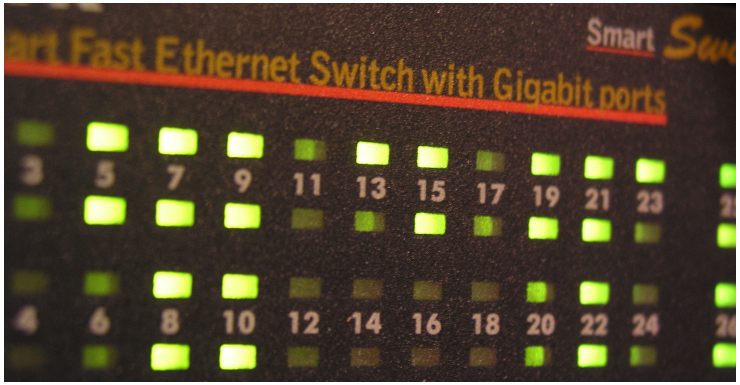
Technikum Wien - Wirtschaftsinformatik - BWI Internet Security

Netzwerksicherheit



René Pfeiffer

pfeiffer@luchs.at

28. Oktober 2006

http://web.luchs.at/, Vienna, AT

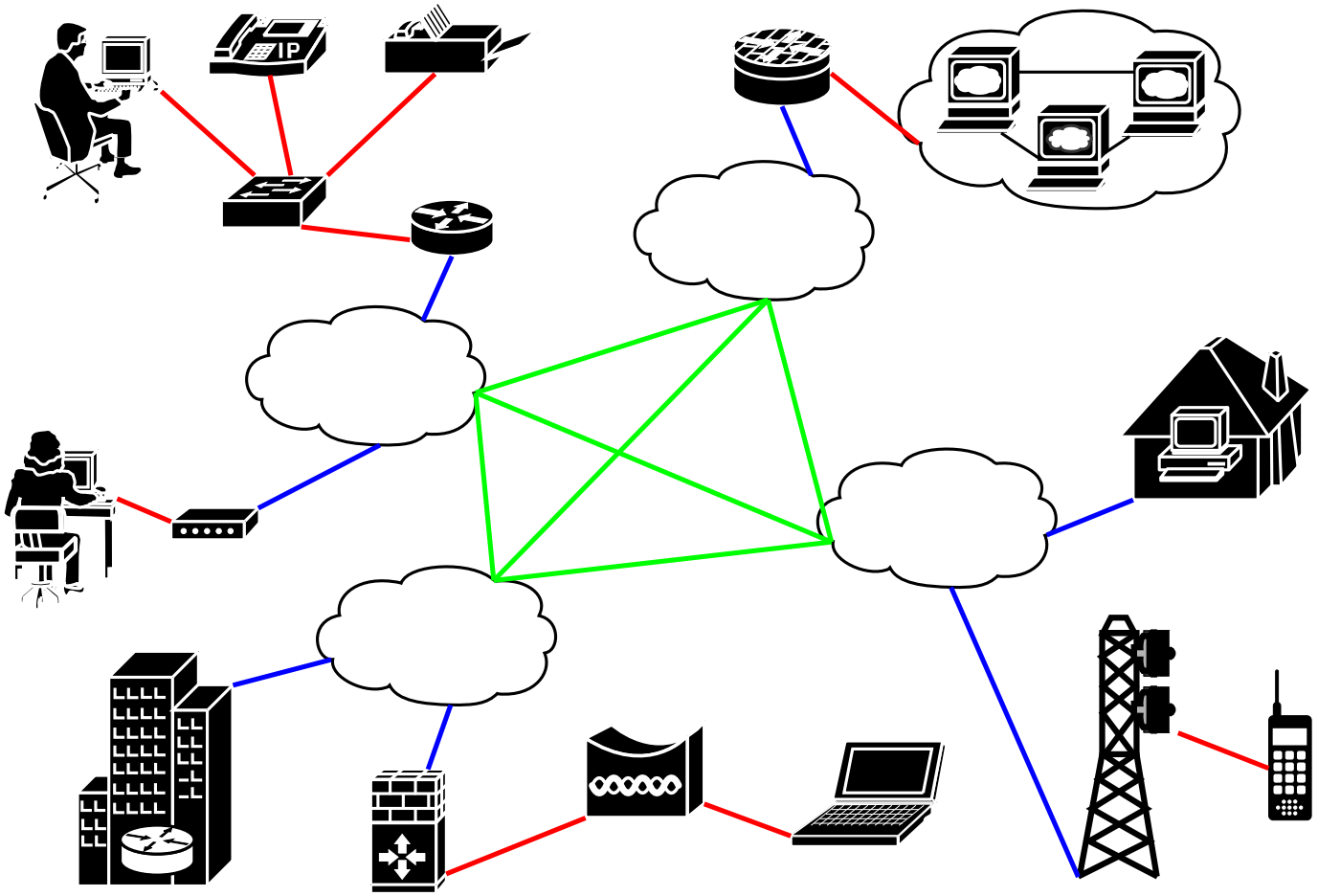
Netzwerksicherheit - Schwerpunkte

- vernetzte Systeme in Internet und anderen Netzwerken
- mögliche Bedrohungen in vernetzten Umgebungen
- Schutzmaßnahmen in Netzwerken
- Design-/Architekturfragen
- Kontrolle und Regelung

Vernetzte Unsicherheit

- Netzwerke enthalten eine Vielzahl von Komponenten
- Netzwerke dehnen sich über „klassische Grenzen” aus
- Netzwerke haben oft tückische Abhängigkeiten
- Netzwerke können Probleme potenzieren und leicht verbreiten

Ausdehnung von Netzwerken



Bedrohungen in Netzwerken

- Attacken auf verfügbare Netzwerkdienste

  Verändern von Kommandos, datengesteuerte Angriffe („data driven attacks”), Paßwortraten, *man-in-the-middle* Attacken (MITM), Denial of Service, . . .

- indirekte Attacken

  Hijacking von Datenverbindungen, Paketschnüffler („sniffer”), Einschleusen und Veränderung von Daten, Replay Attacken, passive Aufklärung, . . .

- Social Engineering

  Anschauen bestimmter Dateiformate (ein falscher Klick in Emails oder auf Webseiten), „Phishing” (kurz für *password harvesting fishing*), Telefonieren, Einsatz von Spyware, Key Logger, . . .

Folgen von Attacken

- Erschöpfung von Resourcen

  Bandbreite, Paketlaufzeiten, Speicherplatz, CPU, Quoten, . . .

- Mißbrauch von Privilegien

  Angreifer erhält zusätzliche/andere Zugriffsrechte, . . .

- Manipulation von Daten

  Applikationen in instabile Zustände bringen, Ändern von Benutzerdaten, . . .

- Abfangen von Informationen

  Mithören von Paßworten, kryptografischen Schlüsseln, Emails, Replay Attacken, . . .

Erschweren von Attacken

- Paketfilter
- Proxy Server
- Content Filter
- Virtual Private Networks (VPNs)
- Intrusion Detection/Prevention Systeme
- gute Systemadministration

Firewall ist ein Oberbegriff für eine Sammlung von verschiedenen Maßnahmen.

