler in tieferen Schichten

Nachteile

- Man muß Proxy für einen Service u.U. erst finden
- Zusätzliche Software
- Transparenz nicht immer möglich

17.45

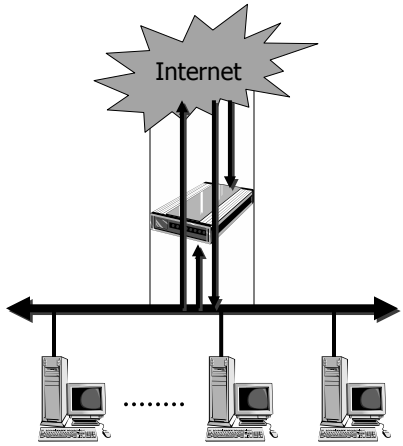
Schutzarchitekturen

- Single Box
  - Screening Router
  - Dual-Homed Host
- Screened Host Architecture
- Screened Subnet
  - Perimeter Network
- Multiple Screened Subnets
  - Split-Screend Subnet
  - Independet Screend Subnets
- Variationen

17.46

Screening Router

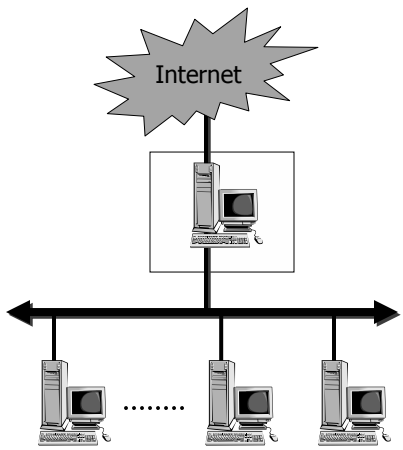
- Spezieller Router
  - Low-Cost
  - Unflexibel
  - Stateful ist kompliziert
  - Wenig Eingriffsmöglichkeiten
- Anwendbar bei
  - Kleinen Netzen
  - Hohe Host Security bereits vorhanden
  - Simple Dienst-Provider
- Produkte
  - z.B. spezielle DSL-Router



17.47

Dual-Homed Host

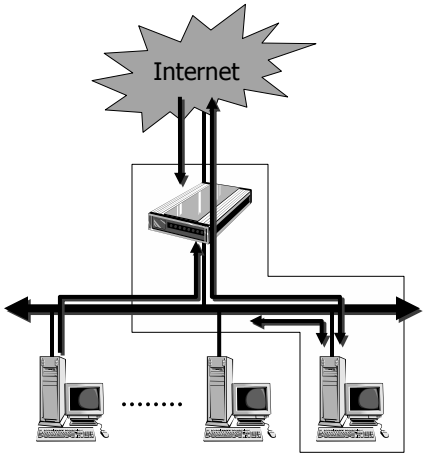
- Höhere Flexibilität
  - Einsatz als Packet Filter
  - Proxy Server
    - Outbound okay
    - Inbound kritisch
  - Stateful möglich
- Nachteile
  - Performance Overhead
  - Ausfallwahrscheinlichkeit
  - Komplexe Software



17.48

Screened Host

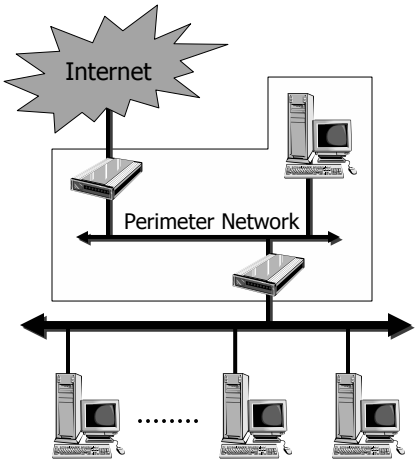
- Zweistufiges Konzept
- Hauptschutz durch Packet Filter
  - Durchlässig nur für Bastion Host
  - Gut absicherbar
- Bastion Host
  - Höhere Policies
- Nachteil
  - Einbruch in Bastion Host ist katastrophal



