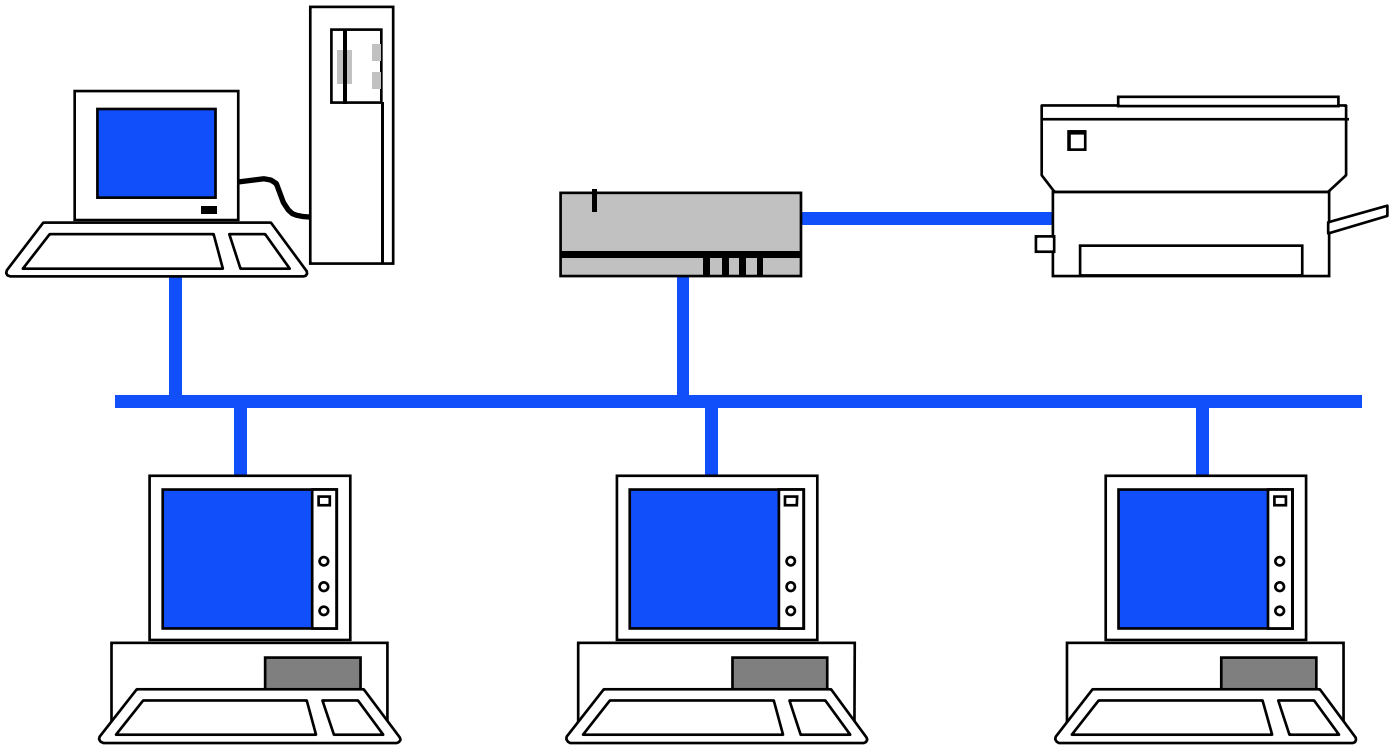
NVS1 B 7+8. Semester

Mag. Dr. Klaus Coufal



Übersicht

- Kompetenzbereiche
  - Netzwerkmanagement und -sicherheit
  - Betriebssystemnahe Programmierung
  - Architektur und Entwicklung verteilter Systeme

I. Netzwerkmanagement und -sicherheit

- Betrieb, Fehlersuche und Verfügbarkeit
- Sicherheitsrisiken und Komponenten von Sicherheitslösungen
- Security Policy und Sicherheitsverwaltung
- Lastverteilung und Performancetuning
- Verwaltungssysteme

II. Betriebssystemnahe Programmierung

- System Calls
- Shell Programmierung

III. Architektur und Entwicklung verteilter Systeme

- Synchronisation und Interprozesskommunikation
- Webapplikationen, Client-Server und Multi Tier Systeme
- Serviceorientierte Architektur (SOA)
- Cloud Computing
- Enterprise Application Architecture und Middleware

I.3. Gefährdungen

- Passive Angriffe
- Aktive Angriffe
- Zufällige Verfälschungen

Passive Angriffe

- Abhören der Teilnehmeridentitäten
  - Wer mit wem
- Abhören der Daten
  - Mißbrauch der Daten
- Verkehrflußanalyse
  - Größenordnungen, Zeitpunkte, Häufigkeit, Richtung des Datentransfers

Aktive Angriffe

- Wiederholung oder Verzögerung einer Information
- Einfügen oder Löschen bestimmter Daten
- Boykott des Informationssystems
- Modifikation der Daten
- Vortäuschung einer falschen Identität
- Leugnen einer Kommunikationsbeziehung

Zufällige Verfälschungsmöglichkeiten

- Fehlrouting von Information
  - Durch „Vermittlungsfehler“ in Knotenrechner
- Fehlbedienung
  - Löschen noch nicht versandter Informationen
  - Ausdrucken sensibler Daten

I.4. Sicherheitsdienste 1

- Aus den Gefährdungen können nun die notwendigen Sicherheitsdienste abgeleitet werden:
  - Vertraulichkeit der Daten
  - Verhinderung einer Verkehrflußanalyse
  - Datenunversehrtheit
  - Authentizitätsprüfung des Kommunikationspartners

Sicherheitsdienste 2

- Authentizitätsprüfung des Datenabsenders
- Zugangskontrolle
- Sendernachweis
- Empfängernachweis

I.5. Sicherheitsmechanismen

- Verschlüsselung
- Digitale Unterschrift
- Hashfunktion
- Authentizitätsprüfung
- Zugangskontrolle
- Sicherstellung der Datenunversehrtheit
- Verhinderung der Verkehrsflußanalyse

Sicherheitsmechanismen 2

- Routingkontrolle
- Notariatsfunktion
- Vertrauenswürdige Implementation
- Abstrahlsichere Endgeräte und Vermittlungseinrichtungen
- Überwachung und Alarmierung (Alert)
- Logbuch

I.6. Authentizitätsprüfung und Schlüsselverteilung

- Schlüsselverwaltung
- Authentizitätsprüfungsverfahren
- Schlüsselverteilung mit Private Keys
- Schlüsselverteilung mit Public Keys

I.6.1. Schlüsselverwaltung

- Schlüsselerzeugung
- Interne Schlüsselverteilung
- Externe Schlüsselverteilung
- Schlüsselinstallation

Schlüsselerzeugung

- Deterministisch
  - Pseudozufallszahlen
  - Rekonstruierbar
- Nicht deterministisch
  - „Echte“ Zufallszahlen
  - „Nicht“ rekonstruierbar

Interne Schlüsselverteilung

- Erfolgt im Netz
  - Abhörgefährdet
  - Gefahr der Fälschung des Schlüssels
  - Fälschung der Identität

Externe Schlüsselverteilung

- Erfolgt durch systemfremde Übertragung (z.B.: Boten)
  - Weniger Abhörgefährdet
  - Geringere Fälschungsgefahr des Schlüssels
  - Fälschung der Identität aufwendiger

Schlüsselinstallation

- Laden der Schlüssel
- Speichern der Schlüssel
- Erschwerung des Zugangs
- Für Richtigkeitsprüfung soll die Kenntnis des Schlüssel nicht notwendig sein

I.6.2. Authentizitätsprüfungs-verfahren

- 2 Arten
  - Schwache Authentizitätsprüfung (Passwort, ...)
  - Starke Authentizitätsprüfung (Verschlüsselung eines „Tokens“ mit kryptographischen Methoden)
- Das Problem der Übertragung ist bei beiden Arten gleich

I.6.3. Schlüsselverteilung mit Private Keys

- Anzahl der Schlüssel: n/2 * (n-1)
- Master-Keys
- Schlüsselverteilzentrale mit zweiseitiger Teilnehmerkommunikation
- Schlüsselverteilung mit einseitiger Teilnehmerkommunikation

Master-Keys

- Masterkeys werden für den Austausch der „Session“-Keys benutzt.
- Allerdings verlagert sich das Problem der Schlüsselverteilung auf die Masterkeys (Seltener im Einsatz).
- Sessionkeys können oft gewechselt werden.

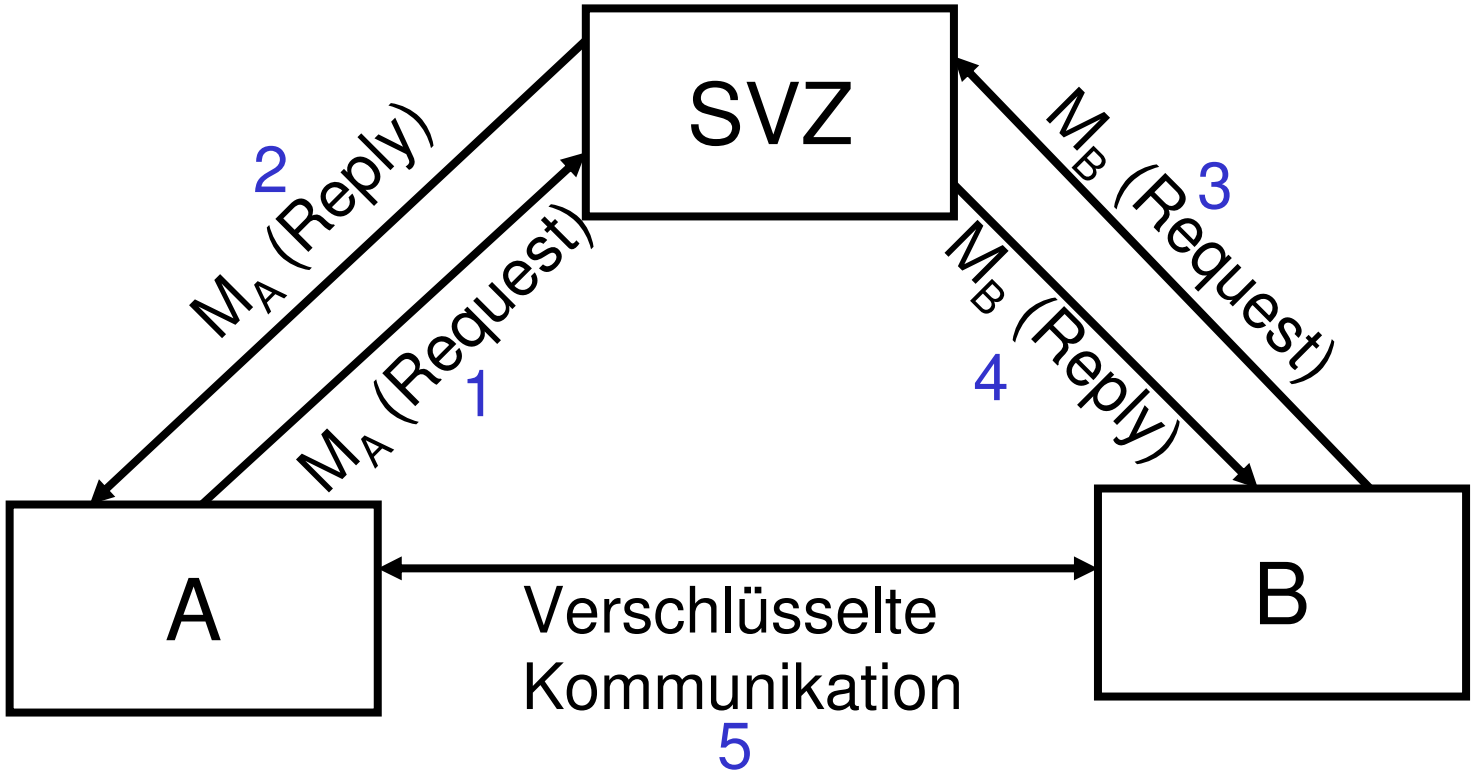
Verteilung mit zweiseitiger Teilnehmerkommunikation 1

- Schlüsselverteilzentrale (SVZ) reduziert den Aufwand für die Verwaltung der Schlüssel bei den einzelnen Teilnehmern.
- A möchte mit B kommunizieren
- M_A und M_B sind die Masterkeys von A bzw. B

Verteilung mit zweiseitiger Teilnehmerkommunikation 2

- A fordert von SVZ einen Sessionkey an
- SVZ schickt den Sessionkey an A
- B fordert ebenfalls von der SVZ diesen Sessionkey an
- SVZ schickt den Sessionkey an B
- A und B kommunizieren
- Jede Station hat nur einen Key

Verteilung mit zweiseitiger Teilnehmerkommunikation 3



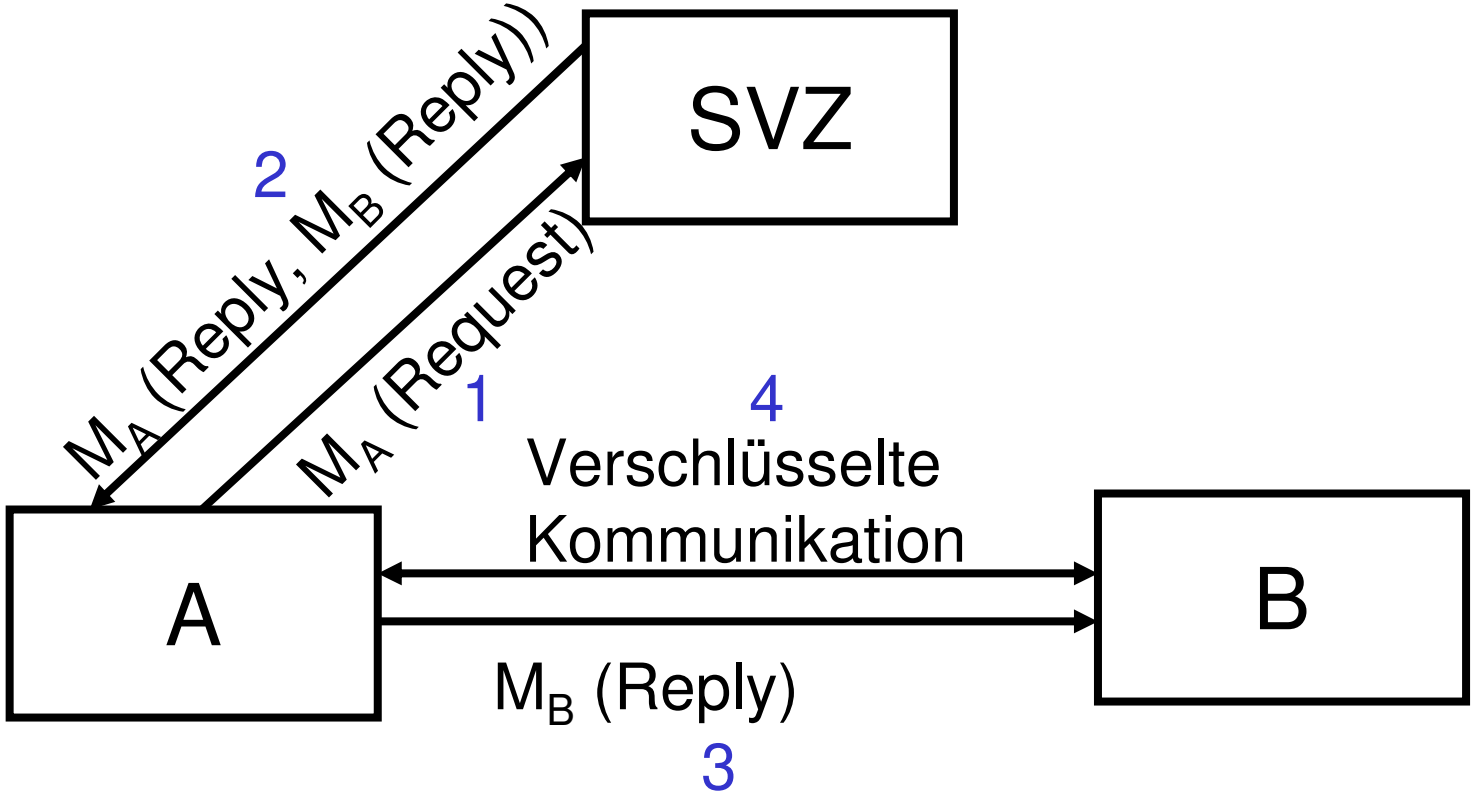
Verteilung mit einseitiger Teilnehmerkommunikation 1

- Schlüsselverteilzentrale (SVZ) reduziert auch hier den Aufwand für die Verwaltung der Schlüssel bei den einzelnen Teilnehmern.
- A möchte mit B kommunizieren
- M_A und M_B sind die Masterkeys von A bzw. B

Verteilung mit einseitiger Teilnehmerkommunikation 2

- A fordert von SVZ einen Sessionkey an
- SVZ schickt den Sessionkey und einen Block für B an A (inkl. Zeitstempel)
- A schickt das für B bestimmte Paket an B (Inhalt ist für A unbrauchbar)
- A und B kommunizieren
- Jede Station hat nur einen Key

Verteilung mit einseitiger Teilnehmerkommunikation 3



I.6.4. Schlüsselverteilung mit Public Keys 1

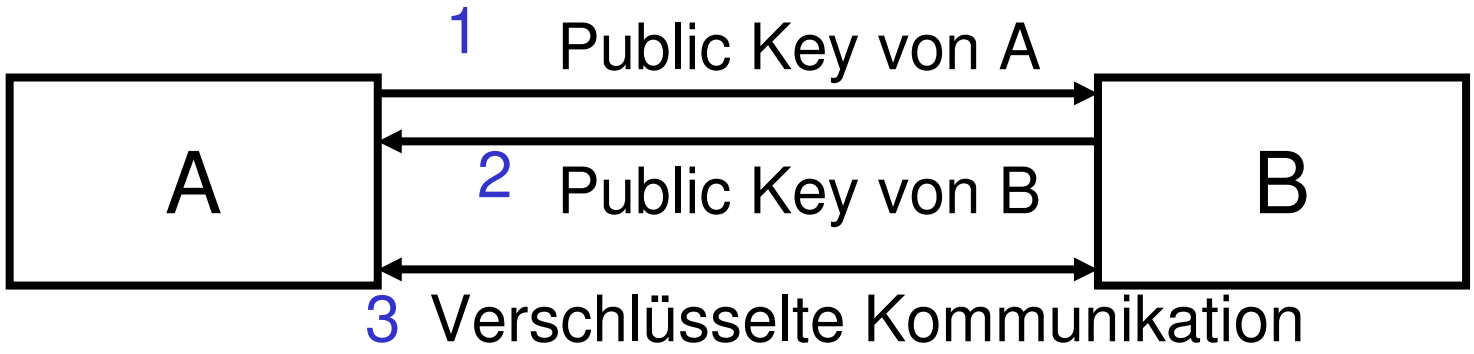
- Schlüsselaustausch zu Beginn der Kommunikation
- Teilnehmer haben Public-Key-Verzeichnis
- Schlüsselverteilzentrale mit zweiseitiger Teilnehmerkommunikation
- Schlüsselverteilung mit einseitiger Teilnehmerkommunikation

Schlüsselverteilung mit Public Keys 2

- Authentizitätsprüfung bei mehreren Schlüsselverteilzentralen
- Normung
- Schlüsselerzeugung
- Hardwarelösungen (Chipcard)

Schlüsselaustausch zu Beginn der Kommunikation

- Austausch der jeweiligen öffentlichen Schlüssel vor der eigentlichen Kommunikation



Teilnehmer haben Public-Key-Verzeichnis

- Vorteile
  - Sichere Kommunikation ohne Schlüsselaustausch möglich
  - Authentizität ist „gewährleistbar“
- Nachteile
  - Verzeichnis wird rasch umfangreich
  - Alle Teilnehmer müssen von einem Schlüsselwechsel informiert werden

Verteilzentrale mit zweiseitiger Teilnehmerkommunikation

- Analog zum Private Key-Verfahren, statt der Masterkeys werden aber auch für die Kommunikation zur SVZ Public-Keys verwendet
- Wenig praktische Bedeutung, da hier mit hoher Sicherheit auf das „einseitige“ Verfahren ausgewichen werden kann.

Verteilung mit einseitiger Teilnehmerkommunikation 1

Annahmen:

- SVZ kennt alle Public-Keys und den eigenen Secret-Key
- SVZ_{pk}, A_{pk}, B_{pk} ... Public Keys
- SVZ_{sk}, A_{sk}, B_{sk} ... Secret Keys
- A möchte gesichert mit B kommunizieren

Verteilung mit einseitiger Teilnehmerkommunikation 2

- A fordert von SVZ den öffentlichen Schlüssel von B an (Anfrage ist mit SVZ_{pk} verschlüsselt und enthält Teilnehmerkennungen von A und B sowie Datum/Uhrzeit)
- SVZ antwortet mit einer Nachricht, die zwei Zertifikate enthält

Zertifikat für A

- Enthält:
  - Teilnehmernummer von B
  - Den öffentlichen Schlüssel von B: B_{pk}
  - Datum/Uhrzeit
- Ist von SVZ digital unterschrieben und mit A_{pk} verschlüsselt

Zertifikat für B

- Enthält:
  - Teilnehmernummer von A
  - Den öffentlichen Schlüssel von A: A_{pk}
  - Datum/Uhrzeit
- Ist von SVZ digital unterschrieben und mit B_{pk} verschlüsselt

Auswertung durch A

- A entschlüsselt sein Paket, überprüft die Unterschrift und übernimmt B_{pk}.
- Das zweite Zertifikat wird an B weitergeleitet
- Eine Kontrollnachricht mit den Daten im Zertifikat der SVZ wird ebenfalls an B geleitet (von A unterschrieben und mit B_{pk} verschlüsselt).

Auswertung durch B

- B prüft das Zertifikat und die Kontrollnachricht
- B sendet seinerseits eine analoge Kontrollnachricht an A
- Nach Prüfung dieser kann die gesicherte Kommunikation beginnen.

Authentizitätsprüfung bei mehreren SVZs

- Analog zur gesicherten Kommunikation zwischen A und B muß eine verschlüsselte Kommunikation zwischen den SVZs hergestellt werden und die öffentlichen Schlüssel der Teilnehmer zwischen den SVZs ausgetauscht werden.

Schlüsselerzeugung

- Erzeugung der Schlüssel durch die Teilnehmer selbst
- Erzeugung der Schlüssel durch die SVZ (Transport der Schlüssel?)
- Erzeugung der Schlüssel durch Dritte (Signaturstellen, ... ,Transport der Schlüssel?)

Hardwarelösungen (Chipcard)

- Sichere Aufbewahrung der Secret-Keys in einer Chipcard
- Hardware zu Lesen der Karte notwendig

I.7. Einbindung in ein Referenzmodell

- ISO-Referenzmodell
  - Application Layer
  - Presentation Layer
  - Session Layer
  - Transport Layer
  - Network Layer
  - Data Link Layer
  - Physical Layer

I.7.0. Problem des Routings

- Für den Verbindungsaufbau sind Daten notwendig, die nicht verschlüsselt sein dürfen
- Sicherung bis zur Schicht 3 schwierig
- Paketvermittlung
- Leitungsvermittlung

I.7.1. Schicht 1

- Dienste:
  - Vertraulichkeit der Verbindung
  - Verhinderung einer Verkehrflußanalyse
- Mechanismen
  - Verschlüsselung (außer Start- und Stopbits) zwischen nächsten Nachbarn (meist auf HW-Ebene).

Schicht 1

- Einsatzmöglichkeiten
  - Nur in Schicht 1 ist der gesamte Verkehrsfluß schützbar.
  - Entzieht sich aber den Möglichkeiten eines Anwenders.
  - Derzeit von keinem Leitungsprovider angeboten.

I.7.2. Schicht 2

- Dienste
  - Vertraulichkeit bei verbindungsorientierten und verbindungslosen Kommunikationen.
- Mechanismen
  - Verschlüsselung der Verbindung (Linkverschlüsselung).

Schicht 2

- Einsatzmöglichkeiten
  - Entzieht sich den Möglichkeiten eines Anwenders
  - Derzeit von keinem Leitungsprovider angeboten (anders im Funkbereich)

I.7.3. Schicht 3

- Dienste
  - Authentizitätsprüfung der Instanz des Kommunikationspartners
  - Zugangskontrolle
  - Vertraulichkeit bei verbindungsorientierten und verbindungslosen Kommunikationen
  - Verhinderung einer Verkehrsflußanalyse
  - Datenunversehrtheit ohne Recovery
  - Authentizitätsprüfung des Absenders der Daten

Schicht 3

- Mechanismen 1
  - Die Authentizitätsprüfung wird durch eine Kombination aus kryptographischen Methoden, digitaler Unterschrift, Paßwörtern und ein eigenes Authentizitätsprüfungsprotokoll unterstützt.

