Juventus Technikerschule HF

Hacking Exposed

HE1: Perimeter Security

Pascal Knecht

Video 0: Überblick

Die 3 grundlegenden IT-Security Massnahmen

- Starke Passwörter
- Software Updates
- Backup (und Restore)

Wichtiger Hinweis

- Dies ist eine Lehrveranstaltung.
- Die im Rahmen der Hacking-Exposed-Vorlesung vermittelten Kenntnisse sollen dazu beitragen, dass Sie Informationssicherheitsaspekte beachten und in Ihren Projekten berücksichtigen.
- Die HE-Vorlesung ist keineswegs als Anstiftung zum Hacken zu verstehen.

Inhalt heute Abend

- Perimeter Security Massnahmen
- Firewalling Grundidee
- Firewalling mit Netfilter und Iptables
- Port Scanning

Ziele

- Sie kennen wichtige Komponenten in der Netzwerksecurity und können deren Zweck beschreiben.
- Sie verstehen wieso es wichtig ist ein Netzwerk in Security-Zonen einzuteilen.
- Sie können Firewall-Regeln lesen, verstehen und selbstständig einfache Regeln schreiben.
- Sie können Nmap bedienen um offene Ports zu finden, Applikationsversionen herauszufinden und eigene Firewall-Regeln überprüfen.