Unterteilen von Netzwerken

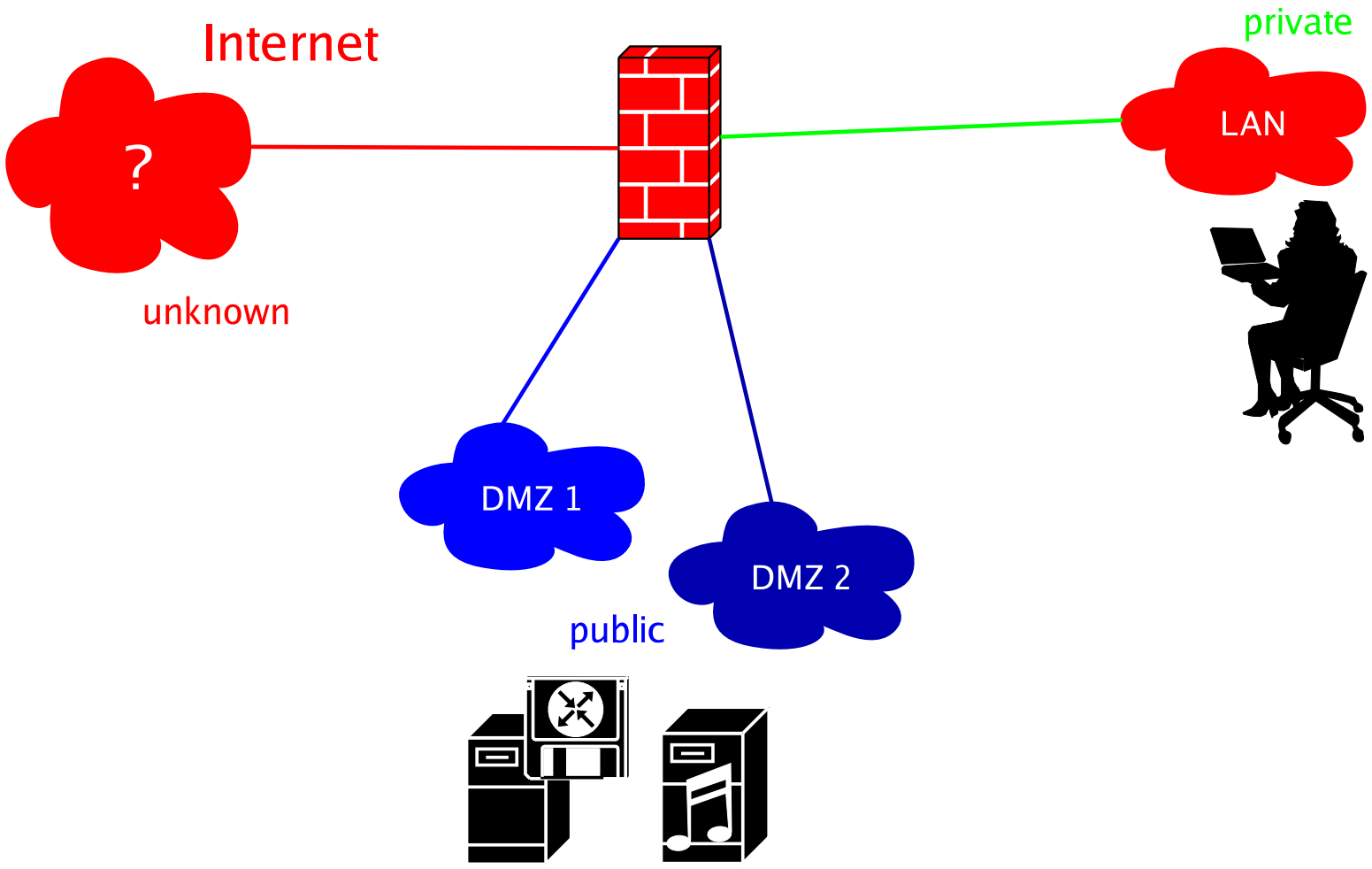
- Aufteilung der Netzwerke nach Vertrauen
- internes, schützenswertes Netzwerk (LAN)
- Perimeternetzwerk oder „Demilitarisierte Zone” (DMZ)

  Quarantänenetzwerk für öffentlich angebotene Dienste
- unbekannte, nicht vertrauenswürdige Netze
  - Internet
  - WLAN Umgebungen
  - WANs und VPNs zu unbekannten Netzen

Vertrauensstufen in Netzwerken

- Datenfluß wird von Vertrauen geregelt
  - vertrauenswürdigeres Netzwerk darf Verbindungen in weniger vertrauenswürdigeres Netzwerk öffnen
  - weniger vertrauenswürdigeres Netzwerk darf nicht oder eingeschränkt in vertrauenswürdigeres Netzwerk
- übersetzt:
  - LAN darf in die DMZ
  - DMZ darf ins Internet
  - Internet darf nur auf ausgewählte Ports in der DMZ
  - Internet darf nicht ins LAN
  - DMZ darf nicht ins LAN

Netzwerke mit Sicherheitsstufen



Sicherheitsstufen mit Einschränkung

- Gefahr durch Spyware & Trojanische Pferde
  - „connect back” Mechanismus
  - Herausschleusen von Daten
- vertrauenswürdige Netzwerke beobachten
- ungeregelten Netzwerkverkehr einschränken/prüfen

Paketfilter

- Inspektion von Datenströmen auf Paketebene (Layer 3/4)
- Regeln prüfen Paketeigenschaften
- moderne Filter
  - prüfen ganze Datenströme
  - führen Buch über detektierte Pakete
  - zustandgesteuertes Filtern („stateful inspection”)
- komplexere Filter verstehen Applikationsprotokolle (Layer 7)

Proxy Server

- arbeiten auf Applikationsebene (Layer 7)
- erlauben sehr gute Kontrolle von Zugriffen
- führen Protokollprüfungen auf Applikationsebene durch
- lassen sich meist mit Filtern kombinieren

  Antivirus, Antispam, Antimalware, etc.

Content Filter

- inspizieren Inhalte von Datenströmen
- führen ebenfalls Protokollprüfungen durch
- werden oft eingesetzt, um Tunnel zu unterbinden
- Anti-Virus / Anti-Spam Filter sind prominentes Beispiel

Intrusion Detection/Prevention Systeme

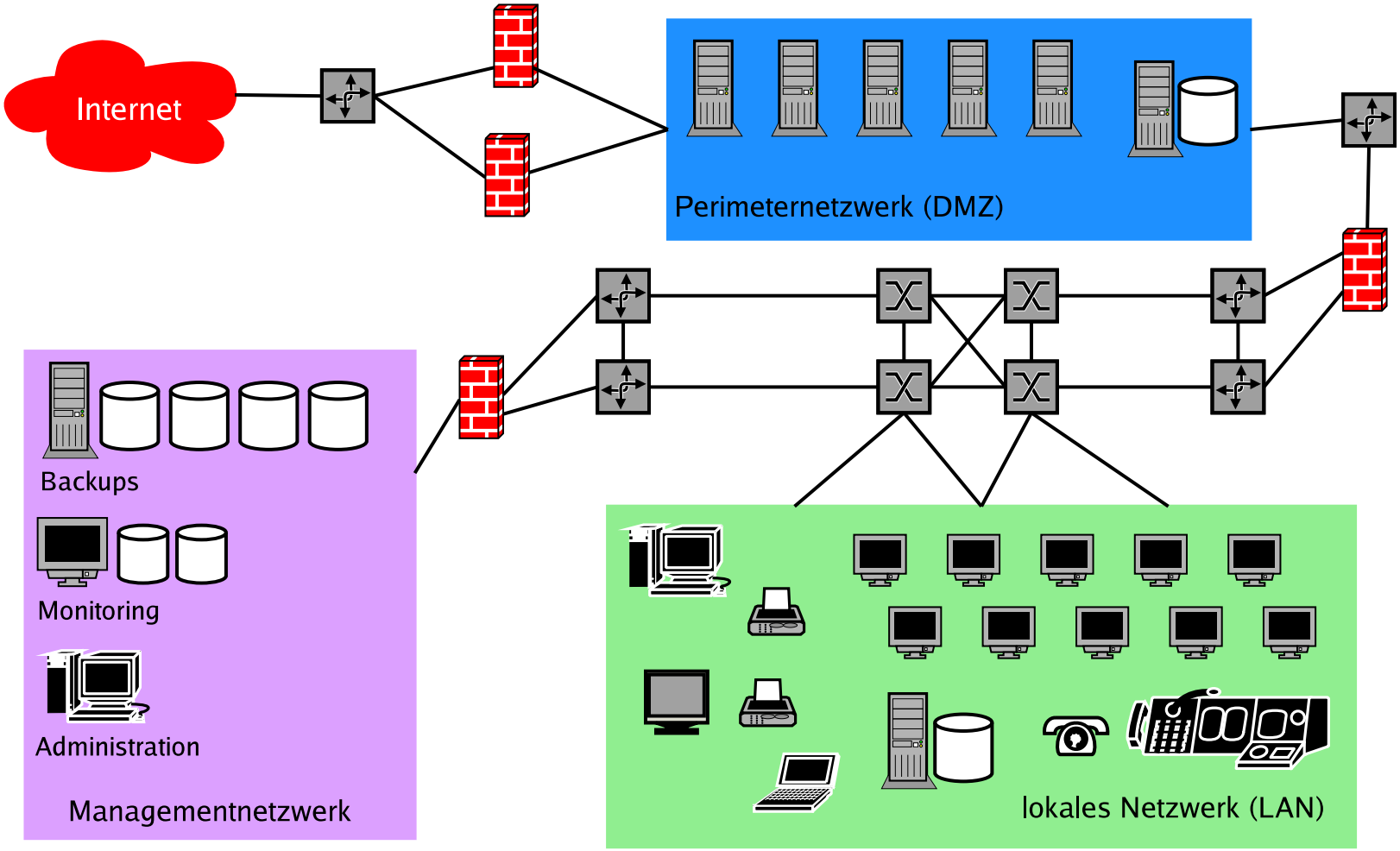
- detektieren Anomalien in Datenströmen
- können „anomale” Datenströme blocken
- sind meist signaturbasiert

  Ähnlichkeit zu Antivirusmethoden; es gibt auch Systeme, die Betriebsdaten beobachten und Abweichungen melden

- benötigen eine Abstimmung auf Einsatzgebiet

  Welche Protokolle? Welche Teile des Netzes müssen überwacht werden?