17.49

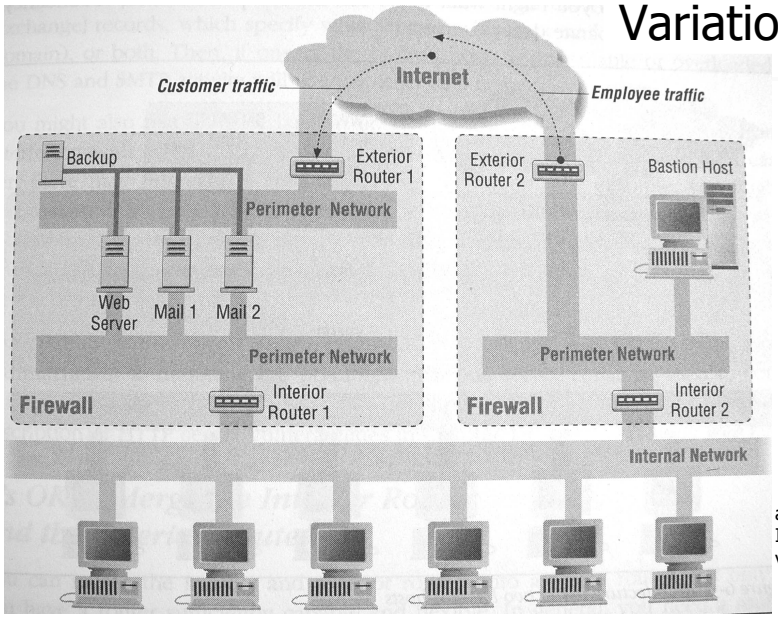
Screened Subnet

- Doppelte Sicherheit
- Bastion Host
  - Ziel für eingehenden Verkehr
  - Proxy für Outbound
- Exterior Router
  - Meist wenig Restriktionen für ausgehende Nachrichten
  - Erkennt „forged source addresses"
- Interior Router
  - Minimale Öffnung schützt inneres Netz
  - Outbound Filter



17.50

Variationen



aus "Building Internet Firewalls" von Zwicky et al.

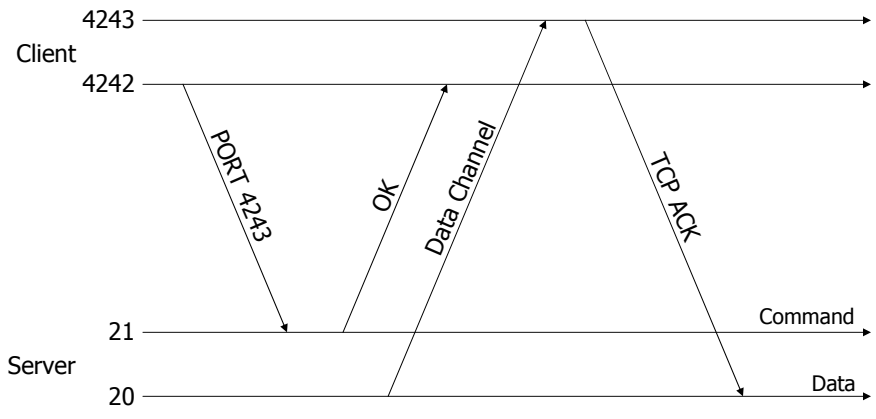
17.51

Paradebeispiel FTP

- FTP ist recht widerspenstig
- Zwei TCP-Verbindungen
  - Command Channel (Server: Port 21, Client: Port > 1023)
  - Data Channel
- Zwei Modi bzgl. Aufbau des Datenkanals
  - Normal Mode
    - Server: Port 20, Client: Port > 1023
  - Passive Mode
    - Server: Port > 1023, Client: Port > 1023
    - Besser, aber nicht alle Clients beherrschen ihn

