JOHANNES KEPLER
UNIVERSITÄT LINZ
INFORMATIK

LANCASTER
UNIVERSITY

Firewalls:
Grundlagen, Schwächen, Möglichkeiten

10. Februar 2006, 14:15
Technologiezentrum Attnang-Puchheim

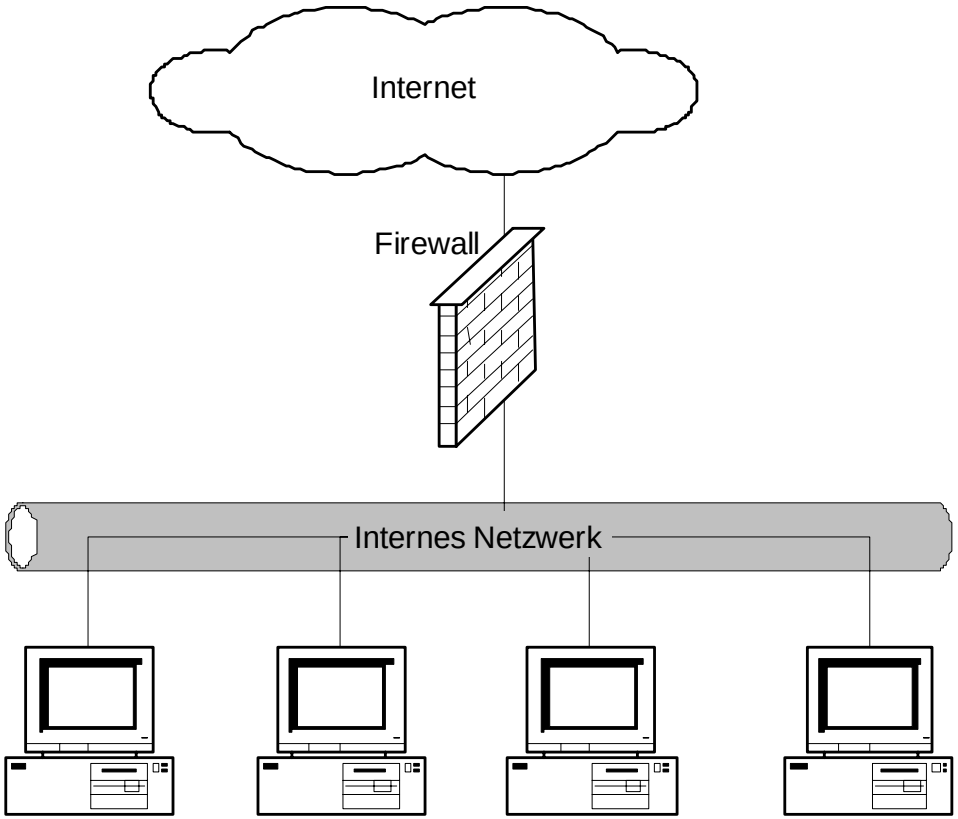
Rene Mayrhofer
Institut für Pervasive Computing, Univ. Linz
Computing Department, Lancaster University
rene@mayrhofer.eu.org

Gibraltar Firewall Entwicklungsleiter
http://www.gibraltar.at/

Vortragsinhalt

- **Grundlagen von Firewalltechnologien**
  - Paketfilter
  - Proxy
  - NAT
  - VPN
- **Potenzielle Angriffsmöglichkeiten (Demo)**
  - ungefiltert
  - mit Paketfilter
  - mit Proxy
  - direkt
- **Abwehrmöglichkeiten**
  - Filter auf verschiedenen Ebenen
  - Verhinderung vs. Verminderung vs. Erkennung
  - Erweiterung auf zusätzliche Ebenen

Was ist eine Firewall?



Firewall - Grundlagen

- Eine Firewall ist der Schwerpunkt der Netzwerk-Sicherheitsmaßnahmen
  - gesamter Verkehr muss Kontrollpunkt passieren
  - Verkehr kann überwacht werden
- Durchsetzen der Sicherheitspolitik
  - verhindert ungewollte Zugriffe von außen
  - verhindert, dass Daten nach außen gelangen
- Protokollierung
  - protokolliert den laufenden Netzwerkverkehr
- Verkleinerung der Angriffsfläche
  - trennt verschiedene Bereiche des Firmennetzwerks
  - DMZ (Demilitarisierte Zonen)