Maßnahmen im Einsatz



Planung von Maßnahmen

1. Identifizieren der Netzwerke und deren Sicherheitsstufe
2. Erfassen der Übergänge

   Router, Bridges, Switches, Accesspoints, ...
3. Gefährliche Szenarien definieren
4. Dimensionieren der Maßnahmen

   Bandbreite, CPU-Last, Festplattenplatz, Verzögerung, ...
5. Umsetzung

6. Testen der Maßnahmen

   Durchspielen aller Szenarien, die betrachtet wurden

7. zurück zu Schritt 1

Sicherheitsmaßnahmen sind niemals statisch, sondern ein stetig ablaufender Prozeß.

Paketfilter

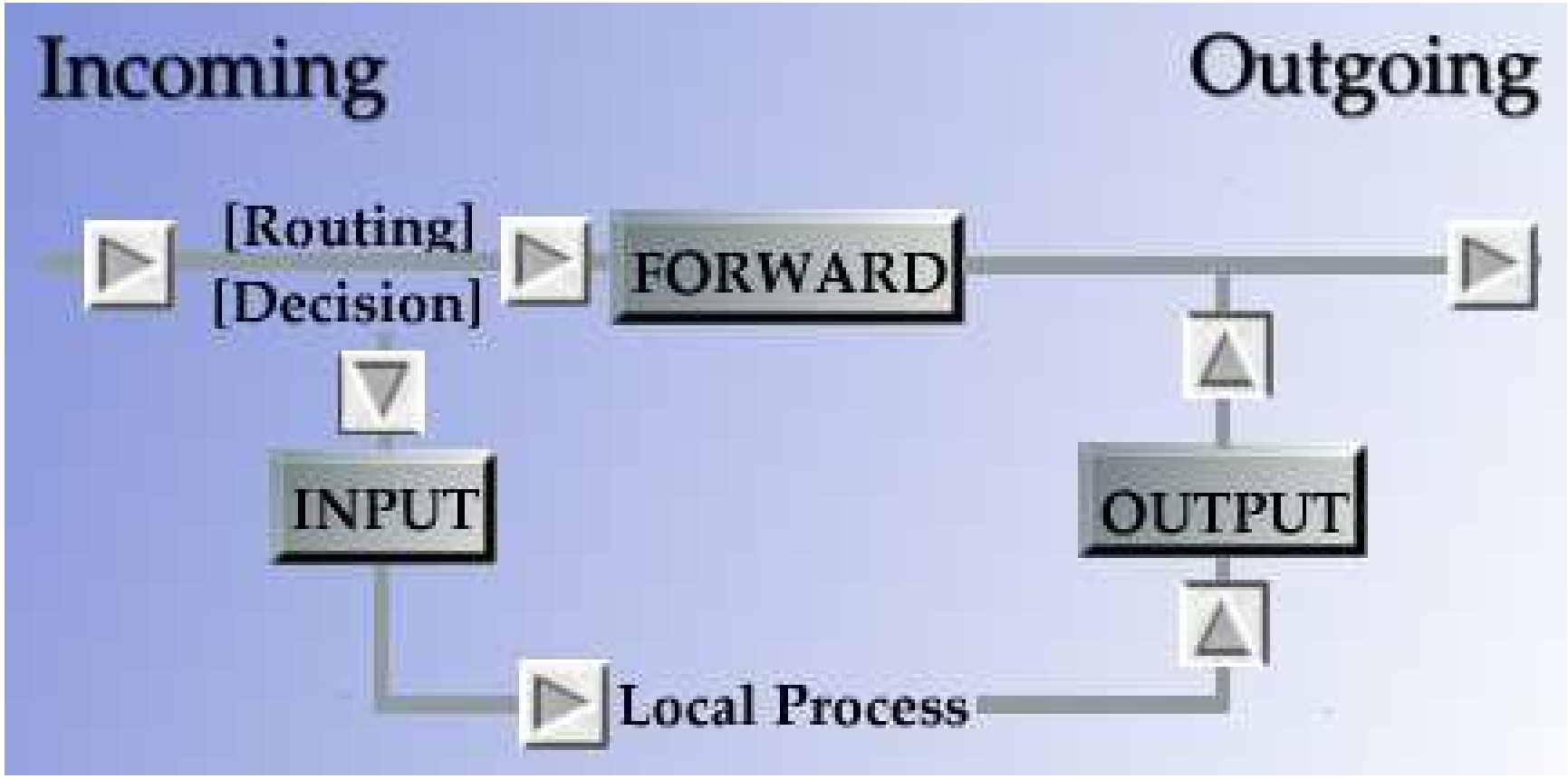
Funktionsweise von Paketfiltern

- Paketfilter inspizieren einzelne Pakete
- Paket wird Prüfungen unterzogen

  Checksummen, Protokolldefinitionen, Flags, etc.
- Paketfilter vergleicht Paket mit Liste von Kriterien

  IP-Adressen, Ports, Codes, Protokolle, etc.
- Paketfilter führt anschließend Aktion durch
  - DROP - Paket verwerfen
  - ACCEPT - Paket durchlassen
  - REJECT - Paket mit Fehlermeldung ablehnen

Paketfluß durch Linux® Netfilter



Stateful Inspection

- oft werden ganze Datenströme übertragen

  Webseiten, Archive, Bilder, Videos, Dokumente, ...

- „stateful inspection” betrachtet Datenströme
- führt zu besserem Filtern und Prüfen der Daten
- Entdecken von eingeschmuggelten Anomalien
- bessere Erzeugung von Fehlermeldungen
- einfachere Filterregeln

„stateful inspection” wird oft auch „zustandsgesteuertes” oder „dynamisches” Filtern genannt.

Default Policy

- DROP - alle Pakete werden geblockt
  Filterregeln definieren Erlaubnisse
- ACCEPT - alle Pakete werden durchgelassen
  Filterregeln definieren Verbote

Beide Ansätze sind legitim, jedoch minimiert DROP die Folgen von Vergeßlichkeit.

Regelaufbau

- Filterregeln müssen eine bestimmte Reihenfolge haben

  Wichtig: falsche Reihenfolge kann Filter unbrauchbar machen

- viele Protokolle sind bidirektional

  „HTTP nach außen” erzeugt damit zwei Regeln

- Filterregeln immer einzeln testen und verändern
- Filterregeln immer komplett neu laden

Protokolle und Ports

- Internetprotokolle
  - Transmission Control Protocol (TCP) für Datenverbindungen mit Fehlerkorrektur
  - User Datagram Protocol (UDP) für einzelne Pakete (verbindungslos)
  - Internet Control Message Protocol (ICMP) für Diagnose- und Fehlermeldungen
- TCP & UDP haben Ports
  - 0-1023 - privilegierte Ports, benutzt von Serverapplikationen
  - 1024-49151 - registrierte/dynamische Ports zur Datenübertragung
  - 49152-65535 - dynamische Ports zur Datenübertragung

  Wird meist zu

  - 0-1023 - privilegierte Ports
  - 1024-65535 - dynamische Ports

  zusammengefaßt.

ICMP Meldungen

- ICMP kennt keine Ports
- ICMP kennt nur Typen und Codes
  - destination-unreachable
    - network-unreachable
    - port-unreachable
    - network-prohibited
    - fragmentation-needed
  - echo-request (ping)
  - echo-reply (pong)
  - time-exceeded (ttl-exceeded)
    - ttl-zero-during-transit
    - ttl-zero-during-reassembly
  - timestamp-request
  - timestamp-reply

Liste ist unvollständig und nur ein Beispiel.

Filtern von TCP Verbindungen

- Secure Shell (SSH) Server lauschen auf Port 22
- eingehende SSH Verbindungen erlauben:
  - eingehend: Quell-IP 0/0 Quell-Port 1024-65535 auf Ziel-IP a.b.c.d Ziel-Port 22 erlauben
  - ausgehend: Quell-IP a.b.c.d Quell-Port 22 auf Ziel-IP 0/0 Ziel-Port 1024-65535 erlauben
- TCP hat immer eine Regel pro Paketrichtung
- TCP Verbindung erfordert immer zwei Regeln

0/0 ist die Kurzform für 0.0.0.0/0.0.0.0 = das ganze Internet

Paketzustände

- NEW - Paket öffnet neue Verbindung
  - immer das erste Paket, welches der Filter sieht
- ESTABLISHED - Paket ist Teil bestehender Verbindung
  - bei TCP alle Folgepakete
  - bei UDP/ICMP Antwortpakete auf das erste Paket
- RELATED - Paket gehört zu bestehender Verbindung
  - ICMP Fehlermeldungen zu bestehenden Verbindungen/Paketen
  - Pakete, die eine neue Verbindung öffnen, aber zu einer bestehenden gehören
- INVALID - ungültige Pakete

Dargestellte Zustände gelten für den Linux® Netfilter, lassen sich aber auf andere Filter übertragen.