Schicht 3

- Mechanismen 2
  - Die Zugangskontrolle erfordert eigene Zugangskontrollmechanismen sowohl in den Vermittlungsknoten (Kontrolle durch den Netzbetreiber) als auch im Zielsystem (Abweisung unerwünschter Verbindungen)

Schicht 3

- Mechanismen 3
  - Knotenverschlüsselung für die Vertraulichkeit der Verbindung und die Vertraulichkeit der Daten. Zusätzlich Routingkontrollfunktionen können dem Benutzer eine Auswahl der Wege erlauben.

Schicht 3

- Mechanismen 4
  - Zur Verhinderung der Verkehrflußanalyse werden vom Netzbetreiber Fülldaten geschickt (müssen verschlüsselt sein oder von einer der unteren Schichten verschlüsselt werden); dabei muß aber durch eine Flußkontrolle gewährleistet bleiben, daß noch immer Daten übertragen werden können.

Schicht 3

- Mechanismen 5
  - Die Datenunversehrtheit kann durch eine Prüfsumme oder Hashwerte sichergestellt werden, dabei ist in dieser Schicht keine „Recovery“ vorgesehen (siehe auch ISO-Referenzmodell).

Schicht 3

- Mechanismen 6
  - Der Sendernachweis wird ebenfalls über die Authentizitätsprüfung (des Senders) erreicht.
- Einsatzmöglichkeiten
  - Durch Netzbetreiber (siehe oben)
  - Durch Anwender (VPN)

I.7.4. Schicht 4

- Dienste
  - Authentizitätsprüfung der Instanz des Kommunikationspartners
  - Zugangskontrolle
  - Vertraulichkeit bei verbindungsorientierten und verbindungslosen Kommunikationen
  - Datenunversehrtheit der Verbindung mit bzw. ohne Recovery
  - Authentizitätsprüfung des Absenders der Daten

Schicht 4

- Mechanismen
  - Siehe Schicht 3 allerdings werden aus den „Next Hop“-Mechanismen „End-to-End“-Mechanismen.
- Einsatzmöglichkeiten
  - Ab dieser Schicht liegt der Einsatz der Mechanismen vollständig in der Verantwortung des Netzbenutzers.

I.7.5. Schicht 5

- Dienste
  - Keine eigenen Sicherheitsdienste aber die Vereinbarung von notwendigen Diensten für die Session
- Mechanismen
  - Keine
- Einsatzmöglichkeiten
  - Keine außer der Vereinbarung

I.7.6. Schicht 6

- Dienste
  - Keine eigenen Dienste
- Mechanismen
  - Sendernachweis
  - Empfängernachweis
  - Notariatsfunktion

Schicht 6

- Einsatzmöglichkeiten
  - Anbieten von Mechanismen um der Anwendungschicht alle notwendigen Dienste zu ermöglichen

I.7.7. Schicht 7

- Dienste
  - Anwendungsabhängig
- Mechanismen
  - Entweder anwendungseigene Mechanismen
  - Nutzung von Mechanismen der darunter liegenden Schichten

I.8. „Normen“

- L2F (Layer 2 Forwarding, Cisco ...)
- PPTP (Point-to-Point Tunneling Protocol, Microsoft ...)
- L2TP (Layer 2 Tunneling Protocol, L2F+PPTP nach RFC 2661)
- IPv4 – IPv6
- IPSec (IP Security Protocol, RFCs 2401 – 2412)

I.8.1. L2F

- Layer 2 Forwarding
- Entwickelt von Cisco (Nortel, Shiva)
- RFC 2341 aus 1998 (historic)
- Reines Tunnelprotokoll (d.h. keine Verschlüsselung)
- Punkt zu Mehrpunktverbindungen möglich
- ISO-Schicht 2

I.8.2. PPTP

- Point-to-Point Tunneling Protocol
- Entwickelt vom PPTP-Forum (Microsoft, U.S.-Robotics, ...)
- Kein Standard, kein Keymanagement, keine Integritätsprüfung
- Verschlüsselung (40, 56 und 128 Bit)
- ISO-Schicht 2

I.8.3. L2TP

- Layer 2 Tunneling Protocol
- Zusammenführung von L2F und PPTP
- RFC 2661 aus 1999 (proposed)
- Unterstützung von Mehrpunktverbindungen und NAT
- Authentifizierung mittels PAP/CHAP

I.8.4. IPv4

- Im IPv4-Protokoll keine Sicherheitsfunktionen implementiert
- Daher kann auch nur innerhalb der Nutzdaten mit Hilfe von Sicherheitsfunktionen (Verschlüsselung) gearbeitet werden
- IPSec (s.u.) später für IPv4 adaptiert.

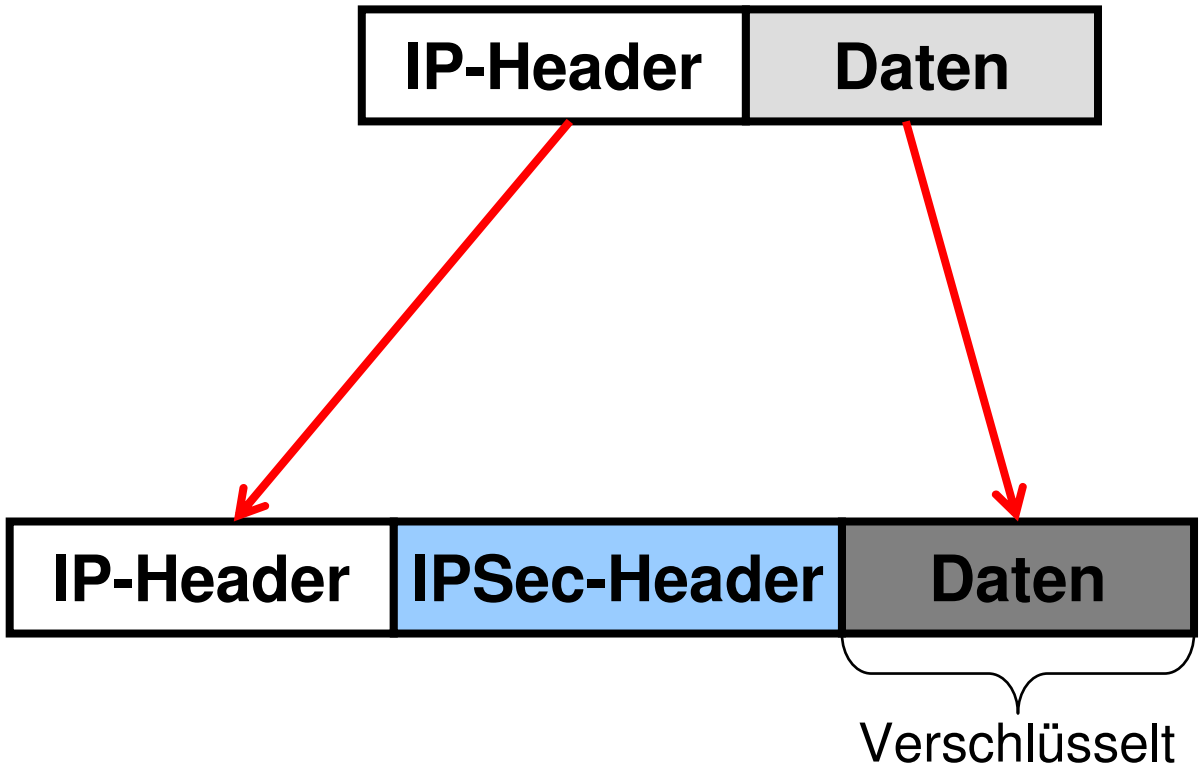
I.8.5. IPv6

- Sicherheitsfunktionen in das Protokoll implementiert.
- Mehr Sicherheit, da ganze Pakete gesichert werden können.
- Sonstige neue Funktionen nicht sicherheitsrelevant.

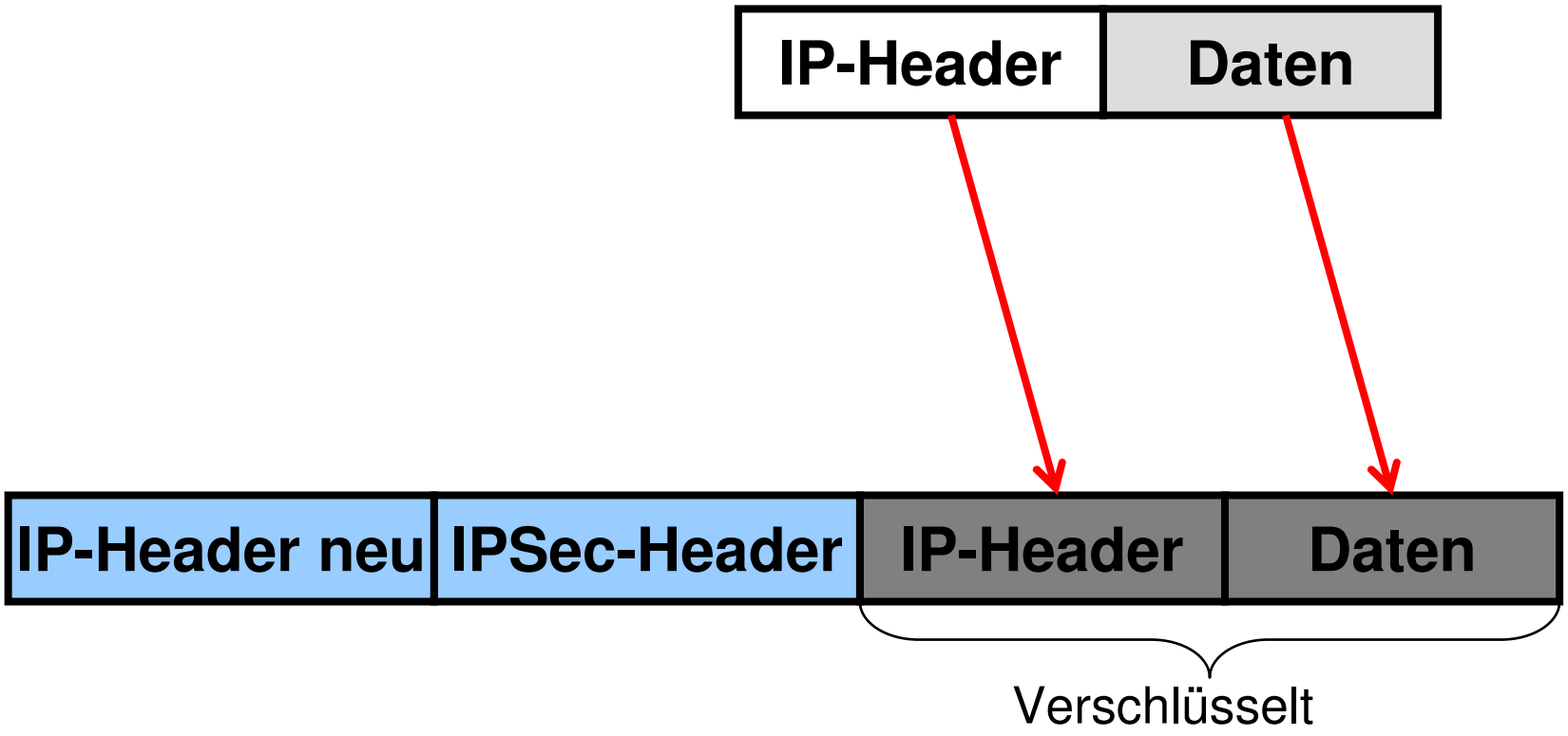
I.8.6. IPSec

- RFCs 2401 – 2412
- IP Security Protocol
- Soll PPTP ablösen
- 2 Modi
  - Transportmodus (nur die Daten werden verschlüsselt)
  - Tunnelmodus (ganzes Paket verschlüsselt)

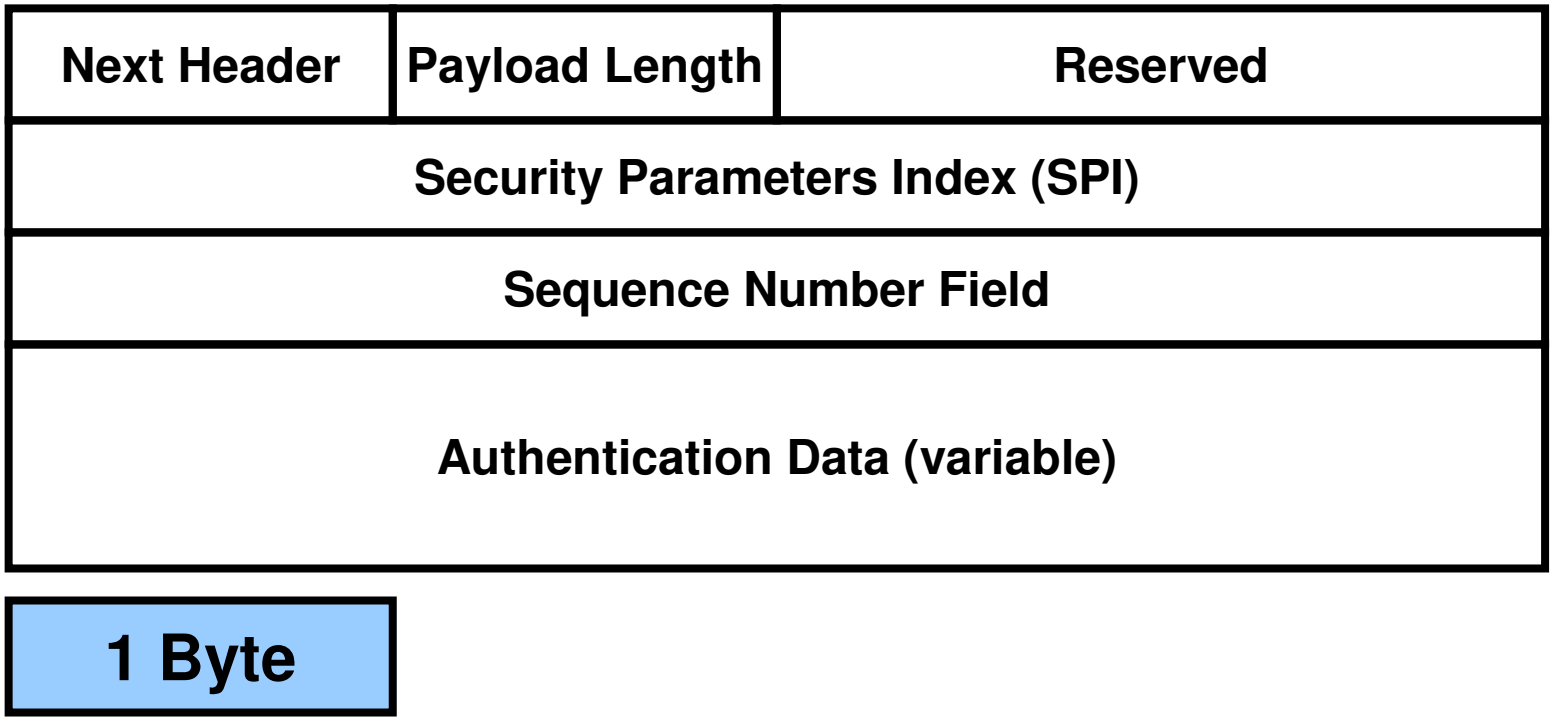
Transportmodus



Tunnelmodus



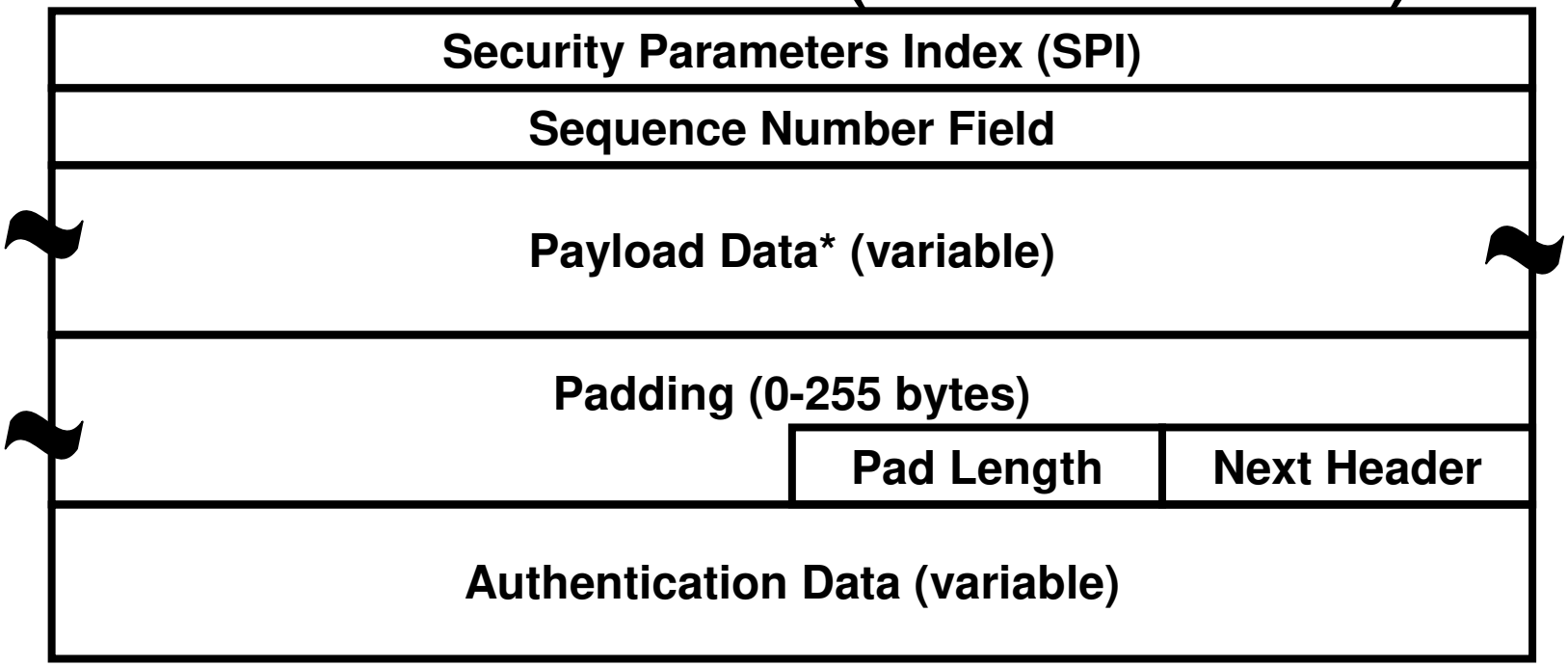
AH-Header (RFC 2402)



Im Header davor steht 51 als Protokolltyp

(IPv4 Protocol- bzw. IPv6 Next Header-Field)

ESP-Packet (RFC 2406)



1 Byte

Im Header davor steht 50 als Protokolltyp

(IPv4 Protocol- bzw. IPv6 Next Header-Field)

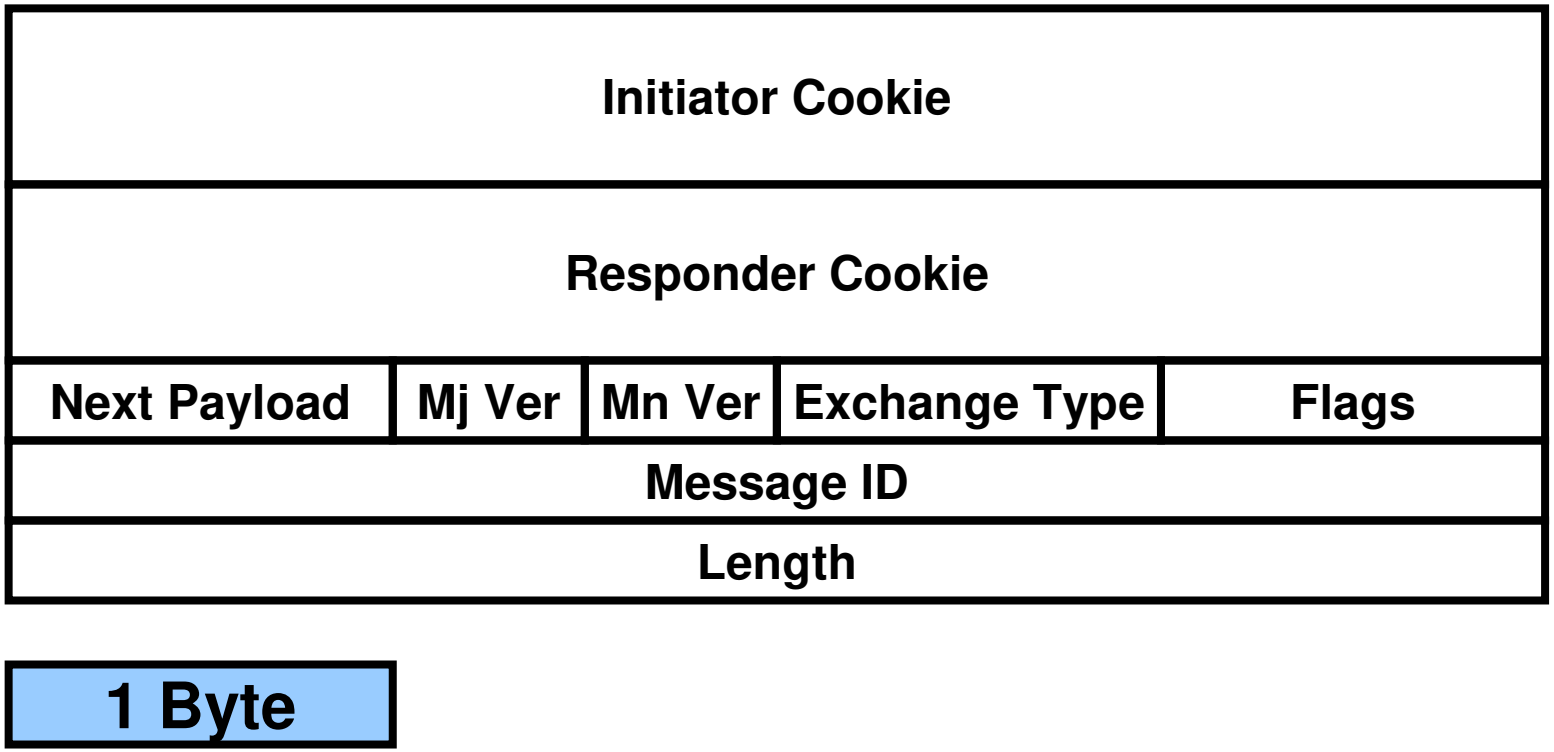
Schlüsselaustausch

- Diffie-Hellman (IEEE Transactions on Information Theory, V. IT-22, n. 6, June 1977)
- Oakley (RFC2412)
- SKEME (IEEE Proceeding 1996)

  Secure Key Exchange Mechanism
- IKE (RFC 2409)

  Internet Key Exchange

ISAKMP-Header (RFC 2408)



Hashfunktionen

- HMAC (RFC2104)

  keyed-Hashing for Message AuthentiCation
- MD5 (RFC 1321)

  Message Digest algorithm
- SHA (FIPS 180-1 1994)

  Secure HAsh standard

Verschlüsselungsalgorithmen

- IDEA (ETH Series in Inf.Proc., v. 1)
- DES (ANSI X3.106)

  Data Encryption Standard
- Blowfish (Dr.Dobb's Journal, April 1994)
- RC4/RC5 (RSA Data Security)

L2TP using IPSec

- RFC 3193 aus 2001 (proposed)
- Verwendet UDP-Port 1701
- Authentifikation, Verschlüsselung, Datenintegrität und Verhinderung von Replayattacken
- Erlaubt freiwillige und verpflichtende Tunnel

Security Policies

- Einführung und Normen
- Ziele von Security Policies
- Arten von Security Policies
- Verantwortung für Security Policies
- Umsetzung von Security Policies
- Labor (Entwickeln von Security Policies für die Musterfirma)

Einführung

- Eine **Security Policy** beschreibt den erstrebten Anspruch einer Institution nach Informationssicherheit.
- In einer Security Policy werden die Verwaltensweisen beschrieben, die zur Erreichung der Ziele notwendig sind (z.B. wie oft muß ein Passwort geändert werden).