- **schützt nur Verbindungen, die durch sie hindurchgehen**
- schützt nicht / nur bedingt gegen Angriffe von innen
- bietet keinen vollständigen Virenschutz
- kann sich nicht selbst einrichten

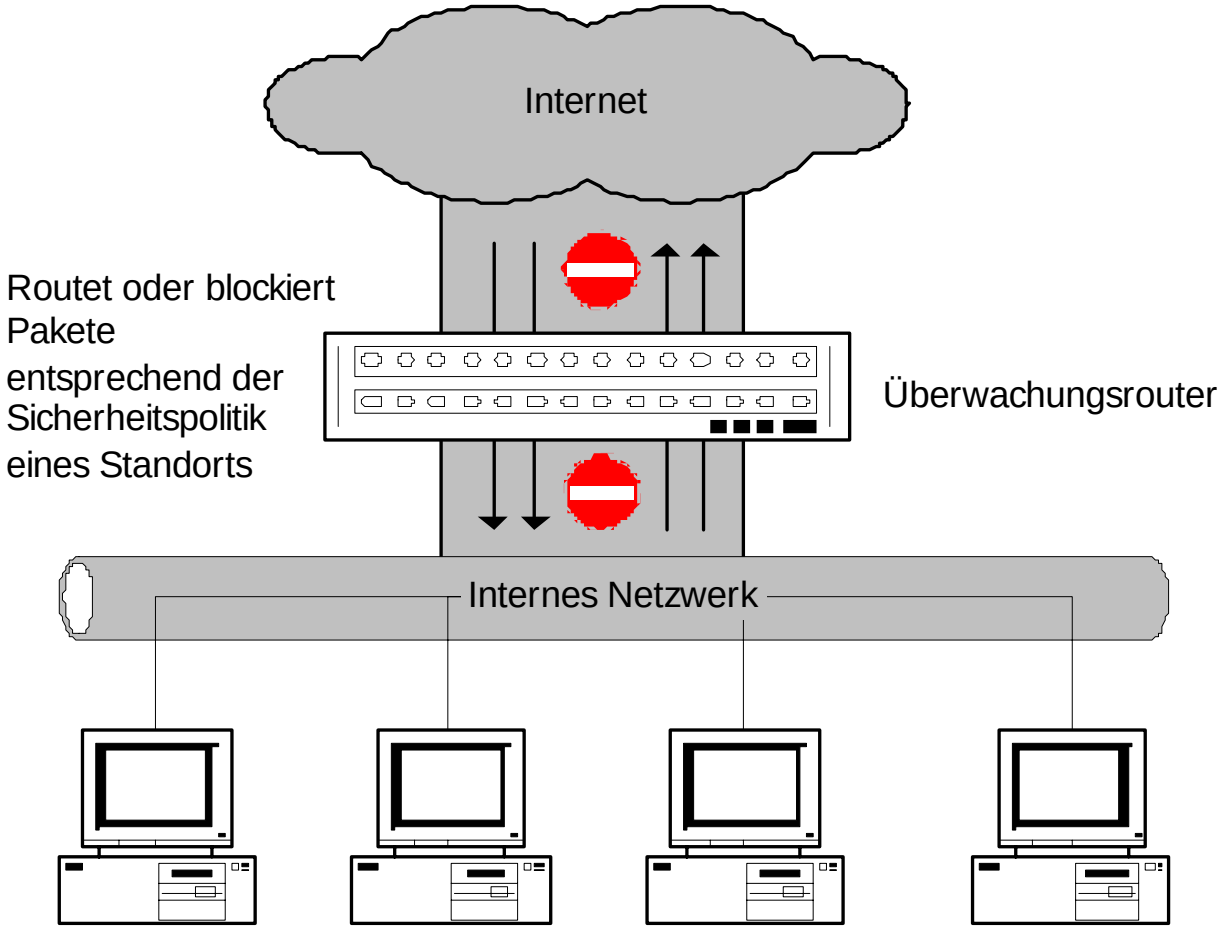
Firewall - Techniken

- **Paketfilterung**: Kern einer jeden Firewall, Umsetzung der Sicherheitsrichtlinien und Überwachung des Netzwerktraffics
- **Proxy-Dienste**: Sicherheits- und Performancefunktion “Application Level Inspection”, “Deep Inspection”, “Content Inspection”, …
- **NAT** (Network – Adress – Translation): Adressübersetzung
- **VPN** (Virtuelle Private Netzwerke): **IPSec**