Filtern mit Paketzuständen (1)

- SSH Verbindung eingehend

  iptables –append FORWARD –protocol tcp –source 0/0 –source-port 1024:65535

  –destination a.b.c.d –destination-port 22

  –match state –state NEW,ESTABLISHED –jump ACCEPT

- SSH Verbindung ausgehend

  iptables –append FORWARD –protocol tcp –source a.b.c.d –source-port 22

  –destination 0/0 –destination-port 1024:65535

  –match state –state ESTABLISHED –jump ACCEPT

Filtern mit Paketzuständen (2)

- SSH Verbindung eingehend

  iptables –append FORWARD –protocol tcp –source 0/0 –source-port 1024:65535

  –destination a.b.c.d –destination-port 22

  –match state –state NEW,ESTABLISHED,RELATED –jump ACCEPT

- SSH Verbindung ausgehend

  iptables –append FORWARD –protocol tcp –source a.b.c.d –source-port 22

  –destination 0/0 –destination-port 1024:65535

  –match state –state ESTABLISHED,RELATED –jump ACCEPT

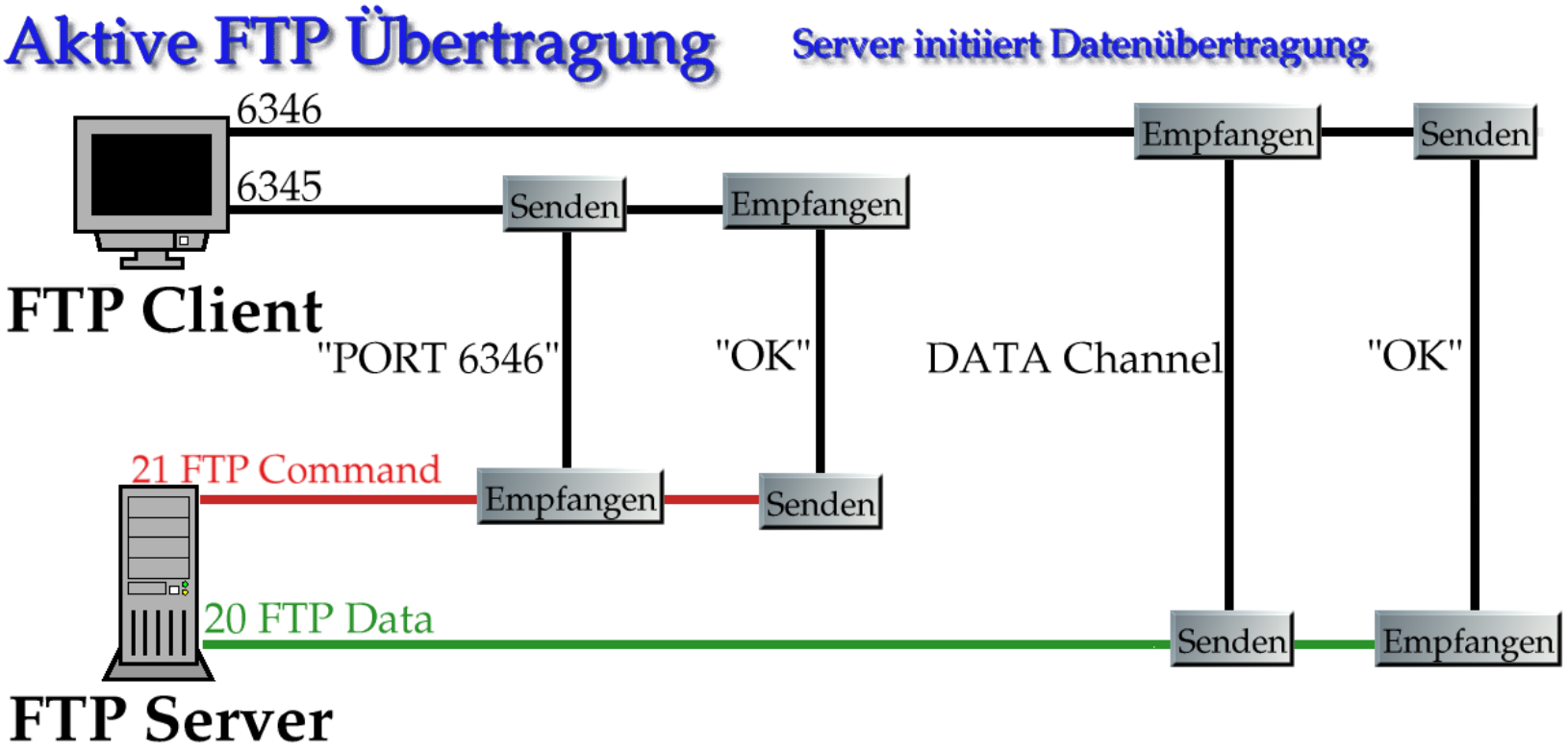
RELATED erfaßt auch ICMP Pakete und filtert diese korrekt.

Filtern von FTP

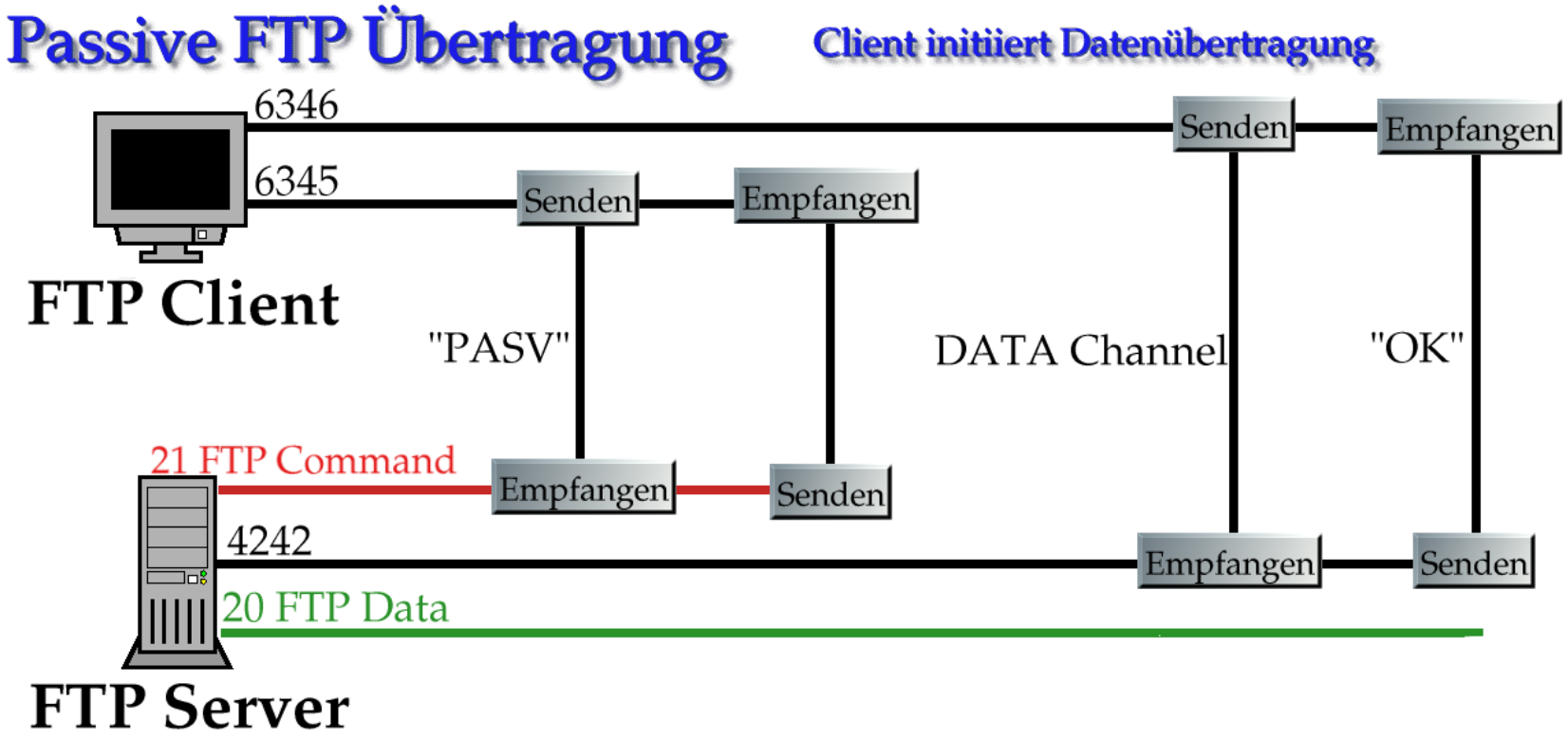
- FTP benutzt zwei TCP Verbindungen
- Server empfängt Kommandos auf Port 21/TCP
- Daten werden seperat im
  - aktiven Modus oder
  - passiven Modus