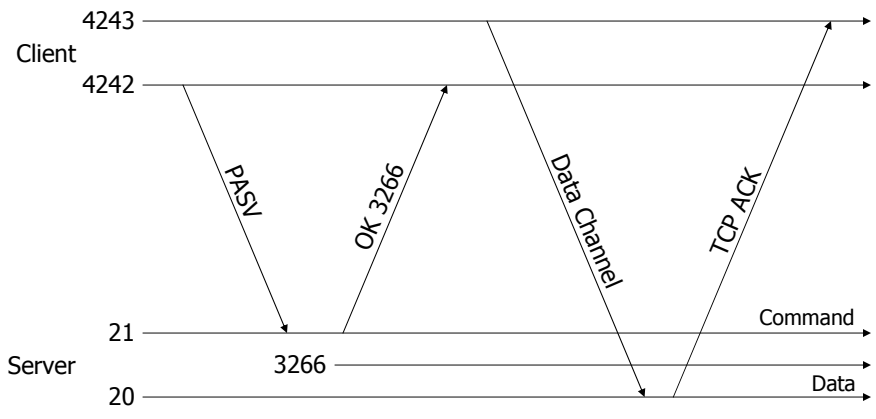
17.52

Ablauf: Normal Mode



17.53

Ablauf: Passive Mode



17.54

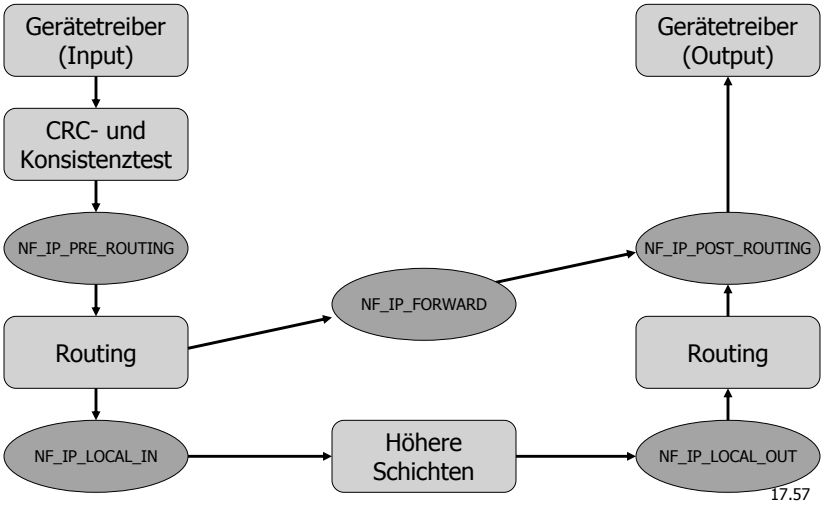
Folgen für

- Packet Filtering FTP
  - Problem Normal Mode: Server öffnet Verbindung zum Client
  - Meisten Filter erkennen PORT-Kommando in den Daten
- Proxying FTP
  - Vergleichsweise einfach möglich
- NAT FTP
  - IP-Adressen und Portnummern in den Daten enthalten
  - Modifizierendes NAT

