Projektvorschlag

BKS>IT-SuM GbR, Thorsten Stöcker

Firewall auf Wechsel-Datenträger, die das lokale OS von der Festplatte in eine Virtualisierungsumgebung lädt.

Übersicht

- Kurzer Überblick über TCP/IP v4
- Grundlagen Firewall, netfilter/iptables
- Grundlagen Virtualisierung
- Alltägliche Risiken, politische Lage
- Projekt-Idee und -Begründung
- Lösungsansätze
- Diskussion

TCP/IP-Historie

- 1969
  Verteidigungsministerim der USA beschließt Computer zu vernetzen

  Kommunikation muß auch bei Teilausfall gewährleistet sein

  ARPANET mit TCP/IP-Vorläufer NCP (Network Control Protocol)

  Drahtgebunden mit 2400Baud/Sekunde

  NCP ist ein Host-to-Host-Protokoll

  Bell-Labs und AT&T programmieren die Software, das spätere Unix

TCP/IP-Historie

- 1976
  Auf Grundlage von ARPANET und NCP wird TCP/IP entwickelt

  TCP/IP = Transmission Control Protocol / Internet Protocol zu deutsch, Übertragungskontrollprotokoll / Zwischennetzprotokoll

  Der Name ist Programm:

  Das Protokoll ist darauf ausgelegt sicher zu stellen, daß die Daten zwischen Rechnern und/oder Teilnetzen ausgetauscht werden.

  Es gibt keine Mechanismen zur Absender- und Empfängerautentifizierung oder Verschlüsselung

TCP/IP-Historie

- 1983
  ARPANET besteht aus mehren Teilnetzen (militärisch und wissenschaftlich)

  Das Netzwerk ist deutlich gewachsen

  NCP wird durch TCP/IP als Protokoll abgelöst

  Nach der Einführung von TCP/IP wird das Netz vermehrt als

  „Internet“

  bezeichnet

TCP/IP-Historie

- 1987
  IBM unterstützt als erster Hersteller TCP/IP neben den eigenen Protokollen
- 1991
  Novell führt als, damals führender Hersteller von PC-Netzwerkbetriebsystemen, TCP/IP ein, obwohl das eigene IPX/SPX einen vergleichenbaren Standard bietet
- 1992
  10.000 Unternehmen benutzen das Internet

Historische Daten

- 1976
  Definition von TCP/IP
- 1977
  Erster Personal-Computer (PC), der Apple-II
- 1983
  Einführung von TCP/IP als Netzwerkprotokoll
- 1984
  Erste GUI für einen PC wiederum von Apple