„Definition“

- Eine Sicherheitsrichtlinie ist ein Satz von Regeln, die definieren **wer** autorisiert ist, auf **was** zuzugreifen und unter welchen Bedingungen diese bzw. die Kriterien nach denen diese Autorisation gewährt oder verwehrt wird.

Normen 1

- Vorreiter BS 7799-1 (British Standard)
- ISO/IEC 27001 und ISO/IEC 27002
- ITU-Empfehlung X.800
- RFC 2196 (Site Security Handbook)
- RFC 4949 (Internet Security Glossary, Version 2)

Normen 2

- ISO/IEC 13335 (Management der Informationssicherheit)
- ITIL (IT Infrastructure Library, http://www.itsmf.at/)
- ISO/IEC 25010 (Softwarequalität)
- PCI-DSS (Payment Card Industry Data Security Standard; Sicherheitsstandard der Zahlungssysteme (VISA, …))

Ziele 1

- AAA
  - Authentication (Authentisierung)
  - Authorization (Autorisierung)
  - Accounting (Zurechenbarkeit)
- CIA
  - Confidentiality (Vertraulichkeit)
  - Integrity (Unversehrtheit)
  - Availability (Verfügbarkeit)

Ziele 2

- Das Ziel der Security Policies ist es „**AAA**“ von Personen und „**CIA**“ von Daten sowie die Nachweisbarkeit aller Änderungen (Protokollierung) sicherzustellen, sowie die Benutzer (Mitarbeiter) zu sensibilisieren und dies durch laufende Audits zu belegen und zu verbessern.

Arten von Security Policies

- Allgemeine Richtlinien (Sicherheitsziele und –strategien)
- Besondere Richtlinien für die einzelnen Einsatzgebiete (Zutrittsschutz, Zugriffsschutz, Schutz vor Verlust)

Allgemeine Richtlinien

- Festlegung von Verantwortlichkeiten
- Festlegung der übergeordneten Ziele
- Auswahl geeigneter Methoden
- Mechanismen zur Kontrolle
- Schulungspläne
- Notfallpläne
- Vorgaben für besondere Richtlinien

Besondere Richtlinien

- S.o.
- Einsatzbereich
- Konfigurationsdetails
- Berechtigungen
- Protokollierungsmaßnahmen
- Je nach Richtlinie weitere Maßnahmen

Benutzerrichtlinien

- Umgang mit Betriebsinterna
- Umgang mit Berechtigungen
- Umgang mit neuen Medien (Internet,…)
- Beiträge der Benutzer zu Schutzsystemen
- Urheberrecht
- Konsequenzen aus Fehlverhalten

Verantwortung für Security Policies

- Die Security Policies werden von der obersten Leitung erlassen, daher ist auch diese für die Vollständigkeit und die Einhaltung verantwortlich.
- Dazu müssen alle Mitarbeiter darüber geschult werden
- Für die Umsetzung sind alle betroffenen Mitarbeiter verantwortlich

Umsetzung von Security Policies

- Anhand von Checklisten analog zu den sonstigen Notfallplänen der Institution
- An Hand von Mustern für viele Fälle diverser einschlägiger Organisationen
- z.B.: BSI (Bundesamt für Sicherheit in der Informationstechnik)
  - https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz_node.htm

Konkrete Beispiele

- http://www.boku.ac.at/fileadmin/data/H05000/H19000/Themen-Content/H19000/Leitung/IT-Guidelines/SecurityPolicy_DE.pdf
- http://www.secupedia.info/wiki/IT-Sicherheits-Policy
- https://www.fu-berlin.de/sites/it-sicherheit/downloads/IT-Sicherheitsrichtlinie.pdf

Labor zu Security Policies

- Entwickeln von Security Policies für die Musterfirma

Sicherheitsverwaltung

- Netzwerkmanagement zur Sicherheitsverwaltung
- Intrusion Detection Systeme zur Sicherheitsverwaltung
- Eigenständige Lösungen z.B. von Antimalwareherstellern
- Berichte zur Sicherheitssituation

Netzwerkmanagement

- Warum
- Anforderungen
- Einordnung in Managementsysteme
- Standards und Protokolle
- Aufbau von Managementsystemen
- OSI-NMS
- SNMP-NMS
- Webbasierendes Management

Warum ?

- In den 80er Jahren wurde durch das Wachstum der Netzwerke der Bedarf nach Netzwerkmanagement immer dringender. Beginnend mit „Remote Login“ wurde ein Framework zur zentralen Verwaltung der Netzwerke geschaffen, um die Administration zu vereinfachen.

Anforderungen

- Faultmanagement
- Configurationmanagement
- Performancemanagement
- Accountingmanagement
- Securitymanagement

Faultmanagement

- Fehler erkennen
- Fehler lokalisieren
- Rekonfiguration zur Umgehung
- Fehler beheben
- Originalkonfiguration wiederherstellen

Configurationmanagement

- Rekonfiguration aktiver Komponenten (z.B.: Router)
- Stilllegung von Komponenten
- Aktivierung neuer Komponenten
- Erkennen neuer Komponenten

Performancemanagement

- Wie groß ist die Auslastung?
- Intensiver Datenverkehr einzelner Stationen?
- Gesamtdurchsatz?
- Flaschenhälse?
- Antwortzeiten?

Accountingmanagement

- Zugangskontrolle und Abrechnung
- Benutzerberechtigungen und deren unerlaubte Weitergabe
- Ineffiziente Nutzung des Netzes durch Benutzer
- Ausbauplanung

Securitymanagement

- Erzeugung, Verteilung und Speicherung von Verschlüsselungsinformationen (CA, PKI)
- Überwachung des Netzes
- Log-File-Analyse

Einordnung in Managementsysteme

- Anwendungsmanagement
- Informationsmanagement
- Systemmanagement
- Netzwerkmanagement
- Facilitymanagement

Anwendungsmanagement

- Anwendungen
- Globale Einstellungen (CI)
- Benutzerspezifische Einstellungen
- Installation
- Deinstallation

Informationsmanagement

- Datenbestände, Datenbanken
- DMS
- Backup and Restore

Systemmanagement

- Host
- Server
- Workstation
- PC
- NC
- Meist inkl. NMS

Netzwerkmanagement

- Aktive Netzwerkkomponenten
  - Hubs
  - Switches
  - Router
- Passive Netzwerkkomponenten
  - Verkabelung
  - Dokumentation

Facilitymanagement

- Gerätemanagement
  - Fax
  - Telephonanlage
  - NIC
- Verbindungsmanagement
  - Schaltschrank und Patchkabel
  - Glasfaserleitung, ...

Standards

- ISO 10164-x (x=1..22)
- SNMP (RFCs 1155, 1157, 1213, 1351..3, 1441..52, 1901..10, 2011..13, davon vieles aber DRAFT, PROPOSAL oder HISTORIC, Übersicht in „http://www.sei.cmu.edu/str/descriptions/snmp.html“)
- CCITT X.700 (=ISO/IEC 7498-4)

Protokolle und Abkürzungen

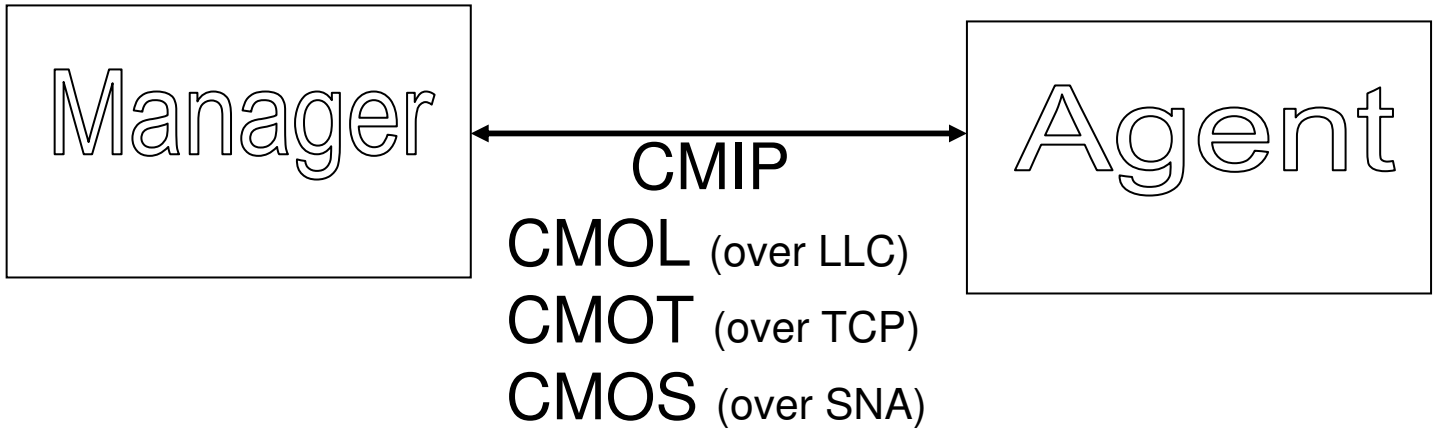
- OSI-CMIP (Common Management Information Protocol)
- OSI-CMIS (Common Management Information Service)
- SNMP (siehe oben)
- RMON (Remote MONitoring)
- MIB (Management Information Base)

Aufbau

- Managementkonsole, Management Station (GUI für das Gesamtsystem)
- Managementserver (Datenbank; Sammlung von Informationen)
- Management Agents (in allen managebaren Geräten bzw. eigene Geräte, die Information sammeln)

OSI-NMS

- Aufbau



SNMP-NMS

- Management Station und SNMP-Agent kommunizieren über SNMP-Nachrichten

Agent/Server	Anwendung	Anwendung
SNMP	IP-Anw.prot.	IPX-Anw.pr.
UDP	TCP	SPX
IP		IPX
MAC		

Webbasierendes Management - Vorteile

- Keine Managementsoftware notwendig (Jede managebare Komponenten beinhaltet Webserver + Browser an der Managementkonsole)
- Herstellerunabhängig
- Geringe Kosten
- Gesicherte Übertragung durch TCP

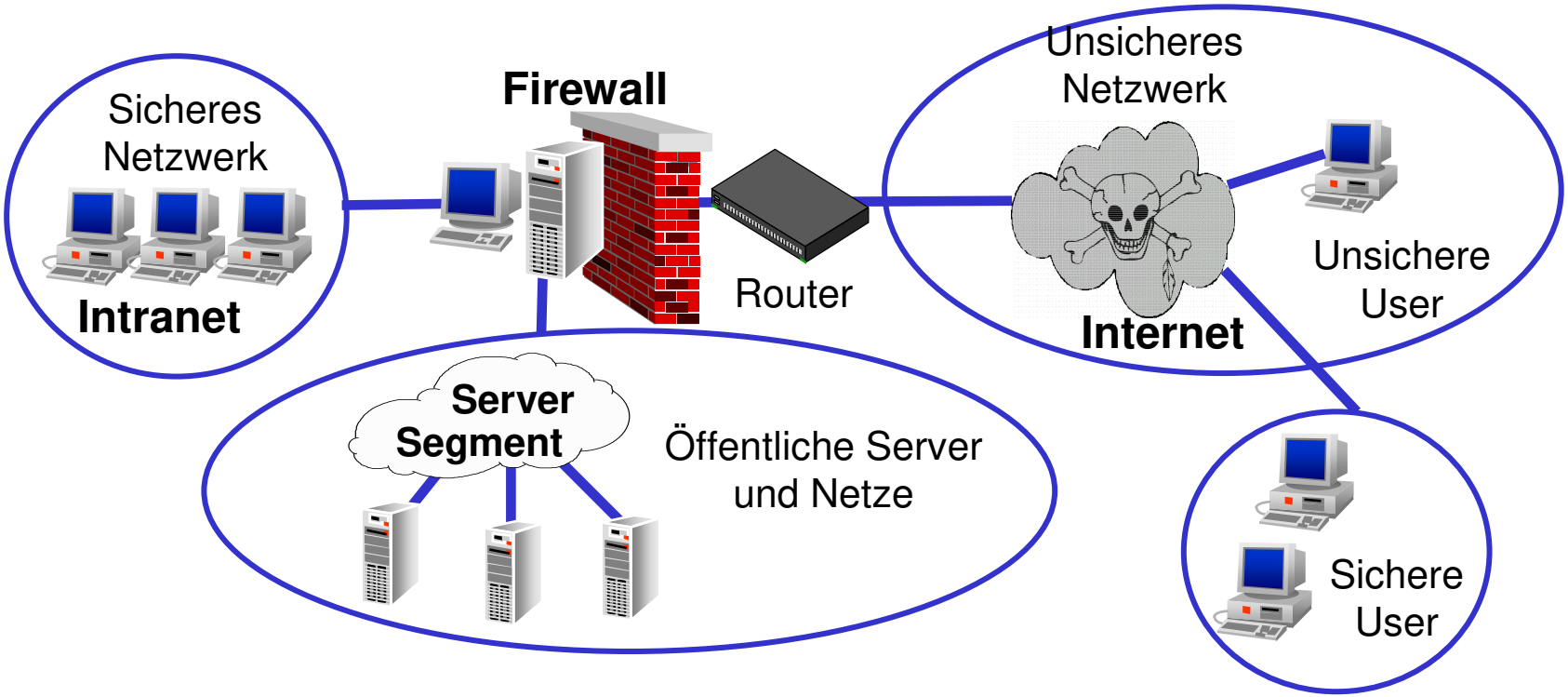
Webbasierendes Management - Nachteile

- Sicherheit
- Keine Traps
- Geringer Funktionsumfang
- Graphische Konfiguration verleitet mehr zum Probieren
- Meist reines Konfigurationsmanagement

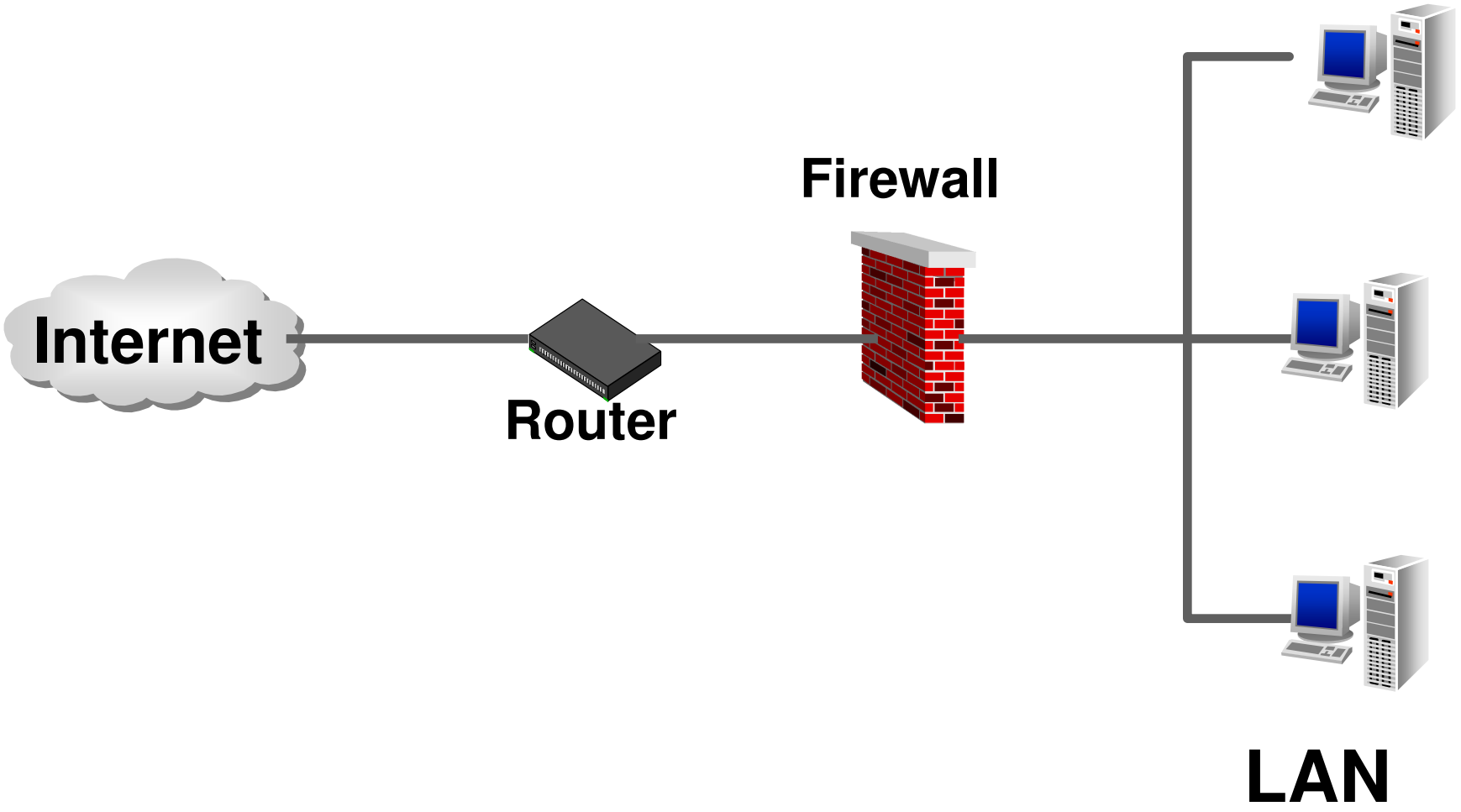
Firewalls

- Firewallarchitekturen
- Funktionsweise
  - Application Layer Gateway
  - Packet Filtering
  - Stateful Inspection

