

<b>WAS BEDEUTET SYSTEMSICHERHEIT? .....</b>	<b>4</b>
<b>EINIGE BEGRIFFE.....</b>	<b>6</b>
SQL-INJECTION .....	6
FOOTPRINTING .....	6
FIREWALKING .....	6
PENETRATION TEST .....	7
<i>Phasen eines Penetration-Tests.....</i>	7
Reconnaissance (Auskundschaftung).....	7
Enumeration .....	8
Exploitation .....	8
Documentation .....	9
<i>Ergebnisse eines Penetration Tests.....</i>	9
<i>Häufige Ursachen für Sicherheitslücken.....</i>	9
SNIFFING.....	10
SESSION- HIGHJACKING .....	11
HACKERPARAGRAPH.....	14
VORRATSDATENSPEICHERUNG.....	16
<b>SYSTEMSICHERHEIT BEI NETZKOMPONENTEN .....</b>	<b>17</b>
ARBEITSWEISE VON NETZKOMPONENTEN.....	17
<i>Arbeitsweise von Hubs .....</i>	17
<i>Arbeitsweise von Switches.....</i>	19
<i>Arbeitsweise von Routern.....</i>	23
<i>Aufgabe 1.....</i>	24
HUBS UND NETZWERKSICHERHEIT .....	25
<i>Aufgabe 2: .....</i>	25
<i>ARP- Spoofing in einer HUB- Umgebung.....</i>	26
Ändern der eigenen MAC- Adresse.....	26
Spoof- Szenario .....	27
SWITCHES UND NETZWERKSICHERHEIT .....	28
<i>ARP- Spoofing Variante 1 .....</i>	28
<i>ARP- Spoofing Variante 2 .....</i>	29
<i>ArpWatch.....</i>	31
ArpWatch in einem HUB-Netz.....	31
ArpWatch in einem Switch-Netz.....	32
<i>Gratuitous ARP .....</i>	33
Example Traffic.....	33
Starten von Vista .....	35
Beschreibung Gratuitous ARP in www.wireshark.org.....	35
<i>Proxy Arp .....</i>	36
ROUTER UND NETZWERKSICHERHEIT .....	37
<i>RIP Version 1 .....</i>	37
<i>Aufgabe RIP .....</i>	38





## Was bedeutet Systemsicherheit?

Die Systemsicherheit gibt darüber Aufschluss wie leicht ein System missbraucht werden kann.

Die Systemsicherheit von IT-Systemen hängt ab von der Sicherheit der:

Kommunikationssysteme

Protokolle, überlisten von Netzkomponenten  
NFS

Betriebssysteme

Schwachstellen, fehlerhafter Programmcode

Server-Dienste

Konfigurationsfehler, Programmierfehler

Anwendungen

Konfigurationsfehler, Programmierfehler

und dem Verhalten der Nutzer

Surfverhalten, schwache Passwörter, Browsereinstellungen  
Öffnen von E-Mail-Anhängen

Welche Personengruppen haben Einfluss auf die Systemsicherheit:

Die Sicherheit wird beeinflusst durch Fehler der einzelnen Komponenten, auf die der **Administrator** in der Regel, außer dem Einspielen von Sicherheits-Patches, keinen Einfluss besitzt, und durch die Konfiguration der Komponenten, für die er selber verantwortlich ist.

**Anwendungsprogrammierer** sind für die Sicherheit ihrer Programme verantwortlich. Sie müssen u.a. auf eine geeignete Vergabe von Zugriffsrechten für das erstellte Programm achten.

Abfangen von böswilligen und fehlerhaften Eingangsdaten

Eine wichtige Rolle für die Systemsicherheit spielt ferner das Verhalten der **Anwender** auf den Systemen. Z.B. Durch das Öffnen von E-Mail-Anhängen oder das Herunterladen von ausführbaren Dateien aus dem Internet kann er die Systemsicherheit gefährden.

## Einige Begriffe

### **SQL-Injection**

Auf einem Webserver befindet sich das Script find.cgi zum Anzeigen von Artikeln. Das Script akzeptiert den Parameter „ID“, welcher später Bestandteil der SQL-Abfrage wird. Folgende Tabelle soll dies illustrieren:

Erwarteter Aufruf	
<b>Aufruf</b>	http://webserver/cgi-bin/find.cgi?ID=42
Erzeugtes SQL	
	SELECT author, subjekt, text FROM artikel WHERE ID=42
SQL-Injektion	
<b>Aufruf</b>	http://webserver/cgi-bin/find.cgi?ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=23
<b>Erzeugtes SQL</b>	SELECT author, subjekt, text FROM artikel WHERE ID=42; UPDATE USER SET TYPE="admin" WHERE ID=23

Wie man erkennen kann, wird dem Programm ein zweiter SQL-Befehl untergeschoben, der die Benutzertabelle modifiziert.

Quelle : WIKIPEDIA

### **Footprinting**

#### 1. Phase eines Angriffs

Informationsbeschaffung ohne direkten Zugriff auf das Zielsystem

Wie besorgt man sich solche Infos?

\* Firmenseite

\* WHOIS-Datenbank -> denic.de

Linux: whois hs-weingarten.de

\* DNS-Einträge [Alle Daten einer Domain werden durch Zonentransfer übertragen]

### **Firewalking**

Pfad durch Firewall finden

Suchen von offenen Ports / Fehlkonfiguration

Mapping des Netzes

## **Penetration Test** (Netz)

Versuch in einen Rechner einzubrechen im Auftrag des Betreibers

## **Phasen eines Penetration-Tests**

Wird aufgeteilt in folgende Phasen:

- \* Reconnaissance (Auskundschaftung)
- \* Enumeration
- \* Exploitation
- \* Documentation

Phasen werden in der Praxis oft vermischt.

## **Reconnaissance (Auskundschaftung)**

Dazu werden Informationen zu folgenden Quellen ausgewertet:

- Homepages
- Google

Google search results for "intitle:Index of" and "intitle:etc parent directory". The result shows a directory listing for '/etc' on a server from November 2006.

Browser screenshot showing a password dump for the '/etc/shadow' file. It lists several user entries, including 'billy:y...JXo:::::::'.

- DNS

Terminal output of 'dig AXFR example.com' command:

```
$ dig AXFR example.com

; <>> DiG 9.3.3rc2 <>> AXFR example.com
;; global options: printcmd
example.com.      86400   IN      SOA     example.com.
                               ; Name Server
example.com.      86400   IN      NS      example.com.
hp3600           86400   IN      A       192.168.1.34
cisco             86400   IN      A       192.168.1.1
client            86400   IN      A       192.168.1.103
www               86400   IN      A       192.168.1.5
peterclient       86400   IN      A       192.168.1.104
```

- Whois

```
usadel@lts2:~$ whois fhwgt.de
Domain:      fhwgt.de
```

wie lange der Eintrag im Name Cache zwischen gesp. wird (in sek.)

```

Domain-Ace: fhwgt.de
Nserver: dns1.hs-weingarten.de
Nserver: dns2.hs-weingarten.de
Nserver: dns1.belwue.de
Nserver: dns3.belwue.de
Status: connect
Changed: 2009-08-25T14:00:11+02:00

[Tech-C]
Type: PERSON
Name: Manfred Dorner
Address: FH Ravensburg-Weingarten
Address: Doggenriedstrasse
Address: Postfach 1261
Pcode: 88241
City: Weingarten
Country: DE
Phone: +49 751 501 9762
Fax: +49 751 501 5 9762
Email: dorner@fh-weingarten.de
Changed: 2005-05-03T11:04:07+02:00

[Zone-C]
Type: PERSON
Name: Manfred Dorner
Address: FH Ravensburg-Weingarten
Address: Doggenriedstrasse
Address: Postfach 1261
Pcode: 88241
City: Weingarten
Country: DE
Phone: +49 751 501 9762
Fax: +49 751 501 5 9762
Email: dorner@fh-weingarten.de
Changed: 2005-05-03T11:04:07+02:00

```

## Enumeration

Suchen/Finden von Angriffsmöglichkeiten

- Port-Scanning -> nmap
- DFN-Cert: meldet Schwachstellen
- Verwendbare Versionen von Diensten und Betriebssystemen ermitteln
- OS-Fingerprint -> welche OS-Version
- Konfigurationsfehler
  - Webserver "Nikto"; Systeme "Nessus" (BSI)
- SNMP: Management von Netzkomponenten
  - > GET: Auslesen von Parametern
  - > SET: Setzten von Parametern
    - (Zugriffsrechte über Community-Strings
    - lesen: public
    - schreiben: wird im Klartext übertragen)
- DNS-Spoofing (Vorgeben, ein anderer zu sein)

## Exploitation

- Ausnutzen von Sicherheitslücken
- Zero Day Exploit
- Was kann dadurch erreicht werden?
- > z.B. root-Rechte erhalten

## Documentation

Der Abschlussbericht:

- Umfangreiche Dokumentation des gesamten Tests
- Schwachstelle
- Details
- Risikoeinstufung
- Lösungsvorschläge
- Managementkurzbericht
- ToDo-Liste: Was kann sofort gemacht werden?

## Ergebnisse eines Penetration Tests

- Schnelle Identifizierung der Schwachstellen
- Lösungsvorschläge / Risikoanalyse
- Awareness
- Schulungseffekt
- Überprüfen des Sicherheitskonzepts

## Häufige Ursachen für Sicherheitslücken

- Veraltete Software, insbesondere Software ohne automatische Online Updates
- Nicht mehr vom Hersteller gepflegte Software/Betriebssysteme
- Schwache Passwörter
- Unsichere Konfiguration
- Admins wird oft nicht genug Zeit gelassen, um alles sicher zu konfigurieren
- Nur an den Außenrändern des Netzes Firewalls, IDS, etc.  
IDS: Intrusion Detection System
- Zu viele Dienste auf einem Server
- Unnötige Dienste aktiviert
- Windowsfreigaben im Netz für alle les- und schreibbar
- Unsicheres WLAN
- Verdächtiges wird nicht weitergegeben
- Incident Response nicht vorhanden -> Notfall-Pläne

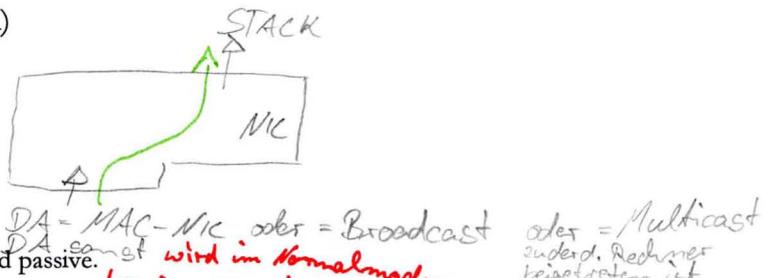
## Sniffing

Sniffer untersuchen und analysieren den Netzwerkverkehr. So haben Administratoren die Chance Schwachstellen und Sicherheitslücken zu erkennen und so auch zu beheben. Im verkehrten Fall können leider natürlich auch Angreifer per Sniffer das Netz nach Schwachstellen absuchen.

Freie Produkte:

- \* Ettercap
- \* NETCORtools (TCP Trace basierend)
- \* Tcpdump
- \* Wireshark (früher bekannt als Ethereal)
- \* NetworkMiner

### Promiscuous Mode



Es gibt zwei Arten von WLAN-Sniffern: aktive und passive.

#### Aktive WLAN-Sniffer

Zu dieser Kategorie gehört der recht verbreitete Netstumbler, der vor allem auf Windows-Systemen genutzt wird. Aktive WLAN-Sniffer senden sogenannte Probe-Request-Pakete an den Access-Point, welcher daraufhin mit einem Probe-Response-Paket antwortet. Es findet also eine explizite Abfrage statt. Anschaulich kann man das vielleicht folgendermaßen erklären: Der Sniffer ruft auf jedem Kanal „Hallo, ist da jemand?“ und jeder Access-Point, der diesen „hören“ kann (im aktuellen WLAN-Kanal), antwortet „Ja, hier ist ein Netz!“

#### Passive WLAN-Sniffer

Der bekannteste Sniffer dieser Kategorie ist u.a. der unter GNU/Linux weit verbreitete Sniffer Kismet.