ISO/OSI – 7-Schichtenmodell

Schicht	ISO/OSI-Modell		TCP/IP-Modell
7	Applikationsschicht	Applikations-Protokolle	telnet, ftp, nfs rlogin, DNS smtp, snmp X-Windows Socket library
6	Präsentationsschicht		
5	Kommunikations-Steuerungsschicht		
4	Transportschicht	Transport-Protokolle	TCP UDP
3	Netzwerkschicht	Internetwork-Protokolle	IP EGP, RIP ICMP ARP, RARP
2	Sicherungsschicht	Network-Access-Protokolle	Ethernet CSMA/CD Token Ring FDDI
1	Bitübertragungsschicht		

Paketfilterung



Paketfilter

- Paketfilter arbeiten auf **Ebenen 3 und 4** des ISO/OSI Schichtenmodells
- routen Pakete zwischen internen und externen Hosts
- arbeiten selektiv
- erlauben und blockieren Pakete

- Paket-Header für IPv4:
  - IP-Quelladresse
  - IP-Zieladresse
  - Protokoll
  - TCP oder UDP-Quellport
  - TCP oder UDP-Zielport
  - ICMP-Meldungstyp
  - Paketgröße
  - ...

- Prinzipielle Unterscheidung zwischen **stateless** und **stateful** Filterung

ISO/OSI – 7-Schichtenmodell

Schicht	ISO/OSI-Modell		TCP/IP-Modell
7	Applikationsschicht	Applikations-Protokolle	telnet, ftp, nfs rlogin, DNS smtp, snmp X-Windows Socket library
6	Präsentationsschicht		
5	Kommunikations-Steuerungsschicht		
4	Transportschicht	Transport-Protokolle	TCP UDP
3	Netzwerkschicht	Internetwork-Protokolle	IP EGP, RIP ICMP ARP, RARP
2	Sicherungsschicht	Network-Access-Protokolle	Ethernet CSMA/CD Token Ring FDDI
1	Bitübertragungsschicht		

Paketfilter

- prinzipielle Unterscheidung in stateless und stateful inspection
- Stateless
  - statische Paketfilterung
  - unabhängig von bereits eingetroffenen Paketen
  - Entscheidung über Aktion (Durchlassen oder Blockieren) für jedes einzelne Paket
- Stateful
  - dynamische Paketfilterung
  - zustandsabhängig
  - untersucht nicht nur den Header eines Pakets, sondern auch den Inhalt
  - beobachtet den Status der Verbindung
  - Kontext-Analyse der Verbindung