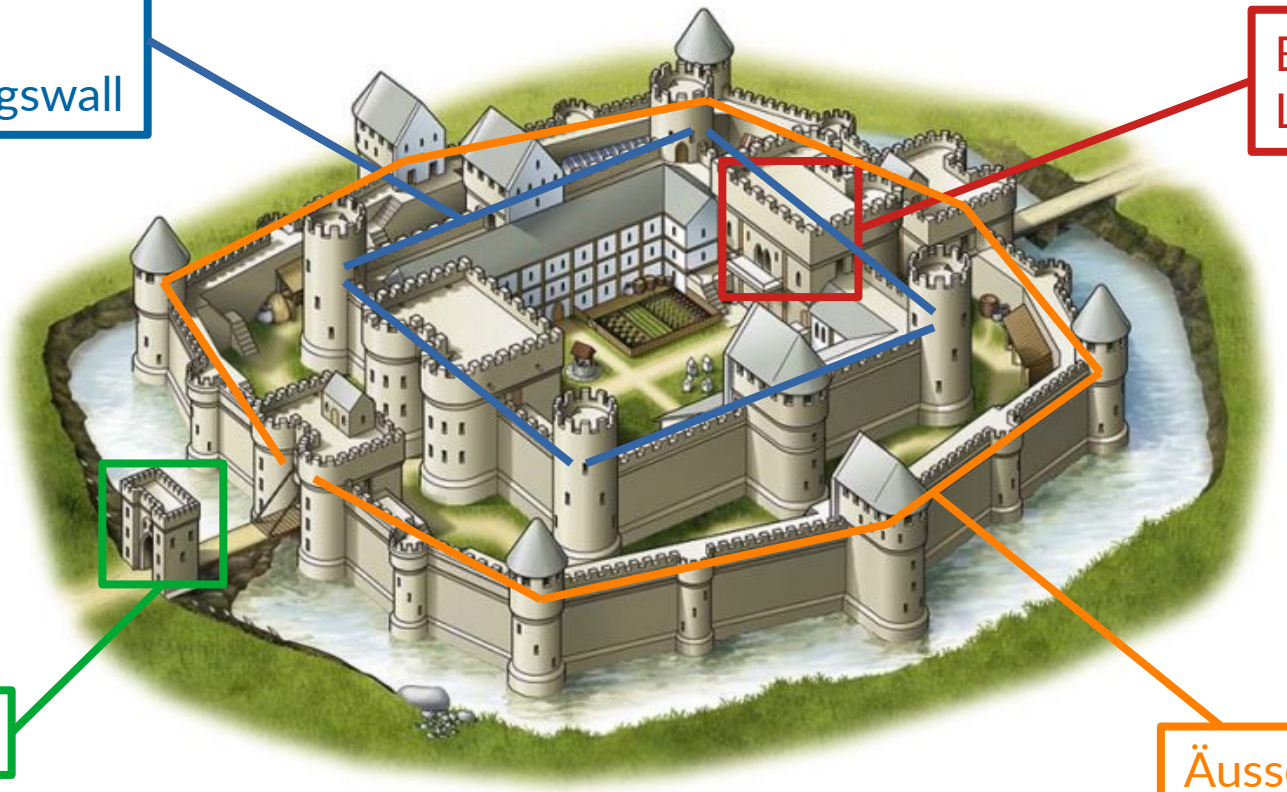
#01 Perimeter Security Massnahmen

Video 1: Perimeter Security Massnahmen

Perimeter Security

Innerer Perimeter
Hauptverteidigungswall

Bergfried
Letzte Bastion

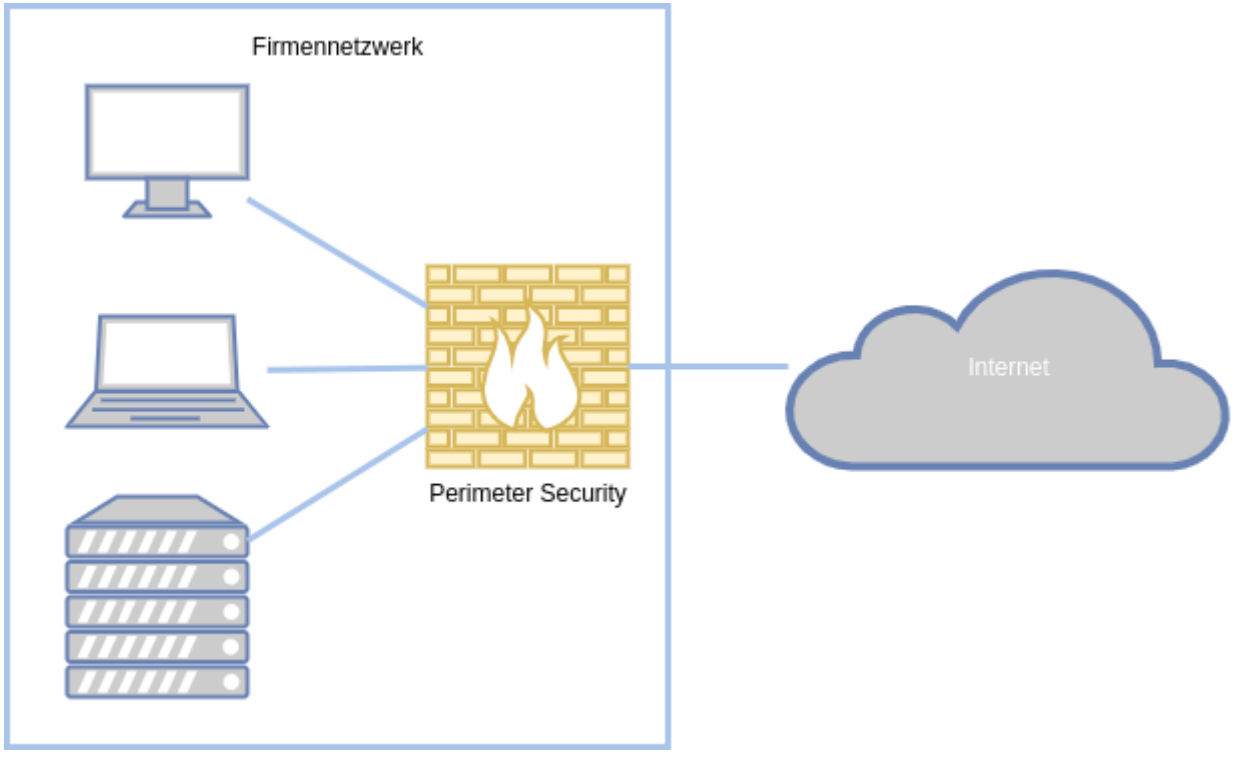


Zutrittskontrolle

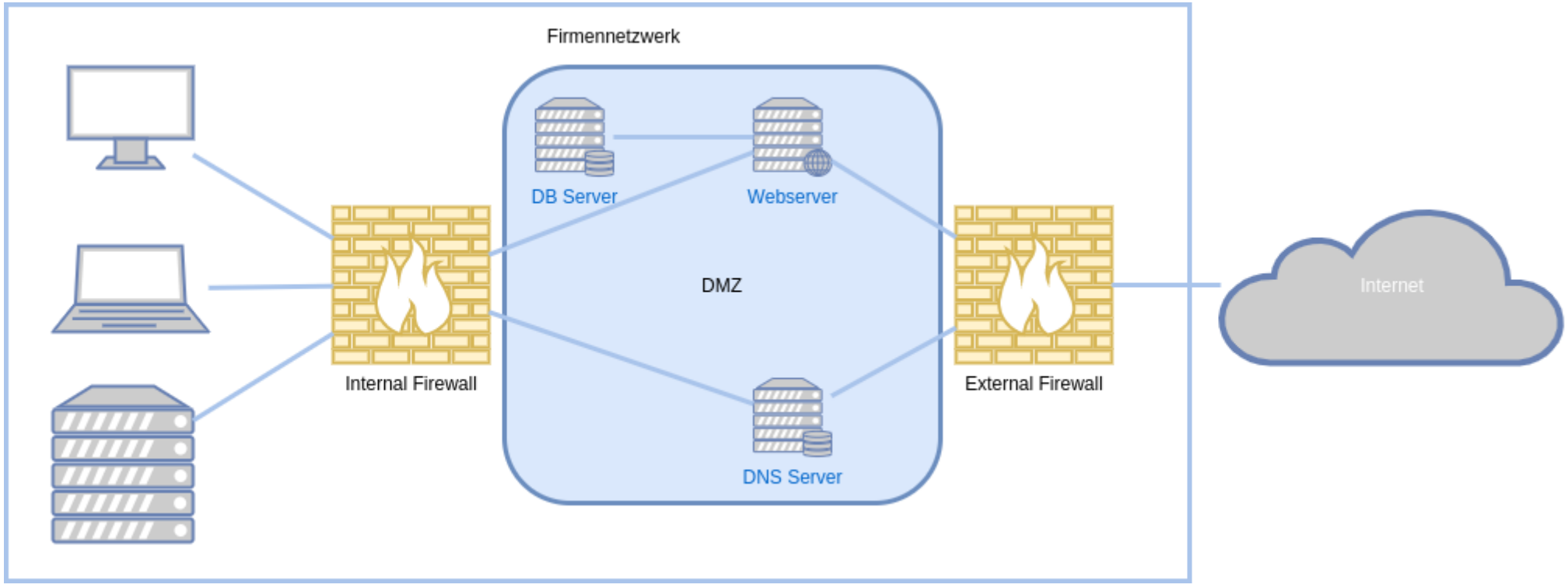
Äusserer Perimeter
Zusätzliche Hürde

https://julianraxworthy.org/wp-content/uploads/2018/01/Construction-of-a-Castle-in-the-middle-Ages-816x479.jpeg