WLAN-Karten wechseln zum Abhören in den Monitor-Modus.

Sie senden selbst keine Pakete aus.

Zum "Knacken" von WEP-Schlüsseln benötigt man ca. 10 Mio Pakete.

- \* Passivscanner können nicht ausgemacht werden, da keinerlei Emissionen vom Scanner ausgehen. WarDriving mit passivem Scanner ist demzufolge nicht in Logfiles (außer dem des Scanners) nachweisbar.
- \* Passivscanner können natürlich Aktivscanner erkennen. So ist es beispielsweise möglich, Intrusion Detection Systeme wie Snort an passive Scanner wie Kismet zu koppeln, um Angriffe auf WLANs zu bemerken.
- \* Passivscanner erkennen auch „exotische“ WLANs, die nicht auf normale Probe-Requests antworten, abgewandelte Protokolle verwenden (Straßenbahnen in manchen Städten), oder deren ESSID verborgen ist.

Das absichtliche Abhören oder Protokollieren von Funkverbindungen ist in Deutschland verboten, sofern es vom Netzbetreiber nicht explizit erlaubt wurde!

### Verbreitete WLAN-Sniffer

POSIX:

- \* dstumbler – BSD
- \* bsd-airtools – BSD, Toolkit (passiv, WEP-Cracker, WLAN-Bibliothek, ...)
- \* wifiscanner – Linux, BSD, Mac OS X (GPL)
- \* Kismet – Linux, BSD, Mac OS X (GPL)
- \* MacStumbler – Mac OS X

Windows:

- \* NetStumbler – aktiv
- \* NetDetect – aktiv/passiv
- \* AirMagnet Laptop – aktiv/passiv (kommerziell)
- \* Airopeek – passiv (kommerziell)
- \* CommView for WiFi – aktiv/passiv (kommerziell)
- \* Sniff'Em – (kommerziell)
- \* inSSIDer – aktiv (Apache-Lizenz)

Windows Mobile:

- \* WiFiFoFum
- \* WiFi Graph

Andere verwendete Programme

- \* Airsnort – Programm zum Brechen der WEP-Verschlüsselung
- \* Aircrack – ein WEP/WPA-Cracker der neuen Generation
- \* fakeAP – simuliert viele falsche WLANs (zum Ärgern von WarDrivern)

## ***Session- Highjacking***

Übernahme von Sitzungen.

Zuerst passives Sniffing -> Infos sammeln

TCP: Sequenznummern und ACK-Nummern ermitteln

## **Übernahme von TCP-Sitzungen**

Grundgedanke bei der Übernahme von TCP-Sitzungen:

Warten bis sich ein Benutzer eingeloggt hat.

Dann wird die Sitzung übernommen.

## **Übernahme von Web-Sitzungen**

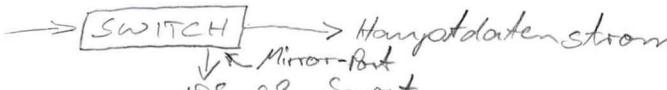
HTTP ist zustandslos.

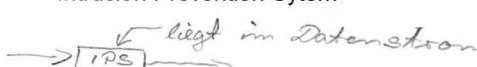
Um z.B. Benutzereingaben zu verwalten wird auf Webserver u.a Cookies verwendet, z.B. Zuteilen einer Sitzungs-ID.

=> Browser übermittelt Cookie bei jeder Anfrage

-> Angreifer entwendet Cookie

**Aufgabe: Welche Bedeutung besitzen die folgenden Begriffe?**

**IDS**      Intrusion Detection System  
  
 Sollen Angriffe, Trojaner, Bauschäden, erkennen.  
 IDS z.B. Snort

**IPS**      Intrusion Prevention System  
  
 Erkennt Angriffe (IDS-Komponente) und reagiert darauf  
**Viren**

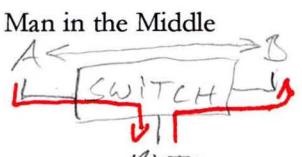
**Würmer**

**Dialer**

**Trojaner, Spyware**

**Hacker**

**Spam**

**Man in the Middle**  
  
 Ettercap } ARP-Spoofing  
 MitM (Angreifer)

Greift Verbindungen an in geschwitzten Netzen mittels  
 Man in the Middle an.

## Hackerparagraph

### § 202c

Vorbereiten des Ausspähens und Auffangens von Daten

(1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er

1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist,

herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft.

(2) § 149 Abs. 2 und 3 gilt entsprechend.

Beweislast liegt beim Admin!

Regeln für die eigene Absicherung:

- Sammlung und Dokumentation der verwendeten Hacker-Tools an einer zentralen Stelle.  
Zugangsbeschränkung auf das Verzeichnis der Tools
- Regelung und Dokumentation der Einsatzzwecke
- Einschränkung der Zugriffsbefugnis
- Dokumentation der Zugriffe auf die Tools und deren Einsätze
- Reguläre Stichproben der Dokumentationen
- Kontrolle der Kontrolleure

## Online-Durchsuchung

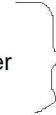
Unbemerkt Zugriff von Strafverfolgungsbehörden auf informationstechnische Systeme.

Bezeichnung der Software für Online-Durchsuchung

RFS - Remote Forensic Software

### Installation der RFS

- Ausnutzung einer unsicheren Konfiguration des Zielsystems
- Ausnutzung eines nicht-allgemein bekannten Fehlers im Zielsystem oder einer Online-Anwendung [Less than Zero Day Exploit] kaufen!



Entfernte Installation

- Automatische Hintergrund-Installation
  - \* Installation über - infizierte Webseite
    - CD/DVD
    - USB-Stick
    - Update (Vorteil: Installation mit Admin-Rechten) -> Microsoft-Update modifizieren
    - E-Mail-Anhang
- Manuelle Installation
  - \* direkter Zugriff auf das Endsystem
  - \* Key-Logger
- Einbau einer Hintertür  
Backdoor, z.B. im BIOS

### Verhindern

- restriktive Konfiguration des \* Betriebssystems
  - \* Browsers
    - keine Script-Sprachen
    - keine Ausführung aktiver Webseiten/Inhalte
- Viren-Scanner
- Personal-Firewalls
- Ständiger Wechsel des Online-Zugangs
- Einsatz von VMs
- Integritäts-Checks -> Prüfsummen von Programmen -> Tripware
- Protokollieren der Kommunikation

## Vorratsdatenspeicherung

März 2010:

GRUNDSATZURTEIL

### Karlsruhe erklärt Vorratsdatenspeicherung für verfassungswidrig

Die Sammlung von Telekommunikationsdaten ist in ihrer jetzigen Form verfassungswidrig. Allerdings schloss Karlsruhe die Vorratsdatenspeicherung nicht grundsätzlich aus.

© Ronald Wittek/dpa



Das Urteil zur Vorratsdatenspeicherung war das letzte des Vorsitzenden des Bundesverfassungsgerichtes, Hans-Jürgen Papier

Die Massenspeicherung von Telefon- und Internetdaten zur Strafverfolgung ist in ihrer jetzigen Form unzulässig. Sie ist dem Urteil der Verfassungsrichter zufolge mit dem Telekommunikationsgeheimnis unvereinbar. Die bisher erhobenen Daten seien unverzüglich zu löschen, verkündeten die Richter in Karlsruhe.

Fertig

Speichern:

Verbindungsdaten, nicht Inhalt

- Festnetz / Handy | jeweils 6 Monate
- Internet-Daten |

-> \* vom ISP zugewiesene IP-Adresse, Anschluss, Dauer, Datum und Uhrzeit der Verbindung  
\* E-Mail: Adresse, Ein- und Ausgangsdaten der Kommunikationspartner (E-Mail-Header)

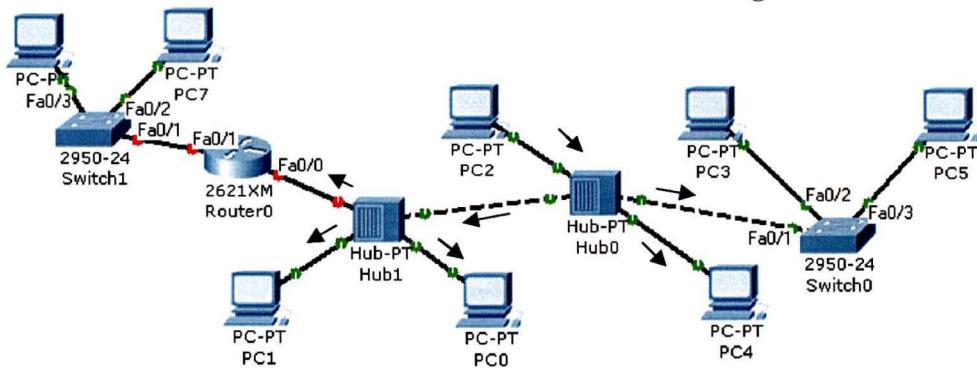
Ausnahme: Geschlossene Benutzergruppe

# Systemsicherheit bei Netzkomponenten

## Arbeitsweise von Netzkomponenten

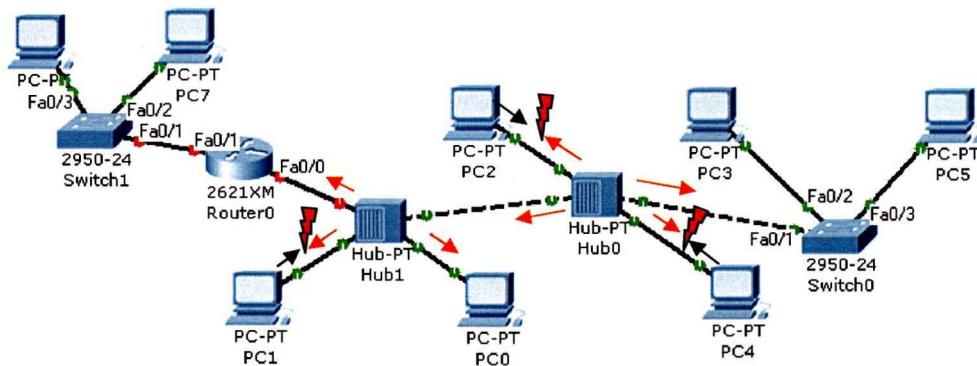
### Arbeitsweise von Hubs

Was ein Hub an einem seiner Interfaces empfängt, leitet er an alle anderen Interfaces weiter. Sendet PC2 in der folgenden Skizze ein Datenpaket, so leitet Hub0 dies an alle anderen Interfaces weiter. Hub1 empfängt dieses Datenpaket über Hub0 und leitet es ebenfalls an alle anderen Interfaces weiter. Der Datenrahmen wird also in die gesamte Kollisionsdomäne verteilt.



Switch0 und Router0 begrenzen die Kollisionsdomäne, die durch die beiden Hubs gebildet wird. In dieser Domäne kann von jedem Netzteilnehmer der gesamte Datenverkehr abgehört werden.

Senden zwei Netzkomponenten (hier PC2 und PC4), die an einen Hub (hier Hub0) angeschlossen sind, gleichzeitig Datenpakete an diesen Hub, so kommt es im Hub zu einer internen Kollision. Der Hub sendet in diesem Fall JAM- Datensendungen an alle seine Interfaces. Der Hub1 leitet die JAM- Sendung von Hub0 an alle seine Interfaces mit Ausnahme des Empfangsinterfaces weiter.