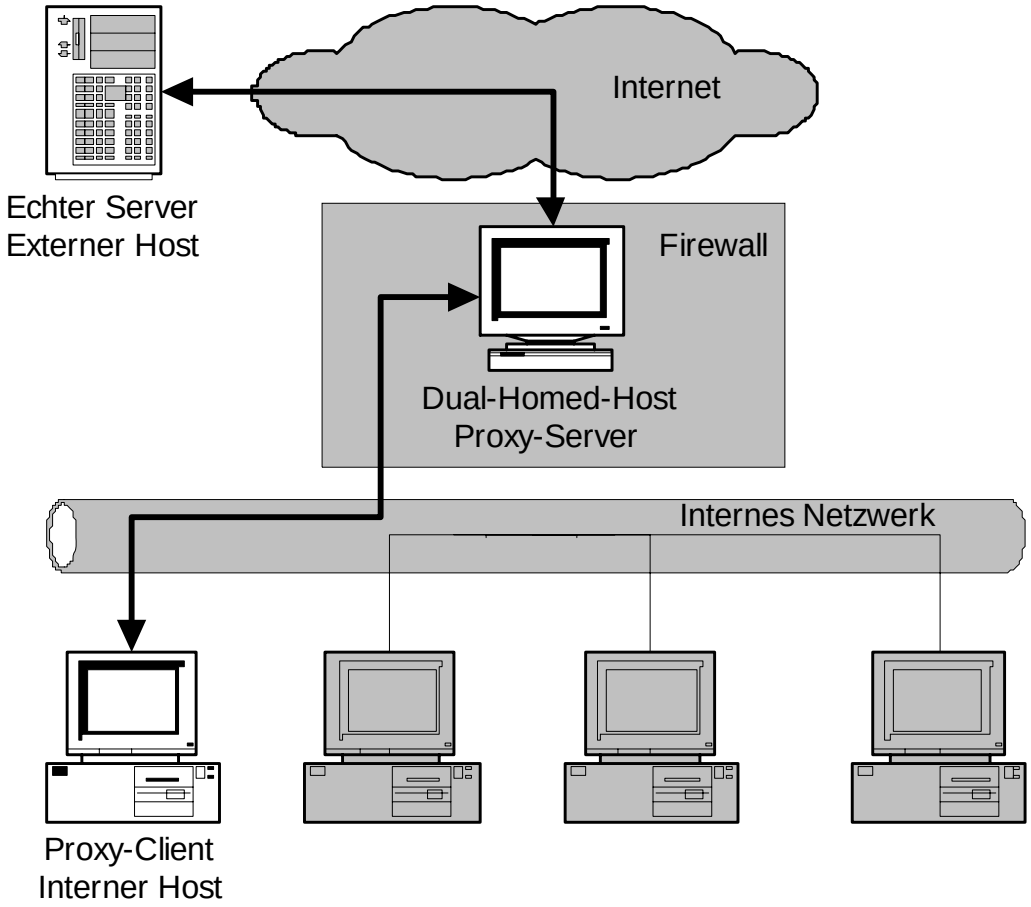
Paketfilter

- bekannte Daten
  - Schnittstelle, an der das Paket empfangen wurde.
  - Schnittstelle, an die das Paket weitergeleitet werden soll.
  - ob das Paket eine Antwort auf ein anderes Paket war **(stateful)**
  - wie viele andere Pakete zuvor zu oder von dem gleichen Host übertragen wurden **(stateful)**.
  - ob das Paket identisch mit einem zuvor gesendeten Paket ist.
  - ob das Paket Teil eines größeren Pakets ist, das in einzelne Teile zerlegt (fragmentiert) wurde (deshalb sehr oft Defragmentierung auf Firewalls, weil die weiteren Fragmente außer dem ersten keinen IP-Header mehr haben).

Paketfilter - Aktionen

- Standard
  - Paket schicken (ACCEPT)
  - Paket verwerfen (DROP)
  - Paket mit Fehlermeldung zurückweisen (REJECT)
  - Informationen über Paket aufzeichnen (LOG)
- Erweitert
  - einen Alarm auslösen
  - Bei stateful: neue Verbindung in Verbindungstabelle eintragen
  - Optionale Zusatzfunktionen: Paket in bestimmter Klasse zählen, Paketgröße zu Quota addieren, Paket in Liste von kürzlich gesehenen Hosts eintragen, ...
  - Paket vor dem Weitersenden verändern !
  - Paket an einen lokalen, transparenten Proxy weitergeben

Proxy - Dienste



Proxy - Dienste

- Proxys arbeiten auf den **Ebenen 5 bis 7**
- Stellvertreter
- spezielle Anwendungen oder Server-Programme, die Benutzeranfragen an Internet-Dienste entgegennehmen und sie an den eigentlichen Dienst weiterleiten.
- Application-Level-Gateways
- Erhöhung der Sicherheit
- höhere Effektivität des Netzwerks bei caching Proxys
- transparent oder nicht transparent
- Kann für bestimmte Protokolle nötig sein, da Eingriff auf Ebenen 5 bis 7 bei NAT nötig sind (z.B. FTP, H.323)

ISO/OSI – 7-Schichtenmodell

Schicht	ISO/OSI-Modell		TCP/IP-Modell
7	Applikationsschicht	Applikations-Protokolle	telnet, ftp, nfs rlogin, DNS smtp, snmp X-Windows Socket library
6	Präsentationsschicht		
5	Kommunikations-Steuerungsschicht		
4	Transportschicht	Transport-Protokolle	TCP UDP
3	Netzwerkschicht	Internetwork-Protokolle	IP EGP, RIP ICMP ARP, RARP
2	Sicherungsschicht	Network-Access-Protokolle	Ethernet CSMA/CD Token Ring FDDI
1	Bitübertragungsschicht		