TCP/IP v 4

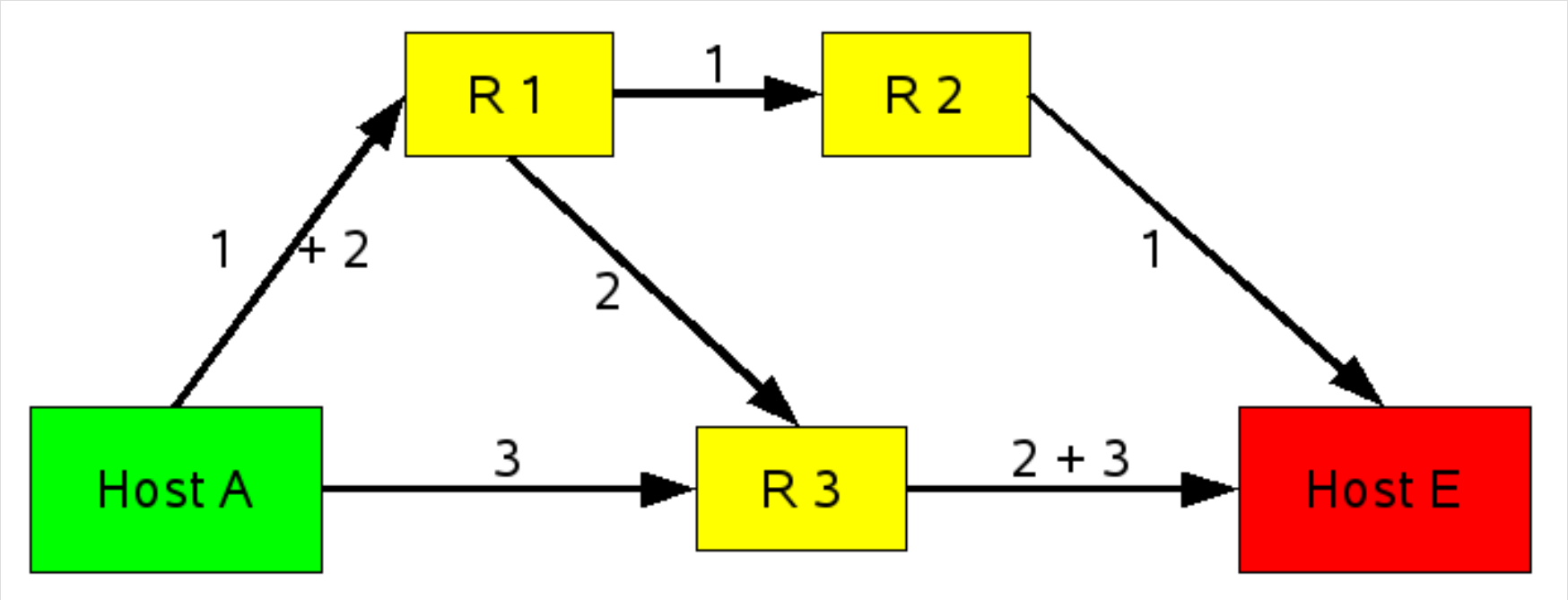
- Adressraum
  rund 4 Milliarden Adressen 4 294 967 296, damals ca. 1 Adresse je Erdbewohner
- Protokoll
  Host-to-Host-Protokoll, ein Host in TCP/IP ist ein Gerät, das mindestens eine TCP/IP-Adresse hat

  Packet-Basiert, größere Datenmengen werden in Pakete zerlegt, nummeriert und auf die Reise geschickt

  Eigentlich kein Protokoll sondern eine Sammlung von Protokollen, eine so genannte Protokoll-Suite

TCP/IP v 4

- Weg der Datenpakete



Weg der Datenpakete 1, 2 und 3 von Host Absender(A) zu Host Empfänger(E)