Die JAM Sendungen von Hub0 kollidieren mit den Sendungen der PCs PC2 und PC4. Im Beispiel sendet PC1 ebenfalls. Seine Datensendung kollidiert mit dem JAM von Hub0, welches Hub1 weitergeleitet hat. Somit erkennen alle gleichzeitig sendenden PCs eine Kollision und können gemäß der Vorgaben des Ethernetprotokolls reagieren.

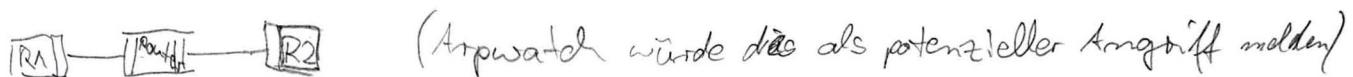
## Arpwatch

Gedacht für Adressen, um Angriffe (ARP-Spoofing) zu erkennen

- Verfolgt den Netzwerkverkehr
- bei Switch muss Monitor-Port eingerichtet sein

## Proxy-ARP (bei Routern einstellbare Funktion)

Wen 2 Rechner miteinander kommunizieren, denken diese, sie seien in selben Subnetz, bei ARP-Requests Antwortet der Router stellvertretend



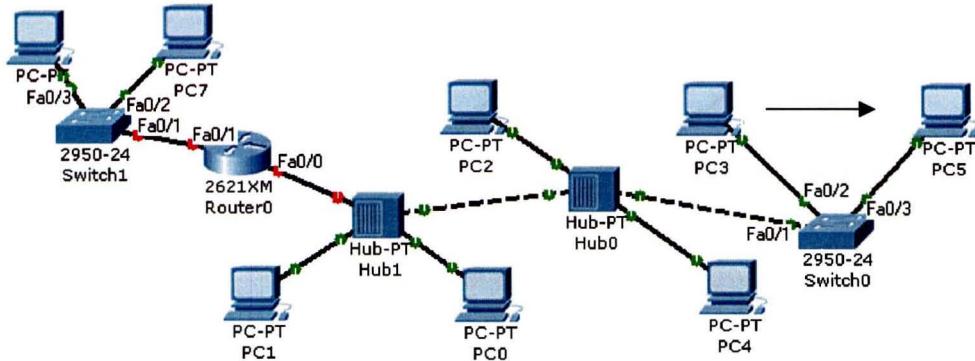
## Gratis-ARP

Host sendet einen ARP-Request, in dem als Ziel- und Quelladresse seine eigene eingetragen hat.  
(Möglichkeit um eine IP-Doppeladressvergabe zu entdecken)

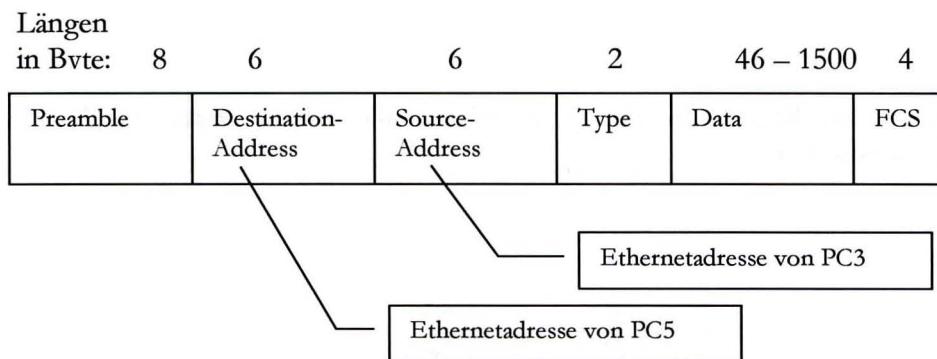
## Arbeitsweise von Switches

Unter anderem, um die in einem Netzwerk verfügbare Bandbreite zu erhöhen, wurden Switches eingeführt. Ein weiterer Effekt bei der Verwendung von Switches ist, dass der Datenverkehr, der an einen Switch angeschlossenen Komponenten, nicht mehr so einfach abgehört werden kann.

Betrachten wir Switch0 in der folgenden Netzskizze. Wird der Switch eingeschaltet, so sind seine Porttabellen (Switching-Tables) - wie bei allen Switches - zunächst leer. Switches lernen selber welche Netzkomponenten an ihre Interfaces angeschlossen sind. Sie identifizieren Rechner und Router anhand der Ethernetadressen dieser Komponenten. Diese Adressen werden dann in die Porttabelle des betreffenden Interfaces eingetragen. Dazu muss die betreffende Komponente mindestens einen Datenrahmen versendet haben. In den Ethernetdatenrahmen werden im Protokollfeld „Source-Address“ die Absendeadressen der Sendequelle eingetragen.



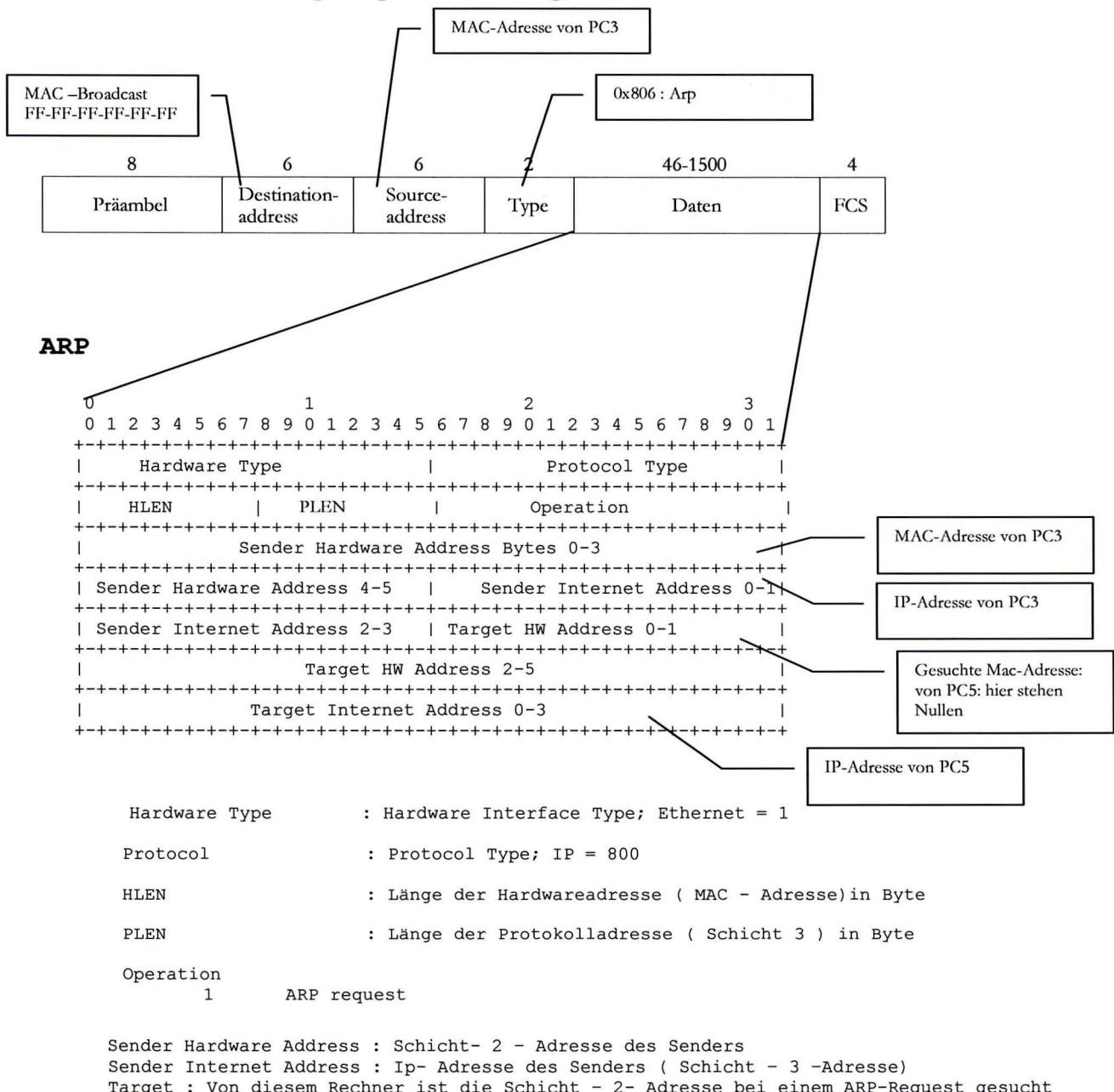
Sendet PC3 nun einen Datenrahmen an PC5, so trägt er in den Ethernetdatenrahmen die folgenden Ethernetadressen ein:



Der Switch empfängt diesen Datenrahmen über sein Interface Fa0/2 und wertet ihn aus. Da seine Porttabellen noch leer sind (er wurde gerade eingeschaltet), trägt er die Adresse von PC3 (Source-Address-Feld) in seine Porttabelle von Fa0/2 ein. Die Adresse von PC5 (Zieladresse) ist dem Switch0 unbekannt. Damit sichergestellt wird, dass der Datenrahmen bei seinem Ziel ankommt, leitet der Switch nun, den von PC3 empfangenen Datenrahmen, über seine anderen aktiven Interfaces (Fa0/1 und Fa0/3) weiter. Dieses Verhalten des Switch (Weiterleiten eines empfangenen Datenrahmens an alle anderen Interfaces bei unbekannter Zieladresse) wird als **flooding** bezeichnet. Der Datenrahmen, der den Switch über Fa0/1 verlässt, wird von den beiden Hubs in der gesamten Kollisionsdomäne, die sie bilden, verteilt. Dieser Datenrahmen kann also in der gesamten Kollisionsdomäne abgehört werden.

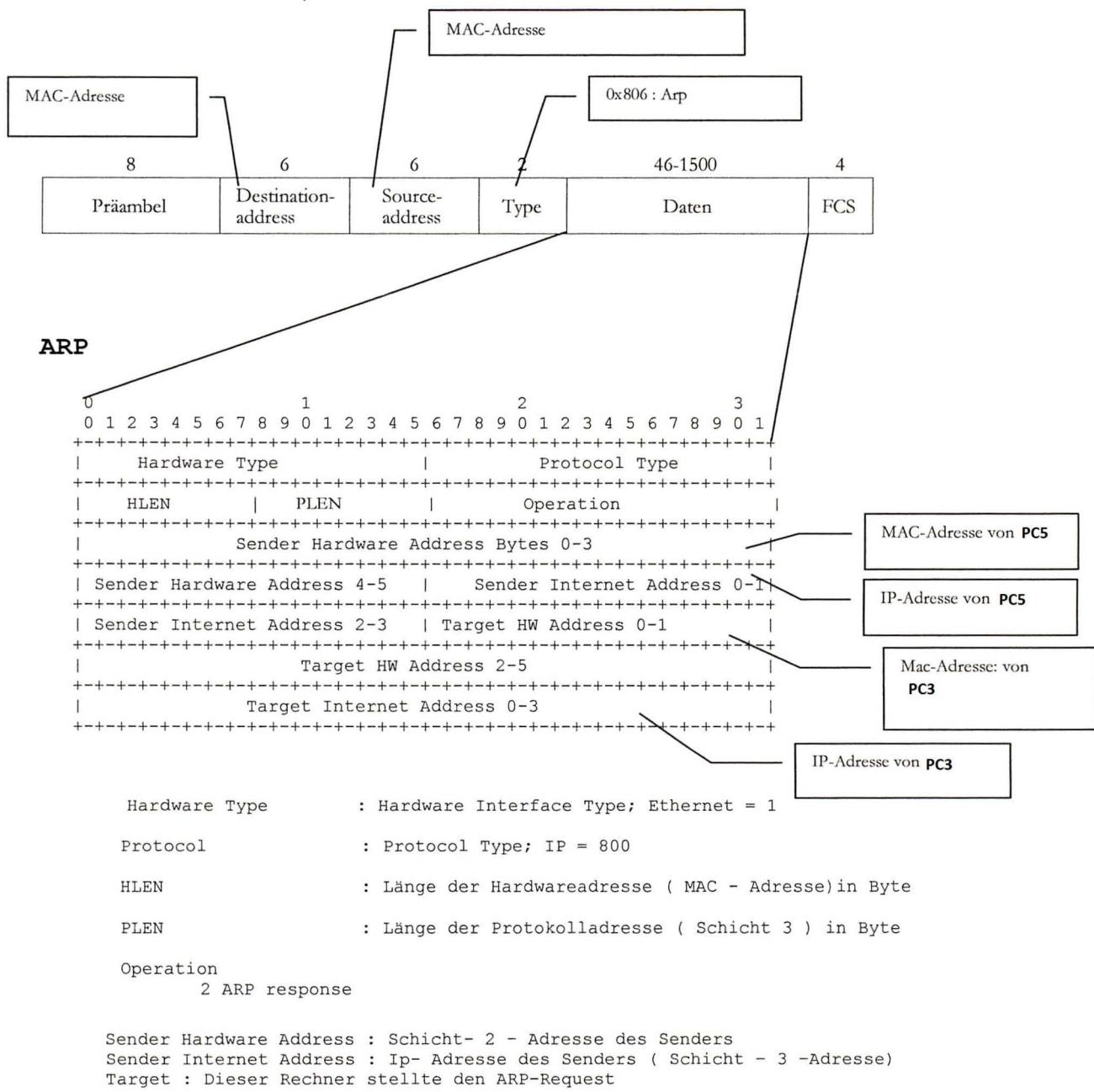
Sendet nun PC5 ein Antwortpaket an PC3, so empfängt Switch0 den Datenrahmen über Fa0/3. In diesem Fall kennt der Switch0 die Zieladresse von PC3. er sendet den Datenrahmen nur über das Interface Fa0/2 weiter. Dieser Datenrahmen kann also in der Kollisionsdomäne nicht abgehört werden. Wenn ein Datenrahmen von einem Switch –wie in diesem Fall- gezielt weitergeleitet wird, bezeichnet man dieses Verhalten als forwarding.