17.55

17.3 Linux als Firewall

Linux als Packet Filter: Netfilter



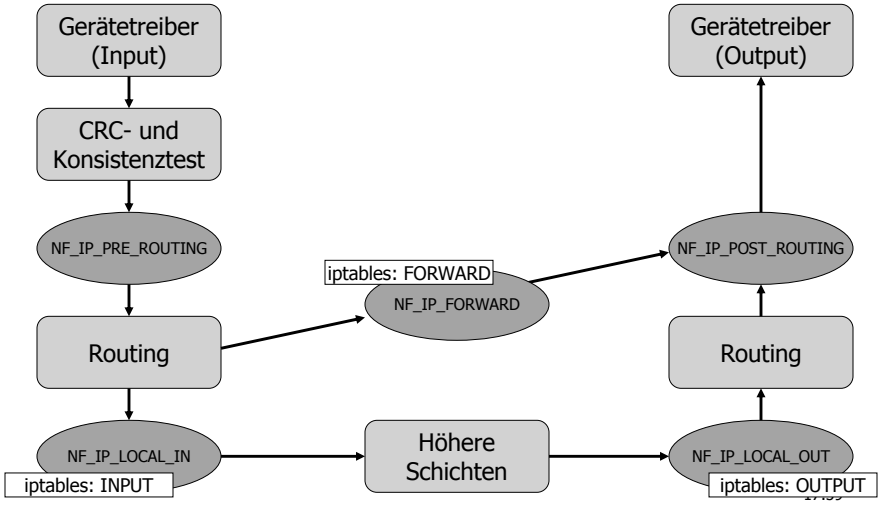
17.57

Netfilter Hooks

- Verankerungspunkte für Filtercode
- PRE_ROUTING
  - Frühestmögliches Abfangen
  - Beispiel: DoS erkennen, Abrechnung, …
- LOCAL_IN, LOCAL_OUT
  - Ein- und ausgehende Pakete des lokalen Rechners
- FORWARD
  - Weiterzuleitende Pakete
- POST_ROUTING
  - Beispiel: Abrechnung, …

17.58

Linux als Packet Filter: iptables

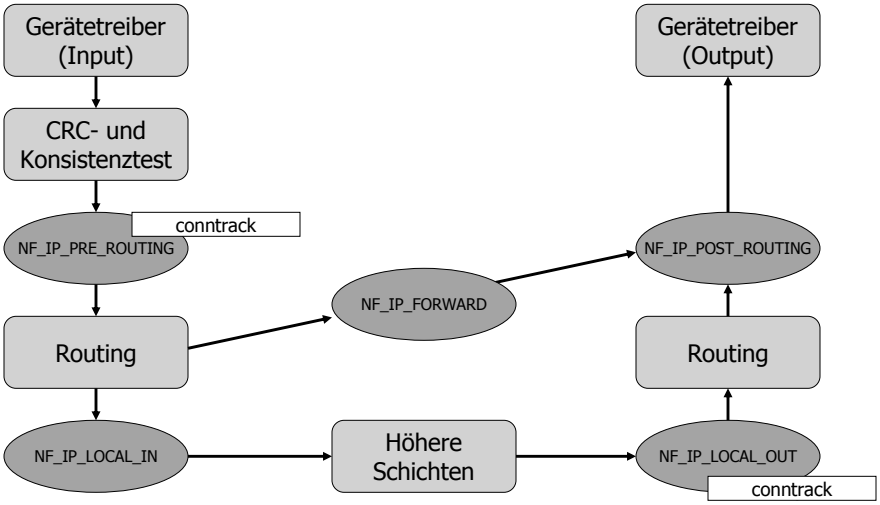


Netfilter-Module

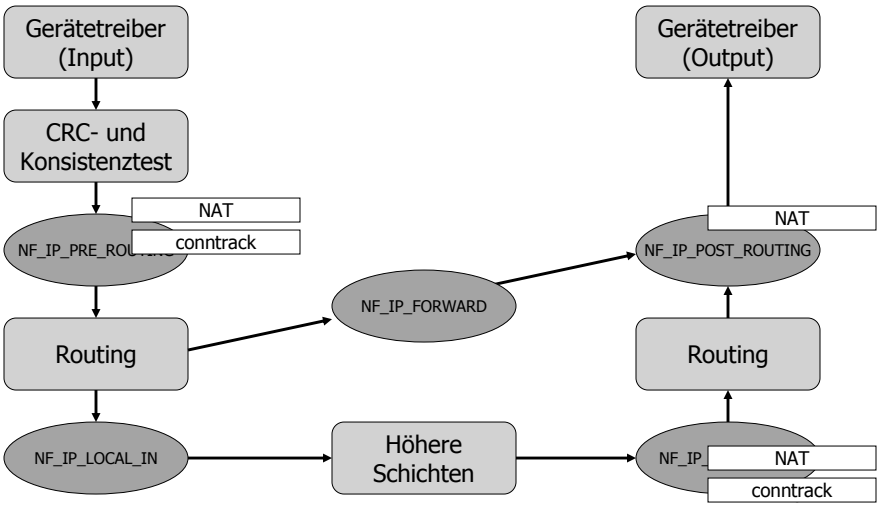
- ip_tables
  - Standardmodul für INPUT, OUTPUT und FORWARD
- ip_conntrack (Connection Tracking)
  - Stateful filter
- ip_conntrack_ftp
- iptable_nat, ipt_MASQUERADE
  - Network Address Translation