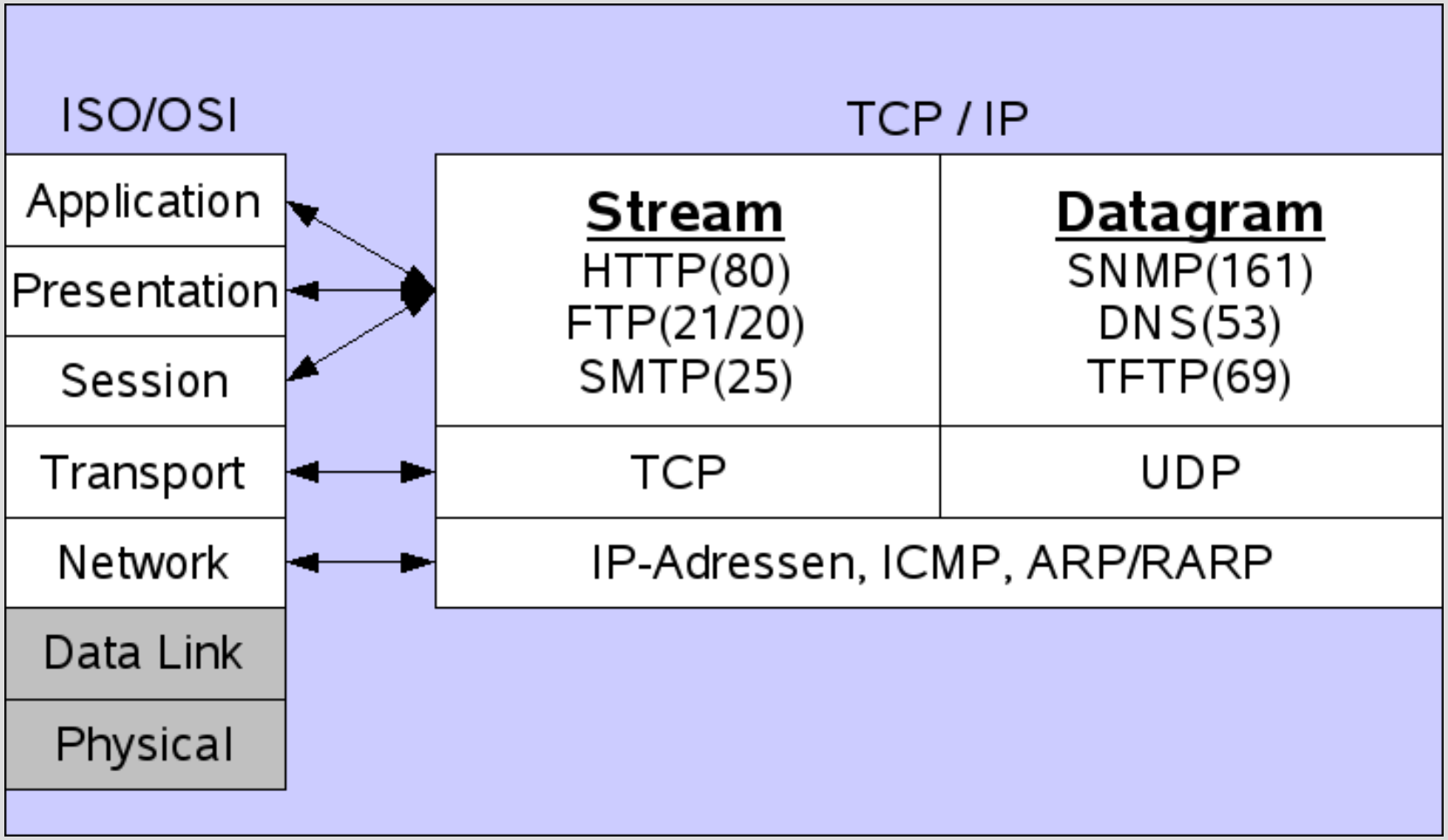
TCP/IP v 4

- IP-Header-Aufbau

Version	Internet HL	Type of Service	Packet Length
Identification		Flags	Fragment Offset
Time to Life		Protocoll	Header Checksum
Source-IP (Absender)			
Destination-IP (Empfänger)			
Options / Padding			
DATA			