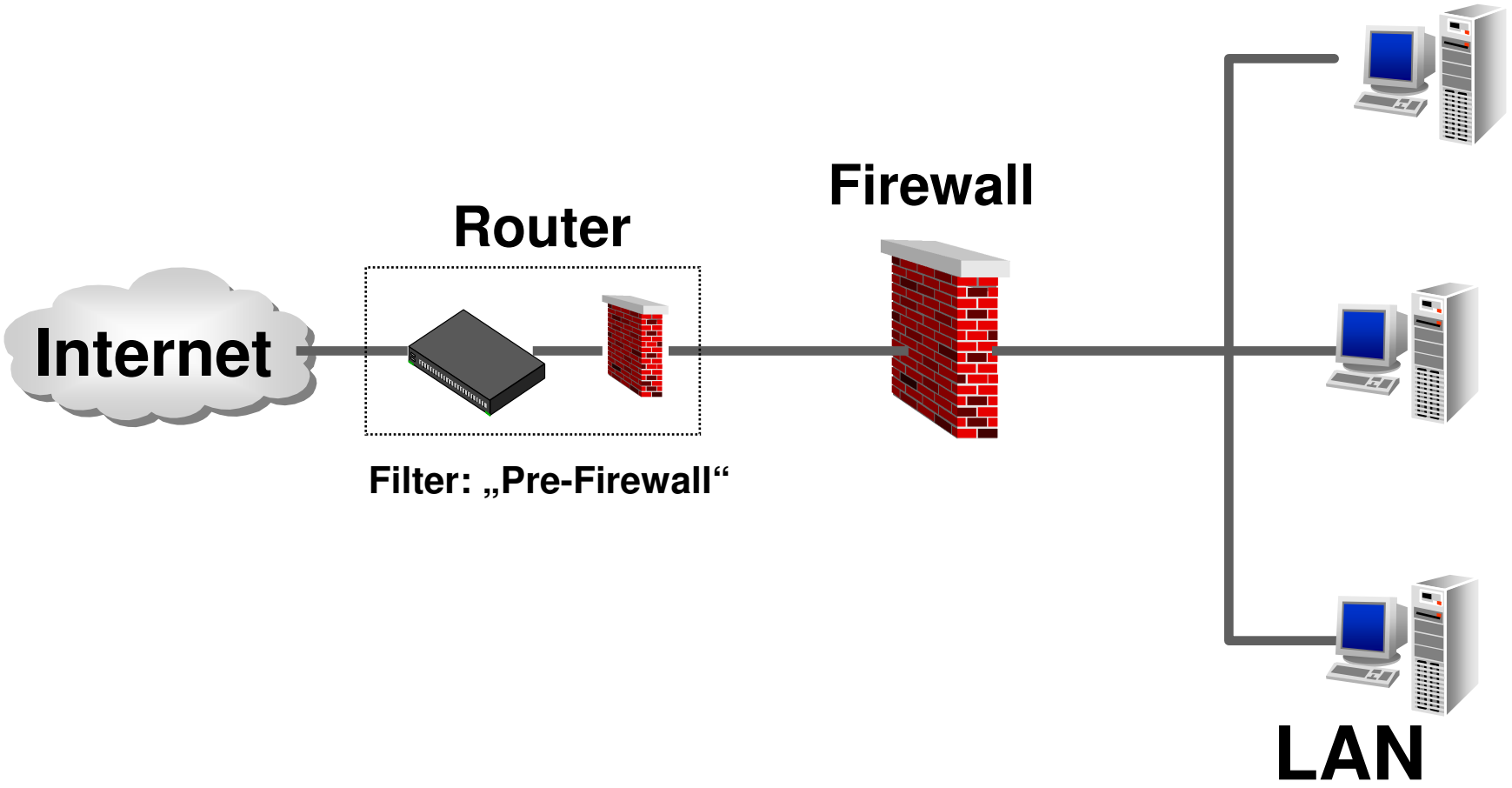
Standardposition der Firewall



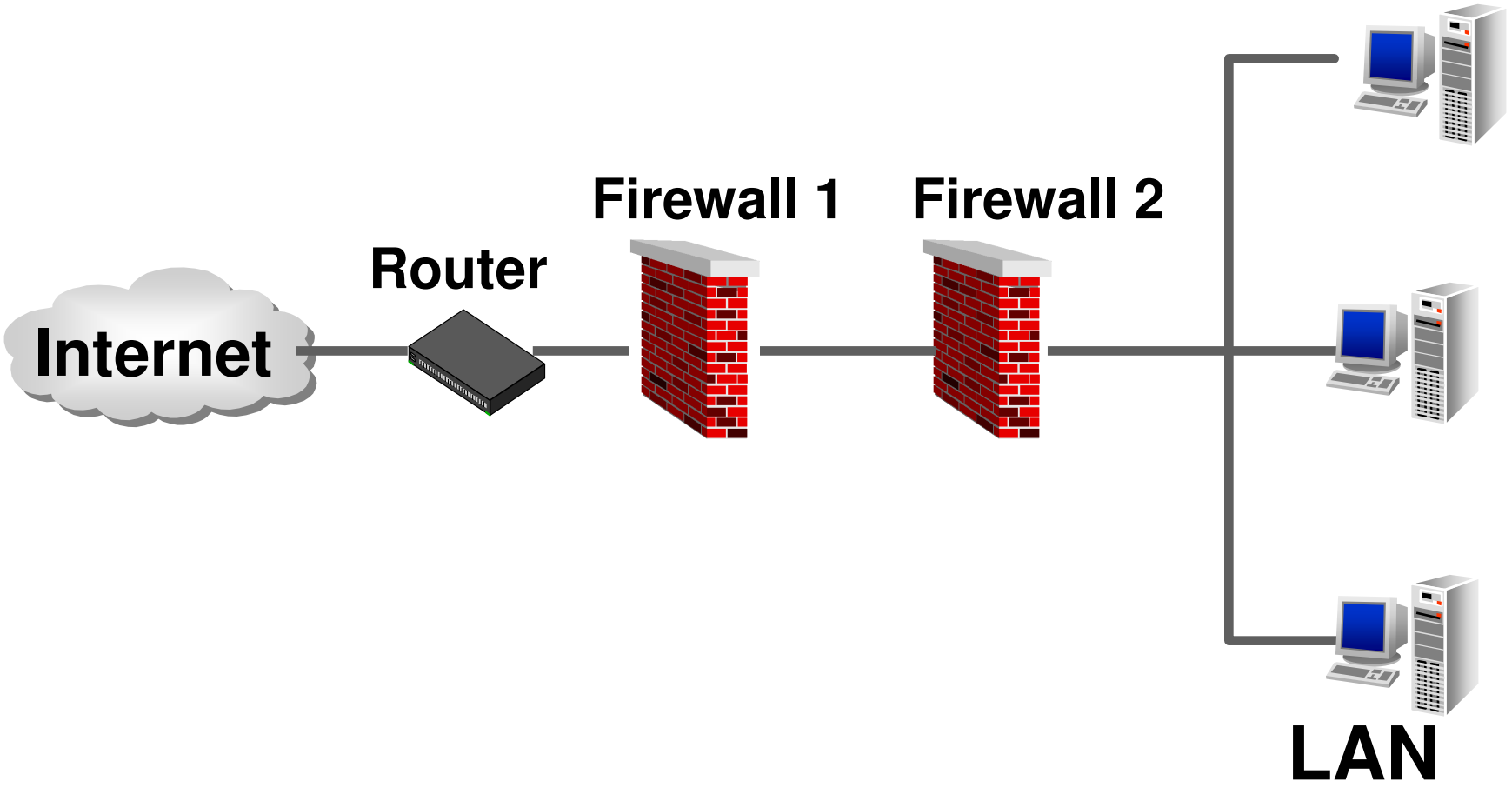
Single Firewall



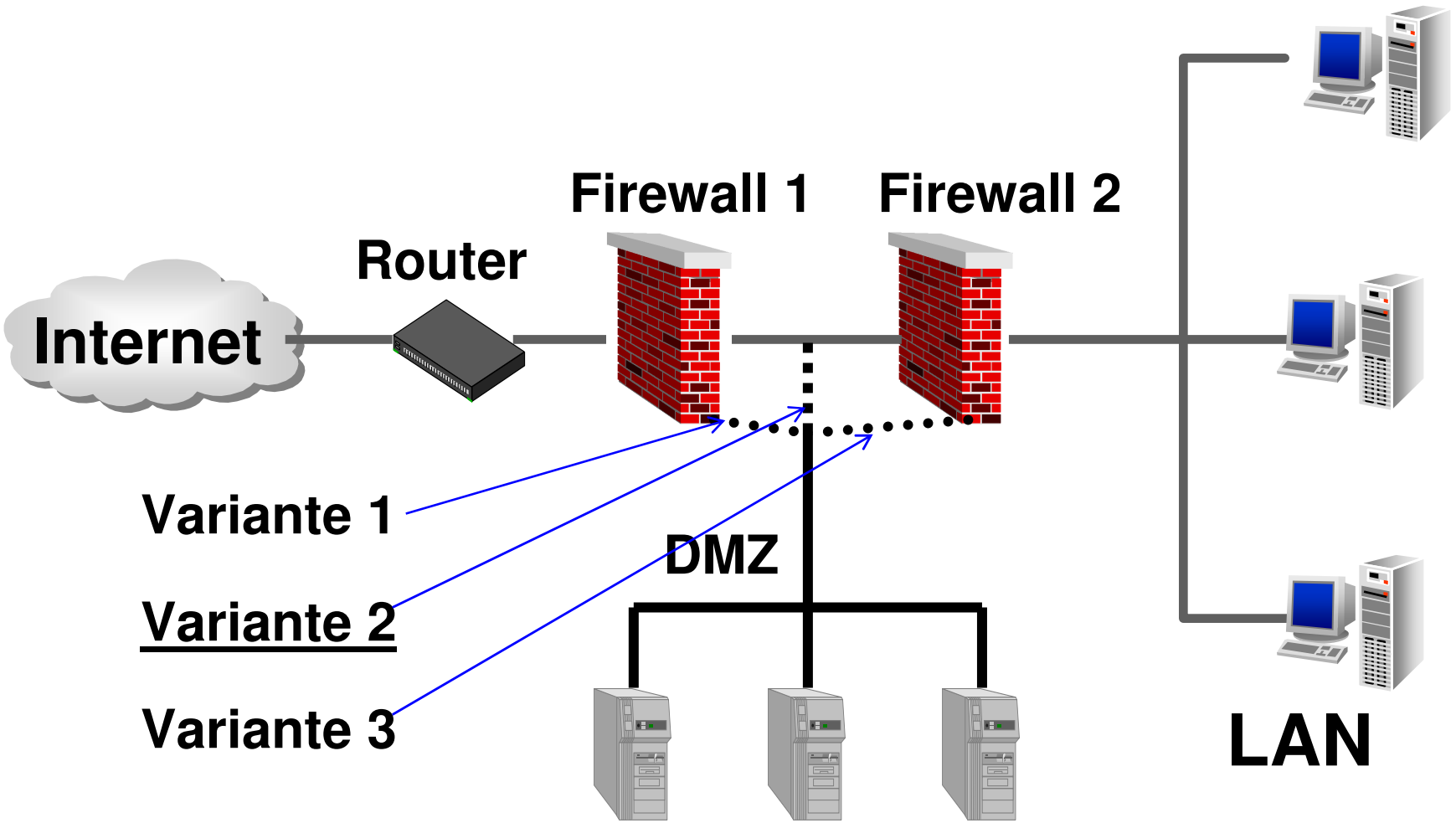
Router als Filter



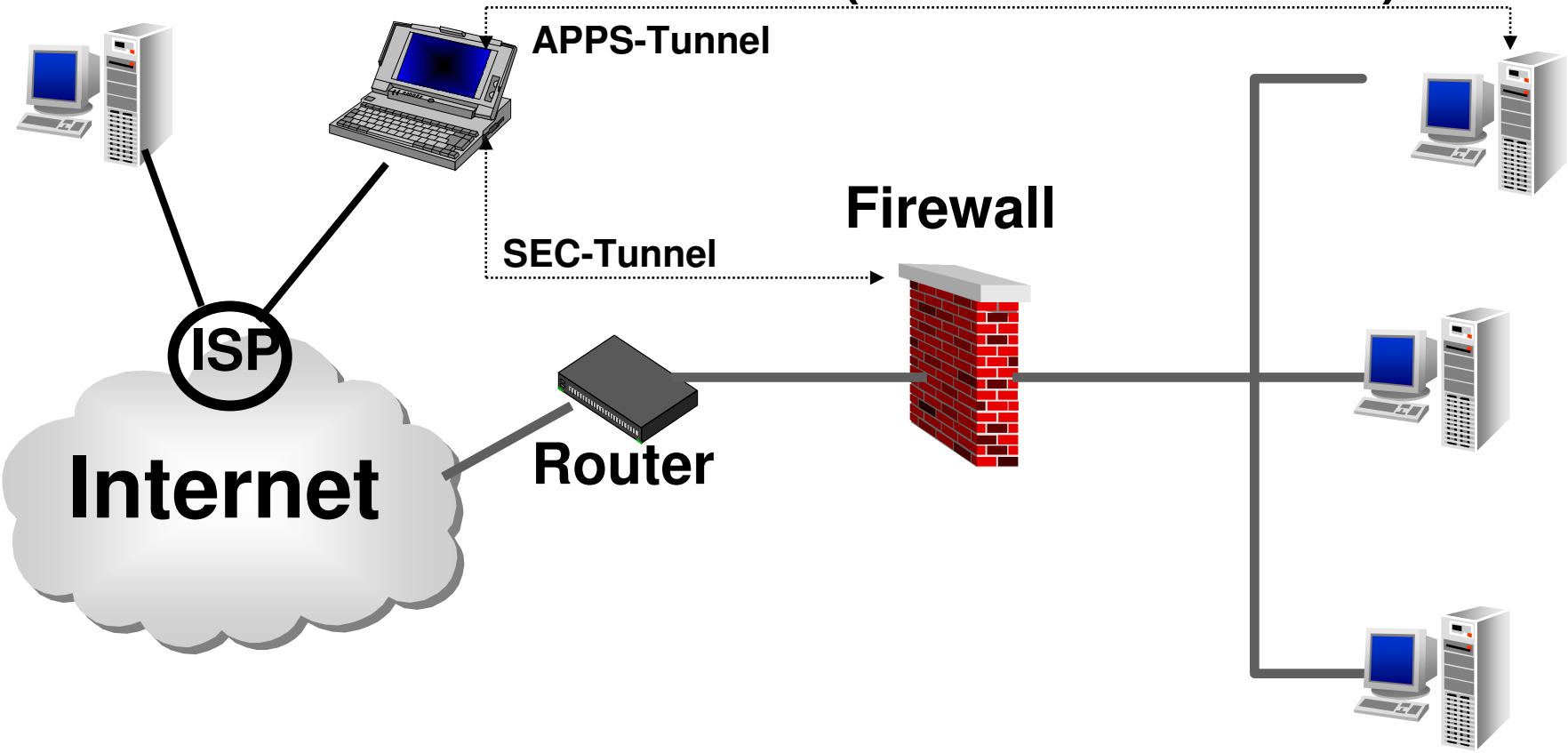
Dual Firewall



Dual Firewall mit DMZ



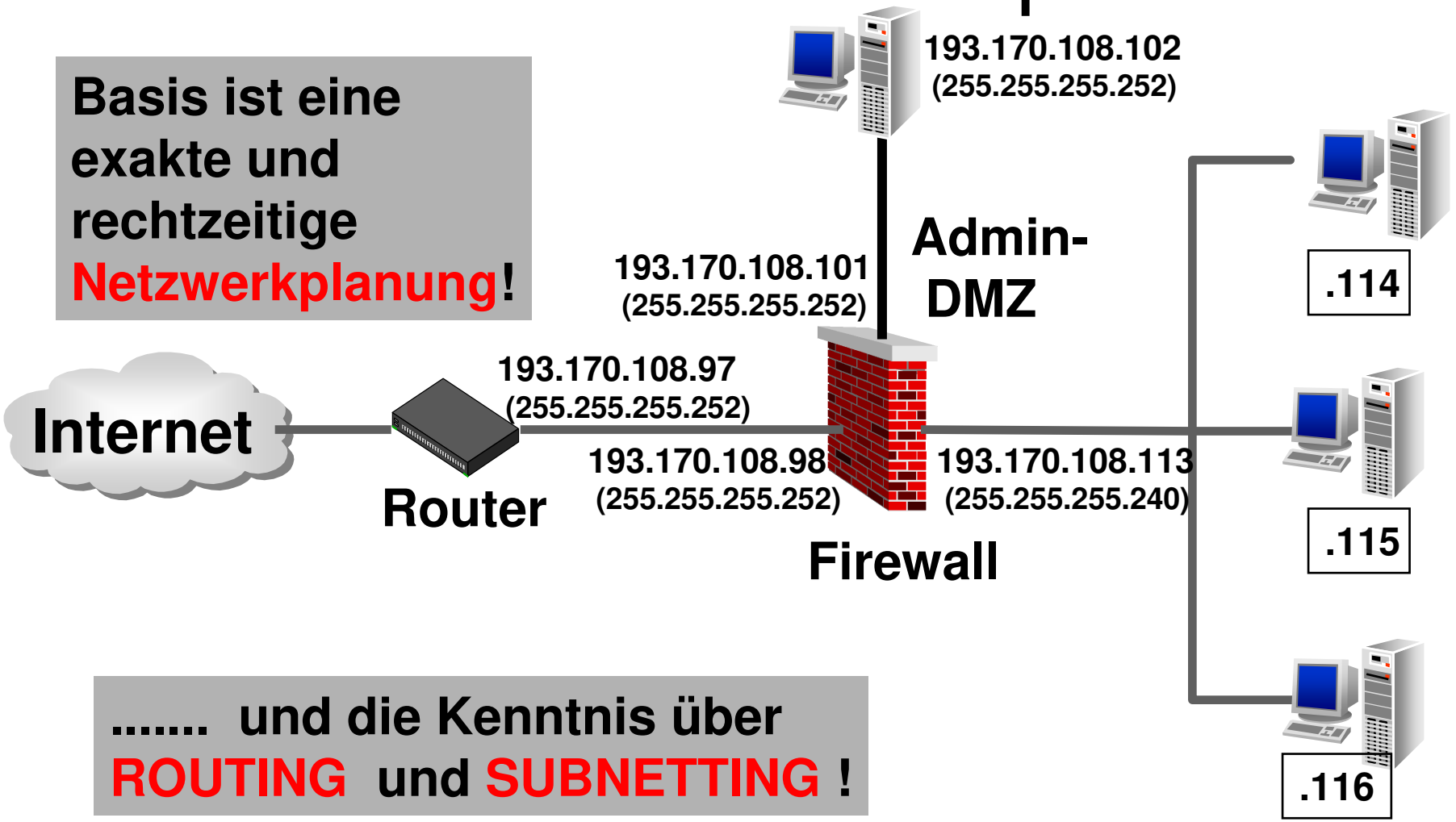
Remote User (Teleworker)



- Remote-User: z.B. SOHO
- Teleworker: „Mobile-Users“

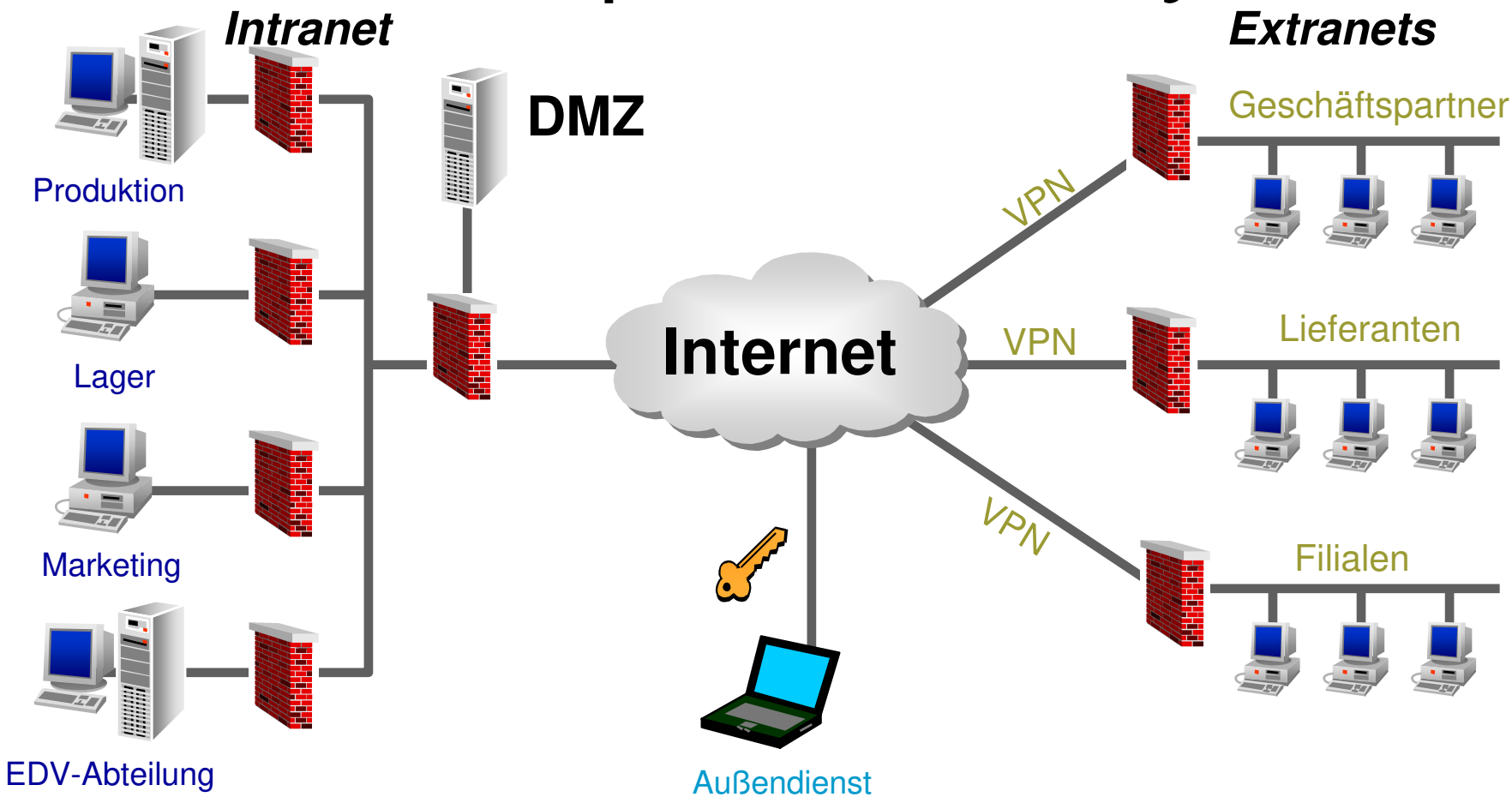
Praktisches Beispiel

Basis ist eine exakte und rechtzeitige Netzwerkplanung!

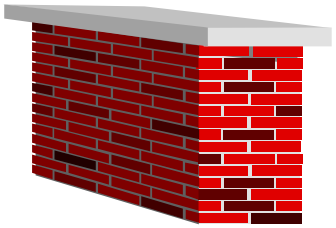


..... und die Kenntnis über ROUTING und SUBNETTING !

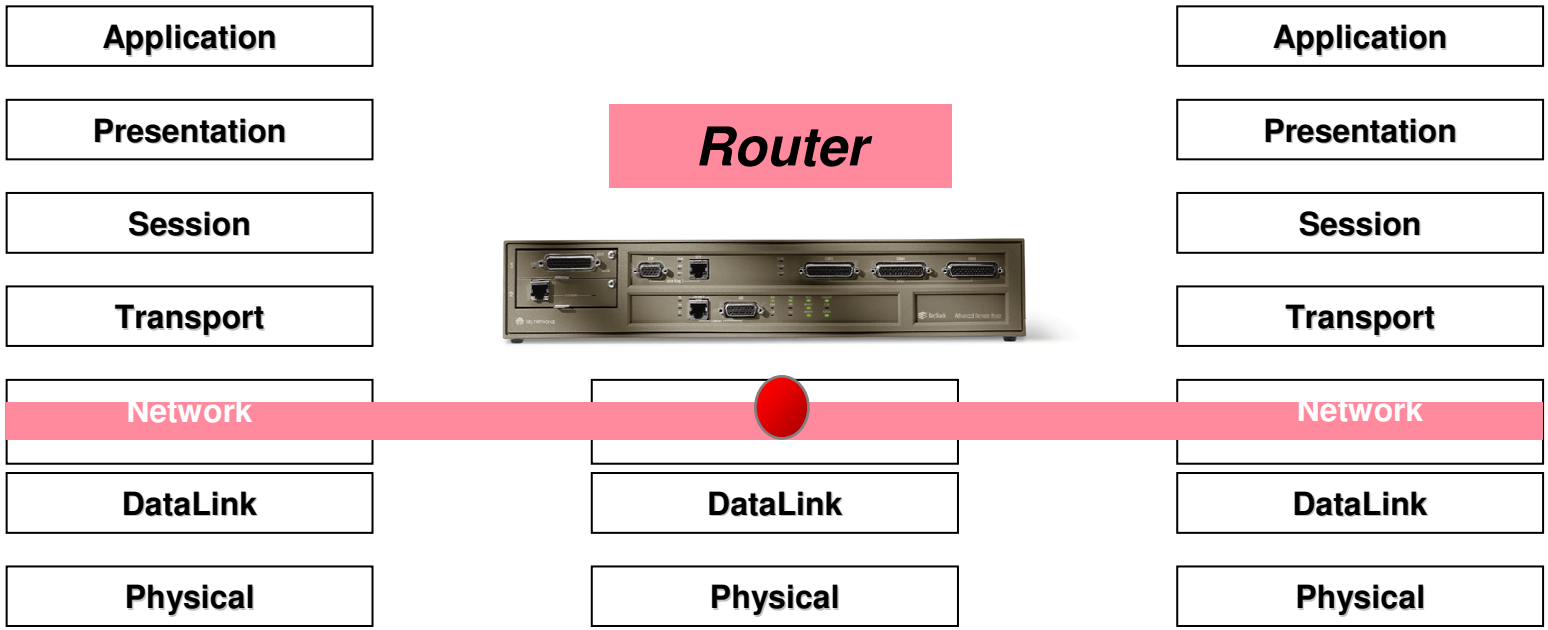
Enterprise Security



Firewall Funktionsweise

Application	**Application Layer Gateway**
Presentation	
Session	
Transport	
Network	**Packet Filtering**
	**Stateful Inspection Engine**
Data Link	
Physical	

Packet Filter (Überwachungsrouter)



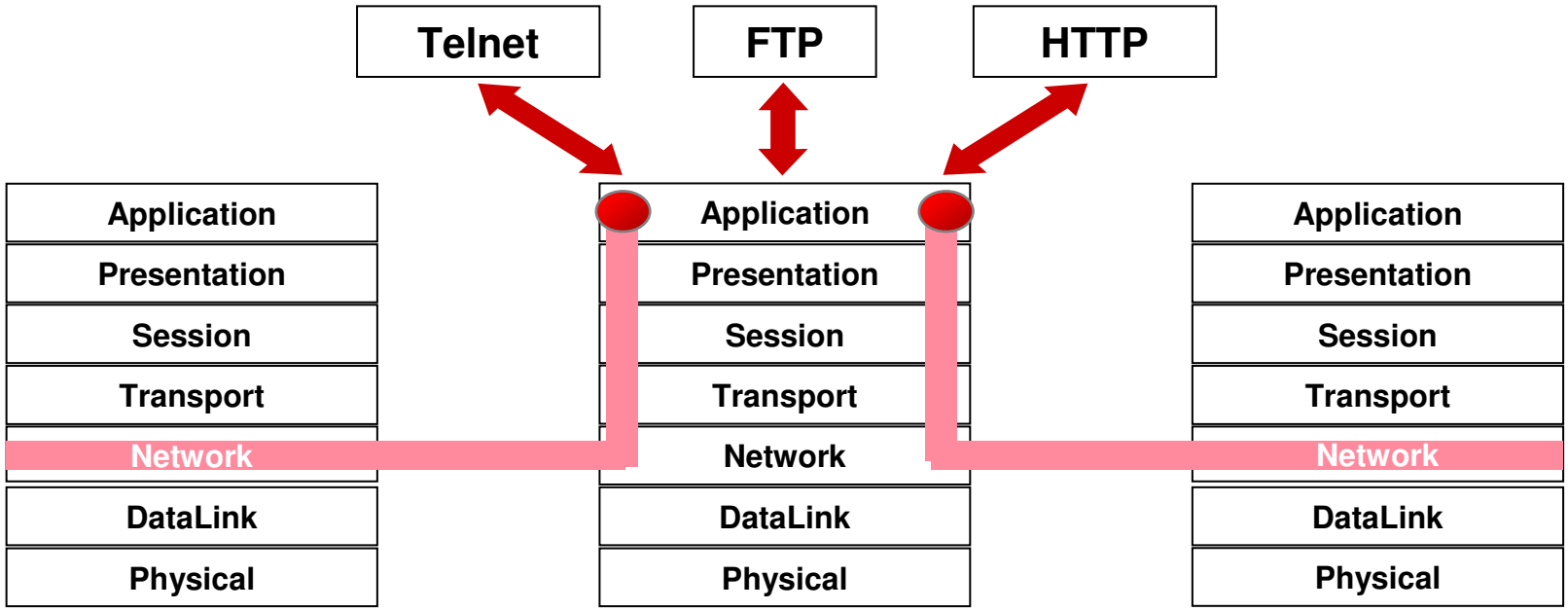
Vorteile:

- Einfach und billig
- Transparent für Applikationen

Nachteile:

- Geringe Security
- IP-Spoofing möglich
- ACLs schwer realisierbar
- Nicht erweiterbar