17.60

Connection Tracking



NAT



iptables: Syntax

- Ein einfaches Beispiel:

  iptables --table filter --flush // Standardfilter löschen
  iptables --policy INPUT DROP // Default Deny Stance
  iptables --policy FORWARD DROP // s.o.
  iptables --policy OUTPUT DROP // s.o.
  iptables -A OUTPUT -p tcp --sport 1024: --dport 80 -j ACCEPT
    - Ausgehende WWW-Requests von Ports größer oder gleich 1024 an Remote-Port 80 akzeptieren

  …
- Im Detail ganz schön kniffelig aber spannend

17.63

Linux als Proxy

- Dedizierte Proxy-Server
  - SQUID: HTTP-Proxy inklusive Cache
- Proxy Package
  - SOCKS, TIS Firewall Toolkit
  - Microsoft Proxy Server

17.64

squid

- Beispiel-Konfiguration

```
# NETWORK OPTIONS
# ------------------------------------------------------
http_port 192.168.42.10:3128
icp_port 0
htcp_port 0


# OPTIONS WHICH AFFECT THE CACHE SIZE
# ------------------------------------------------------
cache_mem 32 MB
cache_swap_low  90
cache_swap_high 95
maximum_object_size 32 MB

# LOGFILE PATHNAMES AND CACHE DIRECTORIES
# ------------------------------------------------------
cache_dir ufs /var/squid/cache 512 16 256
…
```

17.65

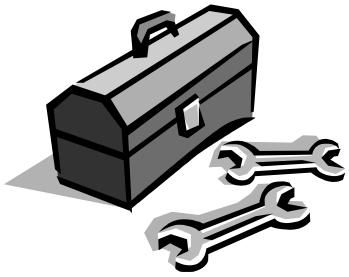
SOCKS

- Bestandteile
  - SOCKS Server
  - SOCKS Client Library
  - SOCKs-ified Versionen von FTP etc.
  - SOCKS Wrapper für ping und traceroute
  - runsocks: SOCKS für dynamisch gebundene Programme ohne Neuübersetzung
- Client auf SOCKS vorbereiten
  - ersetzen der Netzwerkfunktion **`f()`** durch **`Rf()`**
  - f = **`connect`**, **`getsockname`**, **`bind`**, **`accept`**, **`listen`**, **`select`**

17.66

Tools

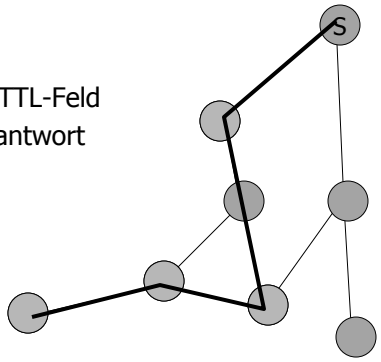
- Konfigurationstools
  - ifconfig, …
- Statistiken
  - netstat, ntop, …
- Abfragen
  - traceroute, nslookup, …
- Sniffer
  - ethereal, tcpdump, …



17.67

Tool: traceroute

- Weg im Internet von einem Sender zu einem Empfänger
- Strategie
  - Nachricht mit wachsendem TTL-Feld
  - Auswertung der ICMP-Rückantwort
- Inkonsistente Sicht möglich