  übertragen

FTP im aktiven Modus



FTP im passiven Modus



Filterregeln für FTP (Server)

- FTP Kommandokanal braucht zwei Regeln

  Client IP 1024-65535 $\longrightarrow$ Server IP 21 (NEW, ESTABLISHED)

  Client IP 1024-65535 $\longleftarrow$ Server IP 21 (ESTABLISHED)

- aktives FTP braucht zwei Regeln

  Server IP 1024-65535 $\longrightarrow$ Client IP 1024-65535 (RELATED, ESTABLISHED)

  Server IP 1024-65535 $\longleftarrow$ Client IP 1024-65535 (ESTABLISHED)

- passives FTP braucht zwei Regeln

  Client IP 1024-65535 $\longrightarrow$ Server IP 1024-65535 (RELATED, ESTABLISHED)

  Client IP 1024-65535 $\longleftarrow$ Server IP 1024-65535 (ESTABLISHED)

Reihenfolge von Filterregeln

- erste zutreffende Regel gilt („first match”)
- mehrere Regeln können auf ein Paket passen
- Gefahr: Pakete können „durchrutschen”
  - allgemeinere Regel steht vor spezieller Regel
  - allgemeinere Regel läßt Paket durch
  - spezielle Regel würde Paket verbieten bzw. nicht erfassen
  - $\longrightarrow$ Paket passiert!
  - Default DROP wird nie erreicht

Reihenfolge (1)

1. 192.168.1.0/24 $\longrightarrow$ 0/0 ACCEPT
2. 192.168.1.234 $\longrightarrow$ 10.0.0.0/24 ACCEPT
3. Default DROP

- 192.168.1.0/24 darf ins Internet
- 192.168.1.234 soll nur ins Netzwerk 10.0.0.0/24 dürfen
- Regeln haben falsche Reihenfolge dafür!

Reihenfolge (2)

1. 192.168.1.234 $\longrightarrow$ 10.0.0.0/24 ACCEPT
2. 192.168.1.234 $\longrightarrow$ 0/0 DROP
3. 192.168.1.0/24 $\longrightarrow$ 0/0 ACCEPT
4. Default DROP

- 192.168.1.0/24 darf ins Internet
- 192.168.1.234 soll nur ins Netzwerk 10.0.0.0/24 dürfen
- Reihenfolge jetzt richtig(er)

Grundregeln für Regelreihenfolgen

- Regeln immer an letzte Stelle anhängen

  beim Linux® Netfilter geschieht dies durch den „Append” Modus

- spezielle Regeln zuerst

  Dies schließt ACCEPT und DROP ein.

- allgemeine Regeln zum Schluß

- Default DROP

  Dinge, die man vergißt, gehen damit automatisch nicht.

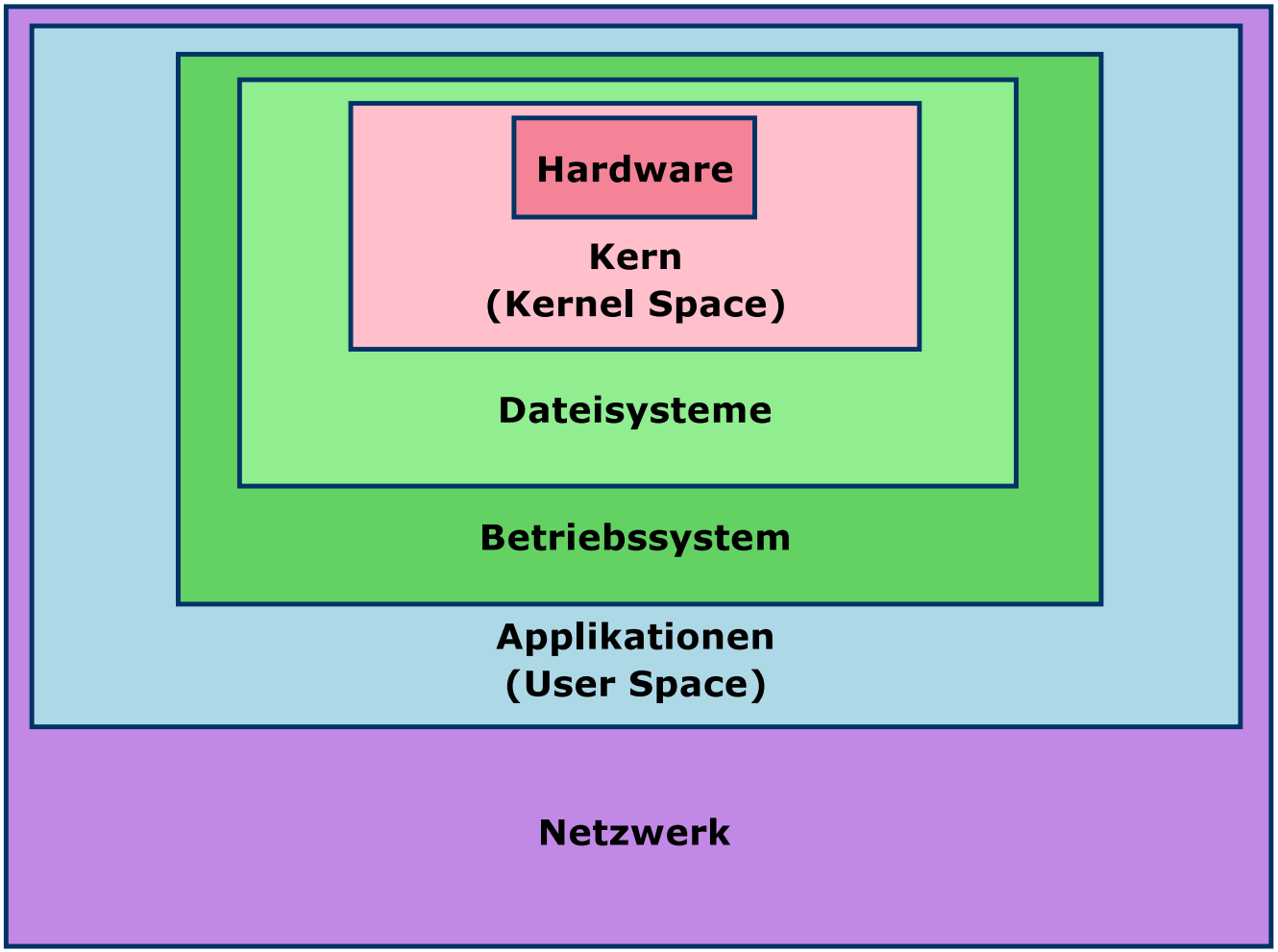
Defence in Depth

- Sicherheit muß durch alle Bereiche gehen
- Sicherheit darf nie am Paketfilter aufhören
- Sicherheit muß bis zum letzten Client durchgezogen werden
- Grundsatz:
  vom äußeren Paketfilter/Modem bis zur letzten Tastatur

Defence in Depth - Beispiel

- externer Router
- externer Paketfilter
- Betriebssysteme der Server in der DMZ
- interne Router/Paketfilter
- Betriebssysteme der Clients
- alle Applikationen
- alle Benutzerrechte und Dateisysteme
- alle Benutzer

Systemabsicherung (Hardening)



Absichern von Server und Client

1. minimale Installation und Absichern der Maschine
2. Deaktivieren aller nicht benötigten Dienste
3. Installieren und Anpassen der benötigten Dienste
4. Rekonfigurieren der Maschine in den Einsatzzustand
5. Testen und Prüfen der Sicherheit
6. Verbinden der Maschine mit dem Einsatznetzwerk

Minimalinstallation und Absichern

- Minimalinstallation heißt: *minimale Sorgen*
- Beheben aller bekannten Systemfehler
- Verwenden bzw. Aufstellen einer Checkliste

  angepaßt an Betriebssystem und Verwendungszweck der Maschine
- Sichern der System-Logs

  Exportieren der Logs auf eine seperate Maschine oder in ein seperates Netzwerk
- Testen der Maschine mit Scannern
- Streßtests, Backup & Recovery Tests

Deaktivieren aller unnötigen Dienste

- Säubern der Init-Einstellungen
- Deaktivieren von Netzwerkdiensten

  Je nach Verwendung kann das sein: NFS, RPC-Dienste, Bootprotokolle, DHCP, Computersuchdienst, Microsoft Netzwerk, IP Routing/Forwarding, FTP-Server, Maildienste, ...
- Idealfall: Löschen der unnötigen Dienste