Firewall als Perimeter Security



Firewall Konzept mit DMZ

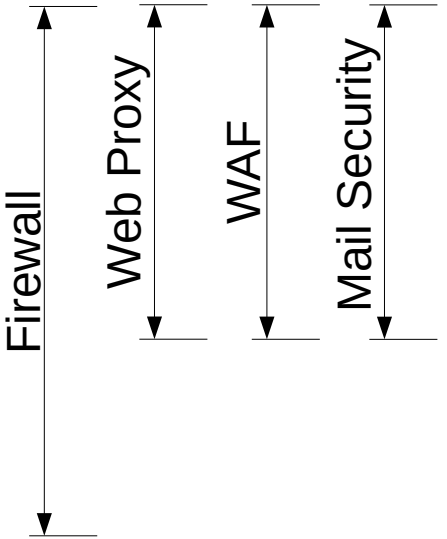


Firewall

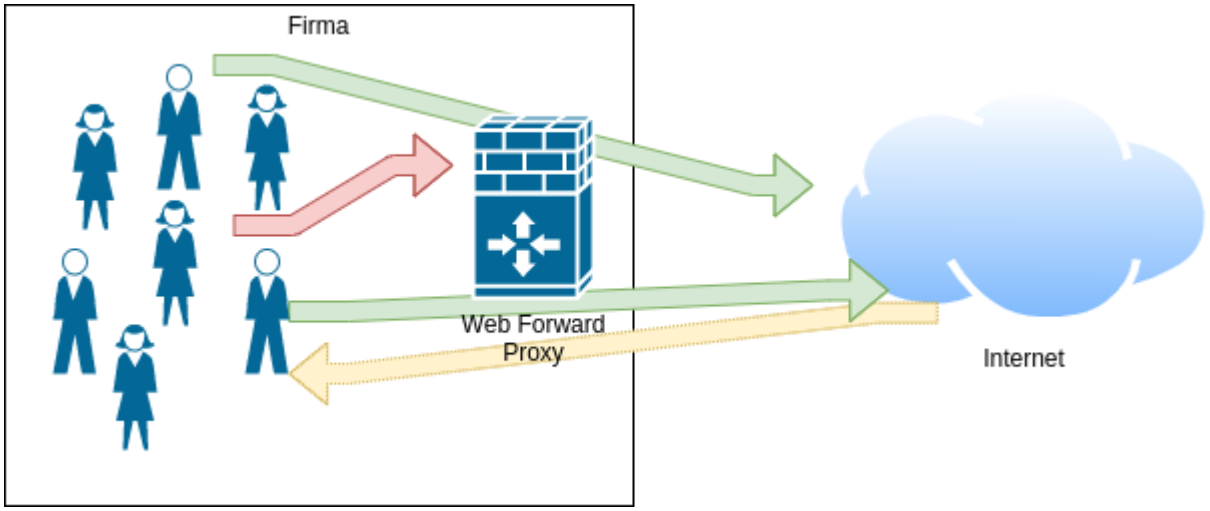
- Ermöglicht Netzwerkszugriffseinschränkung unabhängig von Rechner / Server oder Applikation
- Segmentierung in verschiedene Netze
- IP kontrollieren
  - Vermittlungsschicht, OSI Layer 3
- Ports kontrollieren
  - Transportschicht, OSI Layer 4
- NAT
- Logging von Netzwerkaktivitäten
- Netfilter, CheckPoint, Fortinet etc.

Weitere Security Massnahmen

OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, DHCP, OPC UA
Darstellung (6)		
Sitzung (5)		TLS, SOCKS
Transport (4)	Transport	TCP, UDP, SCTP
Vermittlung (3)	Internet	IP (IPv4, IPv6), ICMP (über IP)
Sicherung (2)	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI
Bitübertragung (1)		