17.68

Beispiel

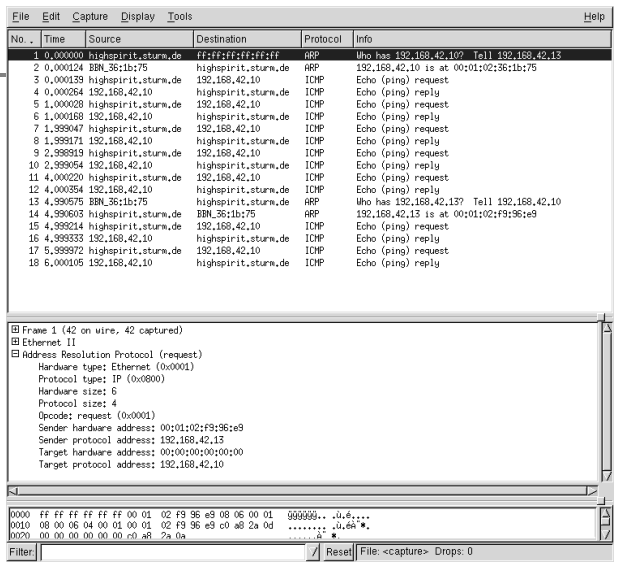
- Von Konz nach Trier:

```
gateway:~ # traceroute tamdhu.uni-trier.de
traceroute to tamdhu.uni-trier.de (136.199.54.243),
     30 hops max, 40 byte packets

1  ffm2-d1-2.mcbone.net (62.104.212.33)  148 ms  149 ms  150 ms
2  G3-0-4.ffm4-gsr.mcbone.net (62.104.212.5)  149 ms  160 ms  150 ms
3  ir-frankfurt2.g-win.dfn.de (80.81.192.222)  159 ms  150 ms  150 ms
4  cr-frankfurt1.g-win.dfn.de (188.1.80.37)  159 ms  150 ms  150 ms
5  cr-muenchen1.g-win.dfn.de (188.1.18.82)  169 ms  169 ms  170 ms
6  cr-stuttgart1.g-win.dfn.de (188.1.18.130)  169 ms  160 ms  160 ms
7  ar-kaiserslautern1.g-win.dfn.de (188.1.76.34)  159 ms  160 ms  160 ms
8  ar-kaiserslautern2.g-win.dfn.de (188.1.77.194)  159 ms  160 ms  160 ms
9  vxr-serial1-0.uni-trier.de (136.199.1.1)  169 ms  160 ms  160 ms
10  sw3rsm-extern.uni-trier.de (136.199.224.226)  169 ms  170 ms  170 ms
11  cisco-224.uni-trier.de (136.199.224.1)  169 ms  170 ms  170 ms
12  tamdhu.uni-trier.de (136.199.54.243)  169 ms  169 ms  170 ms
```