  Alles, was nicht vorhanden ist, kann keine Probleme machen.
- Deinstallation von nicht benötigter Software

Anpassungen

- Ersetzen von Standardsoftware
- Erstellen von Zugriffsbeschränkungen:
  Principle of least privilege
- Konfigurieren von Berechtigungen

  Wichtig: keine Gastkonten, keine unnötigen Benutzerkonten
- Installieren von Überwachungssoftware

  Logüberwachung, Betriebsdatenkontrolle, ...
- Fingerprint aller kritischen Dateien

  kryptographische Checksummen, Archivieren der Fingerprints für späteren Vergleich

Beispiel: Apache Webserver

- GNU/Linux® Basissystem
- Apache Webserver
- MySQL Datenbankserver
- Perl
- Secure Copy (SCP) Server

Minimalinstallation (1)

- Partitionslayout

/	300 MB	Kern & wichtige Programme
/usr	2 GB	Applikationen
/usr/src	1 GB	Quellcodes
/home	40 GB	Webspace
/var	60 GB	Logs, Spool & Datenbank
/tmp	4 GB	temporäre Dateien
Swap	4 GB	Auslagerung (mind. 2faches phys. RAM)

RAID, Logical Volumes, Dateisysteme und dergleichen werden hier nicht betrachtet

- Distribution
  - Standarddistribution mit eingeschränkter Installation
  - „gehärtete” Distribution (Trustix, Adamantix)
  - minimal notwendige Paketauswahl
  - Developmentpakete notwendig für bestimmte Installationen
  - aktuellste Updates installieren, bekannte Probleme beheben
- System nicht mit Internet verbinden!

Minimalinstallation (2)

- Deaktivieren aller unnötigen Dienste
- Deaktivieren von Netzwerkdiensten
  Wir brauchen nur:
  - Port 22/TCP für SCP und SSH
  - Port 80/TCP und eventuell Port 443/TCP für Apache
- Exportieren der Systemlogs auf Logserver
- Scannen, Rebooten, Scannen, Überprüfen
- System nicht mit Internet verbinden!

Anpassen der Installation (1)

- Installieren von Apache und MySQL
  - vom Quellcode (empfohlen) oder als Paket
  - Apache mit *suEXEC* Option installieren
  - eigene Benutzer und Gruppen für Apache, MySQL und CGIs anlegen
  - Verzeichnisse nur für diese zugänglich machen
  - Start-/Stopskripte anpassen

  Wenn möglich, dann Applikationen in *chroot jails* einsperren.
- Installieren von SSH und scponly
  - SSH Zugriff nur mit Public Keys erlauben
  - ausschließlich SSH Protokoll V2 verwenden
- Server mit Zeitquelle synchronisieren
- System nicht mit Internet verbinden!

Anpassen der Installation (2)

- Konfigurieren von Apache und MySQL

  Speichergrenzen, Anzahl gleichzeitiger Verbindungen, . . .

- Konfigurieren von Systemlimits

  beispielsweise `/etc/security/limits.conf`

- Fingerprint aller wichtigen Dateien

  Fingerprints extern speichern

- System nicht mit Internet verbinden!

Testen der Installation

- Scannen des Servers

  `nmap`, nessus, hping2, `nikto`, `...`

- Simulieren von Last
  - geskriptete HTTP Clients um Lastgrenzen zu testen
  - auch URLs mit CGIs testen (Datenbanklast)
- Zugänge prüfen (Passphrases)
- Konfigurationen nochmals prüfen
- Erst jetzt das System mit dem Internet verbinden!

Virtual Private Networks (VPNs)

- erlauben geschützten Transport von Daten
  - Einsatz von Verschlüsselung
  - Verbinden von Netzwerken
  - Nutzen von unsicheren Netzen als Transportmedium
- verhindern Lauschangriffe
- können neue Probleme schaffen, wenn sie nicht richtig eingesetzt werden

  Stichwort: Netzwerke transparent vergrößern und das Gefahrenpotential erhöhen

VPN Implementationen

- IPsec

  stammt aus IPv6 Entwicklung, bei IPv4 nachgerüstet

- Point-to-Point Tunneling Protocol (PPTP)

  frühes VPN Protokoll, leicht zu konfigurieren, schwierig durch Paketfilter zu leiten

- Layer 2 Tunneling Protocol (L2TP)

  Methode zum Aufbauen eines Tunnels auf Layer 2, Verschlüsselung muß über andere Wege zusätzlich konfiguriert werden

- Web VPN

  Tunneln von Resourcen über HTTPS

- OpenVPN

Virtual Private Networks (VPN)

- Virtual Private Networks
  - verbinden Netzwerke über nicht vertrauenswürdige Netze
  - schützen transferierte Daten durch Verschlüsselung
  - kontrollieren Verbindungsaufbau durch Schlüssel
- Realisierung vielfältig
  - IP Security (IPsec) für IPv4 und IPv6
  - Point-to-Point-Tunneling Protocol (PPTP)
  - OpenVPN mittels SSL/TLS Protokoll
  - Layer 2 Tunneling Protocol (L2TP) innerhalb VPNs

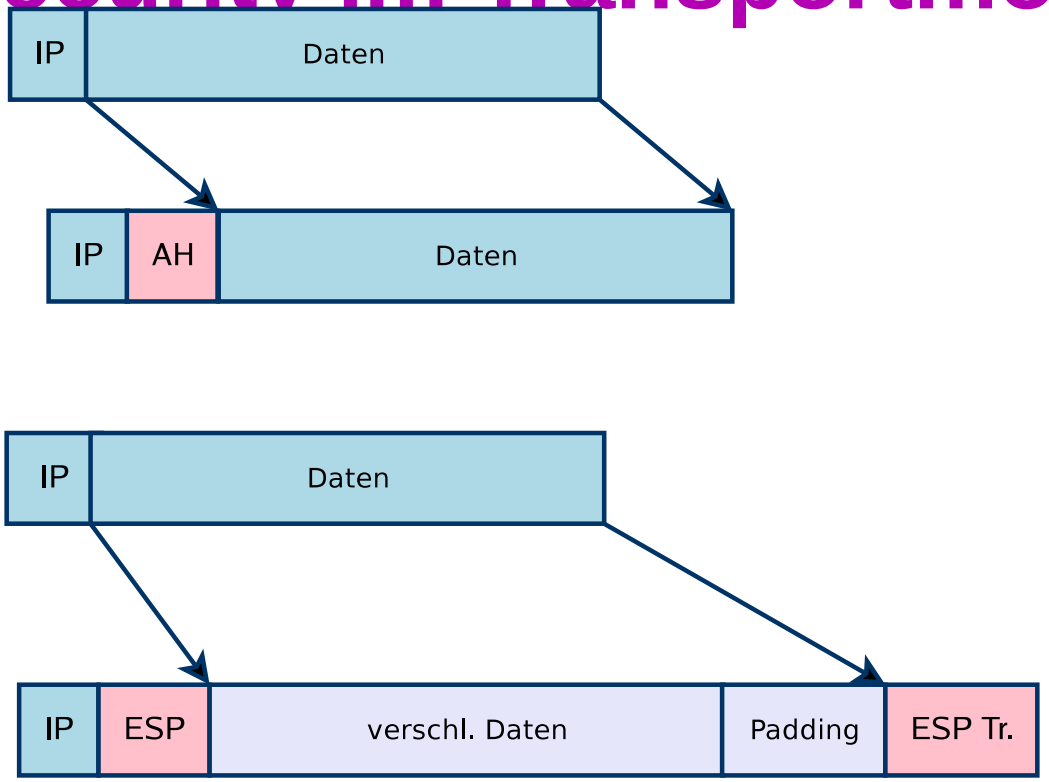
IP Security (IPsec)

- zwei zusätzliche Protokolle
  - Authentication Header (AH) für Integritätsschutz
  - Encapsulation Security Payload (ESP) für Integritätsschutz und Verschlüsselung

  beide verwenden symmetrische Schlüssel
- Datenübertragung geschieht per
  - Transportmodus zwischen zwei Hosts
  - Tunnelmodus zwischen zwei Netzwerken
- Internet-Key-Exchange-Protokoll (IKE)