Anwendungsbeispiel: transparenter HTTP Proxy

- Möglichkeiten für Layer 7-Transparenz:
  - Direkte Untersuchung der einzelnen Pakete im Kernel (äquivalent zur Prüfung der ISO/OSI Schichten 2 – 4)
    $\Rightarrow$ Problem der Komplexität
  - Umleitung der Pakete an einen erweiterten HTTP Proxy
    $\Rightarrow$ besser durch Modularisierung
- Erscheint so, als ob im Web-Client der HTTP Proxy eingetragen wäre, allerdings ohne den damit verbundenen administrativen Aufwand
- Erlaubt Änderungen an den in HTTP übertragenen Daten, z.B.:
  - Filterung nach erlaubten/unerwünschten URLs (wichtig für öffentliche Zugänge, Schulen, etc.)
  - Transparente Entfernung von Viren (on-the-fly)
  - Benutzerauthentifizierung
  - Entfernung ungewünschter HTML-Tags bzw. Inhalte (ActiveX, JavaScript, Cookies, Pop-Ups, etc.)
  - Beschleunigung durch Caching

NAT – Network – Address Translation

- Veränderung von Netzwerkadressen
- Router verändert Pakete
  - nach außen: Quelladresse wird verändert
  - nach innen: Zieladresse wird verändert
- Häufigste Anwendung: Masquerading / Maskierung:
  - Problem: Durch IPv4-Adressknappheit wird von Providern oft nur eine einzige IP-Adresse zur Verfügung gestellt, obwohl mehrere Computer angebunden werden sollen
  - Lösung: Interne Rechner bekommen private, im Internet nicht verwendbare Adressen. Bei der Weiterleitung ins Internet ersetzt die Firewall die Quelladresse aller Pakete durch ihre eigene, Antwortpakete gehen daher direkt an die Firewall. Durch interne Zuordnungstabellen können die Antwortpakete an die richtigen internen Rechner weitergeleitet werden.

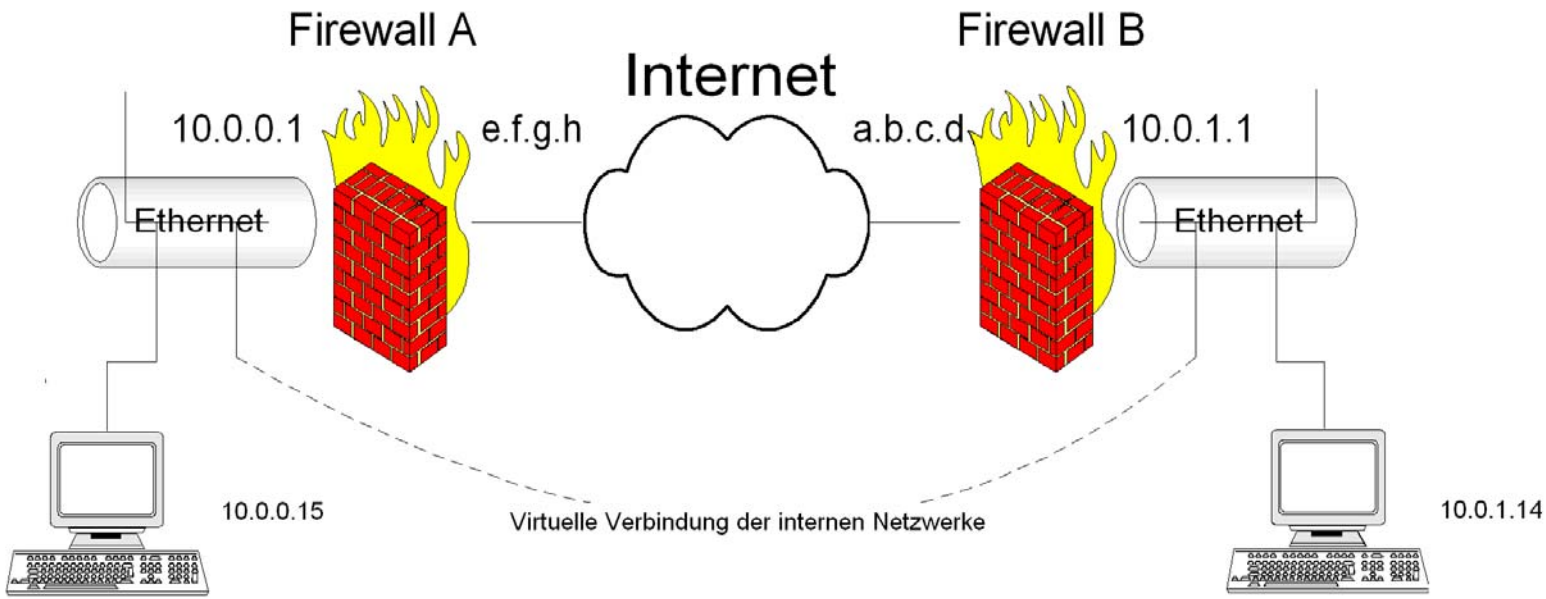
NAT (2)