TCP/IP v 4

- TCP/IP auf dem OSI-Modell



TCP/IP v 4

- Adressräume die im Internet nicht geroutet werden

10.0.0.0 /8 (255.0.0.0) Privates Netzwerk

127.0.0.0 /8 (255.0.0.0) Lokales Netzwerk

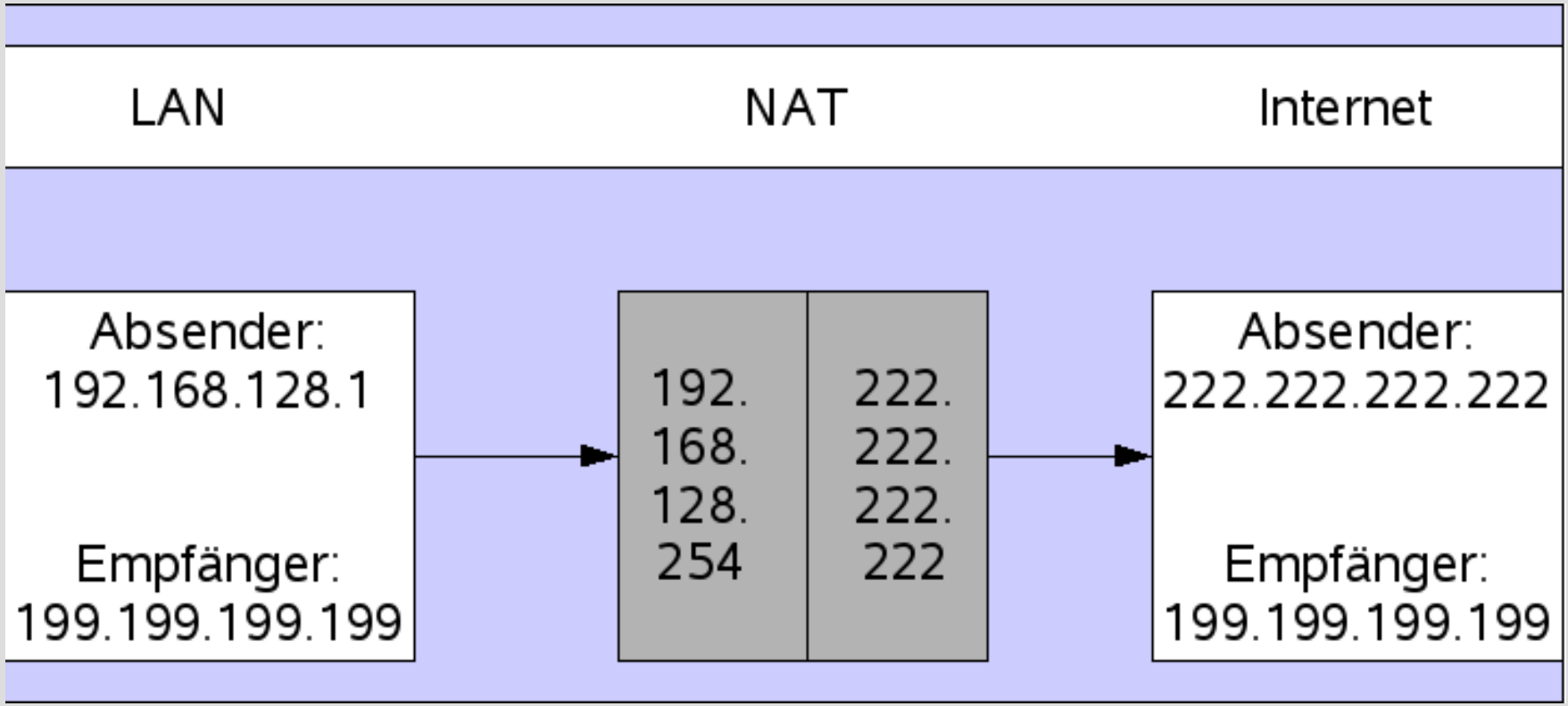
169.254.0.0 /16 (255.255.0.0) ZeroConf

172.16.0.0 /12 (255.240.0.0) Privates Netzwerk

192.168.0.0 /16 (255.255.0.0) Privates Netzwerk

TCP/IP v 4

- Funktion einer NAT



netfilter / iptables

- Warum Linux sich als Firewall eignet

  TCP/IP ist natives Protokoll aller Unixe (auch Linux)

  Native Mehrbenutzerfähigkeit

  kostengünstig

  besonders schnell im Routing

  hohe Flexibilität der Implementation (netfilter/iptables)

  Aufteilung in Kernelspace (netfilter)

  und Userspace (iptables)

netfilter / iptables

- iptables

  beinhaltet 3 Tabellen

  nat Network Address Translation

  filter Packet-Filter (Default, wenn keine Angabe)

  mangle Packet-Manipulation

netfilter / iptables

- iptables

beinhaltet 5 vordefinierte Ketten

Prerouting	Vor Routing-Entscheidung (**nat** + **mangle**)
Input	Für lokalen Prozess (**filter** + **mangle**)
Output	Vom lokalen Prozess
Forward	Packet Routing (**filter** + **mangle**)
Postrouting	Alle Packete (**nat** + **mangle**)

netfilter / iptables

- iptables

  beinhaltet 4 Ziele

ACCEPT	Packet wird angenommen
DROP	Packet wird ohne Meldung verworfen
QUEUE	Packet wird an eine Anwendung weitergeleitet
RETURN	Rücksprung aus einer Kette

netfilter / iptables

- iptables

enthält eine Menge erweiterter Ziele, die wichtigsten

LOG Verbindung im SYSLOG protokollieren

REJECT Packet mit Meldung verwerfen (ICMP + RST)

enthält vordefinierte Vergleichsmuster, wie z.B.

tcp TCP-Packets filtern

mac-source MAC-Adresse filtern

sport Nach Quellport filtern

netfilter / iptables

- iptables

Zum Bereich der Tabelle **nat** gehören weitere Ketten

DNAT	Destination-NAT
SNAT	Source-NAT
MASQURADE	Eine Sonderform der SNAT
REDIRECT	Zur Weiterleitung, z.B. an einen Proxy

netfilter „merkt“ sich die Pfade, somit kann ein connection-tracking einfach etabliert werden

netfilter / iptables

- iptables-Arbeitsweise

Sequenziell von oben nach unten bis zum ersten Treffer

Trifft keine Regel auf das Paket zu, also 0 Treffer so wird die Standard-Policy angewendet