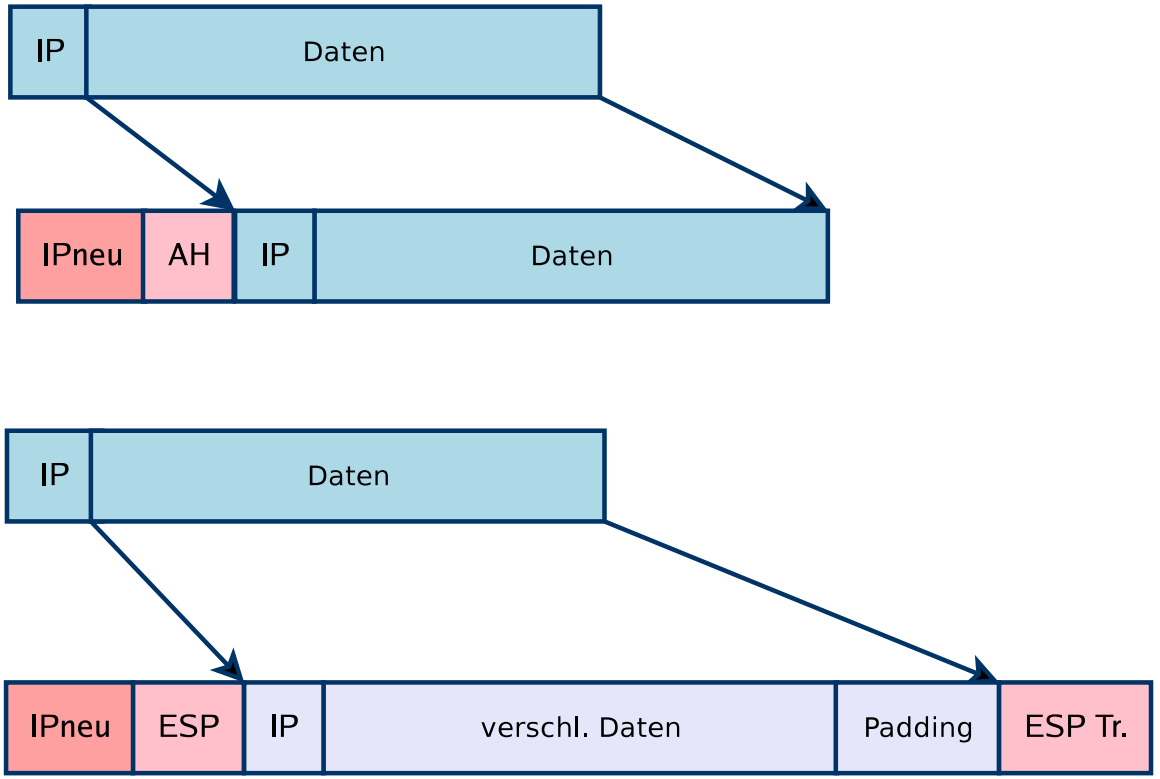
  kann eingesetzt werden, um sicheren Schlüsselaustausch durchzuführen. IKE wird über UDP-Pakete auf Port 500 (Quell- und Zielport) benutzt.

IP Security im Transportmodus



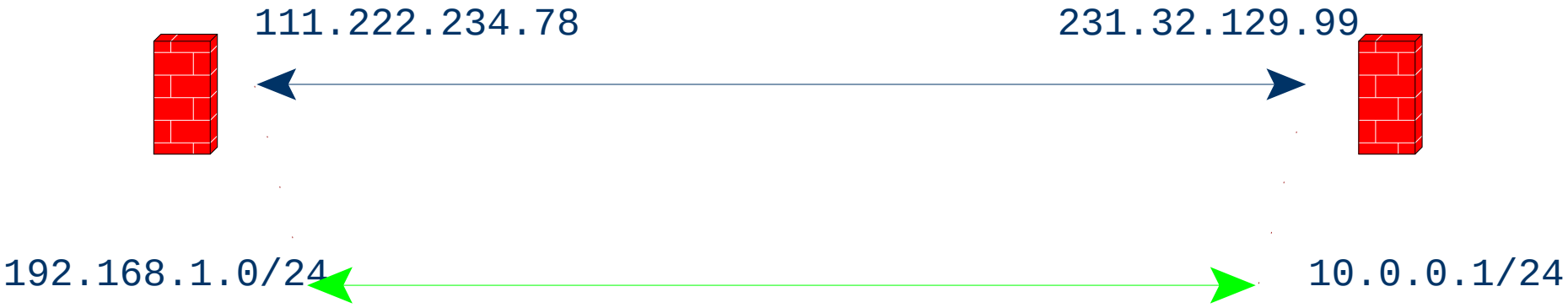
Im Transportmodus bleiben die IP Kopfdaten erhalten. Lediglich der Inhalt wird verschlüsselt. Die IP Kopfdaten werden sowohl bei AH als auch bei ESP durch Hash-Algorithmen vor Fälschung geschützt.

IP Security im Tunnelmodus



Im Tunnelmodus ist nicht erkennbar welche IP mit welcher spricht. Die IP Kopfdaten werden verkapselt und verschlüsselt (im Falle von ESP).

Verbinden von Netzen



VPN Tunnel

VPN Verbindung zwischen zwei Gateways. Der grüne Pfeil markiert den Tunnel, der die Netz zu beiden Seiten der Gateways verbindet. Die dunkelblaue Linie ist die physikalische Netzwerkverbindung, über welche der Tunnel realisiert ist.

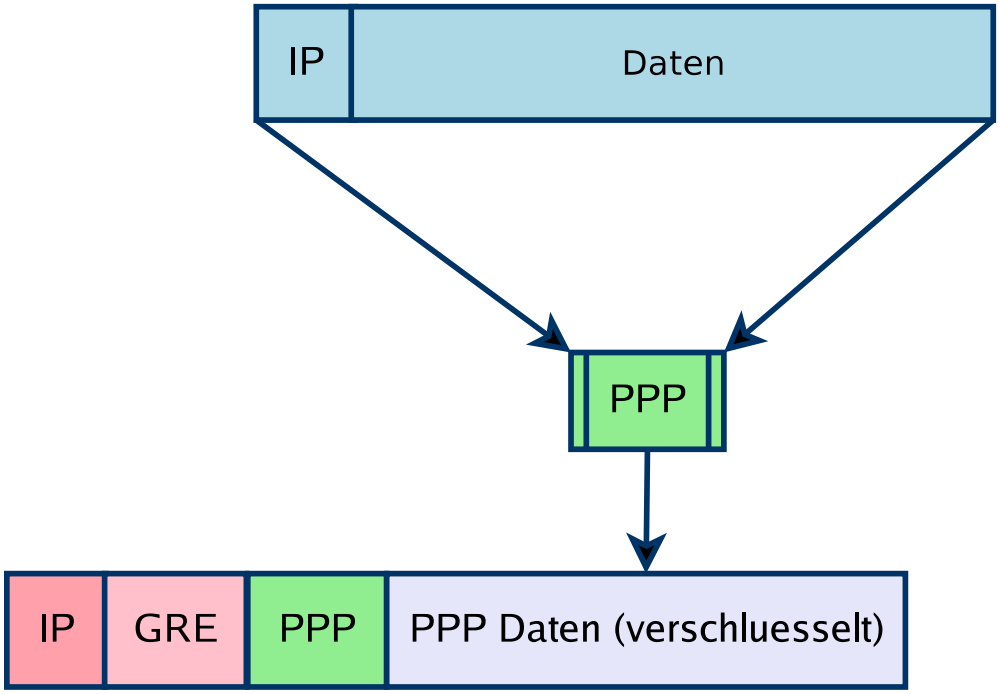
Point-to-Point Tunneling Protocol (PPTP)

- Verkapselung der IP Pakete in PPP Pakete
- Verschlüsseln mit RC4 Algorithmus

  RC4 nicht ausreichend sicher, da Microsofts Point-to-Point Encryption Protocol (MPPE) Paßworte als Basis für Schlüssel verwendet; Extensible Authentication-Protocol (EAP) verbessern die Sicherheit ab MS Windows 2000
- weitere Verkapselung in GRE Pakete