Damit PC3 dem Rechner PC5 einen IP- Datenrahmen senden kann, benötigt er zusätzlich zur IP-Adresse von PC5 dessen MAC-Adresse. Diese erfragt er mit Hilfe eines Arp- Request- Datenrahmens. Dieser Arp- Request hat den folgenden Aufbau:



Arp- Requests werden an die MAC- Broadcast- Adresse gesendet. Datenrahmen, die an die Broadcast-Adresse gesendet werden, werden von einem Switch immer geflutet. D.h. diese Datenrahmen können an jedem Switch- Interface abgehört werden. Sie werden auch durch kaskadierte Switches weiter verteilt. Hubs verteilen Broadcasts –da sie die MAC-Adressen nicht auswerten- sowieso an alle Interfaces mit Ausnahme des Empfangsinterfaces weiter.

Welchen Aufbau hat der ARP-Response zum obigen ARP-Request (PC3 sucht die MAC-Adresse von PC5) ?

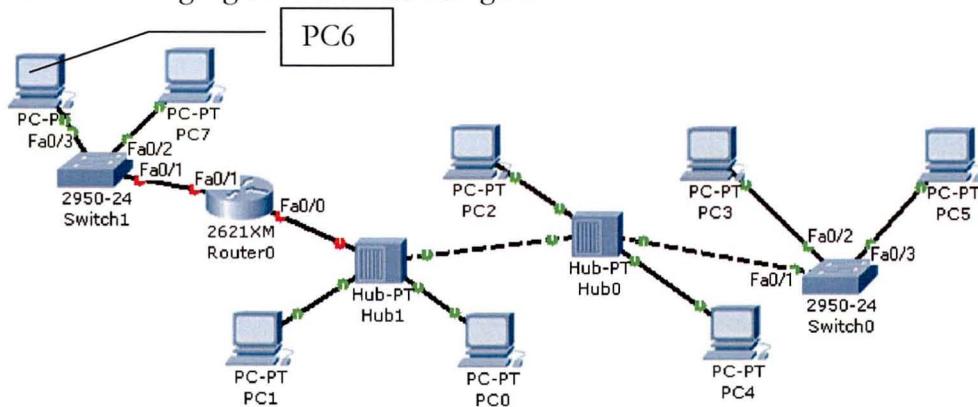


## Arbeitsweise von Routern

Hubs arbeiten in der Schicht 1 des OSI-Modells. Sie werten die Datenrahmen, die sie weiterleiten nicht aus. Switches verwenden Schicht 2 – Adressen (MAC-Adressen, Ethernetadressen...) um ihre Switching-Entscheidungen zu treffen. Sie arbeiten in der Schicht 2 des OSI Modells.

Router arbeiten in der Schicht 3 des OSI-Schichtenmodells. Sie werten daher die Adressen aus, die in dieser Schicht definiert sind. Bei der Verwendung von IP sind es die IP-Adressen. Ein Router muss jedes Schicht 3-Protokoll, das er weiterleiten soll, kennen. Wir werden uns auf IPv4 beschränken. Wenn in diesem Skript eine Netzkomponente mit Router bezeichnet wird, so ist immer ein IPv4-Router gemeint. Das ist ein Router, der IPv4-Pakete weiterleiten (routen) kann.

Switches und Hubs sind für einen Rechner transparent. D.h. er merkt nichts von dem Vorhandensein dieser Komponenten im Netz. Router sind jedoch nicht transparent, sie müssen explizit mit dem Weiterleiten von Datenpaketen beauftragt werden. Diese Beauftragung erfolgt, indem ein Rechner ein weiterzuleitendes Datenpaket an eine der MAC-Adressen des Routers schickt. Damit ein Rechner den richtigen Router für ein Netzwerkziel auswählen kann, ermittelt er diesen aus seiner Routingtabelle. Häufig haben Rechner nur einen Router in ihrer Routingtabelle eingetragen. Dies ist der sog. Default Router. Dies ist regelmäßig dann der Fall, wenn der betreffende Rechner an ein Stub-Netz angeschlossen ist. Dies ist ein Netz aus dem es nur einen Ausgang über einen Router gibt.



Der Router in der obigen Skizze besitzt die IP-Adressen Fa0/0 = 192.168.1.1 und Fa0/1 = 192.168.2.1. Er verbindet also die Netze 192.168.1.0 und 192.168.2.0. PC6 und PC7 geben also 192.168.2.1 als Default-Gateway an, die anderen PCs der Netzkizze 192.168.1.1.

Router leiten MAC-Broadcasts niemals weiter. Daraus folgt, dass Arp-Requests am nächsten Router enden.

Datenverkehr, der sich „hinter“ einem Router abspielt, kann nicht abgehört werden, es sei denn man besitzt einen Rechner im anderen Subnetz, der diese Datenpakete aufzeichnet. Das könnte z.B. ein Rechner sein, der einen aktiven Trojaner besitzt.

## Aufgabe 1

- Beschreiben Sie die Arbeitsweise von HUB / Switch und Router!
- An welche Adressen werden ARP-Responses und ARP-Requests gesendet?
- Welche ARP-Nachrichten können in einer geswitchten Netzmgebung mit einem Snifferprogramm mitgehört werden?
- Was versteht man unter einem Stub-Netz?

## Aufgabe 1

- Beschreiben Sie die Arbeitsweise von HUB / Switch und Router!
- An welche Adressen werden ARP-Responses und ARP-Requests gesendet?
- Welche ARP-Nachrichten können in einer geswitchten Netzumgebung mit einem Snifferprogramm mitgehört werden?
- Was versteht man unter einem Stub-Netz?
- Welche Routen tauchen in der folgenden Routingtabelle auf?

```
C:\Dokumente und Einstellungen\usadel>netstat -r
```

Routingtabelle

```
=====
Schnittstellenliste
0x1 ..... MS TCP Loopback interface
0x2 ...00 50 56 c0 00 08 ..... VMware Virtual Ethernet Adapter for VMnet8
0x3 ...00 50 56 c0 00 01 ..... VMware Virtual Ethernet Adapter for VMnet1
0x4 ...00 18 de 9d 27 cc ..... Intel(R) PRO/Wireless 3945ABG Network Connection -
Paketplaner-Miniport
ort
0x6 ...00 ff bc ab a3 b9 ..... TAP-Win32 Adapter V8 - Paketplaner-Miniport
0x20005 ...00 15 58 7c d5 25 ..... Intel(R) PRO/1000 PL Network Connection -
Paketplaner-Miniport
=====
```

Aktive Routen:

Netzwerkziel	Netzwerkmaske	Gateway	Schnittstelle	Anzahl	Metric
0.0.0.0	0.0.0.0	141.69.100.1	141.69.100.38	10	1 für die Güte der Route
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1	
141.69.100.0	255.255.255.0	141.69.100.38	141.69.100.38	10	
141.69.100.38	255.255.255.255	127.0.0.1	127.0.0.1	10	
141.69.255.255	255.255.255.255	141.69.100.38	141.69.100.38	10	
192.168.224.0	255.255.255.0	192.168.224.1	192.168.224.1	20	
192.168.224.1	255.255.255.255	127.0.0.1	127.0.0.1	20	
192.168.224.255	255.255.255.255	192.168.224.1	192.168.224.1	20	
192.168.234.0	255.255.255.0	192.168.234.1	192.168.234.1	20	
192.168.234.1	255.255.255.255	127.0.0.1	127.0.0.1	20	
192.168.234.255	255.255.255.255	192.168.234.1	192.168.234.1	20	
224.0.0.0	240.0.0.0	141.69.100.38	141.69.100.38	10	
224.0.0.0	240.0.0.0	192.168.224.1	192.168.224.1	20	
224.0.0.0	240.0.0.0	192.168.234.1	192.168.234.1	20	
255.255.255.255	255.255.255.255	141.69.100.38	141.69.100.38	1	
255.255.255.255	255.255.255.255	192.168.224.1		6	
255.255.255.255	255.255.255.255	192.168.224.1		4	
255.255.255.255	255.255.255.255	192.168.224.1	192.168.224.1	1	
255.255.255.255	255.255.255.255	192.168.234.1	192.168.234.1	1	
Standardgateway:	141.69.100.1				

Ständige Routen:

Keine

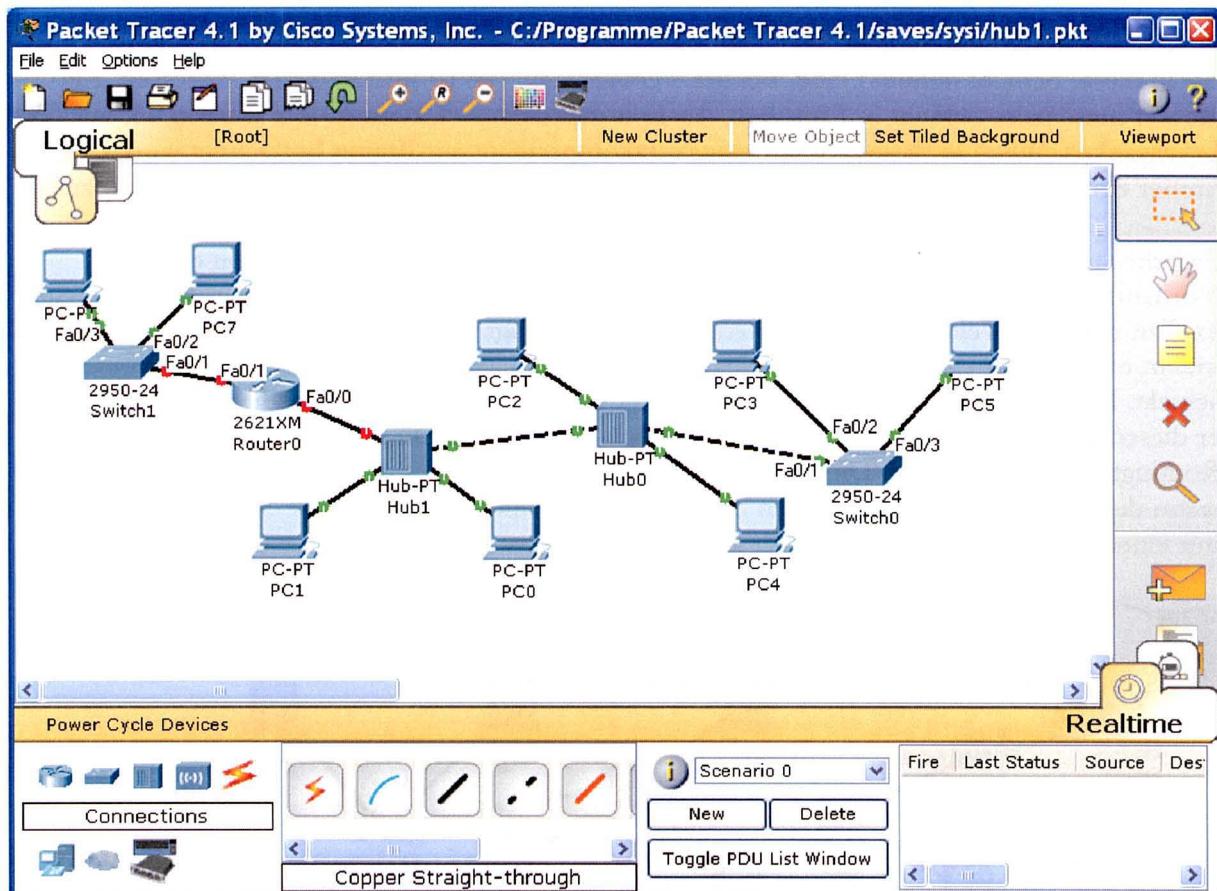
direkter IP Broadcast geht an ein (Sub)-Netz  
(je nach Router-Config. wird weitergeleitet)

(fließt IP Broadcast bis zum nächsten Router)

Über welche IF werden IP Multicast gesendet

## Hubs und Netzwerksicherheit

Hubs senden Datenpakete, die sie auf irgendeinem ihrer Interfaces empfangen, an alle anderen Interfaces weiter. Daher kann ein Rechner, der an einen Hub angeschlossen ist, den Datenverkehr von allen Systemen, die zu einer Kollisionsdomäne gehören, abhören. Zum Abhören werden Snifferprogramme wie z.B. Wireshark verwendet.



### Aufgabe 2:

- Kennzeichnen Sie die Kollisionsdomänen in der obigen Netzskeize.
- Der Router besitzt die IP-Adressen Fa0/0 = 192.168.1.1 und Fa0/1 = 192.168.2.1. Wie müssen die IP-Stacks der PCs konfiguriert werden, damit sie untereinander kommunizieren können.
- Switch0 wird eingeschaltet. PC5 sendet an PC3 3 Ping requests in Folge. Wie werden die Datenpakete von Switch0 verteilt? Wie werden diese Datenpakete von den beiden Hubs verteilt? Wie bearbeitet der Router 0 diese Datenpakete?
- PC0 sendet an PC1 3 Ping requests in Folge. Wie werden diese Datenpakete von den Netzkomponenten der Skizze bearbeitet?
- Untersuchen Sie das Verhalten der Netzkomponenten mit Hilfe der Packet-Tracer Simulation: Packet-Tracer-Datei dieser Netzkonfiguration: /saves/sysi/hub1.pkt

### Fazit:

Was kann in einer Kollisionsdomäne abgehört werden?

Was kann von Rechnern abgehört werden, die am selben Switch angeschlossen sind?

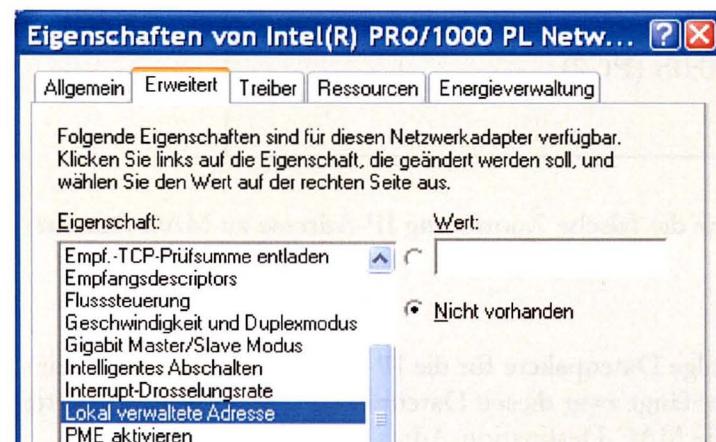
Was kann man vom Datenverkehr abhören, der sich „hinter“ einem Router oder mehreren Routern befindet?

## ARP- Spoofing in einer HUB- Umgebung

Beim Spoofing (Manipulation, Verschleierung) gibt ein Rechner vor ein anderer Rechner zu sein. Beim ARP- Spoofing erhält der angreifende Rechner die MAC- Adresse eines anderen oder eines nicht vorhandenen Rechners.

### Ändern der eigenen MAC- Adresse

Bei Windows XP kann man seine MAC- Adresse unter „Eigenschaften von LAN-Verbindung“ ändern. Dazu muss man in das Untermenü der Konfiguration der Netzwerkkarte wechseln.



Dort findet sich der Menüpunkt „Lokal verwaltete Adresse“. An dieser Stelle lässt sich die eigene MAC- Adresse ändern.

**Es ist also keine gute Idee irgendeine Sicherheitsmaßnahme darauf zu gründen, die Identität eines Rechners an seiner MAC- Adresse erkennen zu wollen.**

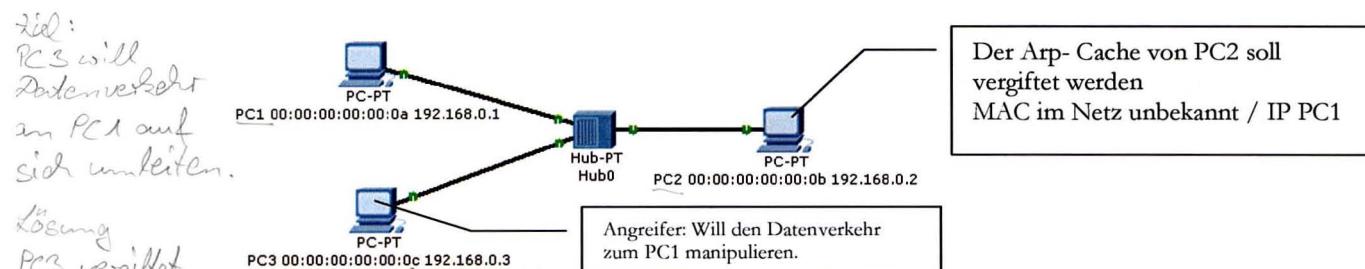
Bei Linux ändert man die MAC- Adresse des Interfaces eth0 mit folgenden Befehlen:

```
I
ifconfig eth0 down
ifconfig eth0 hw ether 0a:0a:0a:0a:0a:0a
ifconfig eth0 192.168.0.23
ifconfig eth0 up
```

## Spoof-Szenario

Gegeben ist das folgende Netz. Für die PCs sind sowohl deren MAC- als auch deren IP-Adressen angegeben. PC3 sei der Angreifer, er will den Datenverkehr von PC2 an PC1 auf sich ziehen und dann verändern.

- 1.) PC2 erzeugt einen ARP-Request: PC2 sucht die MAC-Adresse von PC1



- 2.) PC3 generiert dazu einen gefälschten ARP-Reply auf einen ARP-Request von PC2 mit folgendem Inhalt:

```

dst-ether 00:00:00:00:00:b
src-ether 00:00:00:00:00:d
-----
sender HW-Address: 00:00:00:00:0d(im Netz unbekannt)
sender-ip: 192.168.0.1 (PC1)
target HW-Address: 00:00:00:00:0b (PC2)
target-ip:192.168.0.2 (PC2)

```

- 3.) PC2 trägt daraufhin in seinen arp-cache die falsche Zuordnung IP-Adresse zu MAC-Adresse ein:

192.168.0.1 / 00:00:00:00:00:d

Er sendet aufgrund dieses Eintrages in Folge Datenpakete für die IP-Adresse 192.168.0.1 an die MAC-Adresse 00:00:00:00:0d. PC1 empfängt zwar diesen Datenrahmen. Seine Netzwerkkarte verwirft jedoch dieses Datenpaket, weil die MAC-Destination-Adresse nicht mit der eigenen MAC-Adresse (00:00:00:00:0a) übereinstimmt.

PC3 hat seine Netzwerkkarte in den promiscuous- Mode gesetzt (damit empfängt diese Karte den gesamten Datenverkehr). Er empfängt also auch den Datenverkehr, der an die Adresse 00:00:00:00:0d gesendet wird. Er kann diesen Datenverkehr abhören, verändern und an das eigentliche Ziel (PC1) weiterleiten.

Wie kann der gefälschte ARP-Reply dem PC2 untergeschoben werden? PC3 bekommt den gesamten Datenverkehr mit, damit auch den ARP-Request von PC2. Er muss auf diesen schneller antworten als PC1 und gegebenenfalls PC1 durch eine DoS- Attacke lahm legen.

Eine andere Art des Angriffs besteht darin, dass sich der Angreifer die Adresse der zu spoofenden MAC-Adresse gibt.

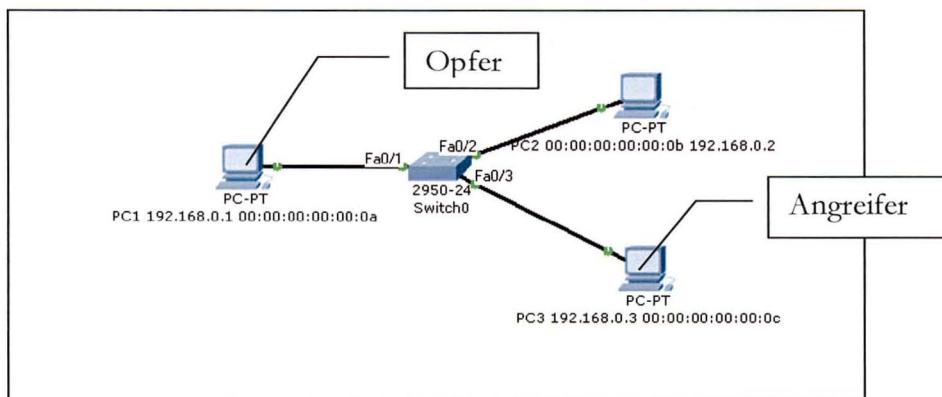
## Switches und Netzwerksicherheit

Wir hatten gesehen, dass es bei Switches schwerer als bei Hubs ist, den Datenverkehr im gleichen Subnetz abzuhören. Mit Hilfe des ARP- Relaying ist es jedoch möglich, Datenverkehr, der für andere Rechner bestimmt ist, selbst in einer geswitchten Umgebung abzuhören. Dazu gibt es zwei Ansätze.

### ARP- Spoofing Variante 1

Der Angreifer gibt sich die MAC- Adresse des anzugreifenden Rechners.

Der Datenverkehr kommt daraufhin bei ihm an. Diesen Verkehr hört er ab. Danach leitet er den Datenverkehr an den eigentlichen Bestimmungsort weiter.