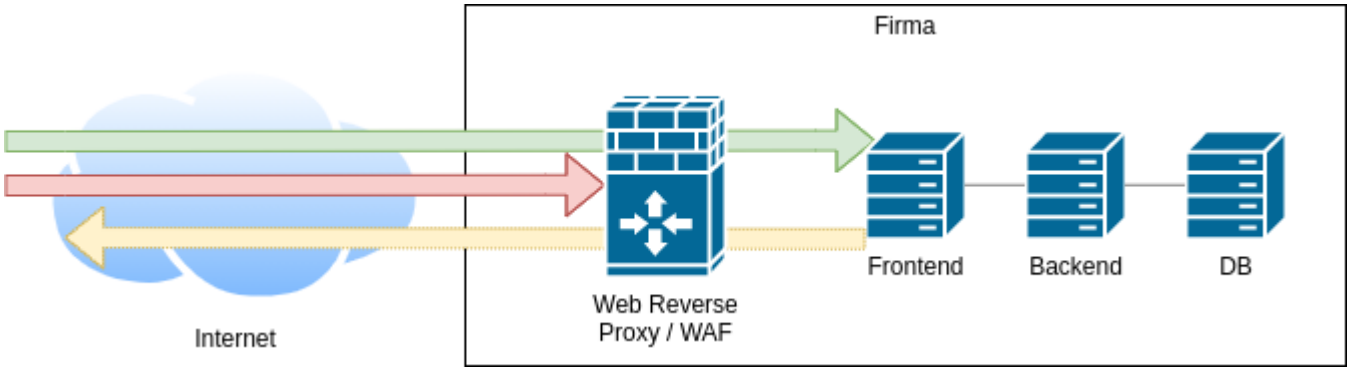


Web Proxy



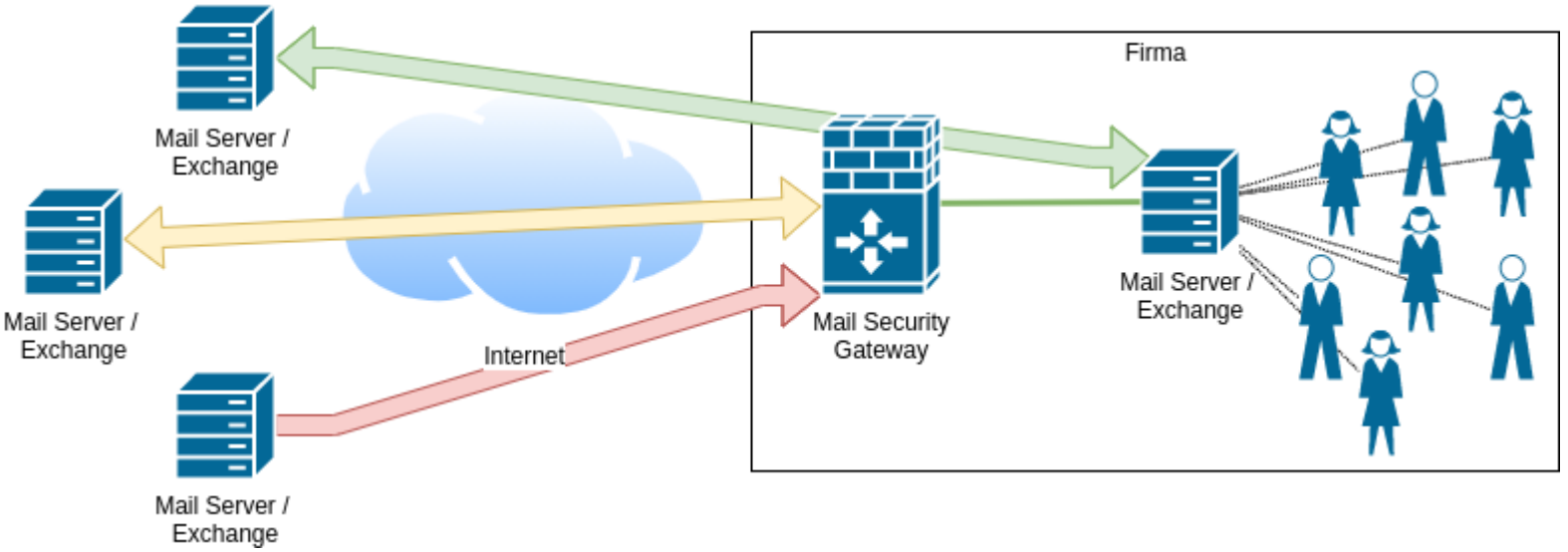
- Anti-Virus Scanning
- URL Filtering
- Connection- & Content-Rating
- Caching
- Logging
- User / Group Verwaltung
- Bandwidth Management

Web Reverse Proxy / Web Application Firewall (WAF)



- TLS-Termination
- Anti-Virus Scanning
- Logging
- Policy Enfocement bspw. OWASP Top 10
- Connection- & Content-Rating
- Caching
- Load Balancing

Mail Security Gateway



- Anti-Virus Scanning
- Deny- & Allow-Listing
- Anti-Spam
- Adress Harvesting Protection
- Logging
- Mail Routing

#02 Firewalling Grundidee

Video 2: Firewalling Grundidee

Server-Applikation und Betriebssystem

- Server-Applikationen hören auf einem Port auf eingehende Anfragen
  - Beispiel: Der Prozess Apache Webserver kann auf allen IPv4 Interfaces (0.0.0.0) für die Ports 80, 443 und 9000 hören. Eine andere Applikation kann dann nicht mehr auf derselben IP/Port Kombination hören.
- Eine Applikation/Prozess pro Port
  - Applikation fordert diesen vom Betriebssystem an
    - Well-Known Ports setzen Root-/Administratorenrechte voraus
- Mehrere Applikationen pro IP-Adresse

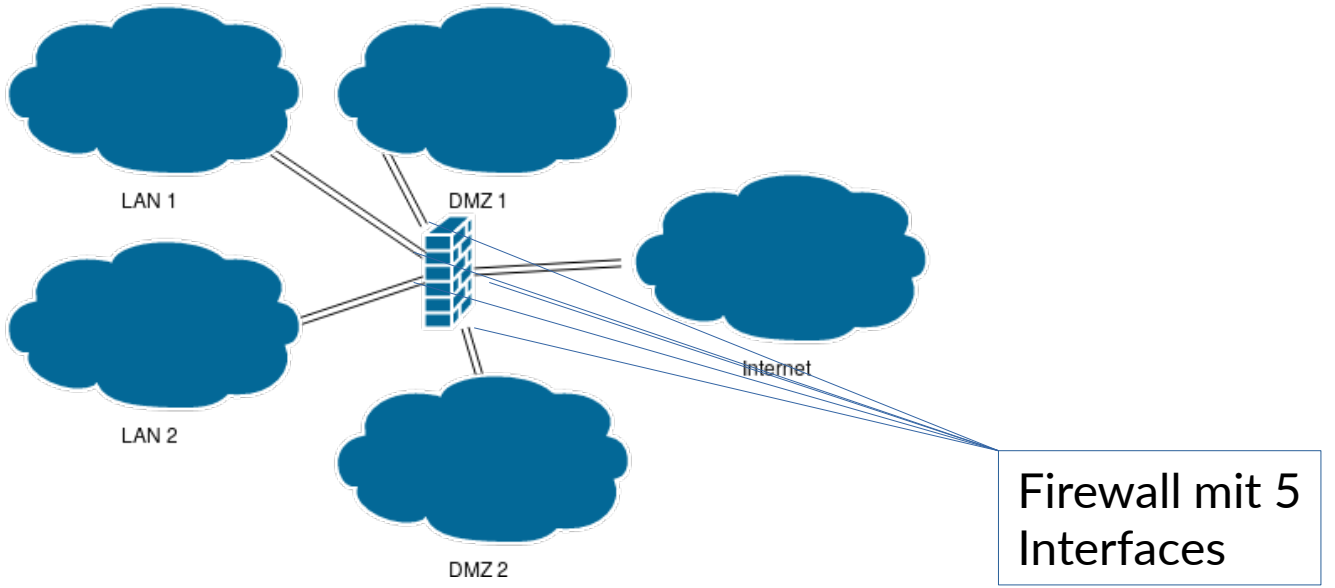
	IPv4	IPv6
Localhost	127.0.0.1	::1
Spezifisches Interface	129.168.1.210	2a02:168:4090::196
Alle Interfaces	0.0.0.0	:::