- Vorteile
  - NAT unterstützt die Kontrolle der Firewall über nach außen gerichtete Verbindungen
  - eingehender Verkehr kann eingeschränkt werden
  - interne Konfiguration des Netzwerks wird verborgen
- Nachteile
  - ev. Problem mit eingebetteten IP-Adressen
  - Verschlüsselung und Authentifizierung erschwert
  - Protokollierung bei dynamischer Adresszuweisung
  - Dynamische Zuweisung von Ports stört Paketfilterung
  - Diverse Protokolle übertragen IP-Adressen der Clients auf Anwendungsebene (z.B. FTP, H.323) ⇨ spezielle Unterstützung muss in NAT eingebaut werden

Virtuelle Private Netzwerke (VPN)

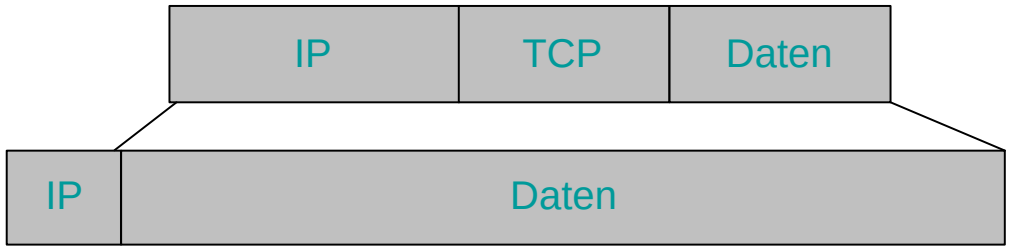
- öffentliches Netz wird privat genutzt
- **Vertraulichkeit** durch Verschlüsselung
- **Integrität** wird geschützt
- **Authentizität** wird sichergestellt
- Daten werden gekapselt
- Methoden
  - End-zu-End Verschlüsselung
  - Tunnel

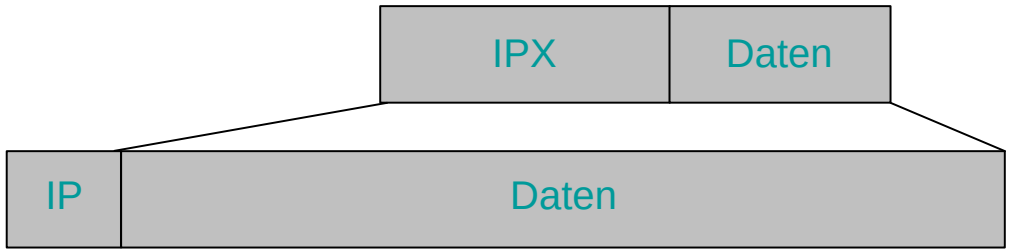
VPN - Prinzip



VPNs für Tunnel

- Meist im Tunnel-Modus betrieben: Rechner hinter den jeweiligen Gateways können transparent miteinander kommunizieren, obwohl die Gateways keine direkte Verbindung haben
- Methode: „Verpacken“ der Pakete, die zwischen den internen Rechnern ausgetauscht werden sollen in IPv4-Pakete





Tunneling

- Verschiedene Implementierungen von Tunneling (Beispiele):
  - GRE (unverschlüsselt)
  - IPv6-over-IPv4 (unverschlüsselt, Übergangsmaßnahme zu IPv6)
  - PPP-over-Ethernet (unverschlüsselt)
  - PPP-over-ATM (unverschlüsselt)
  - L2TP (**unverschlüsselt!**)
  - PPTP (nur bedingt sicher)
  - **IPSec**
  - OpenVPN
  - VTun (unsicher!)
  - CIPE (unsicher!)
  - Tinc (unsicher!)
  - .... (höchstwahrscheinlich unsicher!)