17.69

Tool: ethereal

- Packet Sniffer
- Capture
- Offline Analyse
  - Mehrere Detailstufen

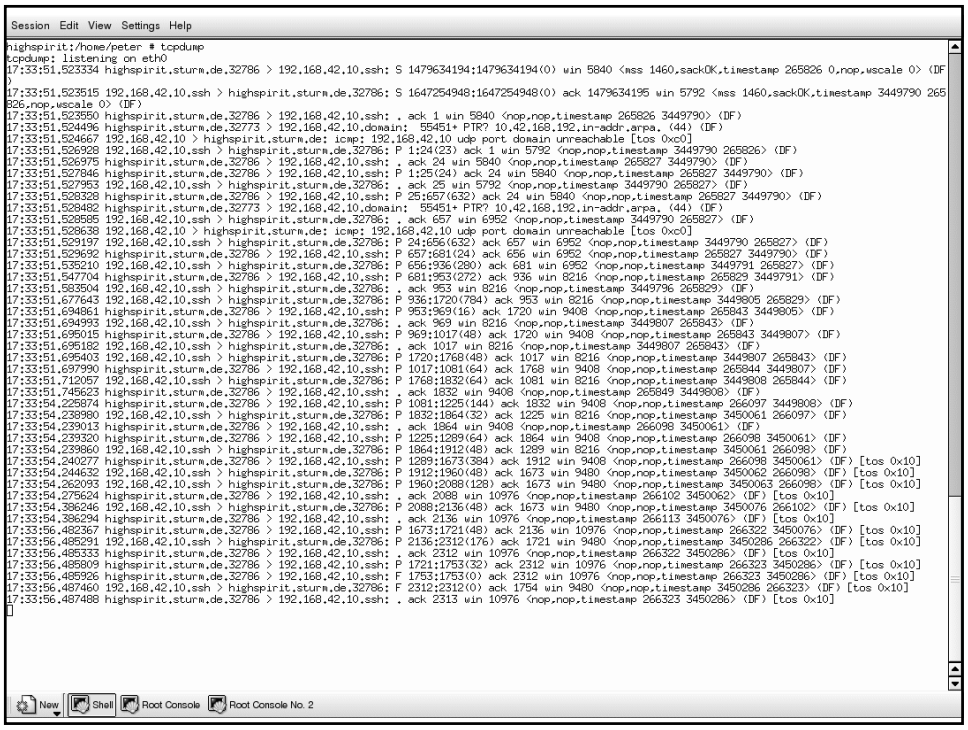


17.70

Tool: tcpdump

- Kommandozeilen-Sniffer
- Konfiguration des Beobachtungsmodus
  - Beschränken auf bestimmte Netze, Hosts, Ports
  - Richtung (src oder dest)
  - Protokoll
  - Ethernet
  - Gateway (Netzempfänger ungleich Empfänger)
  - Broadcast, Multicast
  - …
- Ausgabe ist z.B. über Perl-Skripte auswertbar

17.71



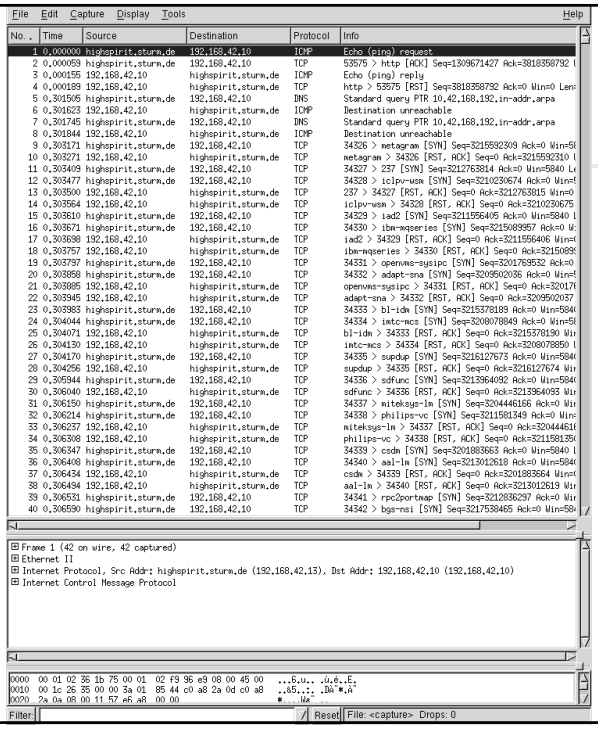
Tool: nmap

- Port Scanner
  - UDP
  - TCP
- u.U. lange Laufzeit
- Vorsicht!

```
highspirit:/home/nfs_export/suse_8.0/suse # nmap -sU 192.168.42.10

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
caught SIGINT signal, cleaning up
highspirit:/home/nfs_export/suse_8.0/suse # nmap -sT 192.168.42.10

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.42.10):
(The 1543 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
3128/tcp   open        squid-http


Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
highspirit:/home/nfs_export/suse_8.0/suse # nmap -sU 192.168.42.10

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.42.10):
(The 1450 ports scanned but not shown below are in state: closed)
Port       State       Service
111/udp    open        sunrpc
32770/udp  open        sometimes-rpc4


Nmap run completed -- 1 IP address (1 host up) scanned in 1470 seconds
highspirit:/home/nfs_export/suse_8.0/suse #
```