Prozesse und deren Sockets unter Linux anzeigen

```
$ sudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5433          0.0.0.0:*               LISTEN      1385/postgres
tcp        0      0 127.0.0.1:5434          0.0.0.0:*               LISTEN      1382/postgres
tcp        0      0 127.0.0.1:5435          0.0.0.0:*               LISTEN      1383/postgres
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1944/stubby
tcp        0      0 127.0.0.1:8853          0.0.0.0:*               LISTEN      1333/stunnel4
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1189/cupsd
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      1388/postgres
tcp6       0      0 ::1:53                  :::*                    LISTEN      1944/stubby
...
```

Segmentierung von Netzen

- Verschiedene Bereiche eines Netzwerkes haben unterschiedliche Sicherheitsstufen
- Nicht jeder Rechner soll jeden anderen Rechner «sehen»



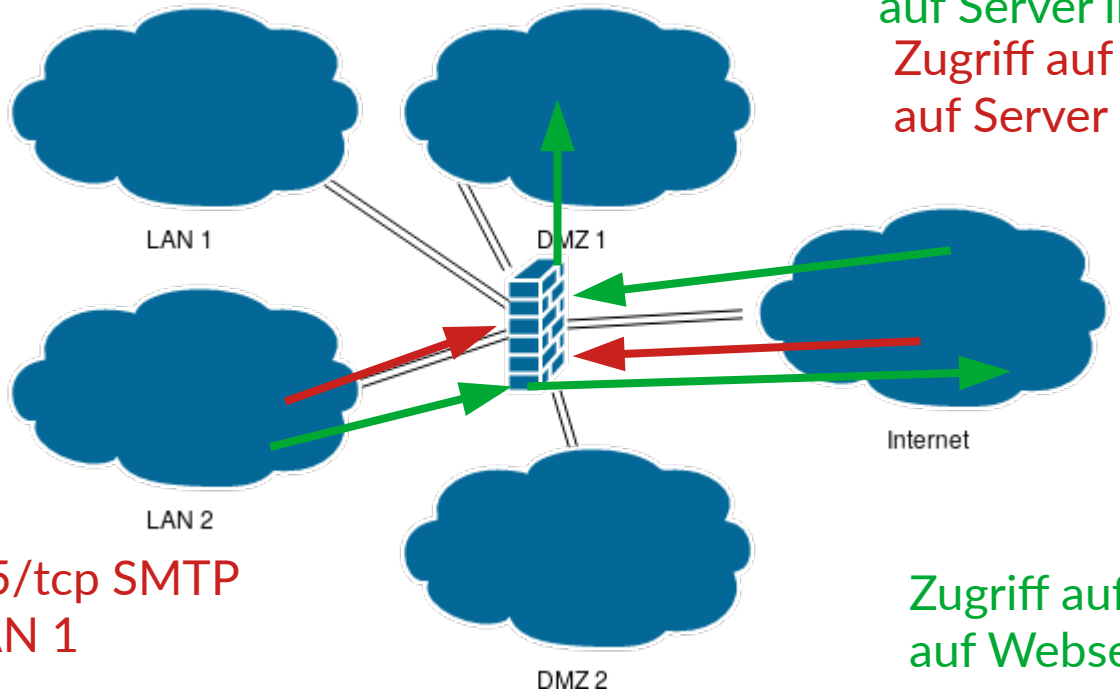
Typische Segmente

- Im Firmenumfeld wird das Netzwerk meist in verschiedene Teilbereiche aufgeteilt:
  - Internet: Nicht vertrauenswürdige Zone grösste Restriktion
  - DMZ: Demilitarisierte Zone mit strengen Zugriffskontrollen
    - «Was von extern und von intern erreichbar ist»
    - Server, Proxy, Mail-Gateway etc.
  - LAN: Interne Zone die aus dem Internet nicht erreichbar ist
    - Arbeitsplatzrechner, Drucker etc.
  - Server Netz: Interne Zone mit internen Servern die nicht (direkt) aus dem Internet erreichbar sind
    - AD, Exchange, File Share etc.

Firewalling Überblick

Zugriff auf Port 443/tcp HTTPS
auf Server in DMZ 1

Zugriff auf Port 445/tcp SMB
auf Server in DMZ 1



Zugriff auf Port 25/tcp SMTP
Auf Rechner in LAN 1

Zugriff auf Port 80/tpc HTTP
auf Webserver

Allgemeine Funktionsweise einer Firewall

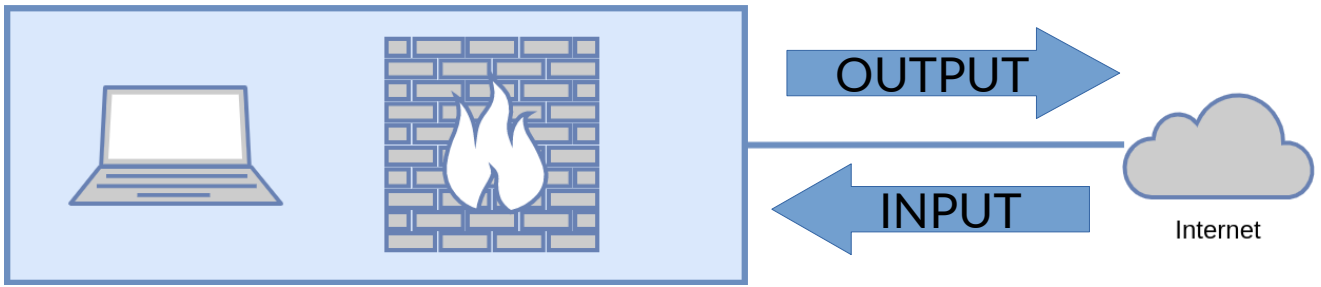
- Anwenden einer Regel eines Regelsets auf eintreffende Netzwerkpakete