JOHANNES KEPLER
UNIVERSITÄT LINZ
INFORMATIK

LANCASTER
UNIVERSITY

Live-Demo

Abwehrmöglichkeiten

- Abwehr für ungefilterten Fall:
  - Einsatz von Paketfiltern ⇒ Firewalls
  - nur diejenigen Ports öffnen, die auch benötigt werden!
- Abwehr von Attacken auf spezielle Protokolle:
  - defensive Programmierung
  - Einsatz von Proxies zur unabhängigen Kontrolle des Protokollflusses ⇒ Deep Inspection / Layer 7 Firewalls
- Abwehr von Attacken auf Applikationsebene:
  - defensive Programmierung
  - Einsatz von musterbasierten Verfahren statt regelbasierten ⇒ Update wie bei Virensignaturen nötig!
- Abwehr von direkten Attacken:
  - Layer 3 Switches ⇒ Firewalls mit je einem Port pro Endgerät
  - VLANs
  - Erkennen statt Verhindern

Warum VLANs?

- Viele verschiedene Netzwerksegmente an einer Firewall
  - z.B. extern, intern, DMZ Fileserver, DMZ Authentication Server, DMZ Log Server, WLAN, ...
  - Extremfall: jedes Netzwerkgerät in eigenem Ethernet-Segment zur kompletten Trennung (Angriffe von Innen, Wurmbefall etc.)
- Problem: max. 12 – 16 Netzwerkinterfaces bei PC Hardware
- Lösung bei Ethernet: VLANs
  - mehrere virtuelle Segmente auf einem physischen Switch / einer physischen Netzwerkkarte
  - Trennung durch (12 Bit) ID im Ethernet Header
  - Switch wertet ID aus und sendet in entsprechendes Segment
  - „Tagged“ Ports auf dem Switch dürfen IDs mitsenden bzw. bekommen Pakete aus mehreren Segmenten mit dem ID Header
  - „Untagged“ Ports bekommen Ethernet-Pakete ohne ID aus dem jeweils zugewiesenen Segment

Warum Traffic Shaping?

- Netzwerkverbindungen sind nie schnell genug
- Aber: verschiedene Arten von Traffic müssen nicht gleich schnell behandelt werden ⇒ Priorisierung kann bestehende Verbindung besser ausnutzen
- Daher: Übertrage ausgewählte Pakete vor anderen

- Abwehr von Attacken:
  - Denial of Service kann begrenzt werden
  - Würmer breiten sich oft rasant aus ⇒ Traffic Shaping kann bremsen
  - Teergruben

Grenzen des Traffic Shaping

- **Mantra**: Man kann nur Traffic kontrollieren, den man selbst erzeugt! (outgoing)
- Grundprinzip des Internet: Paketvermittlung, nicht Leitungsherstellung
- Daher: es kann nicht kontrolliert werden, was Andere an das eigene System senden (incoming)
- z.B. Media Streaming Server (Webradio, Video, ...) sendet UDP Stream schneller als die eigene Downstream-Bandbreite $\Rightarrow$ Leitung überlastet, keine Kontrollmöglichkeit
- (D)DoS Attacken auf Server können verhindert bzw. gemindert werden, (D)DoS Attacken auf die Internet-Anbindung aber nicht!

Gibraltar - Entstehung

- Diverse Debian-basierte Firewalls seit 1999 im Einsatz
- Projektbeginn Juli 2000 von Rene Mayrhofer
- 2000 – 2002: permanente Weiterentwicklung, gestützt auf Verbesserungsvorschläge aus der wachsenden Community
- 2001: Erstentwicklungen zur Web-basierten Administrationsoberfläche, Entwicklung eines entsprechenden Frameworks an der Johannes Kepler Universität Linz
- 2002: erste Ideen zu einer kommerziellen Version
- 2/2003: Partnerschaft von Rene Mayrhofer mit der eSYS Informationssysteme GmbH. Start der kommerziellen Entwicklung
- 11/2003: Präsentation der Version 1.0. Erste Version mit Webinterface
- 5/2004: Gibraltar v2
- 11/2004: Gibraltar v2.1
- 04/2005: Gibraltar v2.2
- 08/2005: Gibraltar v2.3
- ca. 03-04/2006: Gibraltar v2.4

Gibraltar – Zahlen und Fakten

- geschätzte Installationen der freien Version: über 1000
- kommerzielle Installationen: ca. 200
- Testinstallationen (Testlizenzen) seit 11/2003: ca. 3000
- tägliche Anzahl von Zugriffen auf Homepage: 600-1000
- Mailingliste: knapp 500 Mitglieder

- seit 11/2004: ca. 20 Vertriebs- und Supportpartner in
  - Österreich
  - Deutschland
  - Schweiz
  - Italien
  - USA
  - Finnland
  - Griechenland