PC2 sucht von PC1 die MAC- Adresse. Er sendet folgenden ARP- Request:

dst-ether ff:ff:ff:ff:ff:ff	ARP-Request
src-ether 00:00:00:00:00:0b (PC2)	
<hr/>	
sender HW-Address: 00:00:00:00:00:0b (PC2)	
sender-ip: 192.168.0.2 (PC2)	
target HW-Address: 00:00:00:00:00:00 (wird gesucht)	
target-ip: 192.168.0.1 (PC1)	

Da der ARP-Request an die MAC- Broadcast- Adresse gesendet wird, verteilt der Switch diese Datensendung an alle seine Interfaces mit Ausnahme des Empfangsinterfaces. PC3 erhält also ebenfalls diesen Request. PC3 generiert daraufhin einen gefälschten ARP- Reply mit folgendem Inhalt:

dst-ether 00:00:00:00:00:0b (PC2)	ARP-Response
src-ether 00:00:00:00:00:0c (PC3)	
<hr/>	
sender HW-Address: 00:00:00:00:00:0c (PC3)	
sender-ip: 192.168.0.1 (PC1)	
target HW-Address: 00:00:00:00:00:0b (PC2)	
target-ip: 192.168.0.2 (PC2)	

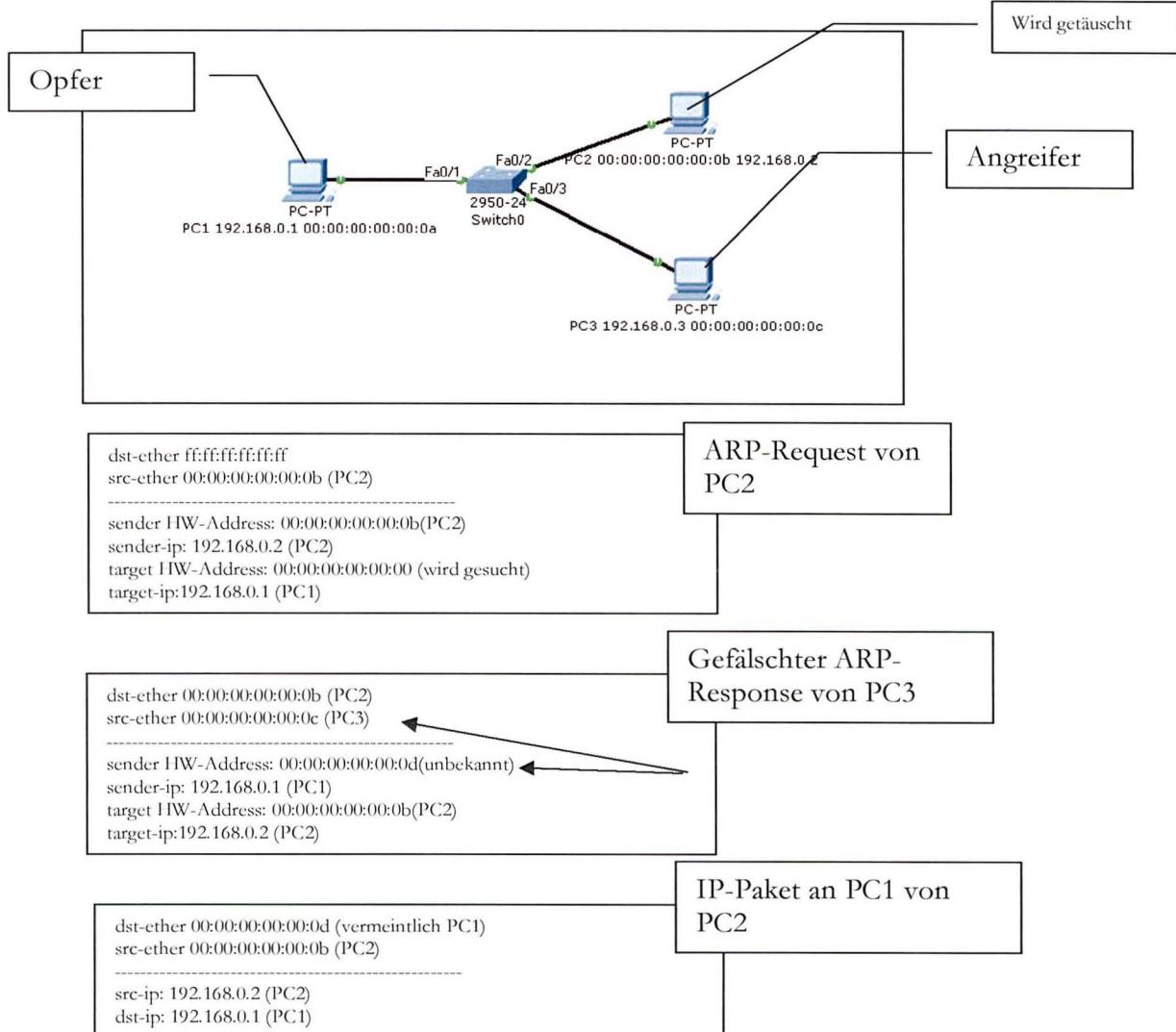
PC2 trägt daraufhin in seinen arp-cache die falsche Zuordnung IP- Adresse zu MAC- Adresse ein:

192.168.0.1 00:00:00:00:00:0c

Datenpakete, die PC2 an PC1 senden will, werden in Folge an die MAC-Adresse 00:00:00:00:00:0c gesendet. Sie gelangen also zum PC3. Dieser leitet die Datenpakete nach dem Auswerten dann an das eigentliche Ziel PC1 weiter.

## ARP-Spoofing Variante 2

Bei dieser Variante wird die MAC-Adresse des Angriffsopfers mit einer nicht existierenden MAC-Adresse gespoofed.



Ein Paket, welches an das Opfer gesendet wird, kommt bei allen Systemen im geschwitzten Netzsegment an, da ein Switch eine unbekannte Adresse niemals lernen kann. Damit ist diese Adresse auch keinem Switch bekannt.

Das Paket wird von allen Netzwerkkarten mit Ausnahme der Netzwerkkarte des Angreifers (diese befindet sich im Promiscuous-Mode) verworfen.

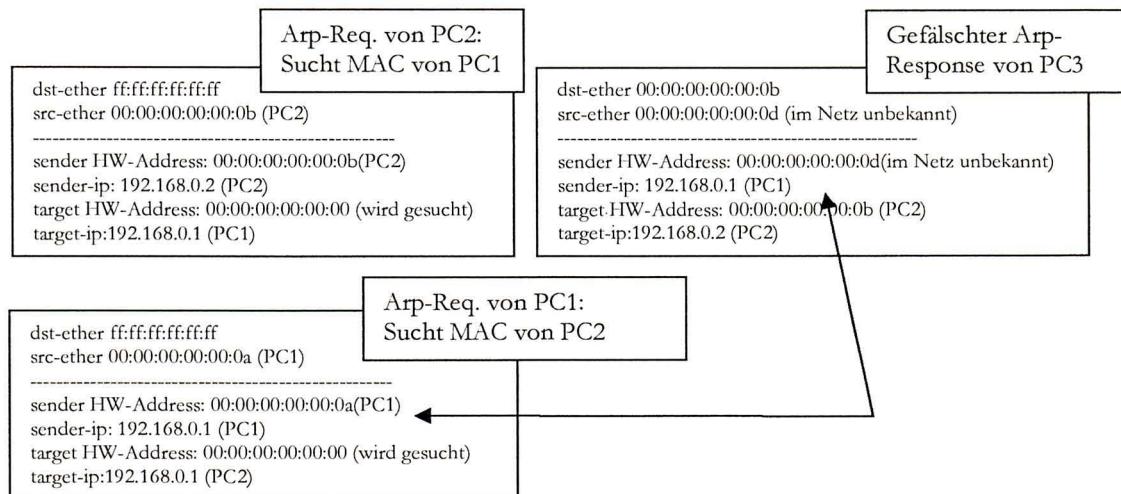
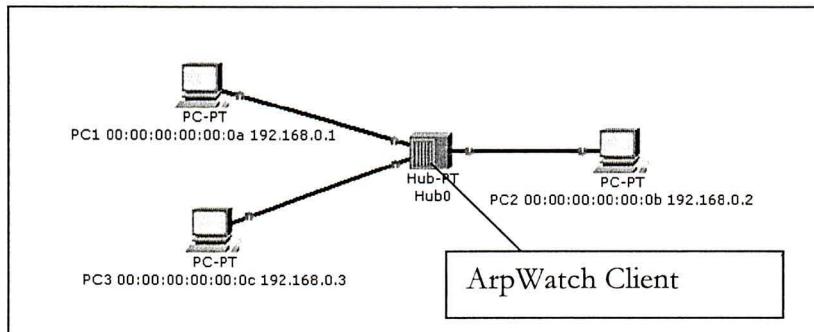
Der Angreifer modifiziert das Paket und sendet es an das eigentliche Ziel weiter.

## ArpWatch

Arpwatch ist ein Tool, welches die Address Resolution Protocol-Pakete (ARP-Pakete) überwacht. Auf diese Weise können neu angeschlossene Rechner erkannt werden, und es können ebenfalls Situationen erkannt werden, in denen IP-Adressen doppelt vergeben werden. Diese Ereignisse werden dem zuständigen Administrator automatisch per E-Mail mitgeteilt, so dass dieser zeitnah eingreifen kann. Das Rechenzentrum erhält diese Hinweise ebenfalls und kann bei Bedarf dem jeweiligen Administrator mit Rat und Tat zur Seite stehen.

## ArpWatch in einem HUB-Netz

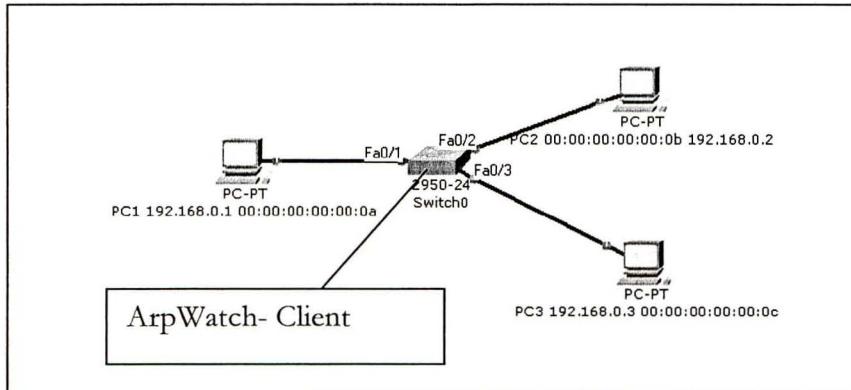
In einer Netzumgebung, die aus Hubs besteht, kann ArpWatch alle ARP-Requests und alle ARP-Responses aufzeichnen. Um alle Responses aufzeichnen zu können, setzt ArpWatch die Netzwerkkarte des Rechners, auf dem ArpWatch arbeitet, in den Promiscuous-Modus.



## ArpWatch in einem Switch-Netz

Überwacht man die ARP- Requests mit ArpWatch in einer geswitchten Umgebung, so kann man folgende Arp- Datenrahmen aufzeichnen:

ARP-Requests, weil diese an die MAC-Broadcast- Adresse gesendet werden. Durch die Auswertung der ARP- Protokollfelder „sender HW-Address“ und „sender-ip“ erhält man eine Liste von IP- Adressen und zugehörigen MAC- Adressen.



## Aufgabe 3

- Wie kann Arpwatch in einer Kollisionsdomäne ARP- Spoofing -Angriffe aufdecken?
- Was kann Arpwatch in einer geswitchten Umgebung aufdecken?