  GRE = Generic Routing Encapsulation Protokoll

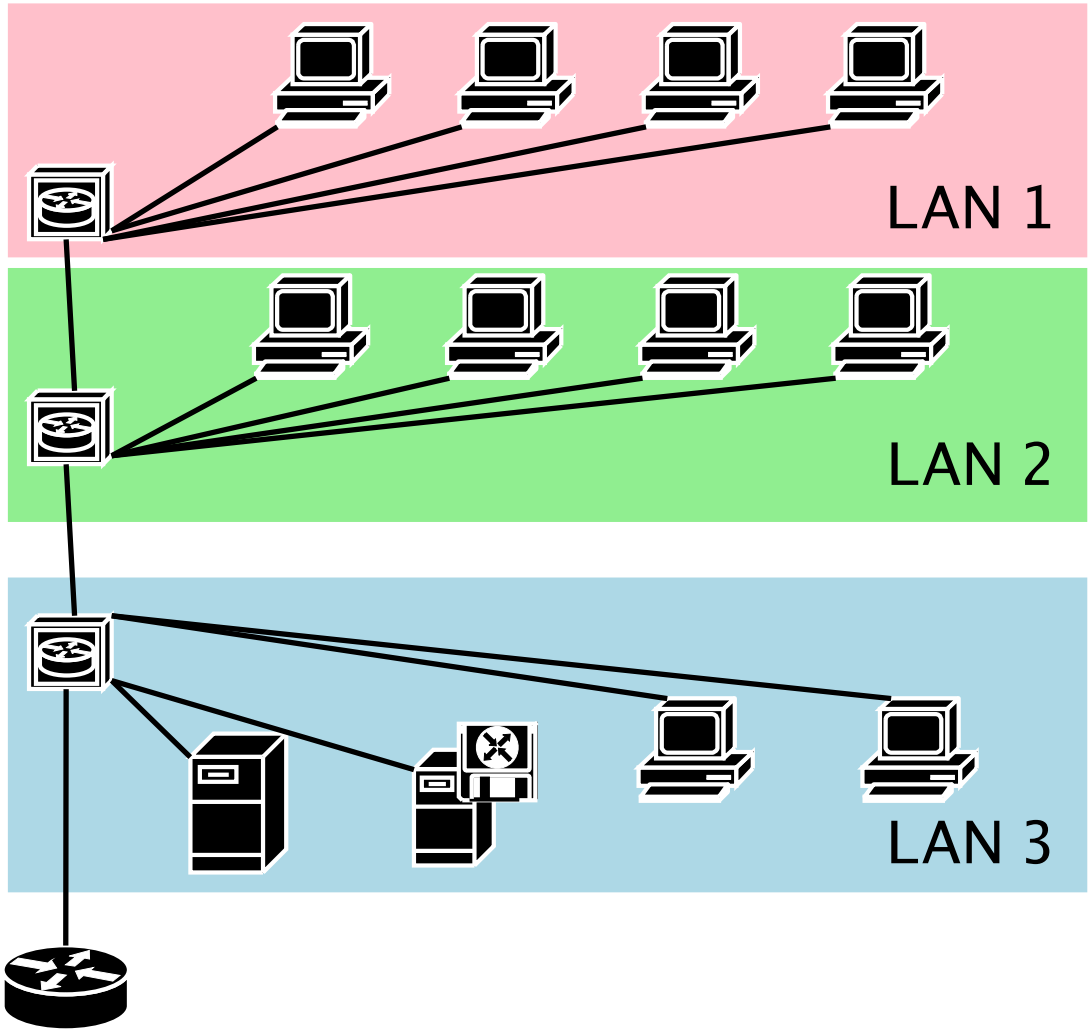
PPTP Verkapselung



Dargestellt ist die mehrfache Verkapselung der ursprünglichen Daten in GRE Pakete.

Trennen von Netzwerken (Layer 2)

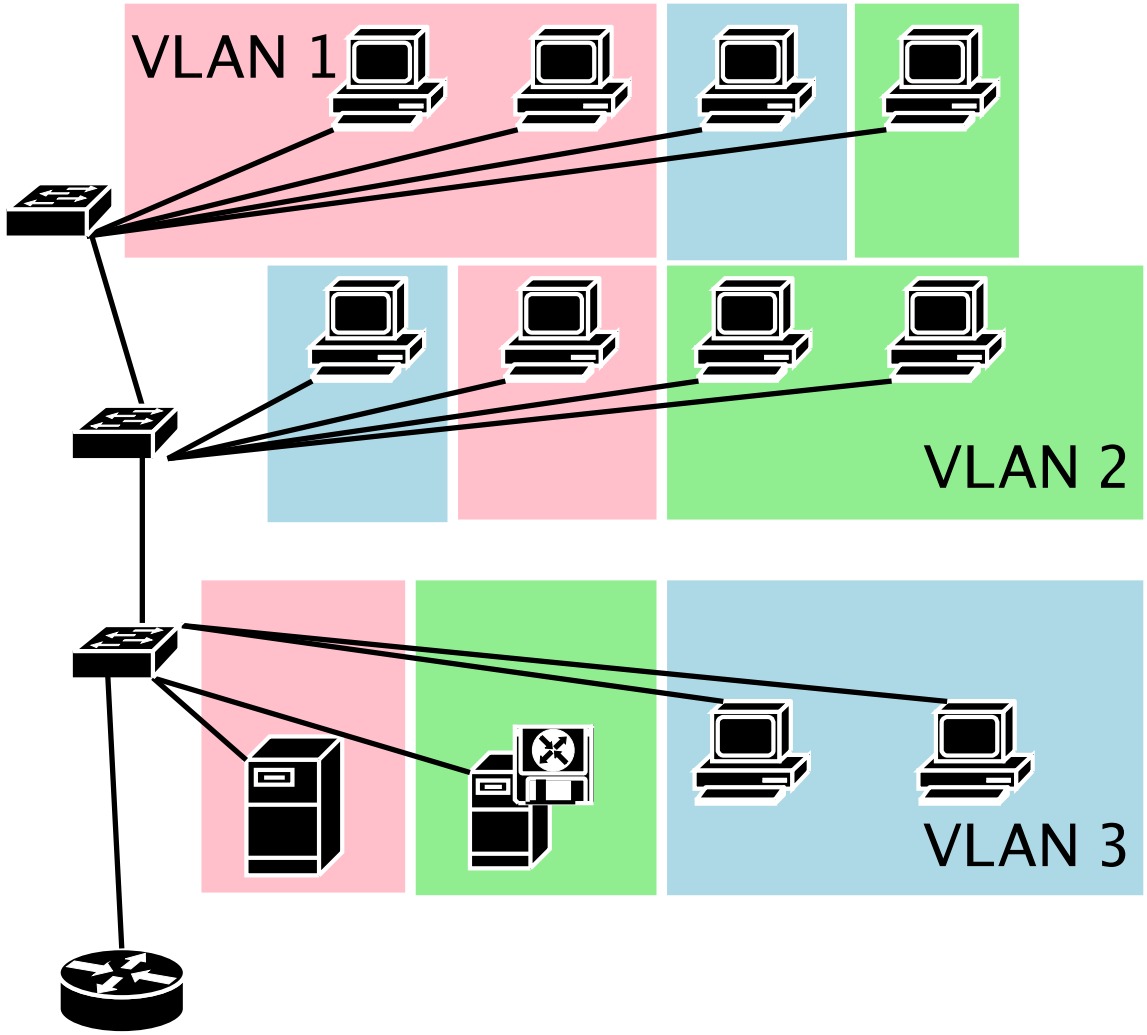
Trennen von Netzwerken traditionell



Virtual LAN (VLAN)

- Virtual LANs unterteilen Netze lokal
- wirksam in Ethernetsegmenten
  - Ethernetframes bekommen ein zusätzliches VLAN-Tag
  - Switches/Systeme erkennen Tag und leiten Frames weiter
- Trennen nach
  - Protokollen (IP, IPX, AppleTalk, etc.)
  - MAC Adressen
  - IP Subnetzen
  - Switchports
- VLAN arbeitet ohne Verschlüsselung
- VLAN arbeitet nur mit passenden Komponenten

Trennen von Netzwerken mit VLANs



VLAN Hopping

- VLAN Technologie dient zur Trennung von Segmenten
- VLANs sind keine reine Sicherheitsmaßnahme
- VLAN Trennung kann temporär aufgehoben werden
- richtige Konfiguration beachten

VLAN Insecurity

Cisco Documentation zum Thema

Referenzen

- Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman, *Building Internet Firewalls*, 2nd Edition, O'Reilly & Associates, Inc., 2000.
- Deborah Russel, G. T. Gangemi, *Computer Security Basics*, O'Reilly & Associates, Inc., 1rst Edition 1991.
- Kevin D. Mitnick, William L. Simon, *The Art of Deception*, Wiley Publishing, Inc., 2002.
- S. Garfinkel, G. Spafford, *Practical UNIX® and Internet Security*, O'Reilly & Associates, Inc., 2nd Edition April 1996.
- Bruce Potter, Bob Fleck, *802.11 Security*, O'Reilly & Associates, Inc., 2003.
- Ralf Spenneberg, *VPN mit Linux®*, Addison-Wesley Verlag, 2004.
- I. Ristic, *Apache Security*, O'Reilly, ISBN 0596007248, März 2005.
- IT-Sicherheit (Publikationen des Bundeskanzleramts)
- The Open Web Application Security Project (OWASP)
- Windows 2000 Security Hardening Guide: Operating System Installation
- Hardening and Tightening Security on Your Server/Network

Über dieses Dokument

- Autor: René Pfeiffer
- Erstellt mit LaTeX und FoilTeX
- Dokumentensammlung unter

  http://web.luchs.at/information/docs.php