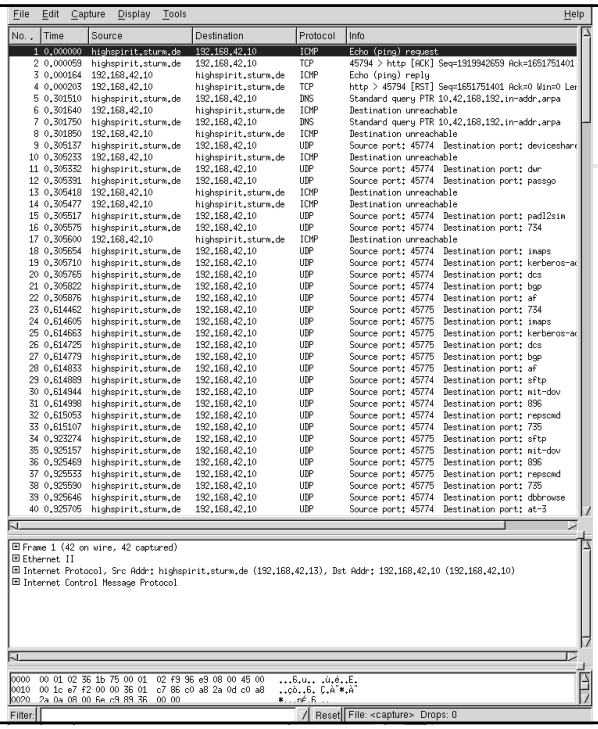
17.73

nmap: TCP-Scan



17.74

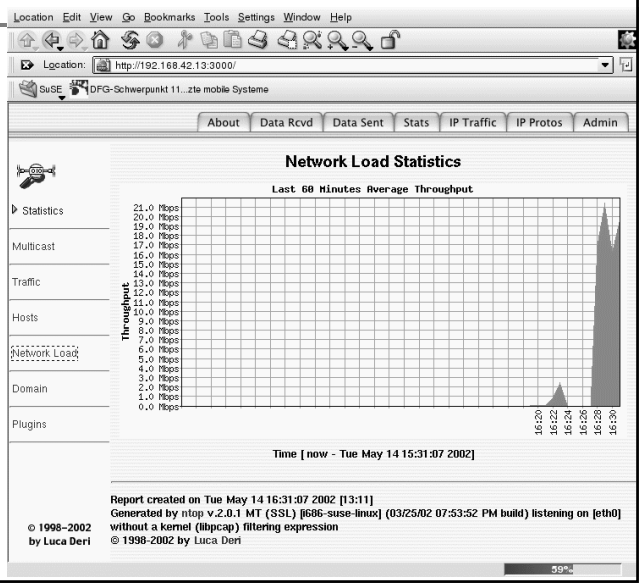
nmap: UDP-Scan



17.75

Tool: ntop

- Netzwerk-Top
- Zwei Modi
  - HTTP
  - Interaktiv



Literatur

- D.E. Comer, D.L. Stevens
  Internetworking with TCP/IP, Volume I-III
  Prentice-Hall, 1993
- W. Stevens
  TCP/IP Illustrated, Volume 1: The Protocols (1994)
  TCP/IP Illustrated, Volume 3: T/TCP, HTTP, ... (1996)
  Addison-Wesley
- G. Wright, W. Stevens
  TCP/IP Illustrated, Volume 2: The Implementation
  Addison-Wesley, 1995
- E.D. Zwicky, S. Cooper, D.B. Chapman
  Building Internet Firewalls
  2. Auflage, O'Reilly, 2000
- Wehrle, Pählke, Ritter, Müller, Bechler
  Linux Netzwerkarchitektur
  Addison-Wesley, 2002

17.77