Professionelle All-in-One Security Lösung

- Router
- Stateful Inspection Firewall
- Deep Inspection Firewall
- Professioneller Viren Schutz von Kaspersky
- Erweiterter Spam Schutz
- Secure Proxy Server
- Inhaltsfilter
- URL-Filter
- Virtuelle Private Netzwerke
- Traffic Shaping
- Bridging / Transparentes Firewalling
- ab v2.4: VLANs
- ab v2.4: Monitoring und Statistiken
- ab v2.4: Anonymisierung
- ab v2.4: Hot-Failover

Deep Inspection Firewall

- Secure SMTP Relay: eingehend, ausgehend, Attachment Blocking, Block Lists, Viren- und Spamschutz postfix (+TLS+IPv6+SASL++)
- Transparenter HTTP Proxy: keine Clientkonfiguration notwendig, Virenschutz squid (+erweiterte Filter-Patches)
- User Authentifizierung: Benutzerliste, Active Directory Integration, LDAP
- Content Caching
- Content Scanning: Antivirus, Cookies, JavaScript, Active X, optional Filterung nach URLs, Datentypen, ...
- FTP Proxy: transparent ausgehend, eingehend SuSE ftp-proxy + frox
- Transparenter POP3 Proxy: Antivirus, Spamschutz, Schutz vor gefährlichen Attachments p3scan

Vorteil gegenüber Hardware-Lösungen (Watchguard, Sonicwall, Cisco, Zyxel,...)

- Preis
- Skalierbarkeit
- Flexibilität
- Erweiterbarkeit
- Sicherheit durch Open Source
- Sicherheit durch Live-CD-Technology

Vorteile gegenüber Softwarepaketen (Astaro, Checkpoint, Smoothwall,...)

- Preis
- Einfache Installation
- Sicherheit durch Live-CD-Technology
- Keine Festplatte notwendig
- Höhere Ausfallsicherheit

Facts

- Gibraltar ist nicht dauerhaft angreifbar: durch physisch schreibgeschütztes System ist es nicht möglich, so genannten „malicious code“ dauerhaft zu platzieren
- Gibraltar ist ausgereift: seit dem Jahr 2000 wird Gibraltar weltweit von Linux-Experten verwendet, getestet und weiterentwickelt. Gibraltar verwendet tausendfach getestete Komponenten, deren Quellcode frei verfügbar ist.
- Gibraltar ist skalierbar und flexibel: je nach Anforderung kann geeignete Hardware verwendet und auch erweitert werden. Gibraltar unterstützt Load-Balancing und Fail-Over.
- Gibraltar reduziert das Spam-Aufkommen um ca. 95%: durch die Kombination mehrerer Anti-Spam-Maßnahmen (RBL-Listen, Inhaltsanalyse, Bayes-Filter, Razor, DCC, SPF, ...) kann Gibraltar wirksam Spam-Mails erkennen und darauf reagieren.

Gibraltar – ausgewählte Referenzen

- **Referenzen Österreich**
  - Universität Linz
  - Fachhochschule Kufstein
  - Fachhochschule Hagenberg
  - Technikum Wien
  - Stadtgemeinde Vöcklabruck
  - Doubrava
  - COPYright by Josef Schürz
  - Kirsch – Muchitsch und Partner
  - GIG Karasekgroup
  - Wolf Systembau
  - Stubai Werkzeugindustrie
  - Financial Adivsory GmbH
  - Ebnerbau Mondsee
  - HGS Unternehmensberatung
  - Profactor Steyr
  - Datacontact
  - CARE Österreich
- **Referenzen International**
  - Universität Washington
  - Universität der Bundeswehr
  - Universität Stuttgart
  - Universität Oxford
  - P&T Luxemburg
  - Graziano Transmissioni, Italien
  - Scotcomms, GB
  - ARIS AG, Schweiz
  - Calistel, Frankreich
  - COOPService Noncello, Italien
  - Kniel System Electronics
  - Noske-Kaeser GmbH, Deutschland
  - Städtische Überlandwerke Coburg
  - Scene Double, GB

JOHANNES KEPLER
UNIVERSITÄT LINZ
INFORMATIK

LANCASTER
UNIVERSITY

Fragen?

Folien: http://www.gibraltar.at/

Spätere Fragen: rene.mayrhofer@gibraltar.at
office@gibaltar.at

OpenPGP Schlüssel: 0xC3C24BDE
7FE4 0DB5 61EC C645 B2F1 C847 ABB4 8F0D C3C2 4BDE