## Gratuitous ARP

Quelle: [http://de.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](http://de.wikipedia.org/wiki/Address_Resolution_Protocol)

*Gratuitous ARP* (engl. „unaufgefordertes ARP“) bezeichnet eine spezielle Verwendung von ARP. Dabei wird von einem Host ein ARP-Anforderungs-Broadcast gesendet, bei der er seine eigene IP-Adresse als Quell- und Ziel-IP-Adresse einträgt. Damit teilt er seine ggf. neue MAC-Adresse unaufgefordert mit. Das kann mehreren Zwecken dienen:

Z funktioniert  
das wirklich  
so einfach

1. Normalerweise darf keine Antwort kommen, denn eine IP-Adresse muss in einem Netz eindeutig sein. Bekommt er trotzdem eine Antwort, ist das für den Administrator ein Hinweis darauf, dass ein Host nicht richtig konfiguriert ist.
2. Jeder Host aktualisiert seinen ARP-Cache. Das ist beispielsweise dann nützlich, wenn die Netzwerkkarte eines Rechners ausgetauscht wurde und die anderen Hosts über die neue MAC-Adresse informiert werden sollen. *Gratuitous ARP* geschieht deshalb normalerweise beim Booten eines Computers.
3. Wenn zwei Server aus Gründen der Ausfallsicherheit als Server und Ersatzserver aufgebaut sind und sich eine IP-Adresse teilen und der aktive Verkehr vom einen auf den anderen geschwenkt werden soll, ist die IP-Adresse jetzt über eine andere MAC-Adresse zu erreichen. Diese neue MAC-/IP-Adress-Zuordnung muss bekannt gemacht werden. Sonst bekommt niemand den Wechsel mit.
4. In einem *Mobile IP*-Szenario sendet der *Home Agent* einen *Gratuitous ARP*, wenn sich der *Mobile Host* aus dem Heimatnetz entfernt, um die Pakete stellvertretend für diesen zu empfangen. Analog sendet der *Mobile Host* einen *Gratuitous ARP*, sobald er sich wieder im Netz befindet.

## Example Traffic

```
Ethernet II, Src: 02:02:02:02:02:02, Dst: ff:ff:ff:ff:ff:ff
  Destination: ff:ff:ff:ff:ff:ff (Broadcast)
  Source: 02:02:02:02:02:02 (02:02:02:02:02:02)
  Type: ARP (0x0806)
  Trailer: 0000000000000000000000000000000000000000000000000000000000000000
Address Resolution Protocol (request/gratuitous ARP)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: 02:02:02:02:02:02 (02:02:02:02:02:02)
  Sender IP address: 192.168.1.1 (192.168.1.1)
  Target MAC address: ff:ff:ff:ff:ff:ff (Broadcast)
  Target IP address: 192.168.1.1 (192.168.1.1)
```

0000 ff ff ff ff ff 02 02 02 02 02 02 08 06 00 01	.....
0010 08 00 06 04 00 01 02 02 02 02 02 c0 a8 01 01	.....
0020 ff ff ff ff ff c0 a8 01 01 00 00 00 00 00 00 00	.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

## Starten von Vista

```

# Frame 28 (42 bytes on wire, 42 bytes captured)
# Ethernet II, Src: VMware_2c:d1:b8 (00:0c:29:2c:d1:b8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
# Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: vmware_2c:d1:b8 (00:0c:29:2c:d1:b8)
  Sender IP address: 0.0.0.0 (0.0.0.0)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.0.171 (192.168.0.171)

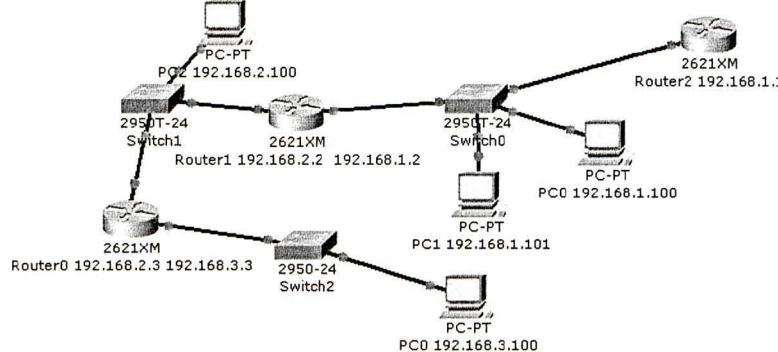
```

Der Vista-Rechner hat die IP-Adresse 192.168.0.171

## Proxy Arp

### Aufgabe4 Proxy Arp

- Was versteht man unter dem Begriff Proxy ARP?
- Was meldet ein ArpWatch-Daemon im Subnetz 192.168.1.0, wenn Router1 Proxy Arp aktiviert hat und PC0 und PC1 die Subnetzmaske 255.255.0.0 verwenden?



## Router und Netzwerksicherheit

Router treffen ihre Routingentscheidungen aufgrund ihrer Routing-Tabellen. Diese werden entweder vom Administrator oder automatisch mit Hilfe von Routing-Protokollen gefüllt. Zu einem Ziel kann es mehr als eine Route geben. Welche der Routen verwendet wird, entscheidet sich daran, welche die beste Metrik aufweist.

Angriffe auf Routingtabellen:

Angreifer kündigt Sessore Route zum Ziel an  
als andere Router im Subnetz

Einige wichtige Routing-Protokolle:

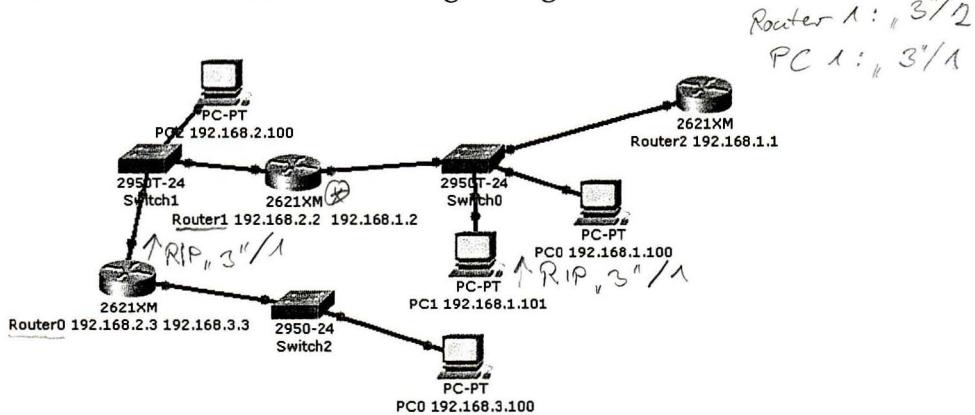
- Routing Information Protocol (**RIP** Versionen 1 und 2)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced Interior Gateway Routing Protocol (**EIGRP**)
- Open Shortest Path First (**OSPF**)

Protokolle mit Authentifizierung: Fett gedruckt

Nur von authentifizierten Partnern werden  
Routing-Updates akzeptiert.

## RIP Version 1

Mit Hilfe des RIP werden Routen angekündigt.



Router 1 hat gleich gute Kandidaten für 192.168.3.0  
Cisco: Load Balancing (Pakete werden gleichmäßig aufgeteilt)

## Aufgabe RIP

Gegeben ist die obige Netzskeze.

a.) Wie kündigt Router1 die Route ins Netz 192.168.3.0 in das Netz 192.168.1.0 an? Tragen Sie diese Ankündigung in die folgenden RIP- Protokollfelder ein.

b.) Wie können Sie den Datenverkehr vom Netz 192.168.1.0 in das Netz 192.168.3.0 auf den PC1 mit Hilfe von RIP umleiten?

c.) Implementiert die Netzwerksimulation packet-tracer

das RIP-Protokoll (rip.pkt)? Welchen Inhalt haben die RIP-PDUs?

0	1	2	3	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1	
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
command (1)   version (1)   must be zero (2)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
address family identifier (2)   must be zero (2)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
192.168.2.0   IP address (4)   192.168.3.0				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
1   metric (4)   2				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
address family identifier (2)   must be zero (2)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
IP address (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
metric (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	

0	1	2	3	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1	
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
address family identifier (2)   must be zero (2)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
IP address (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
must be zero (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	
metric (4)				
+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	+-----+ +-----+ +-----+ +-----+	

command :  
1: request  
2: response

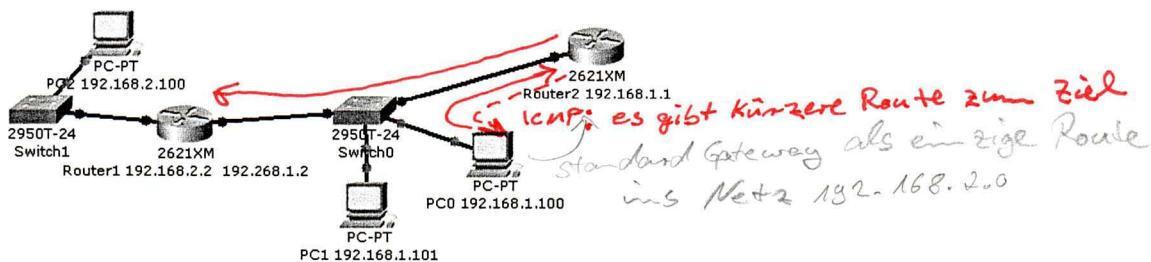
address  
family  
identifier:  
2: IP- family

RFC 1058

## Schwächen von Protokollen ausnutzen

### Angriffe über ICMP

#### ICMP Redirection



Eine ICMP redirect - Nachricht wird gesendet, wenn ein Router eine kürzere Route zum Ziel kennt. Der Router erkennt die kürzere Route an dem Sachverhalt, dass ein Datenpaket, welches er zum Weiterleiten erhält, über das Empfangsinterface weiter gesendet wird.

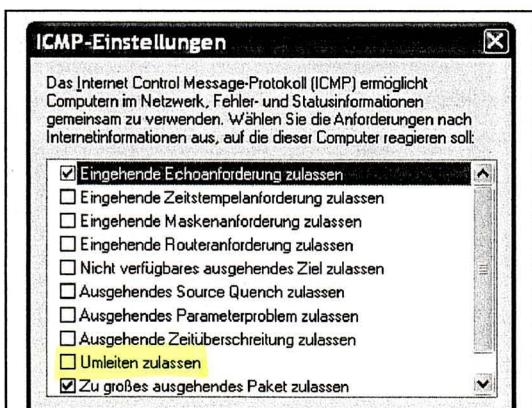
In der obigen Netzkizze habe PC0 den Router2 als Standard- Gateway eingetragen, ferner hat er keine weiteren Routen neben den Routen in das direkt angeschlossene Netz eingetragen. Router2 kennt die Route ins Subnetz des PC2 über Router1.

Sendet PC0 ein Datenpaket an PC2, so sendet er es an sein Standardgateway Router2. Dieser leitet das Datenpaket an Router1 weiter, er sendet zusätzlich noch eine ICMP- Redirect Nachricht an PC0. Diese Nachricht enthält den Hinweis, dass es eine kürzere Route zum Ziel über Router1 gibt.

#### Aufgabe ICMP redirect

a.) Wie setzt Router2 die ersten 64 Bit der ICMP redirect- Nachricht, wenn PC0 an PC2 ein Ping-Paket sendet?

0	1	2	3
Type	Code	Checksum	
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Gateway Internet Address			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
	Internet Header + 64 bits of Original Data Datagram		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+



b.) Testen Sie ob die Netzwerksimulation ICMP-Redirects simuliert! Das Netzszenario ist unter `redir.pkt` gespeichert.

c.) Wie verhält sich XP, wenn eine ICMP-redirect Nachricht eintrifft?

d.) Wie können Sie den Datenverkehr von PC1 auf PC0 umleiten? PC1 verwendet XP als Betriebssystem.

e.) Schützt die Windows-XP-Firewall vor ICMP-redirect-Nachrichten?