- Funktion der einzelnen Teilbereiche
- - die **Ketten** bestimmen wo ein Paket geprüft wird
- - die **Muster** bestimmen welches Paket geprüft wird
- - die **Ziele** bestimmen was mit dem Paket passieren soll.

netfilter / iptables

- iptables – eine sichere Firewall

Ein sichere Firewall beinhaltet NUR 3 Statements

Sichere Firewall

/usr/sbin/iptabels -P INPUT DROP
/usr/sbin/iptabels -P FORWARD DROP
/usr/sbin/iptabels -P OUTPUT DROP

(**-P** definiert die Standard-Policy einer Kette)

netfilter / iptables

- iptables, Regel für http-Zugang per Script

$FW -A in2ext -p tcp --sport $HIGHP --dport 80 -j ACCEPT

$FW	Variable für /usr/sbin/iptables
-A in2ext	selbstdefinierte Kette „in2ext“
-p tcp	Protokoll tcp
--sport	Quellport
$HIGHP	Variable für Port-Range 1024-65535
--dport 80	Zielport 80 (http)
-j ACCEPT	Ziel Paket akzeptieren

netfilter / iptables

- iptables, vorherige Regel ausgeschrieben

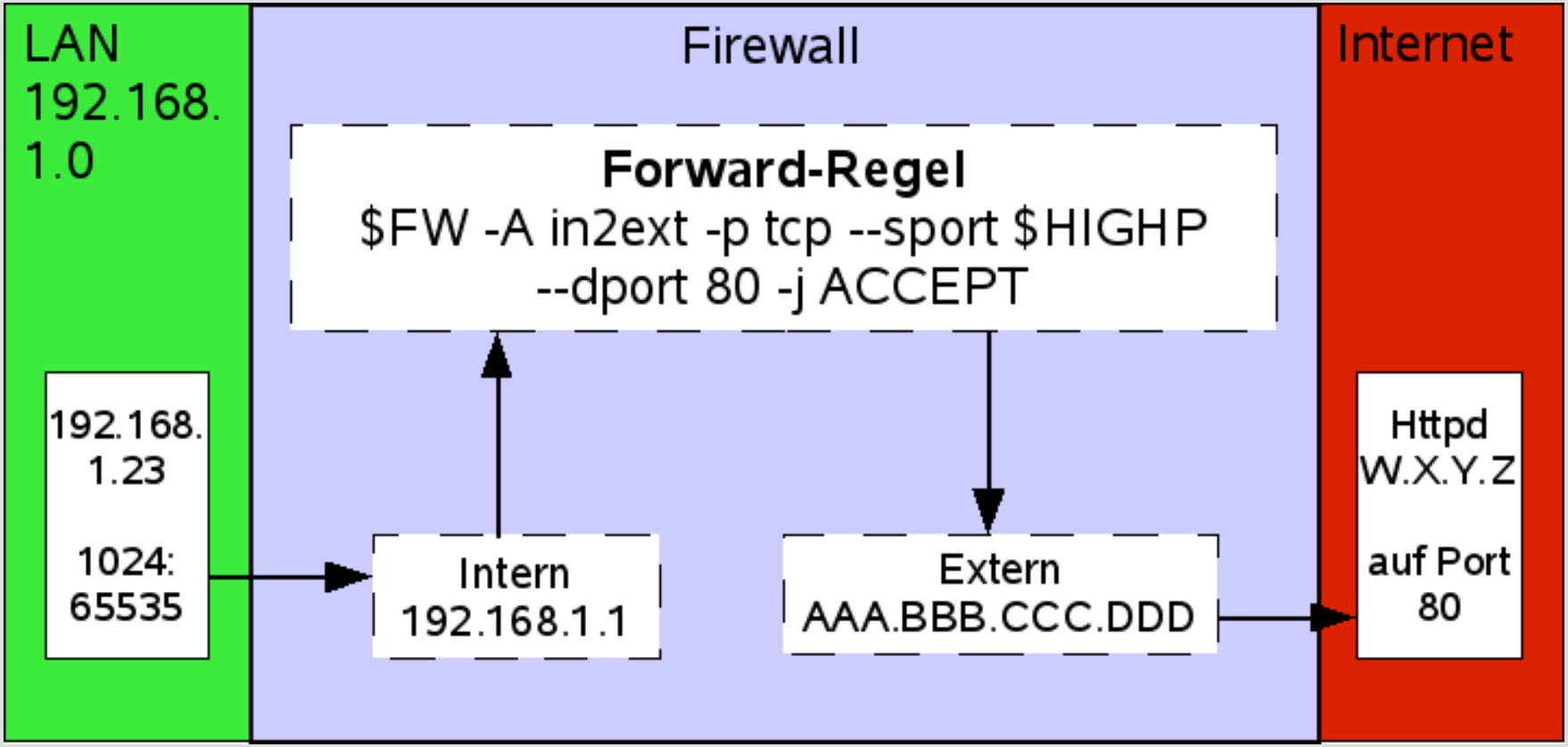
/usr/sbin/iptables -t filter -A FORWARD -i 192.168.1.1 -o AAA.BBB.CCC.DDD -s 192.168.1.0/255.255.255.0 -p tcp --sport 1024:65535 --dport 80 -j ACCEPT