Application Layer Gateway



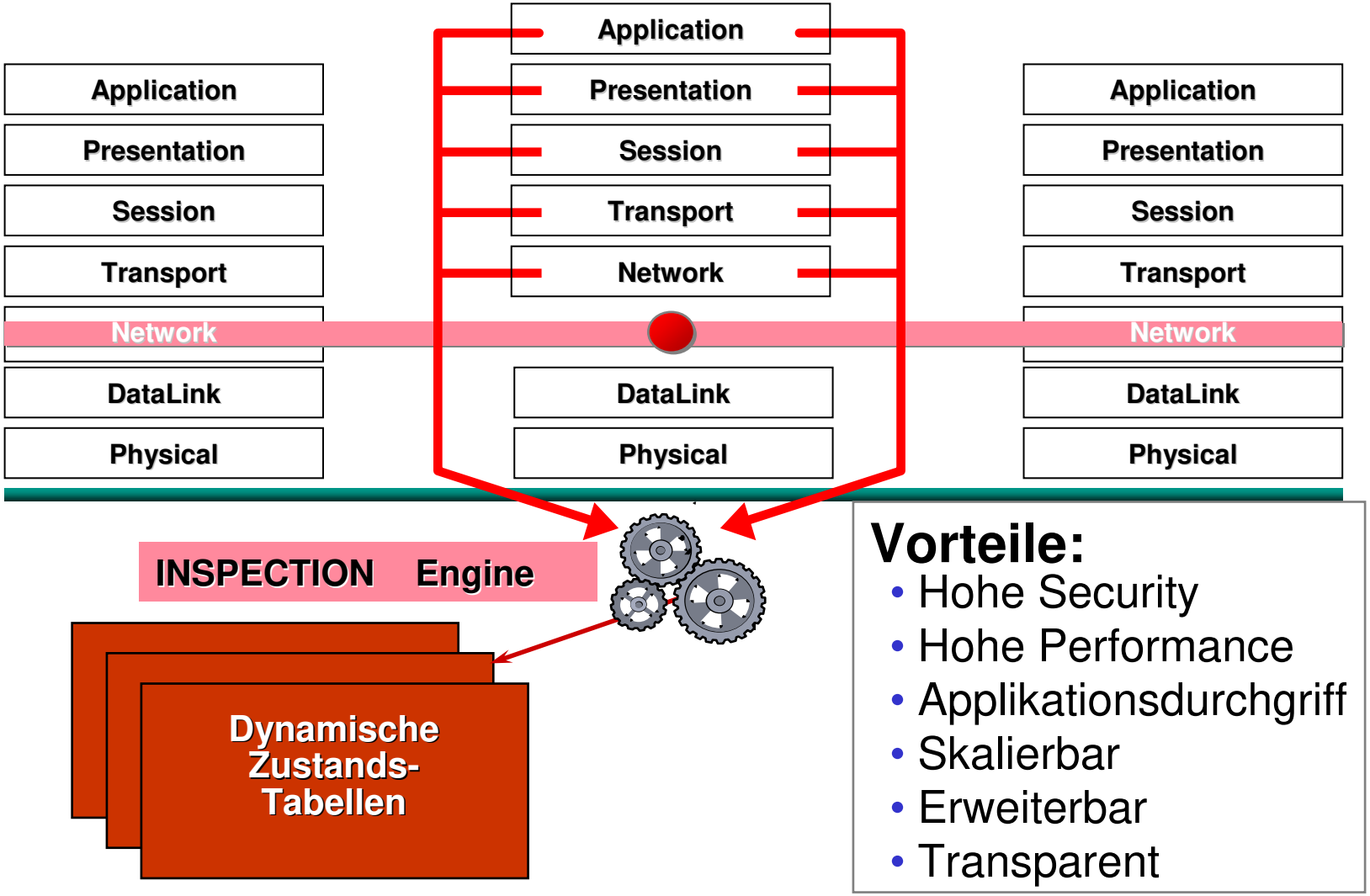
Vorteile:

- Hohe Sicherheit
- Im Bereich der Applikationen

Nachteile:

- Schwache Performance
- Nicht Transparent
- Schwer Skalierbar
- Eingeschränkte Applikationsunterstützung

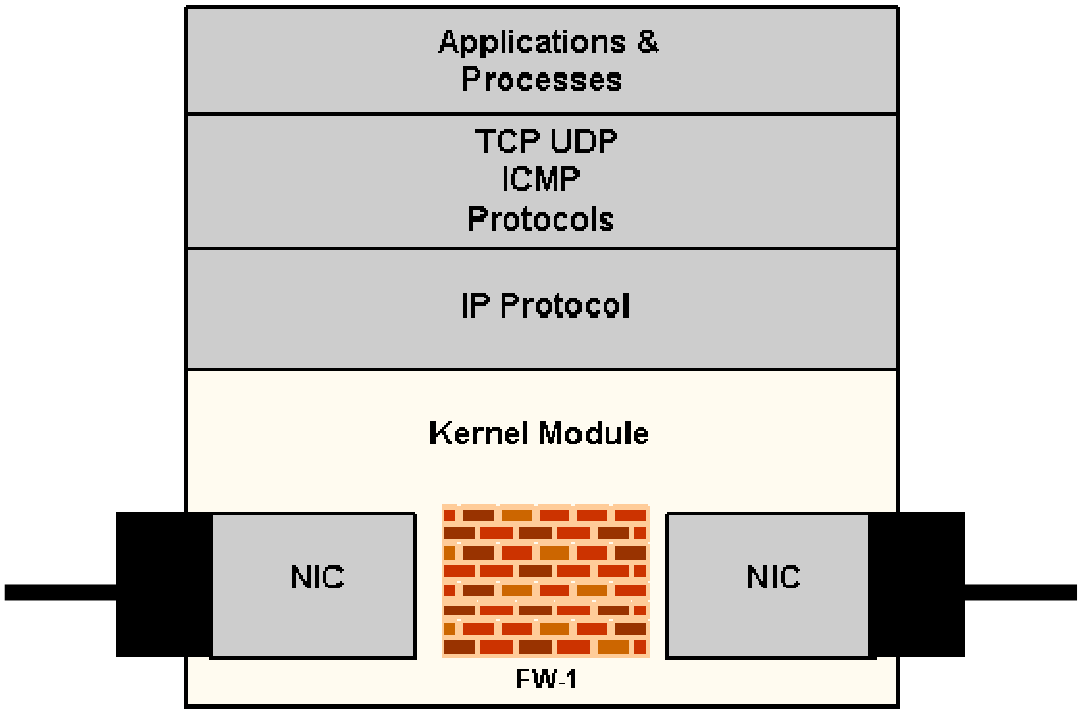
Stateful Inspection



Vorteile:

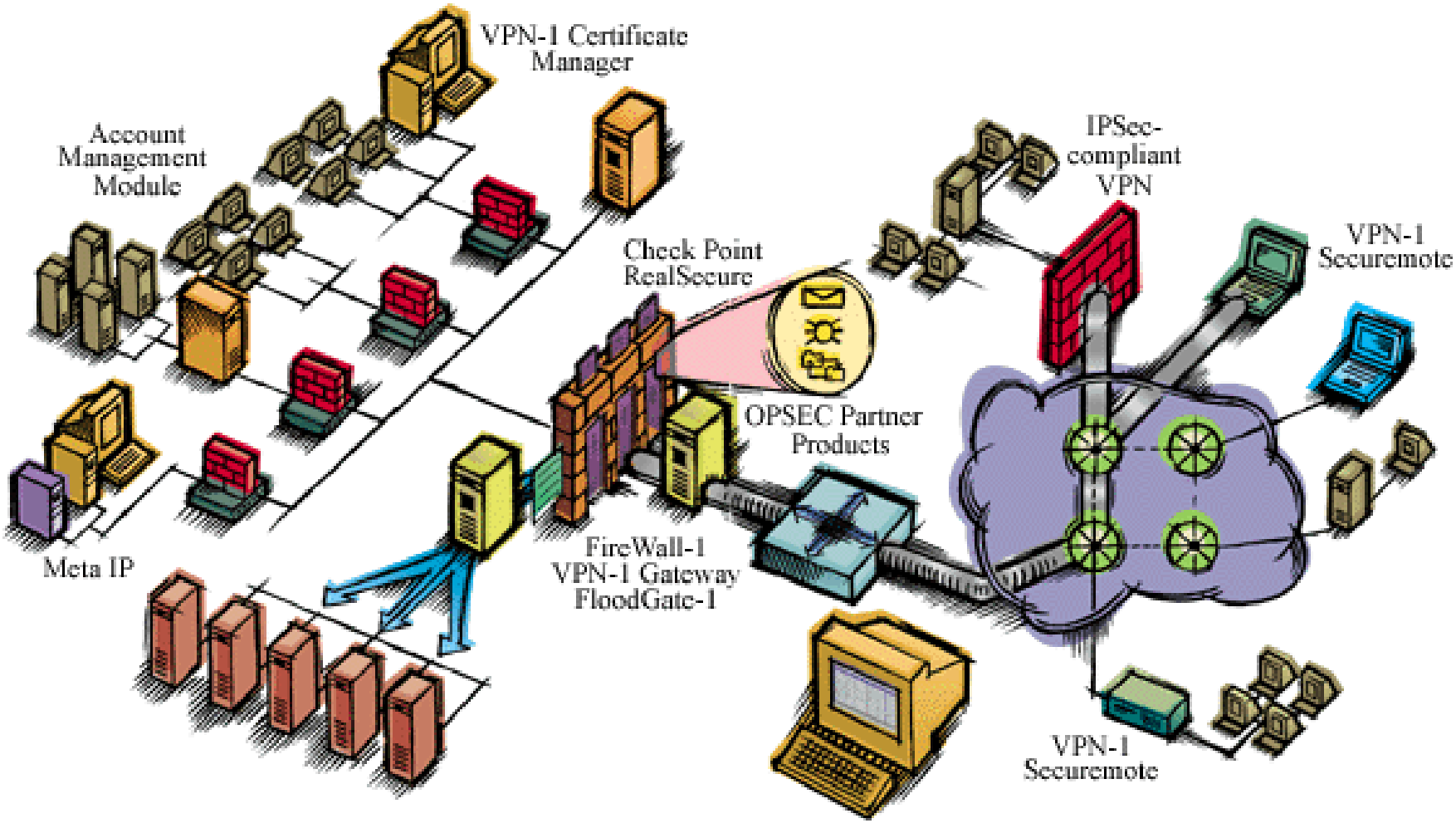
- Hohe Security
- Hohe Performance
- Applikationsdurchgriff
- Skalierbar
- Erweiterbar
- Transparent

Inspection Engine (FW-1)



- Ist im Kernel als Modul integriert
- Akzeptiert Pakete, wirft sie zurück oder vergißt sie
- Schont die Systemressourcen

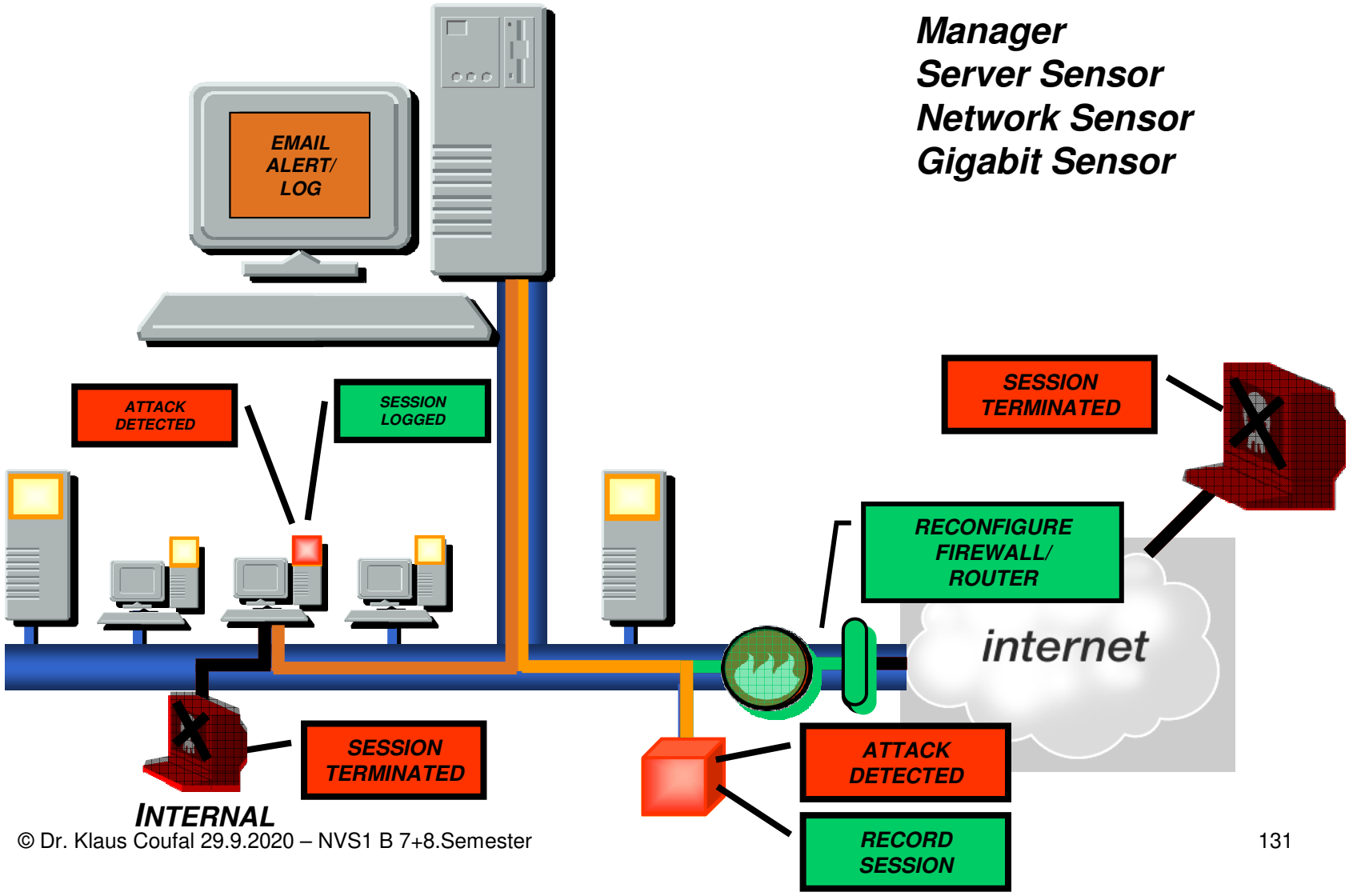
Firewall-1 Beispiel



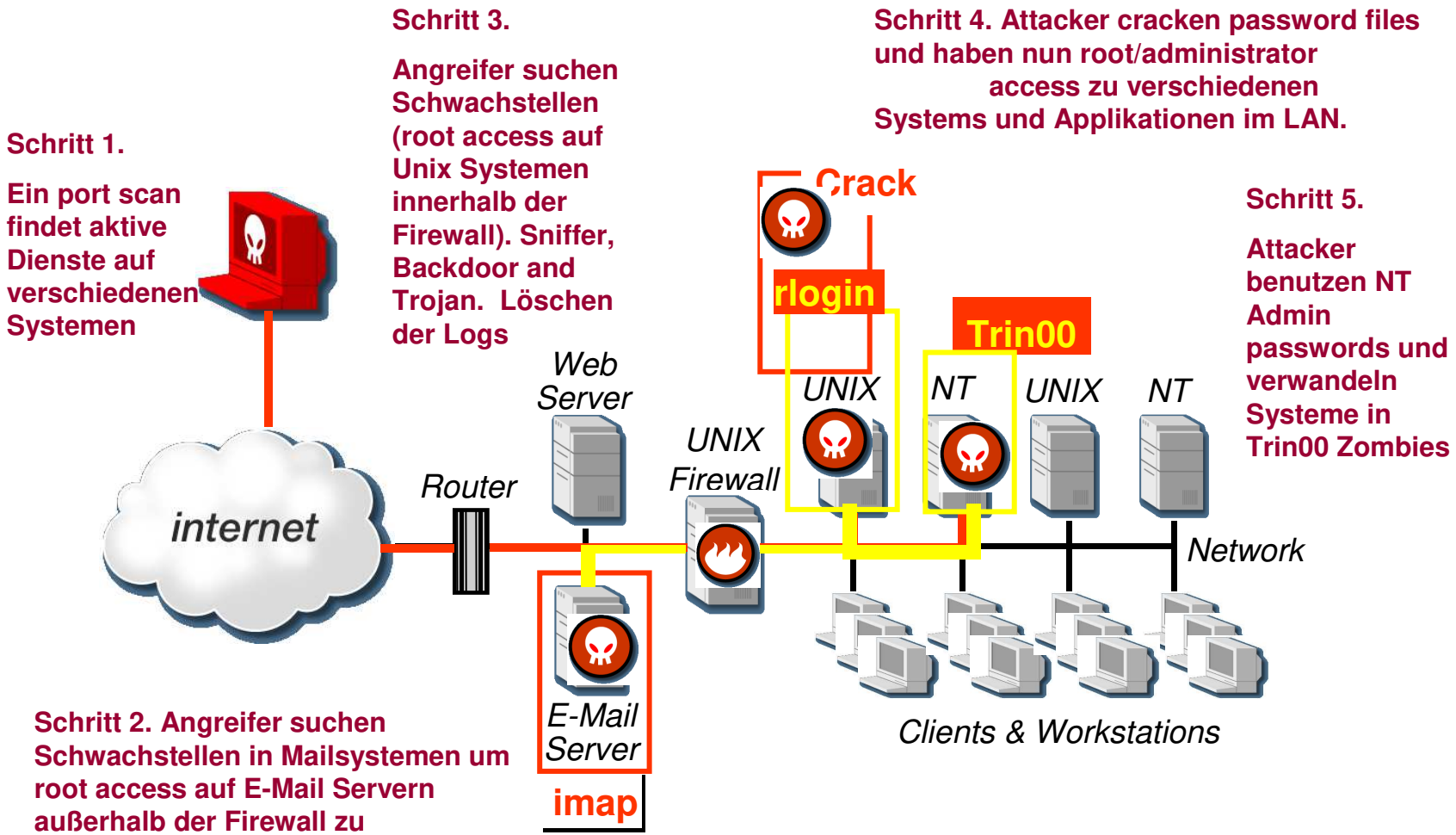
IDS

- Angriffsabwehrmanagement
- Hackerarbeitsweise
- IDS (Intrusion Detection System)
- IRS (Intrusion Response System)
- Arbeitsweise
- Honeypot

Angriffs-Abwehr- Management



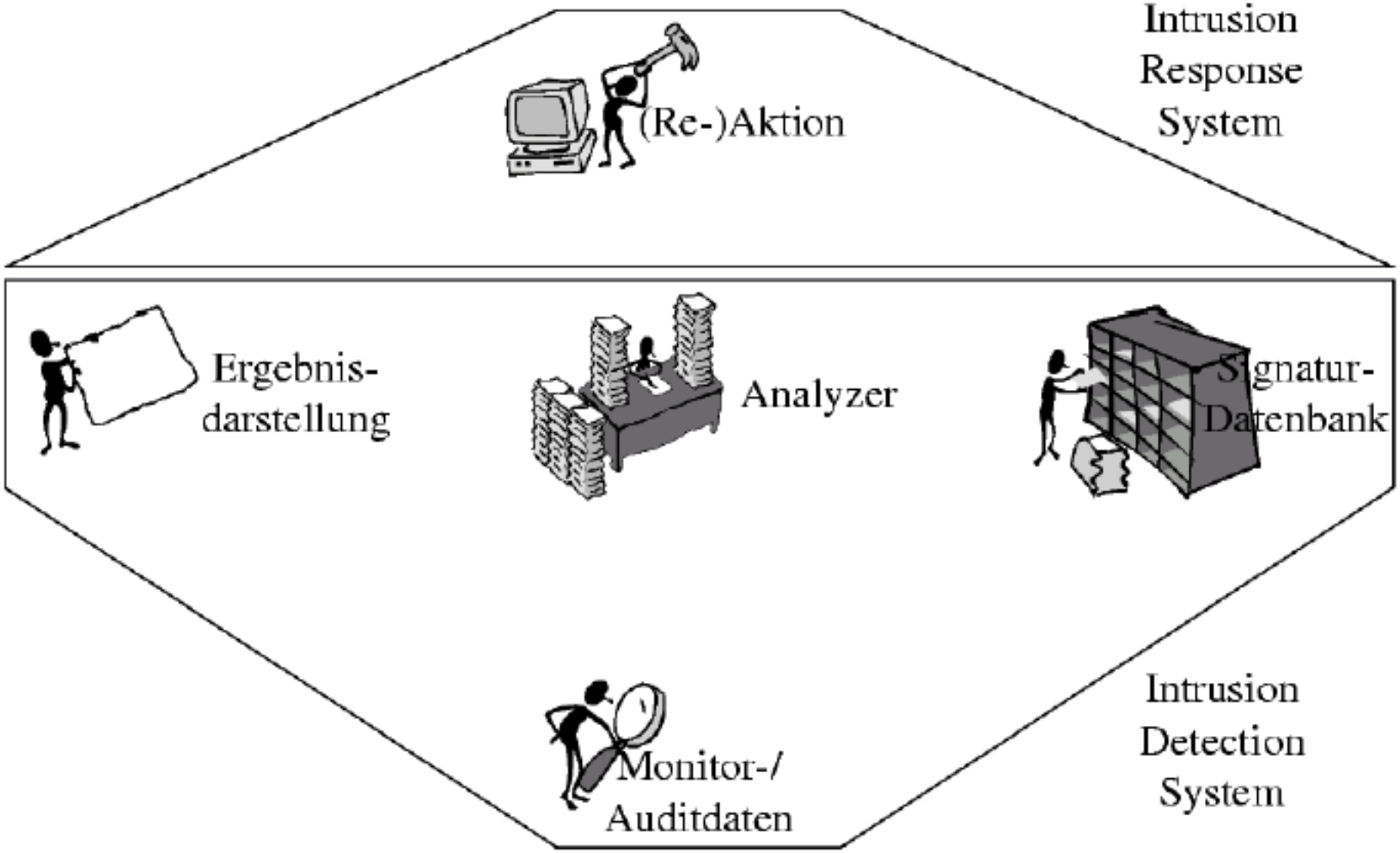
Hackerarbeitsweise



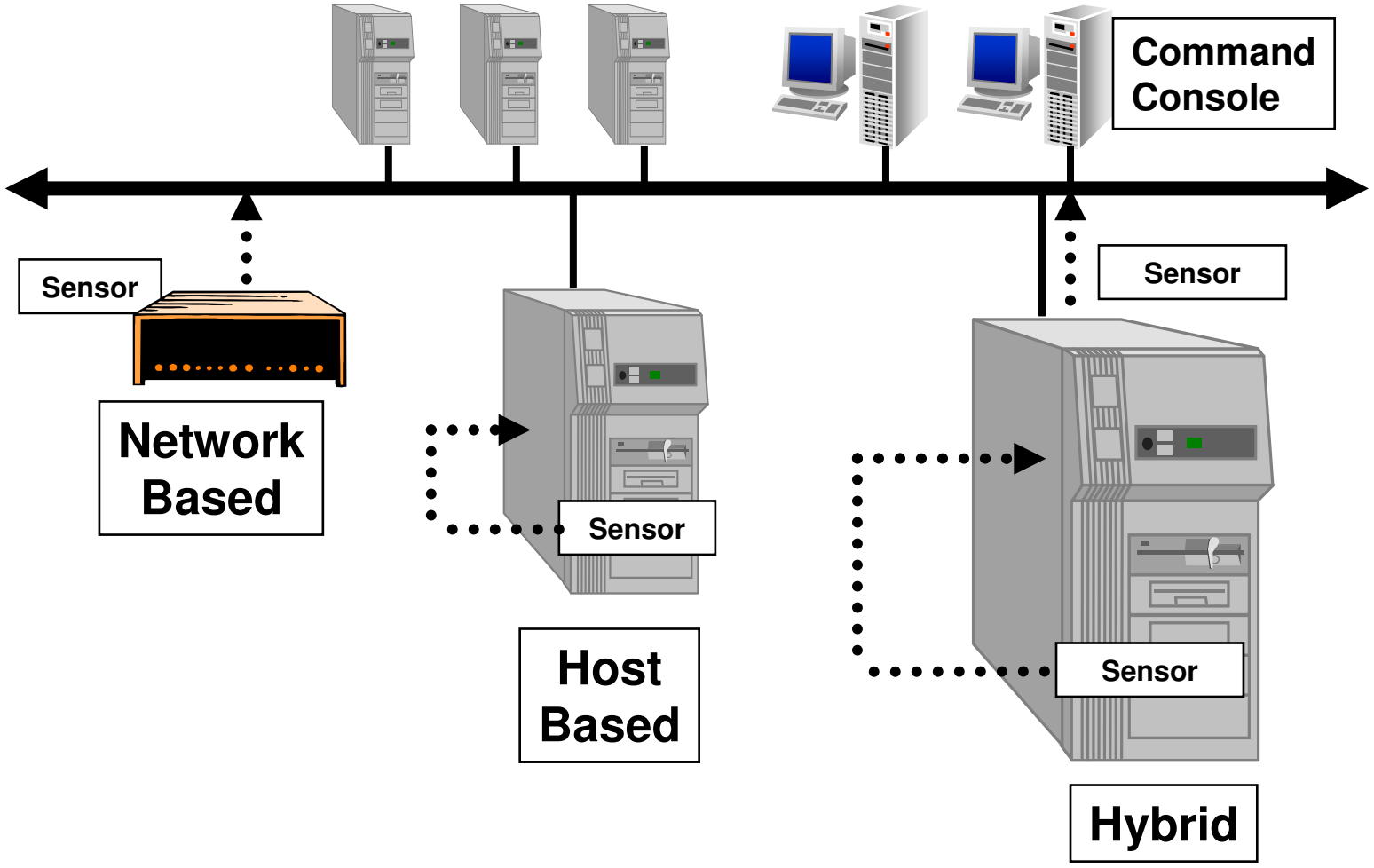
IDS – Definition

- Ein Intrusion Detection System ist ein Konglomerat von Möglichkeiten, Angriffe zu erkennen und – im Gegensatz zu statischen Firewallsystemen - darauf reagieren zu können.
- The ability to detect inappropriate, incorrect, or anomalous activity