Interface	INPUT / OUTPUT	Source	Destination	Port

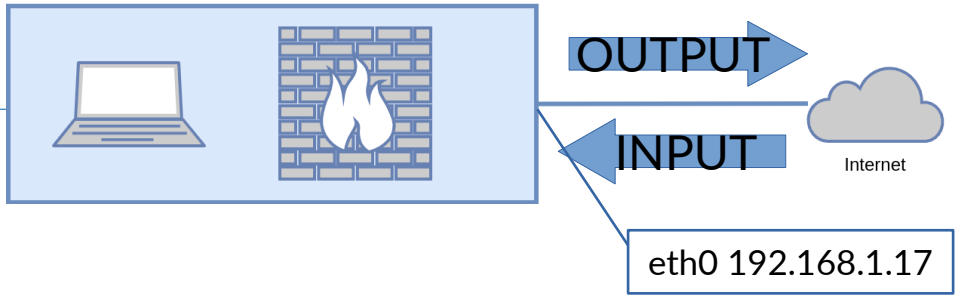
  - Logisch-UND verknüpft
- Inbound: Aus Sicht des Gerätes kommt der Pfeil von extern auf das Gerät
- Outbound: Aus Sicht des Gerätes geht der Pfeil Richtung extern
- Ausführen einer Aktion
  - Allow: Verbindung zulassen
  - Deny/Block: Verbindung blockieren und keine Antwort zurücksenden
  - Reject: Verbindung blockieren und Sender informieren
    - Bspw. TCP Reset Paket oder ICMP Fehlermeldung
- Firewall allgemein gilt: First Match Action und Deny-All-Rule als Clean-up Rule

Fokus: Firewall auf lokalem System

- Wir konzentrieren uns auf ein Setup, in dem die Firewall direkt auf dem Server / Notebook konfiguriert wird:



Beispiel



Interface	INPUT / OUTPUT	Source	Destination	Port	Action
eth0	INPUT	192.168.1.0/24	192.168.1.17	80/tcp	Allow
eth0	INPUT	192.168.1.0/24	192.168.1.17	443/tcp	Allow
eth0	INPUT	Any	Any	Any	Deny
eth0	OUTPUT	192.168.1.17	Any	443/tcp	Allow
eth0	OUTPUT	192.168.1.17	Any	53/udp	Allow
eth0	OUTPUT	Any	Any	Any	Reject