Diese sehr übersichtliche Schreibweise hat die selbe Bedeutung wie die vorherige, die da lautet:

- Nimm die Tabelle filter (-t). Benutze die Kette Forward.
- Wenn das Paket auf Deinem Interface 192.168.1.1 (-i) ankommt und auf Deinem Interface AAA.BBB.CCC.DDD (-o) ausgegeben werden soll und diese Paket aus dem lokalen Netz 192.168.1.0/24 (-s) stammt und ein TCP-Paket ist, dann prüfe, ob das Paket von einem hohen Port an Port 80 gesendet wurde.
- Trifft dies alles zu, akzeptiere das Paket und leite es weiter.

netfilter / iptables

- Weg eines HTTP-Pakets durch die FW



Virtualisierung

- Arten der Virtualisierung

Hardware

Großrechner (Partionierung), seit 2005 (intel) bzw. 2006 (AMD) auch im Bereich x86

Software

Betriebsystem-, System-, Anwendungs-, Speicher- und Netzwerkvirtualisierung, Domaining

Virtualisierung

- Software-Virtualisierung

  Betriebsystem-Virtualisierung (BSD-Jails, VServer)

  System-Virtualisierung (VMware, Xen)

  Anwendungs-Virtualisierung (SoftGrid, Java VM)

  Speicher-Virtualisierung (SAN, LVM)

  Netzwerk-Virtualisierung (VLAN, VPN)

  Domains (Multidomain-Server (HTTP))

Virtualisierung

- System-Virtualisierung

Typische Vertreter der System-Virtualisierung via Software:

VMware

VirtualPC

Xen

QEMU

VirtualBox

Virtualisierung

- System-Virtualisierung

  Baut Zwischenschicht (Hypervisor) ein, der entweder direkt auf der Hardware läuft oder auf einem Host-Betriebsystem aufsetzt.

  System-Virtualisierung erlaubt es, mehre Betriebsysteme zu laden und stellt jedem Gast-Betriebsystem eine eigene Hardware-Umgebung zu Verfügung.

  3 Formen der System-Virtualisierung, die sich durch die Qualität der Hardware-Emulation unterscheiden.

Virtualisierung

- System-Virtualisierung - Formen

Hardware-Emulation

Emuliert vollständige Hardware, es laufen unmodifizierte Betriebsysteme, auch für „fremde“ Hardware

Hardware-Virtualisierung

Emuliert Teile der Hardware, es laufen unmodifizierte Betriebsysteme, die für diese Hardware entwickelt wurden

Paravirtualisierung

Es laufen modifizierte Betriebsystem (Anpassung auf den Hypervisor), die für diese Hardware entwickelt wurden