IDS und IRS


Intrusion
Response
System
(Re-)Aktion
Ergebnis-
darstellung
Analyzer
Signatur-
Datenbank
Monitor-/
Auditdaten
Intrusion
Detection
System

Arbeitsweise



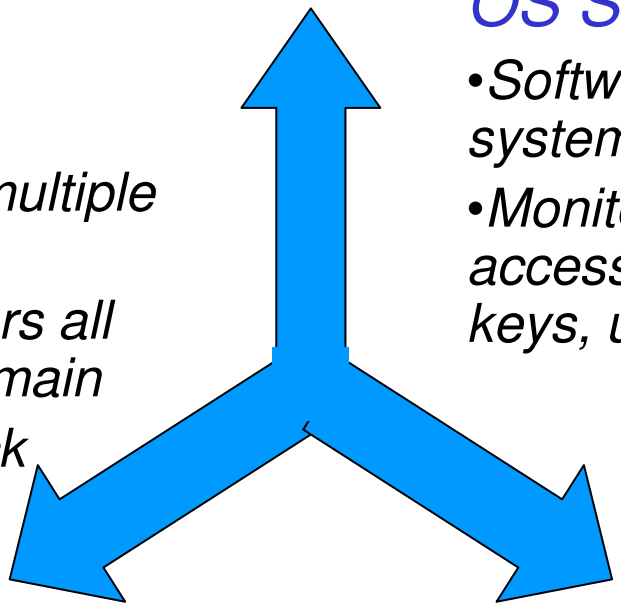
Host- bzw. Network-based

- Host based ID
  - Benötigt LOG-Files und auditing agents
  - Man muß Software auf das zu überwachende System laden
- Network based ID
  - Beobachtet den Netz-Traffic
  - Verwendet Daten-Pakete am Netz für die Informationsgewinnung

Sensoren

Network Sensor

- *Dedicated hardware/ software solution*
- *One sensor protects multiple systems*
- *Promiscuously monitors all traffic on a collision domain*
- *Diverse range of attack signatures*



OS Sensor

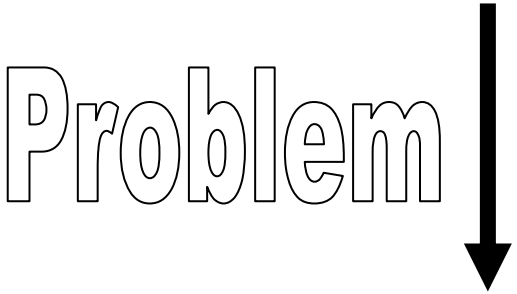
- *Software that runs on <u>each</u> system to be protected*
- *Monitors system logs, file access, port activity, registry keys, user activity*

Server Sensor

- *Combination of host and network sensors*
- *Software that runs on <u>each</u> system to be protected*
- *Tightly integrated with the TCP/IP stack to monitor all traffic to/from the system*

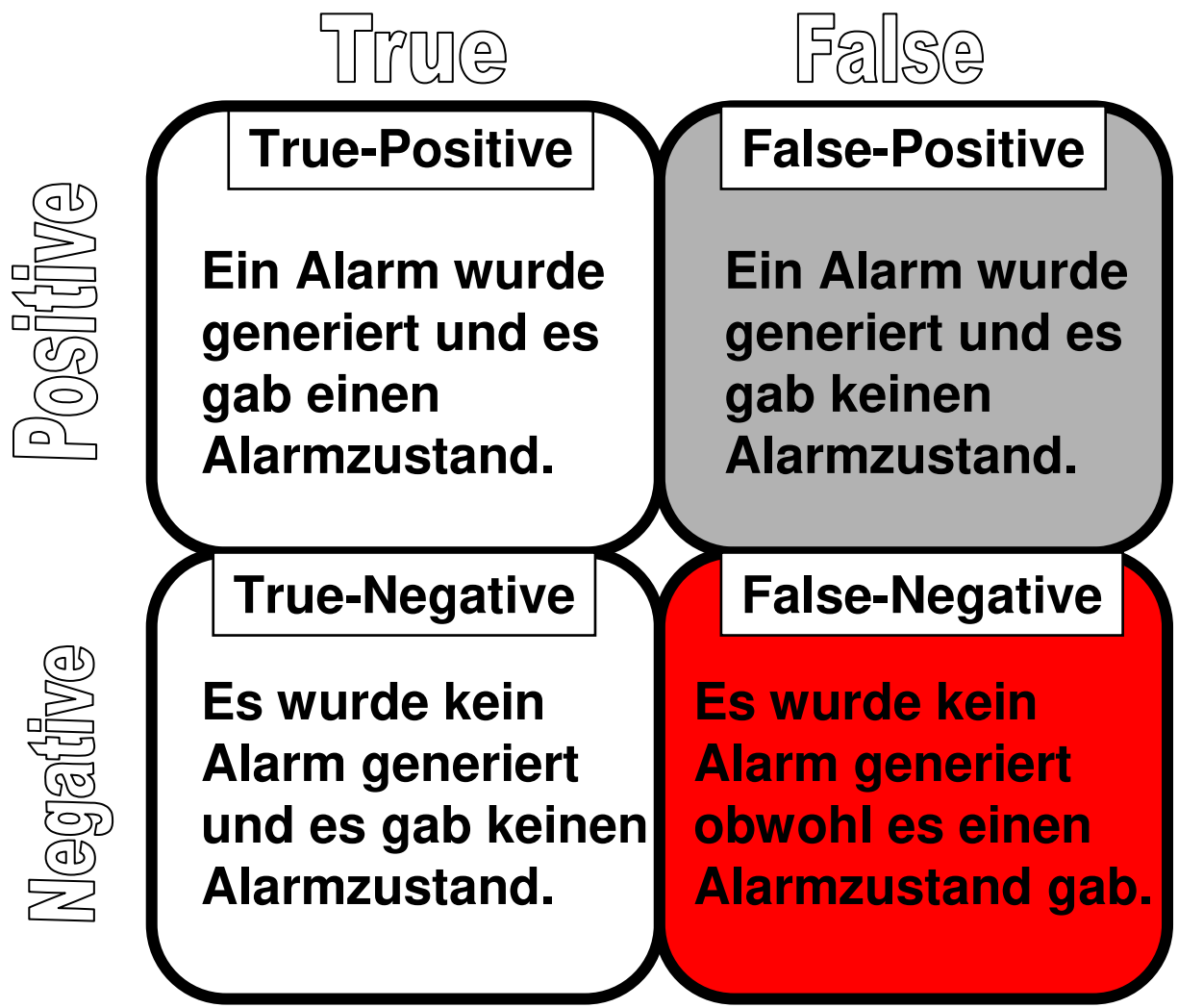
Fehlalarme 1

- **Intrusion Detection**
- **Misuse Detection**
- **Anomaly Detection**



- **False positiv**
- **False negative**

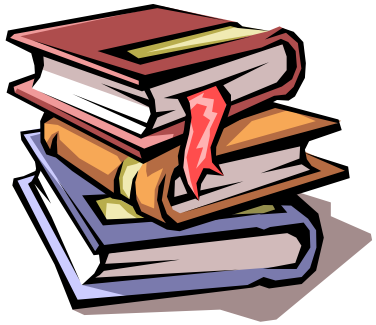
Fehlalarme 2

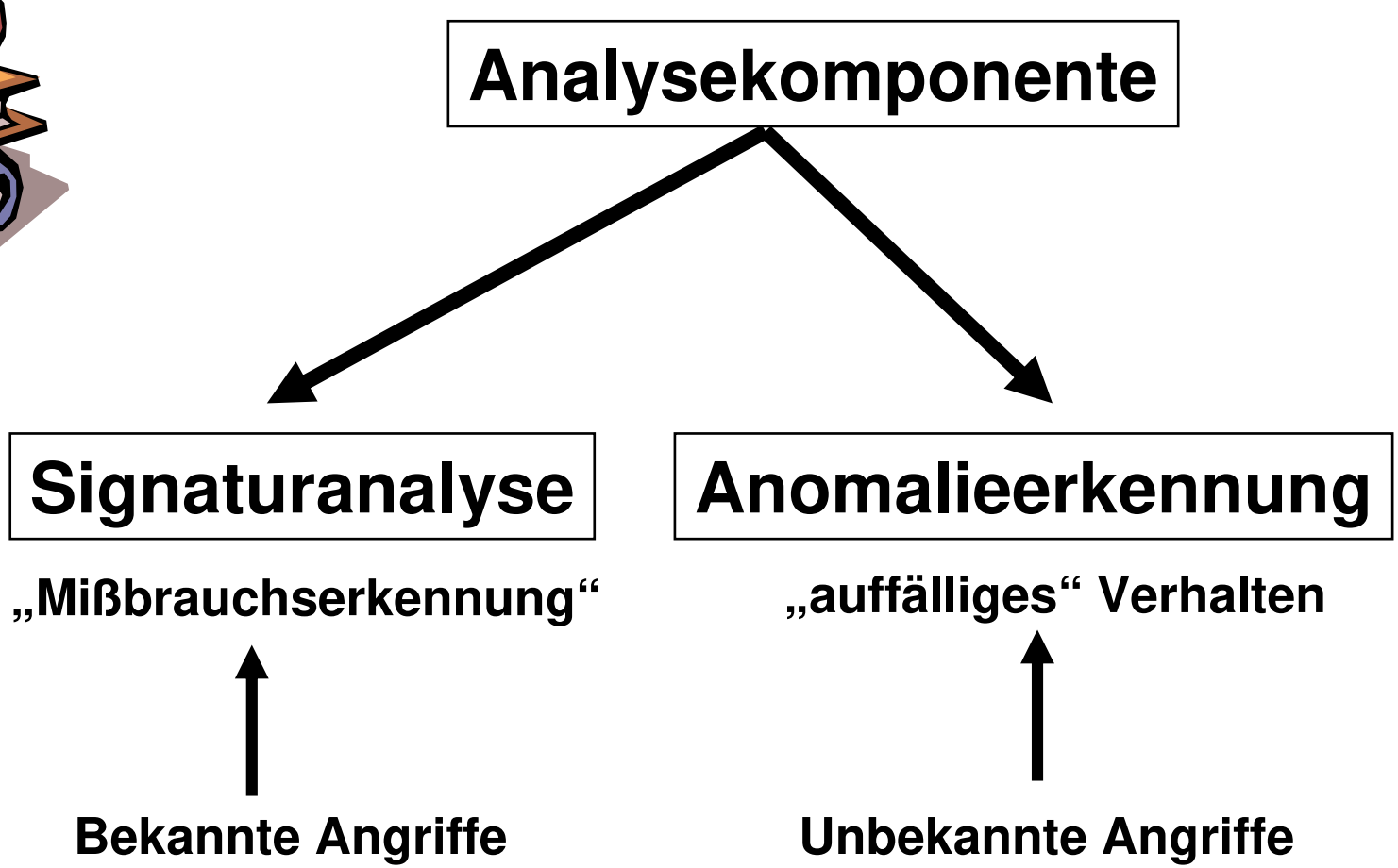


IDS – Varianten

- **Background Operation**
  Kein menschlicher Eingriff notwendig
- **On-demand Operation**
  Reaktion ja – aber erst nach Operator-Anforderung
- **Scheduled Operation**
  „On-Demand“, zu definierten Zeitpunkten
- **Real-Time Operation**
  Automatische Reaktion in Minuten oder Sekunden
- **24*7 Monitoring**
  Ständiges „human“ controlling für neue Situationen
- **Incident Response**
  Reaktion auf Meldungen von „außen“

IDS Aufgaben





Anomalieerkennung 1

Anforderungen:

- **Echtzeitfähigkeit**

  Schnelle Reaktion ist notwendig, da Intruder ihre Spuren verwischen

- **Adaptivität**

  Profile und Schwellwerte müssen ständig aktualisiert werden

- **Einfache Konfiguration**

  Mittels „Default-Profile“ muß ein schnelles Umkonfigurieren möglich sein.

Anomalieerkennung 2

Profilarten-1

- **Benutzerprofile**

  Individuelle Arbeitsprofile, die bei jeder Benutzeraktion aktualisiert werden

  Bsp: **CPU-Auslastung**
  Tippgeschwindigkeit
  Art & Häufigkeit der verwendeten Programme
  bevorzugte Arbeitszeit

- **Benutzergruppenprofile**

  Zusammenfassung von Benutzern mit ähnlichen Arbeitsmustern

Anomalieerkennung 3

Profilarten-2

- **Ressourcenprofile**

  Beschreibung systemweiter, benutzerunabhängiger System-ressourcen.

  Bsp: Speicherbedarf
  Dateizugriffe
  I/O-Aktivitäten an Ports
  verwendete Protokolle

- **Prozeßprofile**

  Überwachung der Systemprozesse, speziell, wenn sie keinem Benutzer zugeordnet sind (z.B: Hintergrundprogramme)

- **statische Benutzerprofile**

  Benutzerprofile, die nur in unregelmäßigen Abständen aktualisiert werden (gegen langsame, gezielte Benutzerveränderung der Hacker)

Anomalieerkennung 4

- **Operationales Modell**

  „Schwellwert-Modell“ – ein Alarm wird ausgelöst, wenn eine Variable einen bestimmten Wert erreicht (z.B.Loginversuche).

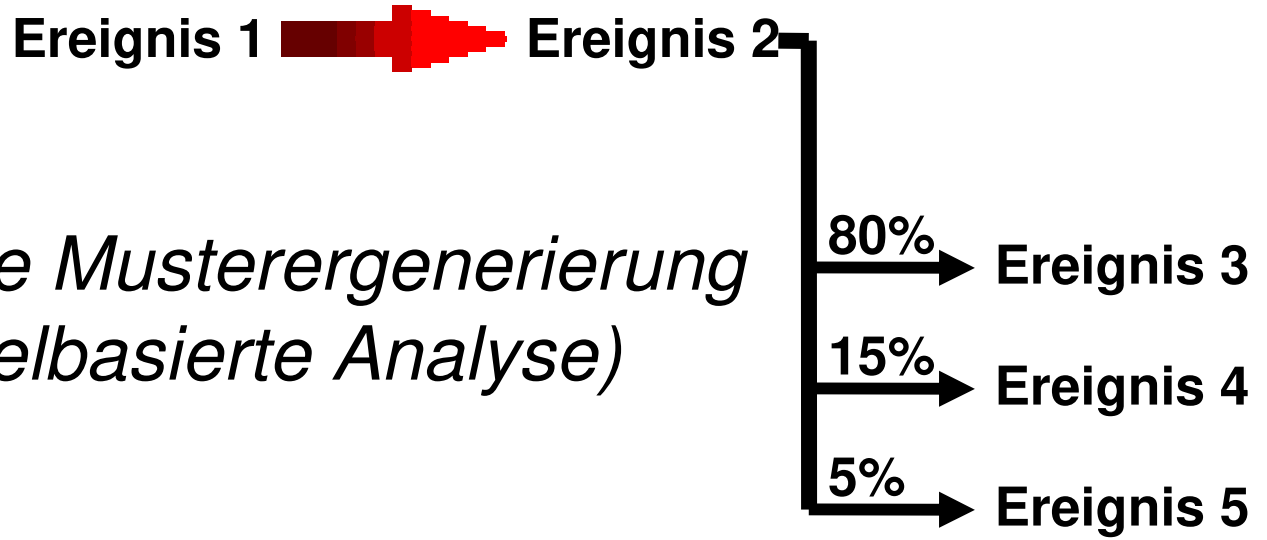
- **Modell von Mittelwert und Standardabweichung**

  Ein Alarm wird ausgelöst, wenn sich eine Beobachtung nicht in einem „Konfidenzintervall“ befindet.

- **Modell von Zeitreihen**

  Die Zeit, zu der ein Ereignis eintritt, fließt in die Entscheidung mit ein

Anomalieerkennung 5



Prädiktive Musterergenerierung
(Regelbasierte Analyse)

Regel:

Wenn das Ereignis 2 unmittelbar nach dem Ereignis 1 eingetreten ist, dann folgt Ereignis 3 mit einer Wahrscheinlichkeit von 80%, Ereignis 4 mit 15% und Ereignis 5 mit 5%

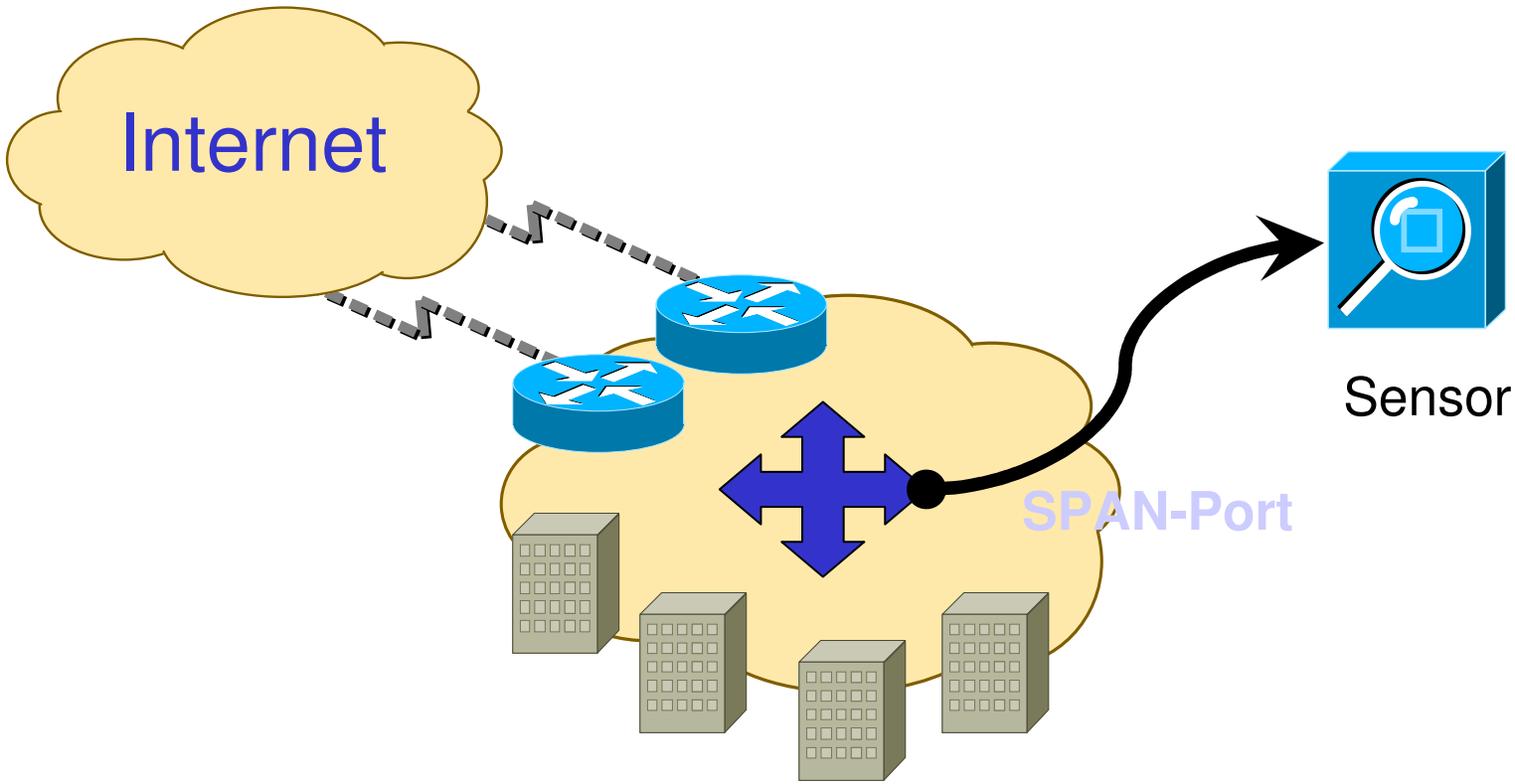
IDS und Neuronale Netze

Lernphase → **Vorhersagephase**

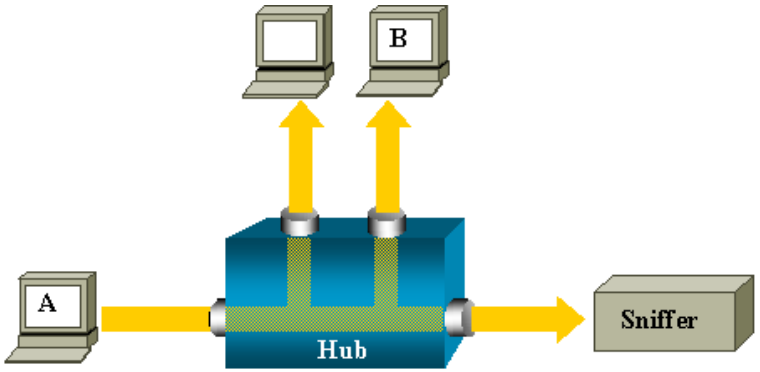
Vorteil:
- **Kann auch mit „verrauschten“ Daten umgehen**

Nachteil:
- **Benötigt viel „trial &error“ in der Lernphase**
- **Angreifbar in der Lernphase**

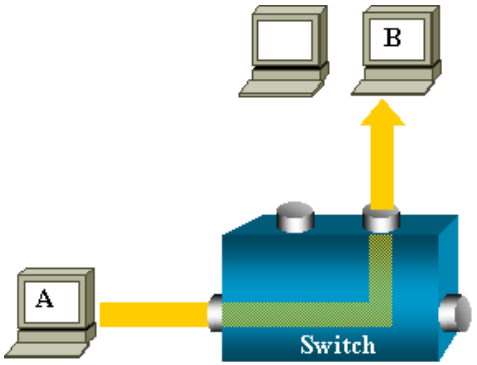
IDS – Beispiele 1



IDS – Beispiele 2

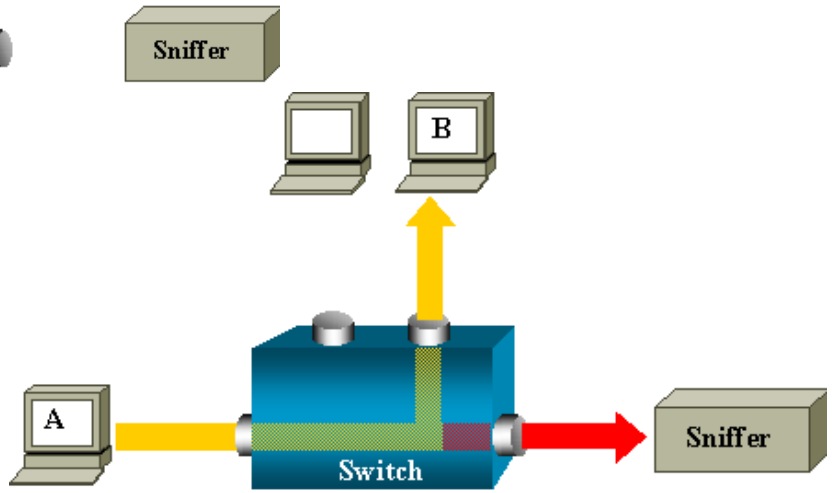


HUB

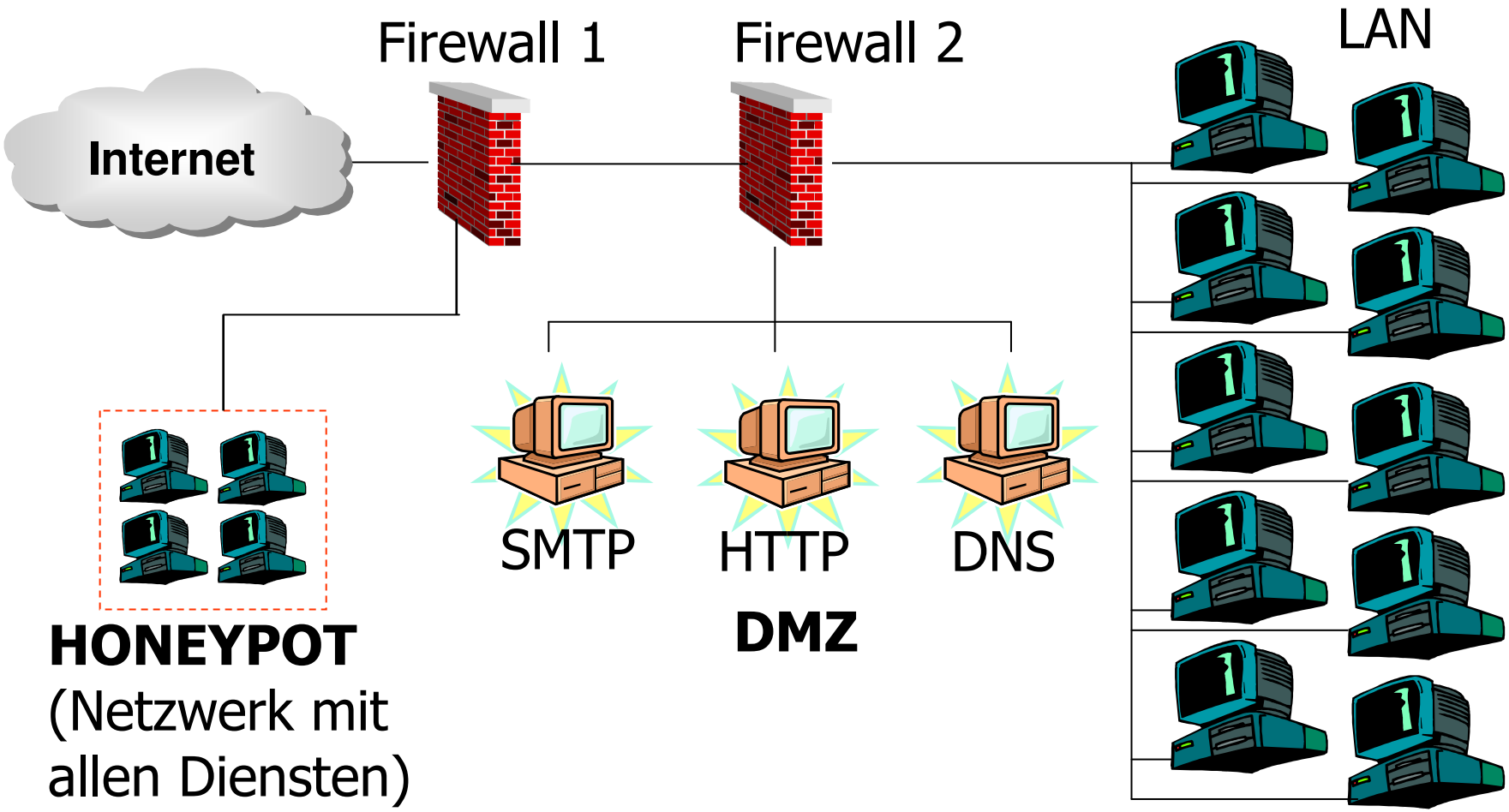


Switch

Switch mit SPAN-Port



Honeypots


Firewall 1
Firewall 2
LAN
Internet
SMTP
HTTP
DNS
DMZ
HONEYPOT
(Netzwerk mit
allen Diensten)