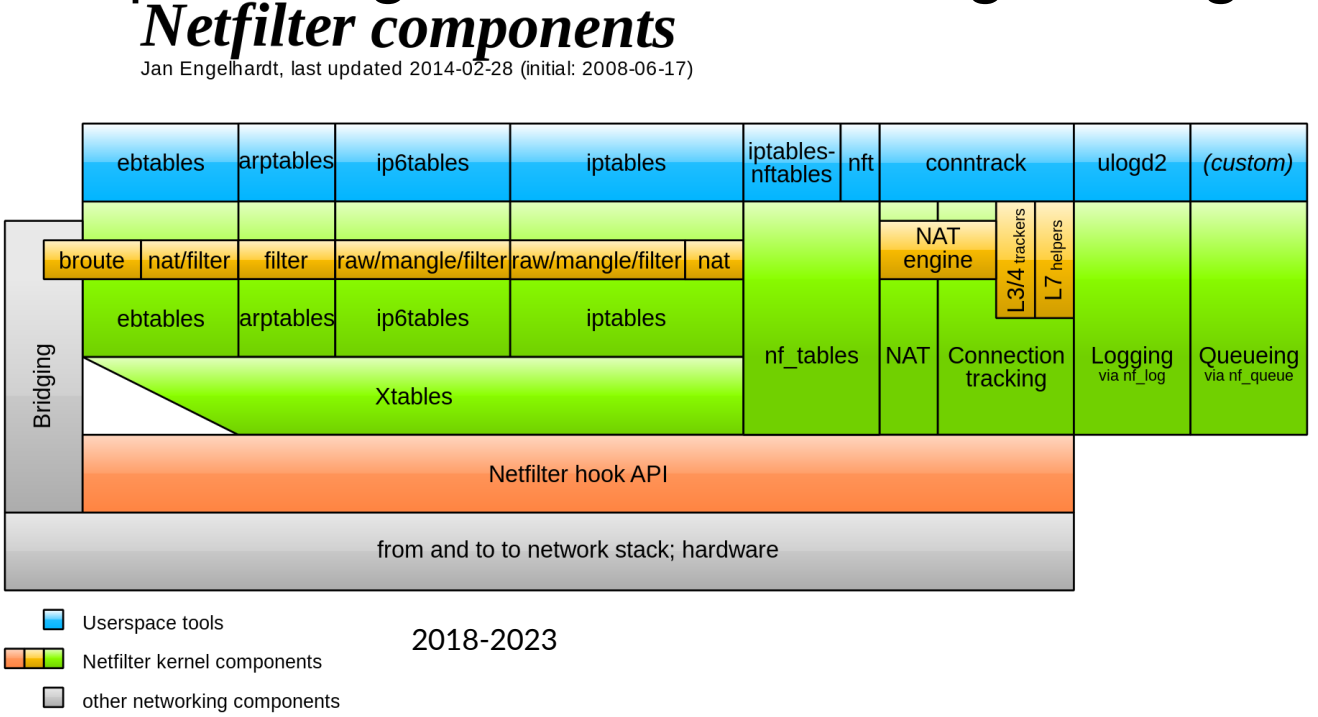
#03 Firewalling Netfilter und Iptables

Video 3: Firewalling Netfilter und Iptables

Juventus Technikerschule HF

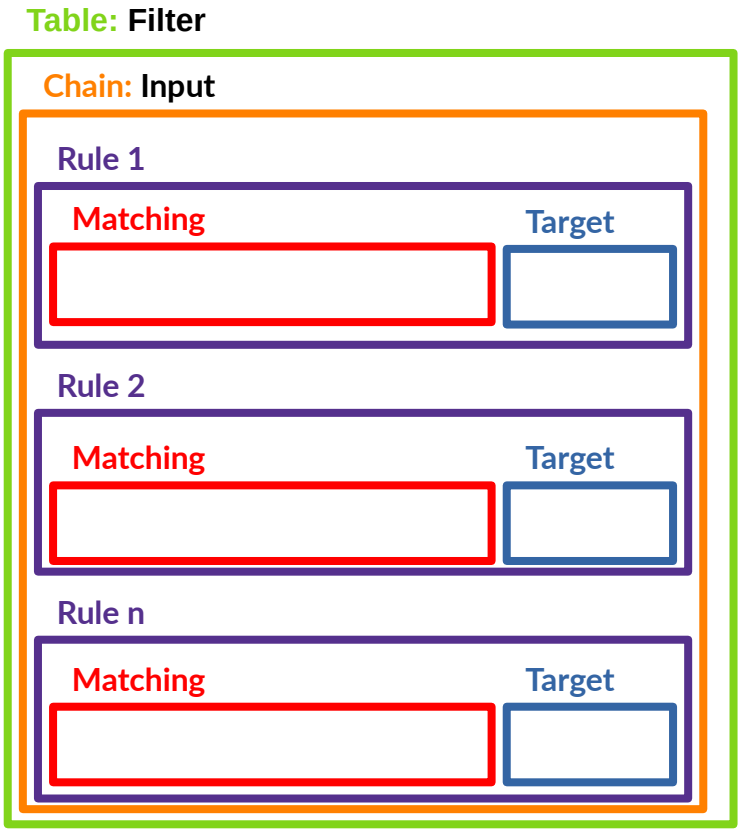
Netfilter und Iptables

- Netfilter ist die Linux Firewall
- Iptables das Userspace Programm zur Verwaltung des Regelsets



Iptables Komponenten

- Tables
  - Verschiedene Tabellen beherbergen verschiedene Rule-Typen
  - Filter, NAT, Mangle, RAW & Security
- Chains
  - Punkt im Traffic-Flow an welchem eine Rule aktiv sein kan
  - Prerouting, Input, Forward, Output & Postrouting
- Rules
  - Bestimmt welche Action für ein Paket vorgesehen ist
  - Matching-Component
    - Protokoll, IP-Adresse, Port, Interface, Headers etc.
  - Target-Component
    - Terminating Targest: Accept, Drop, Reject und einige andere



Iptables Chain

- Jedes Netzwerkpaket (Inbound und Outbound) durchläuft mindestens eine Chain
  - INPUT – eingehende Pakete die für lokalen Socket vorgesehen sind
  - FORWARD – eingehende Pakete die für ein anderes System vorgesehen sind
  - OUTPUT – lokal generierte Pakete welche dieses System verlassen
- Jede Chain hat eine Policy
  - ACCEPT – Paket wird akzeptiert, Standardeinstellung
    - `# iptables --policy INPUT ACCEPT`
  - DROP – Paket wird verworfen
    - `# iptables -P INPUT DROP`

Regelset für einfachen, lokalen Webserver

- Port 80 eingehend erlauben
  - `# iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT`
- FORWARD und INPUT Chain alle Pakete verwerfen
  - `# iptables -P FORWARD DROP`
  - `# iptables -P INPUT DROP`

Regeln

- Gehören zu einer Chain
  - Werden an diese angehängt `-I` am Anfang und mit `-A` am Ende des Regelsets
  - Und mit `-D` gelöscht (selber Befehl)
- Steuern Protokoll, Port und IP-Adressen
- Haben eine Aktion `-j` (für Jump)
  - ACCEPT
  - DROP
  - REJECT
- Ganzes Regelset kann mit `iptables -L` angezeigt werden
- Alle Regeln löschen mit `iptables -F` (ausser Chain Policy)