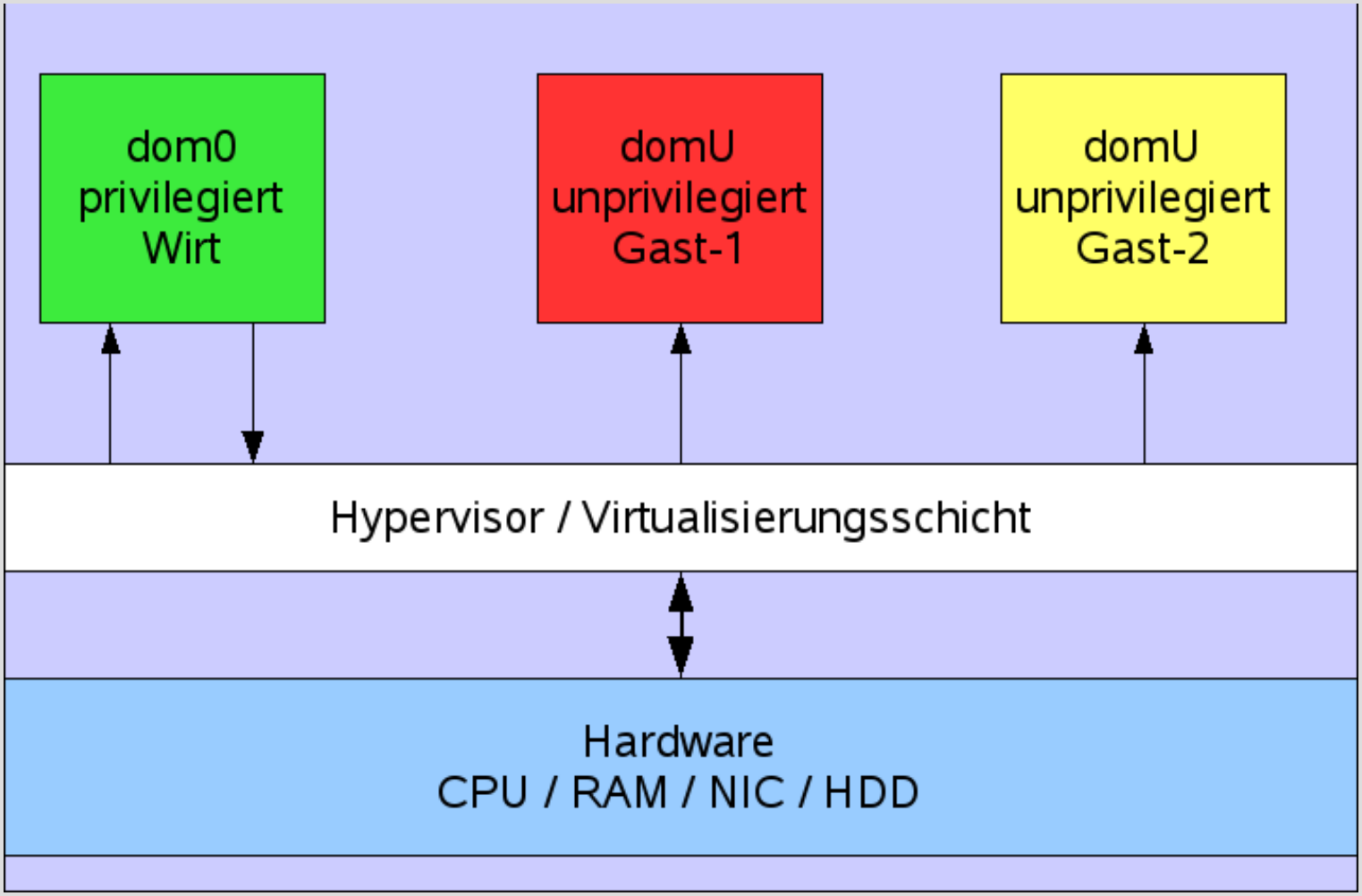
Virtualisierung

- Xen

1. Xen stammt von einer Universität (Cambridge)
2. Xen steht unter der GPL
3. Xen ist für wichtig Distributionen verfügbar / integriert
4. Xen ist ein ausgereiftes Produkt
5. Xen ist servertauglich (SELS, RHEL, Netware)
6. Xen unterstützt die HW-Techniken von intel + AMD
7. Xen hat eine breite Unterstützung in der Industrie
8. Xen soll zum Industrie-Standard werden

Virtualisierung

- Xen Aufbau



Bedrohung

- Bedrohungen

Viren, Würmer, Trojaner, Backdoors, rootkits

Keine oder schlechte Absicherung in „fremden“, „vertrauenswürdigen“ Netzen

Bundestrojaner durch LKA, BKA, Zoll, BND, MAD

Backdoors in proprietärer Software (BIOS)

Hacking durch „Freunde“ (social Engineering)

Keylogger