Labor zu Sicherheitsverwaltung

Auswerten von Logfiles zu einem Sicherheitsbericht

- Welche Logfiles
  - Syslog, Windows Ereignisanzeige
  - Firewall, Intrusion Detection System
- Auswerten der Logfiles
- Erstellen des Sicherheitsberichts

Betriebsparameter

- Betriebsparameter definieren
- Beispiele für Betriebsparameter
- Wichtige Betriebsparameter erkennen
- Wichtige Betriebsparameter überwachen
- Konsequenzen aus der Überwachung

Betriebsparameter – Definition

- Ein Betriebsparameter ist eine Eigenschaft, die den „Betrieb“ charakterisiert
- Oft durch eine Kennzahl oder eine Kenngröße beschrieben
- Abweichungen von den Sollwerten haben i.a. negative Auswirkungen auf den Betrieb

Betriebsparameter – Gliederung

- Hardware
- Dienste
- Betriebssysteme
- Applikationen
- Protokolle
- …

Betriebsparameter – Hardware

- Computer
- Router
- Switch
- Kabel
- Access Point
- Appliance (Firewall, NAS, …)
- …

Betriebsparameter – Dienste

- Webserver
- Mailserver
- VoIP-Server
- Verzeichnisdienst
- DB-Server
- Groupwareserver
- …

Betriebsparameter – OS

- Clientbetriebssysteme
- Serverbetriebssysteme
- Routerbetriebssysteme
- Switchbetriebssysteme
- Appliancebetriebssysteme
- „Telephon“betriebssysteme
- …

Betriebsparameter - Applikationen

- Büroapplikationen (Text, Tabellenkalkulation, Präsentation, …)
- DB-Applikationen (CRM, …)
- Groupware (Kalender, Workflow, …)
- Geschäftsanwendungen
- „Netz“-Anwendungen (Browser, Mail, …)
- Entwicklungsapplikationen (Compiler, …)
- … (Antimalwareprogramme, …)

Betriebsparameter – Protokolle

- Protokolle der Anwendungsschicht
- Protokolle der Transportschicht
- Protokolle der Netzwerkschicht
- Protokolle der Datenverbindungsschicht
- …

Betriebsparameter – Beispiele1

Betriebsparameter – Auswahl

- Kriterien erstellen, nach denen aus der Unzahl von Parametern, die ausgewählt werden, die wichtig (=notwendig für den Betrieb der Firma (Institution, …)) sind.
- Meßbarkeit dieser Parameter sicherstellen
- Definition von Grenzwerten

Betriebsparameter – Überwachung

- Festlegung, wie diese Parameter überwacht und protokolliert werden
- Automatisch / Manuell
- Laufend („Trap“gesteuert) / Zyklisch / Bei Bedarf
- Zentral / Dezentral
- Verantwortung für Überwachung

Betriebsparameter – Konsequenzen

- Passiv: Logfiles bzw. Datenbank
- Aktiv: Meldung bzw. Alarmierung
- Meldung: Popup, Mail, SMS, …
- Einstufung der „Dringlichkeit“
- Reaktion?
  - Wer, Wie, Wann
  - Protokollierung der Reaktion

Überwachungsframeworks1

- HP OpenView (HP Openview)
- IBM Tivoli (IBM Tivoli)
- CA NSM (CA NMS)
- Microsoft Operation Manager (MOM)

Überwachungsframeworks2

- NAGIOS (nagios.org)
- ZENOSS (zenoss.org)
- OpenNMS (opennms.org)
- Cacti (cacti.net)
- Zabbix (zabbix.com)
- Spiceworks (spiceworks.com)
- …

NAGIOS

- Was ist das
- Geschichte
- Features
- Versionen
- Umsetzung

NAGIOS

- Nagios
- Ain't
- Gonna
- Insist
- On
- Sainthood

NAGIOS – Fakten

- OSS (Open Source Software)
- Autor: Ethan Galstad (et al.)
- NAGIOS ist ein Monitoring System zur Überwachung der IT-Infrastruktur
- Schwierigkeiten sollen erkannt werden, bevor sie „kritisch“ werden (d.h. den Betrieb stören)

NAGIOS – Geschichte1

- 1996
  - MS-DOS Applikation, um die Funktion von Netwareservern zu überwachen
- 1998
  - Umstellung auf Linux
- 1999
  - Open Source Projekt „NetSaint“
  - Plugins

NAGIOS – Geschichte2

- 2002
  - Aus Markenschutzrechtlichen Gründen Umbenennung in NAGIOS
- 2007
  - Gründung der Nagios Enterprises, LLC zur Vermarktung der Dienstleistung und der Entwicklung

NAGIOS – Features1

- Umfangreiche Überwachung
  - Überwachung von Applikationen, Diensten, Betriebssystemen, Netzwerkprotokollen, Infrastrukturkomponenten
  - Script APIs erlauben die Überwachung von Nichtstandardkomponenten (z.B.: eigener Software)

NAGIOS – Features2

- Aufbereitete Darstellung
  - Zentrale Sicht auf die gesamte IT-Infrastruktur
  - Detailierte Statusinformationen im Webinterface

NAGIOS – Features3

- Wahrnehmung
  - Rasches Erkennen von Ausfällen einer Infrastrukturkomponente
  - Alarme an die zuständigen Personen mittels eMail oder SMS
  - Eskalationsmöglichkeiten sichern die Zustellung eines Alarms an die richtigen Personen

NAGIOS – Features4

- Problembeseitigung
  - Alarmbestätigung erleichtern die Kommunikation und die Problembearbeitung
  - Ereignissteuerungen (Event handler) erlauben den automatischen Neustart von Anwendungen oder Diensten

NAGIOS – Features5

- „Proaktive“ Planung
  - Erweiterungen zur Beobachtung von Trends und zur Planung der Kapazität bewahren vor Engpässen
  - Geplante Ausfälle erlauben die Verhinderung von Alarmen während des Ausbaus der Infrastruktur

NAGIOS – Features6

- Auswertung und Berichte
  - Verfügbarkeitsberichte sichern die Einhaltung von SLAs.
  - Historische Berichte bieten die Auszeichnung von Alarmen, deren Bestätigung und Beantwortung
  - „Addons“ erweitern die Berichtsmöglichkeiten

NAGIOS – Features7

- Mandantenfähigkeit
  - Der mehrbenutzerfähige Zugriff erlaubt allen Berechtigten die jeweilige Sicht auf ihre Infrastruktur
  - Benutzerspezifische Darstellungen sichern die jeweils notwendige Detailtiefe

NAGIOS – Features8

- Erweiterbare Architektur
  - Einfache Integration in Anwendungen von Drittanbietern mit Hilfe mehrere APIs
  - Erweiterung der Basisfunktionalität durch zahlreiche AddOns (nicht alle OSS)

NAGIOS – Features9

- Bewährte und stabile Plattform
  - Mehr als 10 Jahre in Entwicklung/am Markt
  - Skalierbar auch für viele zu überwachende Knoten (mehrere 1000)
  - Redundanz zur Ausfallssicherung garantiert die unterbrechungsfreie Überwachung kritischer IT-Infrastruktur

NAGIOS – Features10

- Dynamische „Community“
  - Mehr als eine Million Benutzer
  - Kostenloser Support über Mailinglisten
  - Viele Addons

NAGIOS – Features11

- Anpassbarer Programmcode
  - OSS (Open Source Software)
  - GPL (General Public License)
  - Damit Zugriff auf den gesamten Quellcode

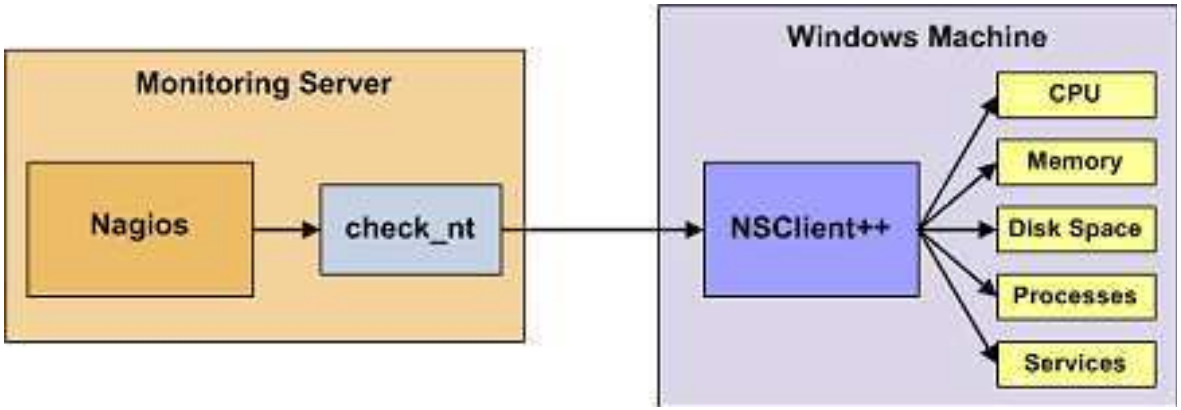
NAGIOS – Installation

- Die Installation erfolgt abhängig von eingesetzten Betriebssystem am Überwachungsserver
- Schnellstartanleitungen für verbreitete Betriebssysteme (z.B.: OpenSuSE, Ubuntu, …) finden Sie unter:

  https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.html

Nagios – Überwachung1

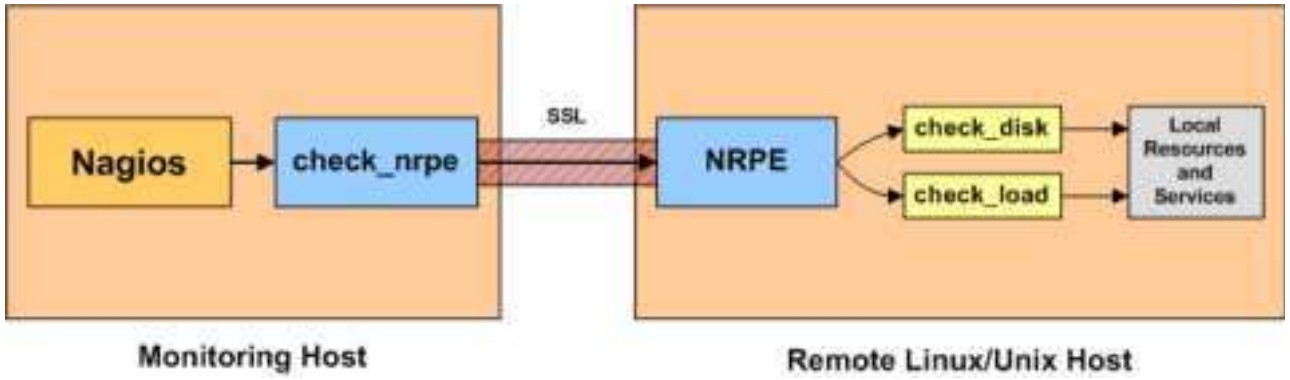
- Agent (NSClient++, NC_Net, …) am Zielsystem Windows



Quelle: http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf

Nagios – Überwachung2

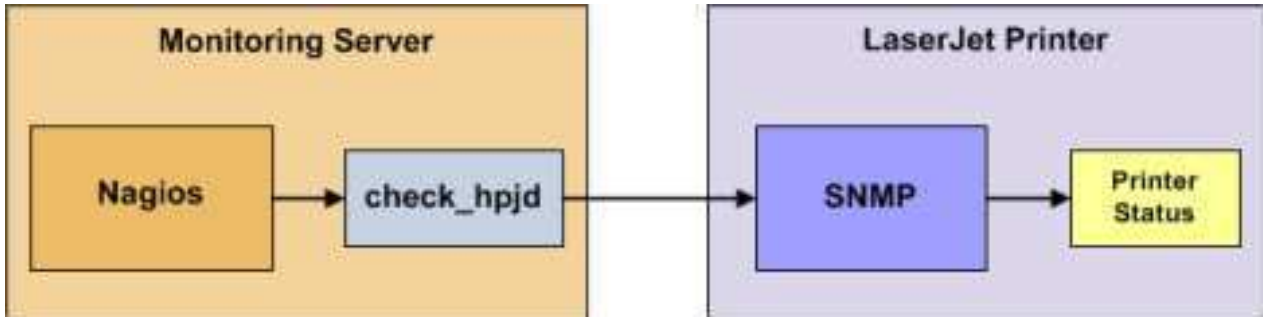
- Agent am Zielsystem Linux



Quelle: http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf

Nagios – Überwachung3

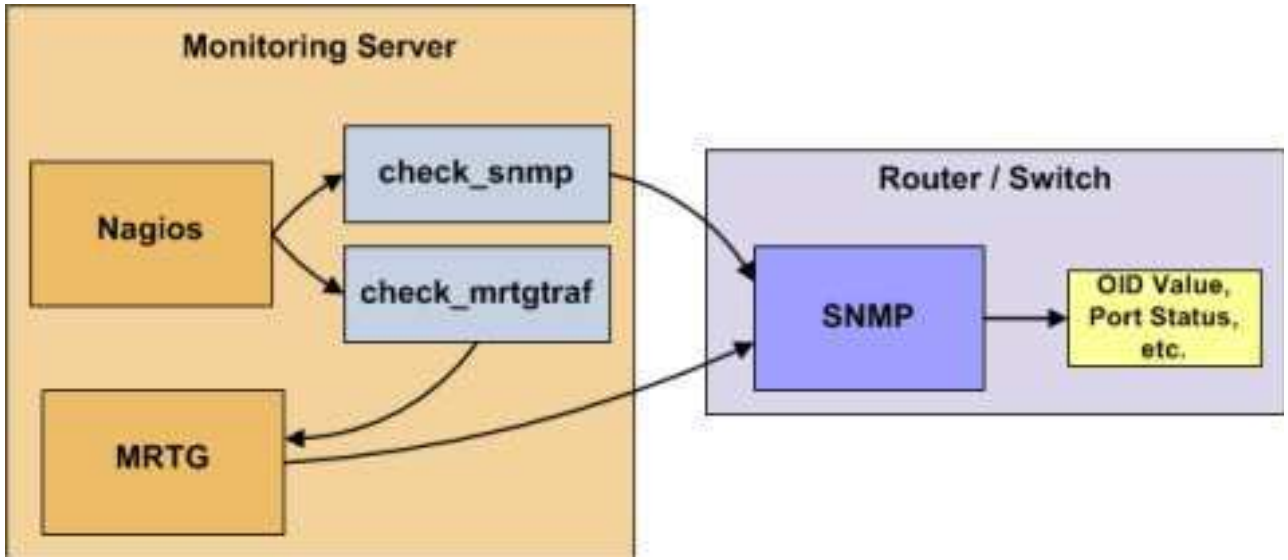
- Agent am Zielsystem Netzwerkdrucker



Quelle: http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf

Nagios – Überwachung4

- Agent am Zielsystem Router/Switch



Quelle: http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf

NAGIOS – Überwachung5

- Konfiguration am NAGIOS-Server
  - ev. Installieren bzw. Aktivieren eines PlugIns (z.B.: check_nt;)
  - Definition des neuen Hosts (unter Verwendung von Templates)
  - ev. Definiton neuer Dienste
  - Neustart des NAGIOS-Dämons

NAGIOS – WinBeispiel1

- Host Definition:

/usr/local/nagios/etc/objects/windows.cfg

```
define host{
use windows-server ; (Template)
host_name spgsrv
alias Server_Spengergasse1
address 192.168.1.2
}
```

NAGIOS – WinBeispiel2

- Überwachung der Agentversion

```
define service{
use generic-service
host_name spgsrv
service_description NSClient++ Version
check_command
   check_nt!CLIENTVERSION
}
```

NAGIOS – WinBeispiel3

- Überwachung der Laufzeit

```
define service{
use generic-service
host_name spgsrv
service_description Uptime
check_command check_nt!UPTIME
}
```

NAGIOS – WinBeispiel4

- Überwachung der CPU-Auslastung

```
define service{
use generic-service
host_name spgsrv
service_description CPU Load
check_command check_nt!CPULOAD!-l 5,80,90
}
```

- 5=5Minuten, 80%=Warnung, 90%=Critical

NAGIOS – PrintBeispiel1

- Host Definition:

/usr/local/nagios/etc/objects/printer.cfg

```
define host{
use generic-printer ; (Template)
host_name hplaserjetnet
alias HP LaserJet 4000 dn
address 192.168.1.30
hostgroups allhosts
}
```

NAGIOS – PrintBeispiel2

- Überwachung des Druckers

```
define service{
use generic-service
host_name hplaserjetnet
service_description Printer Status
check_command check_hpjd!-C public
normal_check_interval 10
retry_check_interval 1
}
```

NAGIOS – SwitchBeispiel1

- Host Definition:

/usr/local/nagios/etc/objects/switch.cfg

```
define host{
use generic-switch ; (Template)
host_name cisco-2960-253
alias Cisco 2960 Switch Etage1
address 192.168.1.253
hostgroups allhosts,switches
}
```

NAGIOS – SwitchBeispiel2

- Überwachung des Pingverhaltens

```
define service{
use generic-service ; (Template)
host_name cisco-2960-253
service_description PING
check_command check_ping!200.0,20%!600.0,60%
normal_check_interval 5
retry_check_interval 1
}
```

NAGIOS – WebBeispiel1

- Host Definition:

/usr/local/nagios/etc/objects/hosts.cfg

```
define host{
use generic-host ; (Template)
host_name webserver
alias External Web Server
address 192.189.51.21
hostgroups allhosts
}
```

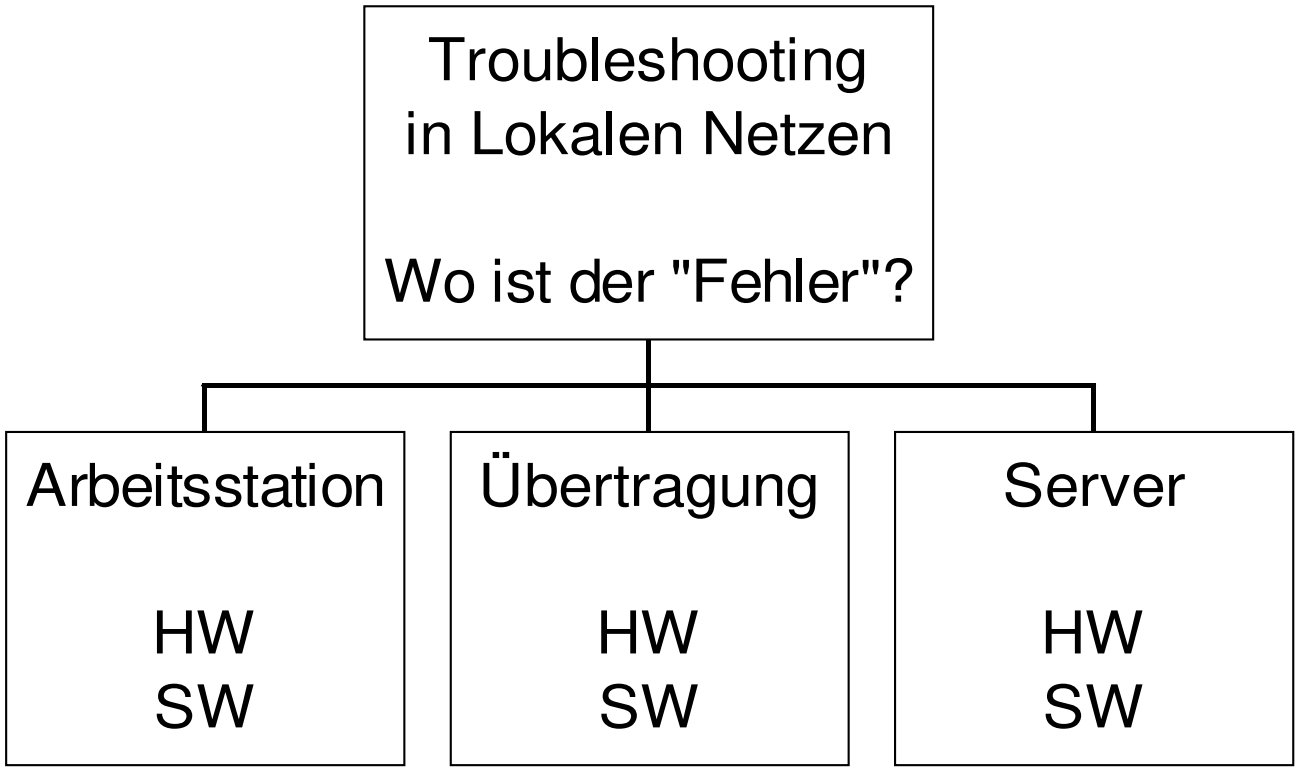
NAGIOS – WebBeispiel2

- Überwachung des Webservers

```
define service{
use generic-service ; (Template)
host_name webserver
service_description HTTP
check_command check_http
}
```

Labor zur Überwachung

Troubleshooting in LANs



Arbeitsstation HW

- Netzwerkkarte
- Einstellungen (PnP, (IRQ, Port, DMA))
- Motherboard
- Peripheriegeräte (Platten, ...)
- Stromversorgung
- ...

Arbeitsstation SW

- Konfigurationsdateien (z.B.: Registry, CONFIG.SYS, /etc/modules.conf)
- Netzwerkkonfiguration (Registry, /etc/sysconfig/network/ifcfg-eth0)
- Treiber
- Anwendung
- ...

Übertragung HW

- Verkabelung (Kabellänge)
- Abschlußwiderstand
- Stecker
- Wackelkontakte
- Kurzschlüsse
- Erdung
- Übertragungstechnik

Übertragung SW

- Topologie
- Routing / Router
- Segmentierung (Physisch / Logisch)
- Adress-Schema
- Konfiguration

Server HW

- Netzwerkkarte
- Einstellungen (PnP, IRQ, Port, DMA, ...)
- Motherboard
- Peripheriegeräte (Platten, ...)
- Stromversorgung
- ...