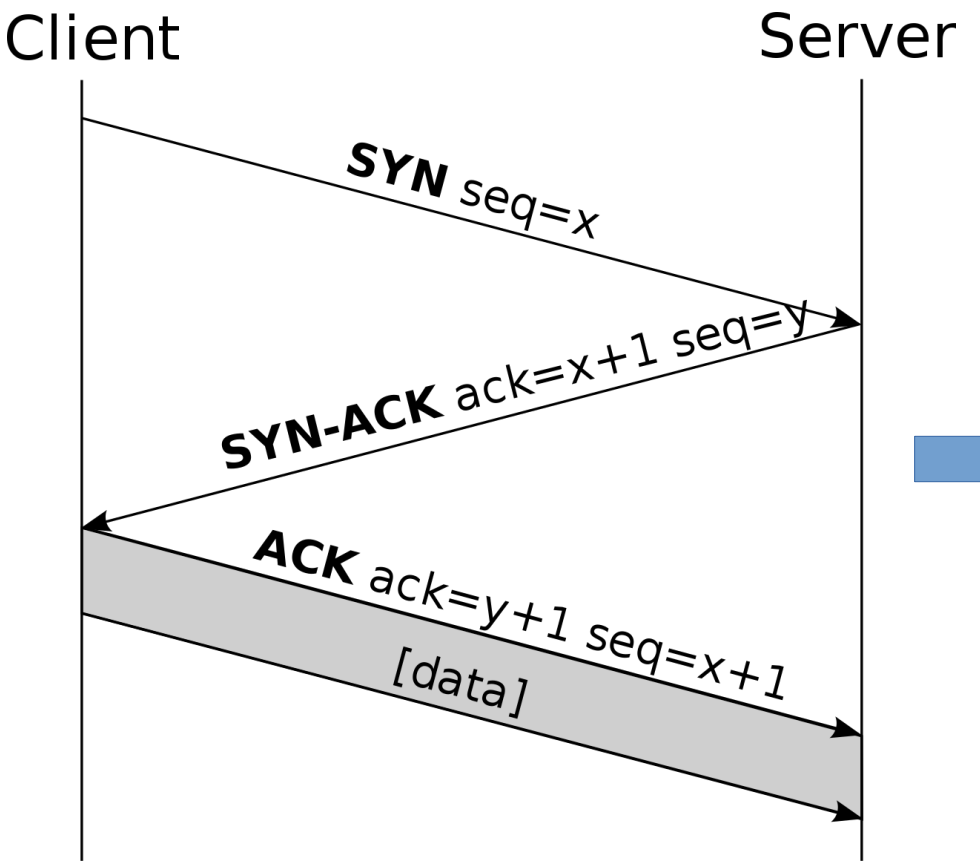
#04 Port Scanning

Video 4: Port Scanning

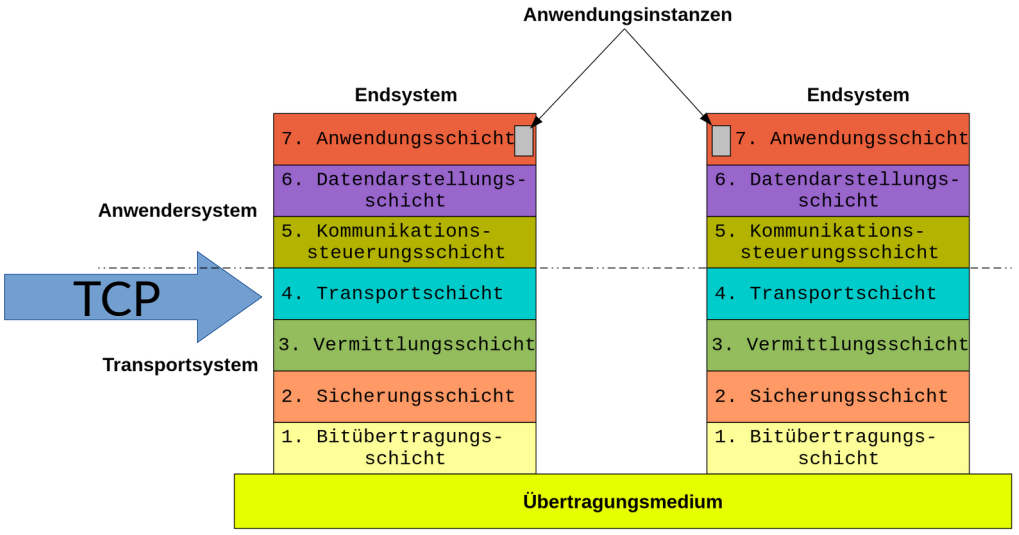
Netzwerk Port

- Ein Port ist der Bestandteil einer Netzwerk-Adresse auf dem Transport Layer (TCP und UDP)
  - Well-Known Ports: 0 bis 1023
  - Registered Ports: 1024 bis 49151
  - High Ports: 49152 bis 65535 (2^{16}-1)
- Server Applikationen sind an einen oder mehrere Ports gebunden (`bind`) und warten auf Anfragen (`listen`)
- Client Applikation wird vom OS ein Source Port zugeordnet

TCP Handshake



OSI-Modell



Quelle: Wikipedia

Der Port-Scanner Nmap

- Ziel ist Finden von offenen Ports
  - Jeder offene Port ist eine mögliche Bedrohung
- Einsatzzweck
  - Security Audits von Netzwerkgeräten
  - Finden von offenen Ports
  - Erstellen eines Netzwerkinventars
  - Finden von Schwachstellen in Software oder Systemen
- Vergewissern Sie sich stets, dass Sie den richtigen Host scannen und scannen Sie nie Systeme die Ihnen nicht gehören

Port Zustände

- Nmap ist in der Lage, Remote Ports in unterschiedliche Zustände einzuteilen. Die wichtigsten sind:
  - **open** – Applikation akzeptiert TCP Verbindungsaufbau
  - **closed** – Port ist erreichbar aber keine Applikation hört auf dem Port (TCP RST) → Host ist online!
  - **filtered** – Firewall blockiert Zugriff auf Port, kein TCP RST

Nmap Usage

- `$ nmap scanme.nmap.org`
  - Scan der 1000 häufigsten Ports
- `$ nmap www.addere.ch -p 1,2,4-8,16`
  - Scannt Ports 1, 2, 4, 5, 6, 7, 8 und 16

Scan-Techniken

- Nmap unterstützt eine Vielzahl unterschiedlicher Scan-Methoden mit spezifischen Einsatzzwecken
- TCP SYN scan (-sS) ist Standard-Scan
  - Nur SYN-Pakete, kein TCP Handshake, sehr schnell, erweiterte Rechte notwendig
- TCP connect scan (-sT)
  - Verwendet OS Network API und baut TCP Verbindung auf, weniger schnell und genau
- Weitere Scan-Methoden: UDP scan (-sU), TCP NULL, FIN und Xmas scan (-sN, -sF, -sX), TCP ACK scan (-sA), etc.

Service & Version Detection

- Service-DB mit ca. 2200 Port und Service Kombinationen
  - 25/tpc ist SMTP
- Schnell
- Ungenau respektive kann falsch sein
  - Webserver kann auch auf Port 25/tcp hören
- Keine Informationen über Service/Daemon
  - Jedoch nötig für Vulnerability Suche

Service & Version Detection

- Version detection mit `-sV` starten
- Verschiedene Anfragen werden auf Port geschickt und Antworten/Reaktionen ausgewertet
  - Service Protocol
    - FTP, SSH, Telnet, HTTP etc.
  - Application Name
    - ISC BIND, Apache httpd, Solaris telnetd etc.
  - Version Number
  - Hostname
  - Device Type
    - Printer, Router etc.
  - OS Family
    - Windows, Linux

Übungen & Labor

Übungen: HE1

Labor: github.com/ryru/HackingExposed

Kurze Einführung in Wireshark

- Packet-Filter
  - `host <ip> and port <port>`
- Drei UI-Bereiche
- Display-Filter
  - `ip.addr == <ip> and tcp.port == 80`
  - Häufige Protokolle filtern
    - `http, ssl, icmp, smtp`
- TCP Flow
- Statistics
  - Protocol Hierarchy
  - Conversations
  - Endpoints

Videoempfehlungen fürs Selbststudium

- Hirne Hacken (43 Min)

CC BY-SA 4.0