Server SW

- Konfigurationsdateien (z.B.: Registry, STARTUP.NCF, AUTOEXEC.NCF, /etc/inetd.conf)
- Treiber und Ihre Parameter
- Netzwerkkonfiguration
- ...

Vorgangsweise

- Einschränken, welche Komponente für den Fehler verantwortlich ist.
- Grobprüfung der Arbeitsstation (Reboot)
- Sichtprüfung des Netzwerkanschlusses
- Sichtprüfung des Servers (OK?)
- Auslesen der Log-Dateien
- Detailprüfung der Arbeitsstation
- Prüfmittel (Software)
- Prüfmittel (Hardware)

Prüfmittel - Software

- Prüfen der Konfiguration der Arbeitsstation
- Prüfen der Übertragungsstrecke
- Prüfen des aktiven Komponenten (HUB, ...)
- Prüfen des Server
- ...

Prüfen Arbeitsplatz

- Konfigurationsdateien mit aktueller Kopie vergleichen
- Funktionsfähigkeit ohne Netz prüfen
- Funktionsfähigkeit im Netz mit einfacher Anwendung prüfen (ev. Bootdiskette)
- ...

Prüfen Übertragungsleistung

- COPY, XCOPY, cp,
- ...

Prüfen aktiver Komponenten

- Repeater
- Bridges
- Hubs
- Switches
- Router
- SNMP, Managementsoftware

Prüfen Server (Windows)

- AUTORUNS.EXE
- REGMON.EXE
- TCPVIEW.EXE
- Eventviewer
- Taskmanager
- Ressourcenmonitor
- …

Prüfen Server (Linux)

- ps
- netstat
- iptraf
- systat
- /var/log/messages
- …

Prüfmittel - Hardware

- Kabeltester (Layer 0,1)
- LAN Troubleshooter (Layer (0,1),2,3)
- LAN Analyzer (Layer 3,4,5,6,7)
- ...

Vorbeugung Arbeitsstation

- Benutzerschulung
- Kopie der Konfigurationsdateien
- Backup
- Zertifizierte Hardware
- Standardisierung (so wenig Typen wie möglich)
- Dokumentation

Vorbeugung Übertragungsstrecke

- Zertifizierte Hardware
- Abnahme vor erster Inbetriebnahme
- Dokumentation aller Komponenten
- Laufende Prüfungen
- Dokumentation aller Änderungen
- Wichtige Systeme mit Ersatzstrecken ausstatten

Vorbeugung Server

- Kopie der Konfigurationsdateien
- Backup
- Zertifizierte Hardware
- UPS, SFT-Level, RAID, Cluster, ...
- Dokumentation

II.1 System Calls

- Definition (Was ist das?)
- Zweck (Warum?)
- Umsetzung
- APIs (Application Programming Interface)
- Bibliotheken
- Beispiel

System Calls – Definition

- System Calls sind vom Betriebssystem zur Verfügung gestellte Aufrufe, um die Funktionen des Betriebssystem von einem Programm aus, benutzen zu können.

System Calls – Zweck

- Umsetzung der Trennung verschiedener Modi (Kernel, User).
- Datenübergabe zwischen Programm und Betriebssystem
- Nutzen der vorhandenen Betriebssystemfunktionen (z.B.: Datei öffnen, …)

System Calls – Umsetzung

- Sehr von der verwendeten Hardware abhängig
- Häufig als Softwareinterrupt implementiert
- Programm meldet damit dem Betriebssystem, welche Funktion benötigt wird

System Calls – APIs

- Für eine möglichst einfache Benutzung der Systemcalls stellen die Betriebssysteme APIs (Application Programming Interface) zur Verfügung, über die der Anwendungsprogram-mierer die Funktionen benutzen kann.

System Calls – Bibliotheken 1

- Da die Betriebssystem immer mehr Funktionen zur Verfügung stellen, werden auch die Zahl der Systemcalls mehr.
- Für unterschiedliche Entwicklungsumgebungen werden Bibliotheken angeboten, die die Verwendung der APIs wesentlich vereinfachen.

System Calls – Bibliotheken 2

- I/O
- Dateioperationen
- Windowmanagement
- Timerfunktionen
- Prozeßmanagement
- …

System Calls – Beispiel

- Lesen/Schreiben einer Registryeinstellung unter Windows mittels Bibliotheksaufruf
- Lesen einer Datei unter C

System Calls – Beispiel 2

- z.B.: in der Scriptsprache Autohotkey
- Lesen:
- RegRead, Value, HKCU, Software\CKTools\sudokum, StartX

- Schreiben:
- RegWrite, REG_DWORD, HKCU, Software\CKTools\sgm, StartX, 100

System Calls – Beispiel 3

- #include <stdio.h>
- …
- datvar=fopen(filetoread, ″r″)
- fgets(vektor, maxzeichen, datvar)
- …
- fclose(datvar)

System Calls – Labor

- Schreiben einer Anwendung, die vom Betriebssystem Datum und Uhrzeit in lokaler Darstellung und eventuelle Lokalisierungs- und Spracheinstellungen abfrägt und diese in einem Fenster darstellt (optional mit einem Button zum Aktualisieren, …)

II.2 Shell Programmierung

- Auftragssteuersprachen
- Je nach Betriebssystem unterschiedliche Skriptsprachen
  - VSE: JCL (Job Control Language)
  - VMS: DCL (Digital Command Language)
  - Windows: BATch und CoMmanD
  - Linux: Shellscripte

Shell Programmierung Windows

- Betriebssystemeigene Sprachen
  - Windows Shell Skript (cmd.exe)
  - Windows Power Shell
- Zusätzliche Programme
  - VBScript, JScript
  - AutoIT, AutoHotKey
  - TCL (Tool Command Language)

Shell Programmierung Linux

- Betriebssystemeigene Sprachen
  - Shellscripte (bash, csh, sh, ksh)
- Zusätzliche Programme
  - TCL (Tool Command Language)
  - …

Beispiel Windows Shell Script

```
@ECHO OFF
IF %1.==. GOTO EXIT
DIR | FIND "%1"
:EXIT
```

- Alle Dateien mit bestimmten Datum anzeigen

Beispiele Windows Powershell

```
Get-Process | where { $_.WS -gt 10MB } |
   Stop-Process
```

- Beendet alle Prozesse, die mehr als 10MB RAM benötigen

```
$processToWatch = Get-Process notepad
$processToWatch.WaitForExit()
```

- Wartet bis Notepad terminiert

Beispiel Autohotkey

```
P1=%1%
Start:
if P1 {
    Filename:=P1
    FileSetTime,,%Filename%,C,0,0}
else {
    MsgBox, 64, SetDate, Programm setzt Zeitstempel,5}
ExitApp
GuiDropFiles:
P1:=A_GuiControlEvent
Goto Start
```

- Programm setzt Datum/Uhrzeitinfos einer Datei auf den aktuellen Wert

Beispiel Bashscript

```
#!/bin/bash
# Usage: addmailuser username prename lastname
if ["$1" = ""]
then
    echo Usage: $0 username prename lastname
else
    useradd -c "$2 $3" -p -u $1
    passwd $1
    mkdir /home/$1
    chown $1:users /home/$1
    echo "Das ist eine Testnachricht" | mail -s Test $1
fi
```

- Neuen Mailuser auf einem Linuxserver anlegen

Beispiel TCL

```
# Erstes TCL-Programm
button .btn -text "Hallo Klaus"
pack .btn
```

- Ausgeben eines Textes

II.3 Regular Expressions

- Reguläre Ausdrücke sind Zeichenketten, die mittels syntaktischer Regeln Mengen von Zeichenketten dient.
- \b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b
- Beschreibt jede e-Mail-Adresse

Regular Expressions 2

. Beliebiges Zeichen (außer Zeilenumbruch)

* kein oder mehr Vorkommen vom vorangegangenen Zeichen

? kein oder ein Vorkommen vom vorangegangenen Zeichen

+ ein oder mehr Vorkommen vom vorangegangenen Zeichen

Regular Expressions 3

{min,max} zwischen min und max Vorkommen vom vorangegangenen Zeichen

[…] Klasse von Zeichen

- [abc] a oder b oder c
- [a-z] beliebiger Kleinbuchstabe
- [A-Z] beliebiger Großbuchstabe
- [0-9] beliebige Ziffer
- [a-zA-Z] beliebiger Buchstabe

Regular Expressions 4

[^…] Zeichen, das nicht in Klasse ist

\d Eine Ziffer

\D Eine „Nichtziffer“

\s Ein „Whitespace“-Zeichen“ (Leerzeichen, Tabulator, Zeilenumbruch)

\S Ein „Nicht-Whitespace-Zeichen“

\w Ein „Wortzeichen“ [a-zA-Z0-9_]

\W Ein „Nichtwortzeichen“

Regular Expressions 5

^	Zeichen muß erstes Zeichen sein
$	Zeichen muß letztes Zeichen sein
\b	Wortgrenze
|	Auswahl (entweder oder)
(…)	z.B.: (Mon|Diens|Donners|Frei)tag
	(ab)+ ein oder mehr „ab“

Regular Expressions 6

\t Tabulator

\r Carriage Return

\n Neue Zeile

\xHHZeichen in Hexcodeangabe

\. Punkt

\\ Backslash (Methode funktioniert für alle Sonderzeichen (.*?+[{|()^$)

Regular Expressions Optionen

- An Anfang einer Regular Expression können Optionen stehen: „opt)RegEx“

i Groß-/Kleinschreibung irrelevant

m Mehrzeilig

s „.“ inklusive Zeilenumbruch

x ignoriert Whitespace im Muster

…

III.1 Webapplikationen

- Architektur von Webapplikationen
- Verteilung der Aufgaben bei Webapplikationen
- Speicherung von Daten bei Webapplikationen
- Entwicklungswerkzeuge für Webapplikationen

Architektur von Webapplikationen

- Webserver
- Applikationsserver
- Authentifikationserver
- Client

Webserver - Aufgaben

- Dokumente (HTML-Dateien, CSS-Dateien, Bilder, …) an Clients ausliefern
- Das verwendete Protokoll ist dabei HTTP (Port 80) bzw. HTTPS (Port 443)
- Heute zunehmend dynamisch erstellte Dokumente (SSI, PHP, JSP, ASP, …)

Webserver - Zusatzaufgaben

- Cookieverwaltung
- Zugriffsbeschränkungen
- Weiterleitung – Rewrite
- Fehlerbehandlung (Fehlerseiten)
- Protokollierung
- (Caching)

Webserver – Produkte

- Apache HTTP Server
  - Apache Software Foundation
- IIS (Internet Information Server)
  - Microsoft
- …
  - CERN httpd
  - lighttpd

Applikationsserver – Aufgaben

- Server auf dem Anwendungsprogramme ausgeführt werden
- Laufzeitumgebung für den Serverteil einer Client-Server-Anwendung
- Daten vom Client verarbeiten und die Ergebnisse wieder dem Client zur Verfügung stellen

Applikationsserver – Produkte

- Adobe ColdFusion
- IBM Websphere
- Oracle (BEA) Weblogic
- Apache Tomcat
- Zope
- …

Authentifikationserver 1

- Webinterface für Benutzer (i.A. https)
- Sicherheit nach außen (Problematik meherer Server im Inneren und damit unterschiedliche Zertifikate)
- Funktionsweise eines „Reverse Proxy“
- Oft Anbindung an Verzeichnisdienst, selten eigene Benutzerdatenbank

Authentifikationserver 2

- Sun ONE (Open Network Environment) -> Sun Java Enterprise System
- Novell iChain -> Novell Access Manager -> NetIQ Access Manager
- Microsoft Forefront

Client

- Browser
  - Mozilla Firefox
  - Google Chrome
  - Microsoft Internet Explorer
  - Apple Safari
  - …
- Apps

III.2 Client-Server-Architektur

- Definition
- Zutaten
- Serverbeispiele
- Schichten
- 2-Schicht-Architektur
- 3-Schicht-Architektur

Definition

- Client-Server-Architektur beschreibt ein Modell Aufgaben in einem Netzwerk zu verteilen.
- Dabei übernehmen Server die zentralen Aufgaben und stellen Sie Clients zur Verfügung

Zutaten

- Server (Dienstleister)
- Client (Nutzer)
- Protokoll (erlaubt dem Client einen Request an den Serverdienst zu stellen und eine entsprechende Response auszuwerten)

Serverbeispiele

- Fileserver (NCP, SMB, NFS, …)
- Printserver (LP, NCP, …)
- CD-ROM-Server (Medienserver)
- Datenbankserver
- Mailserver (smtp, pop, imap)
- Webserver (http)
- Applikationsserver

Schichten

- Um die Komplexität von Anwendungen zu reduzieren bzw. die Modularität zu erhöhen, wird auch bei Softwarearchitekturen ein Schichtenmodell eingesetzt, allerdings mit deutlich weniger Schichten, wie im ISO-Referenzmodell

2-Schicht-Architektur

- „Two-Tier-Architecture“
- Klassische Client-Server-Architektur
- Clientschicht (oft auf Fat-Client, d.h. Client übernimmt auch Logikaufgaben)
- Serverschicht (immer auf Fat-Server)

3-Schicht-Architektur

- „Three-Tier-Architecture“
- Aktuelle Realisierungen beruhen meist auf diesem Modell
- Präsentationsschicht (Client Tier)
- Logikschicht (Middle Tier)
- Datenhaltungsschicht (Database Server Tier)

Client Tier

- „Front End“
- Benutzerschnittstelle (GUI)
- Benutzereingaben
- Kommunikation mit der Middle Tier
- i.A. keine direkte Kommunikation mit dem Back End
- Oft im Webbrowser realisiert

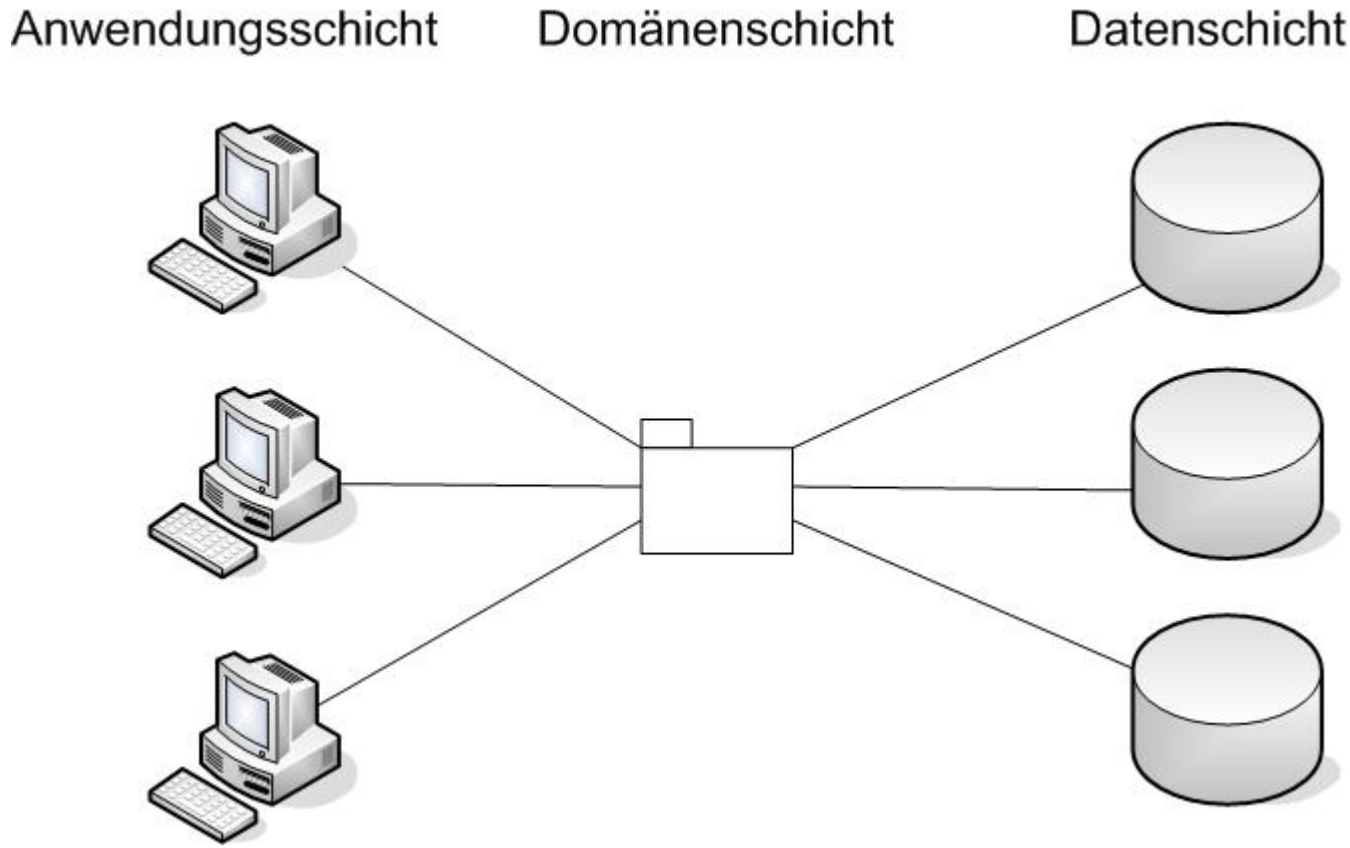
Middle Tier

- „Application-Server-Tier“, „Enterprise Tier“, „Businesslogikschicht“
- Umsetzung der Businesslogik
- Kommunikation mit den anderen beiden Schichten (getrennt von einander)
- Realisierung mit einem Anwendungsserver

Database Server Tier

- „Back End“
- Datenhaltung i.A. auf einem Datenbankserver
- Kommunikation mit der Middle Tier

3-Schicht-Architektur-Beispiel



Quelle: http://de.wikipedia.org/wiki/3-Tier-Architektur#Drei-Schichten-Architektur

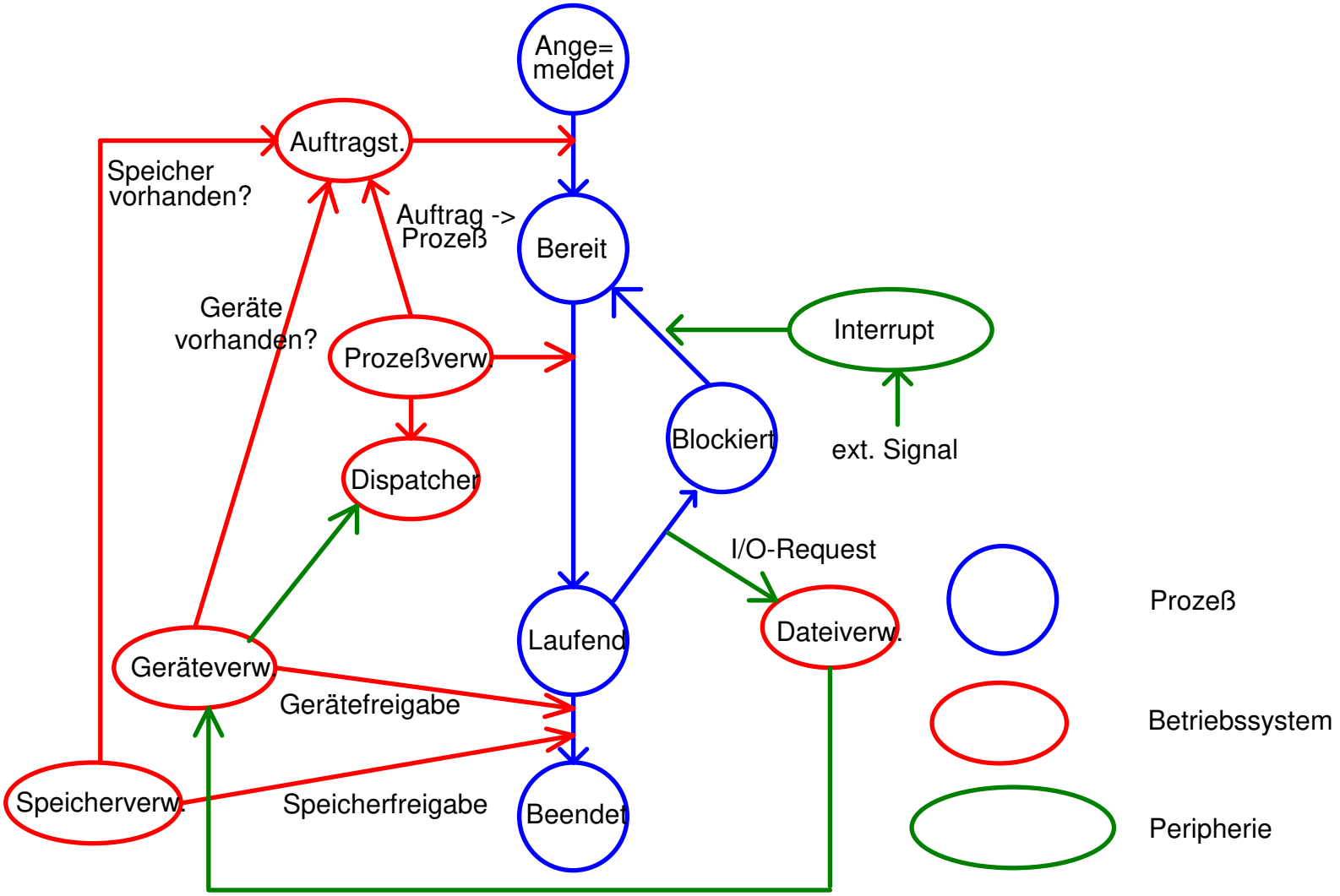
Mehrschichtarchitekturen

- Hier wird das 3-Schichten-Modell erweitert, in dem einzelne Aufgaben aus einer der drei Schichten herausgelöst werden und in einer eigenen Schicht implementiert werden
- z.B.: Präsentation, Steuerung, Datenzugriff

III.3 Algorithmen & Synchronisation

- Prozeßkommunikation
- Semaphoren
- Pipes
- RPC

Prozeßlebenszyklus



Prozess-Synchronisation

Prozesse sind oft voneinander abhängig

z.B.:

- Schreibprozess muß auf Daten warten
- Zwei Prozesse wollen gleich HW nutzen

=> 2 Grundklassen der Synchronisation

- Wechselseitiger Ausschluss
- Prozesskooperation

Das Semaphorkonzept

- Binärsemaphor
  - Ampel,
  - Flag,
  - ...
- allgemeine Semaphoren
  - Zähler

Pipe-Konzept

Manche Betriebssysteme bzw. Prozessoren stellen für die einfachere Synchronisation sogenannte Pipes ("Röhren" für einen Datenstrom) zur Verfügung. Dabei werden die Ergebnisse eines Befehls als Eingabe für den nächsten Befehl verwendet (FIFO).

z.B.: dir | sort | more >filelist.txt

Pipe-Symbole

- | Anonyme Pipe
- < Named Pipe für die Eingabe
- > Named Pipe für die Ausgabe
- >> Named Pipe für die Ausgabe (append)
- [unit]> Angabe einer Ein- oder Ausgabeeinheit (z.B. 2>&1)

RPC

- Remote Procedure Call
- RFC 707, 1057, 5531
- Viele inkompatible Implementierungen
  - XNS Xerox Network System
  - Sun ONC RPC (Open Network Computing)
  - DCE RPC (Distributed Computing Env.)
  - MSRPC → DCOM → .NET Remoting

RPC – Ablauf

- Client sendet eine Anfrage an den Server
- Server verarbeit diese Anfrage
- Server schickt eine Antwort
- Client arbeitet mit der Anwort weiter

RPC: SOAP

- Simple Object Access Protocol → Eigenständiges Acronym
- Weiterentwicklung von XML-RPC
- Spezifikationen: http://www.w3.org/TR/soap/

SOAP Request Beispiel

```
<?xml version="1.0"?>
<s:Envelope
   xmlns:s="http://www.w3.org/2003/05/soap-
   envelope">
    <s:Body>
        <m:TitleInDatabase xmlns:m="http://www.spg.at/db/soap">
          NVS1 und Coufal
        </m:TitleInDatabase>
    </s:Body>
</s:Envelope>
```

SOAP Reply Beispiel

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <s:Header>
        <m:RequestID xmlns:m="http://www.spg.de/db/soap">FC</m:RequestID>
    </s:Header>
    <s:Body>
        <m:DbResponse xmlns:m="http://www.spg.at/db/soap">
            <m:title value=„NVS1 und Coufal">
                <m:Choice value="1">NVS1_H4.PDF</m:Choice>
                <m:Choice value="2">NVS1_B34.PDF</m:Choice>
            </m:title>
        </m:DbResponse>
    </s:Body>
</s:Envelope>
```