## ICMP Destination unreachable

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Type   Code   Checksum	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
unused	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Internet Header + 64 bits of Original Data Datagram	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+

ICMP Fields:

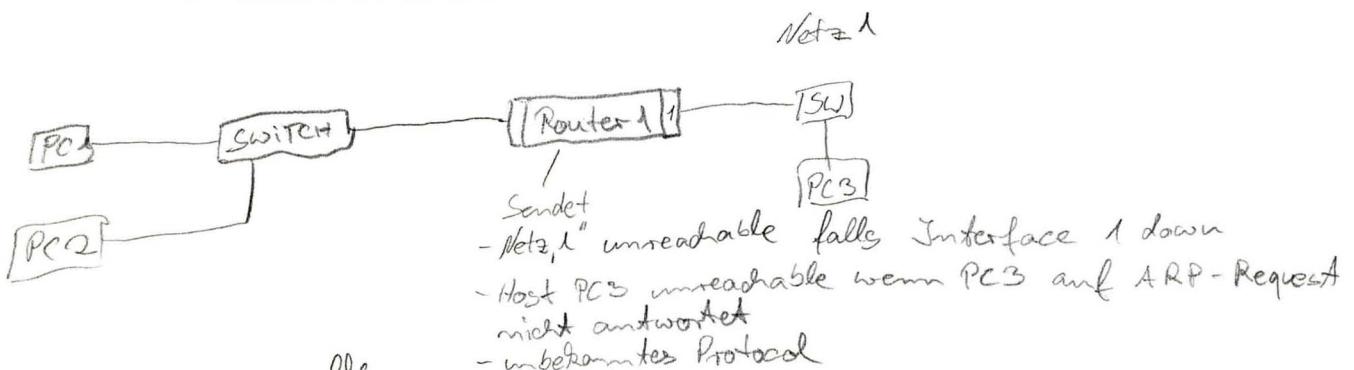
Type 3

Code

- 0 = net unreachable;
- 1 = host unreachable;
- 2 = protocol unreachable;
- 3 = port unreachable;
- 4 = fragmentation needed and DF set;
- 5 = source route failed.

Server: Dienst nicht verfügbar

Absender gibt Route vor (in IP-Option Feld)



Die meisten Firewalls  
Sperren diese Pakete!  
IPv6 (im Moment) nicht mehr vorgeschlagen

## ICMP Time Exceeded Message

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Type   Code   Checksum	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
unused	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Internet Header + 64 bits of Original Data Datagram	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+

Type 11      Code 0 = time to live exceeded in transit;  
                  1 = fragment reassembly time exceeded.

Reassembly vom IP-Fragmentation: IP-Ziel

Type 11 Code 1: Fragment(e) sind nicht im Zeitfenster eingetroffen

## ICMP Router Discovery



Beide Router: Gute Kandidaten für das Default Gateway.  
DHCP gibt nur ein Default GW an.

## ICMP Router Advertisement Message

Wird zeitlich von Routern gesucht oder als Reaktion auf eine Router Solicitation Nachricht

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
<hr/>			
Type	Code	Checksum	
+-----+	+-----+	+-----+	+-----+
Num Addrs	Addr Entry Size   Lifetime		+
+-----+	+-----+-----+	+-----+	Eintrag dieser in Routingtabelle
Router Address[1]			
+-----+	+-----+-----+	+-----+	
Preference Level[1]			Priorität des Routers
+-----+	+-----+-----+	+-----+	wird in Metric angerechnet
Router Address[2]			
+-----+	+-----+-----+	+-----+	
Preference Level[2]			
+-----+	+-----+-----+	+-----+	
.			Angriff: Angreifer bindet
.			sich mit lokaler
.			Priorität an.

### IP Fields:

Source Address	An IP address belonging to the interface from which this message is sent.
Destination Address	The configured AdvertisementAddress or the IP address of a neighboring host.
Time-to-Live	1 if the Destination Address is an IP multicast address; at least 1 otherwise.

### ICMP Fields:

Type	9
Code	0
Checksum	The 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP Type. For computing the checksum, the Checksum field is set to 0.
Num Addrs	The number of router addresses advertised in this message.
Addr Entry Size	The number of 32-bit words of information per each router address (2, in the version of the protocol described here).
Lifetime	The maximum number of seconds that the router addresses may be considered valid.
Router Address[i], i = 1..Num Addrs	The sending router's IP address(es) on the interface from which this message is sent.
Preference Level[i], i = 1..Num Addrs	The preferability of each Router Address[i] as a default router address, relative to other router addresses on the same subnet. A signed, two's-complement value; higher values mean more preferable.

## ICMP Router Solicitation Message

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+			
Type   Code   Checksum			
+-----+-----+-----+-----+			
Reserved			
+-----+-----+-----+-----+			

### IP Fields:

Source Address	An IP address belonging to the interface from which this message is sent, or 0.
Destination Address	The configured SolicitationAddress.
Time-to-Live	1 if the Destination Address is an IP multicast address; at least 1 otherwise.

### ICMP Fields:

Type	10
Code	0
Checksum	see Router Advertisement Message
Reserved	Sent as 0; ignored on reception.

### Übernahme einer TCP-Sitzung:

Soll eine TCP-Verbindung übernommen werden, so muss der Angreifer die richtige Sequenz- und ACK-Nummern verwenden.

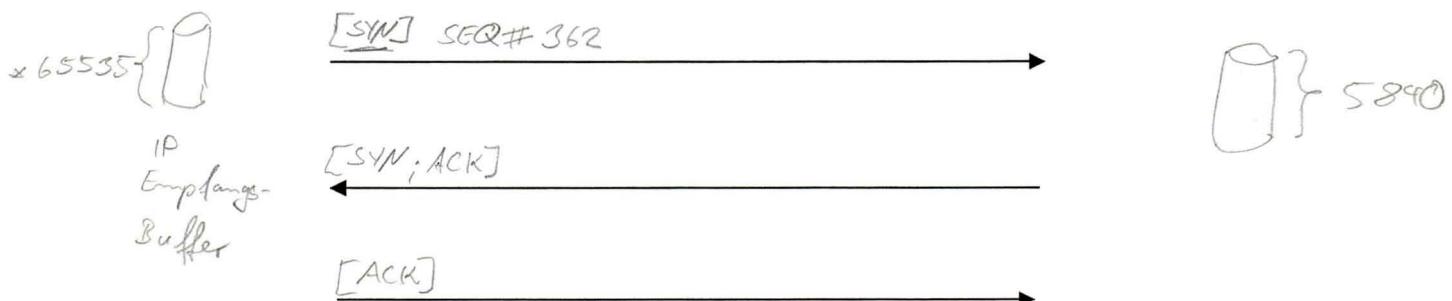
## **TCP/IP - Verbindungen Manipulieren**

## Reguläre TCP - Protokollabläufe

## Verbindungsauflauf

Source	Destination	TCP Ports	Flags	SEQ#	ACK#	Length Data	Window Size
141.69.100.20	141.69.100.101	1978>23	syn	362	ungültig	0	65535
141.69.100.101	141.69.100.20	23>1978	syn, ack	541	363	0	5840
141.69.100.20	141.69.100.101	1978>23	ack	363	542	0	65535

141.69.100.20 Port 1978      141.69.100.101 Port 23



Flaas:

SYN: Es wird das 1. Mal die SEQ # gesendet

[ACK]: ACK# ist gültig; ACK# enthält die nächste erwartete SEQ# des Partners. Wird beim Verbindungs- aufbau um 1 hochgezählt obwohl keine Daten gesendet wurden (Data Length = 0)

WindowSize gibt an: Wie viele Bytes darf der Partner mit pipelining senden bis er auf eine Quittung warten muss.

Stop & Go: sending  
Quitting  
←  
Receiving  
→  
Quitting  
←

Pipelining: Sequencing  
→ →  
Sampling  
←

Verbindungsamtlan: 3-Way Handshake

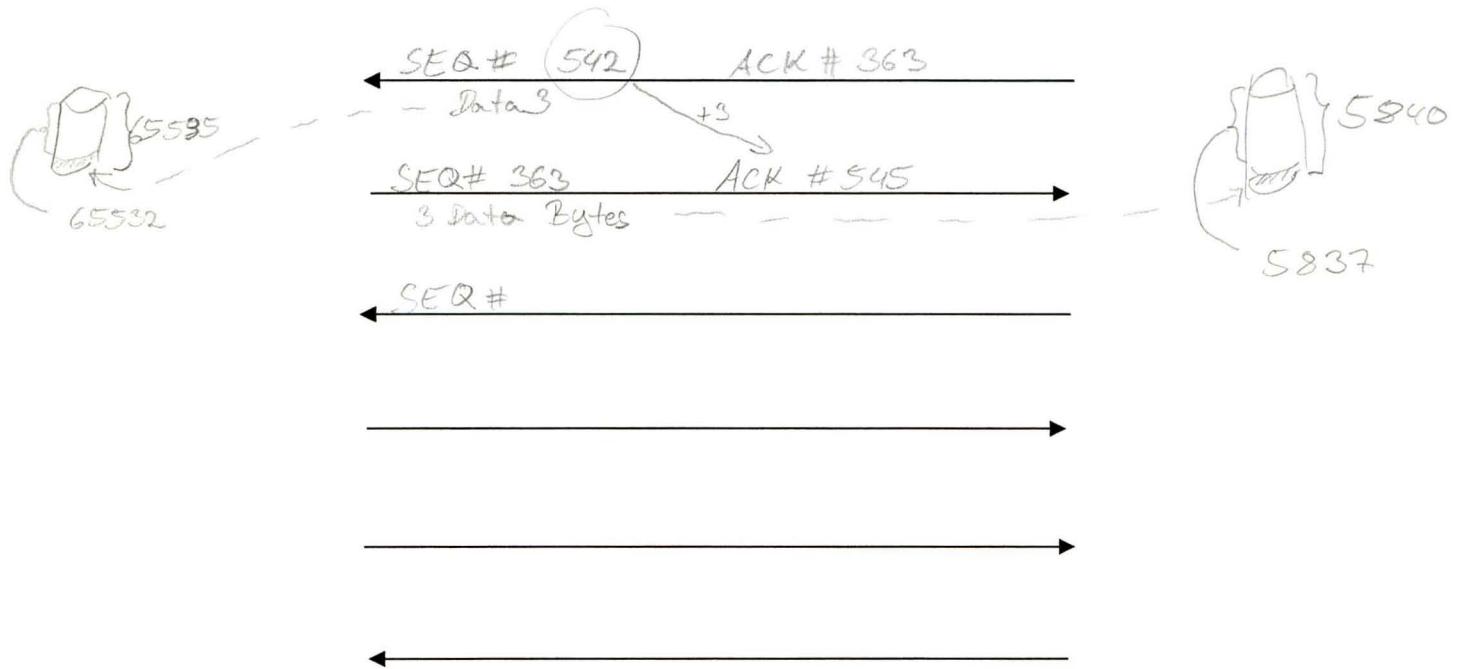
Syn Flood: Angreifer (client) unterschlägt das 3. Datenspaket des 3-Way Handshake [ACK] an den Server. Server speichert die falsche offene Verbindungen und belegt Ressourcen  
viele offene Ressourcen → Buffer voll → Dienst nicht verfügbar  $\Rightarrow$  DoS

## Austausch von Daten

Source	Destination	TCP Ports	Flags	SEQ#	ACK#	Length Data	Window Size
141.69.100.101	141.69.100.20	23>1978	ack	542	363	3	5840
141.69.100.20	141.69.100.101	1978>23	ack	363	545	3	65532
141.69.100.101	141.69.100.20	23>1978	ack	545	366	28	5837
141.69.100.20	141.69.100.101	1978>23	ack	366	573	0	65504
141.69.100.20	141.69.100.101	1978>23	ack	366	573	1	65504
141.69.100.101	141.69.100.20	23>1978	ack	573	367	0	5836

141.69.100.20  
Port 1978

141.69.100.101  
Port 23



$$\text{SEQ\#}_p + \text{Anzahl Datenbytes} = \text{ACK\#}$$

$$\text{WinSize}_{\text{new}} = \text{WinSize}_{\text{alt}} - \text{Anzahl Datenbytes}$$

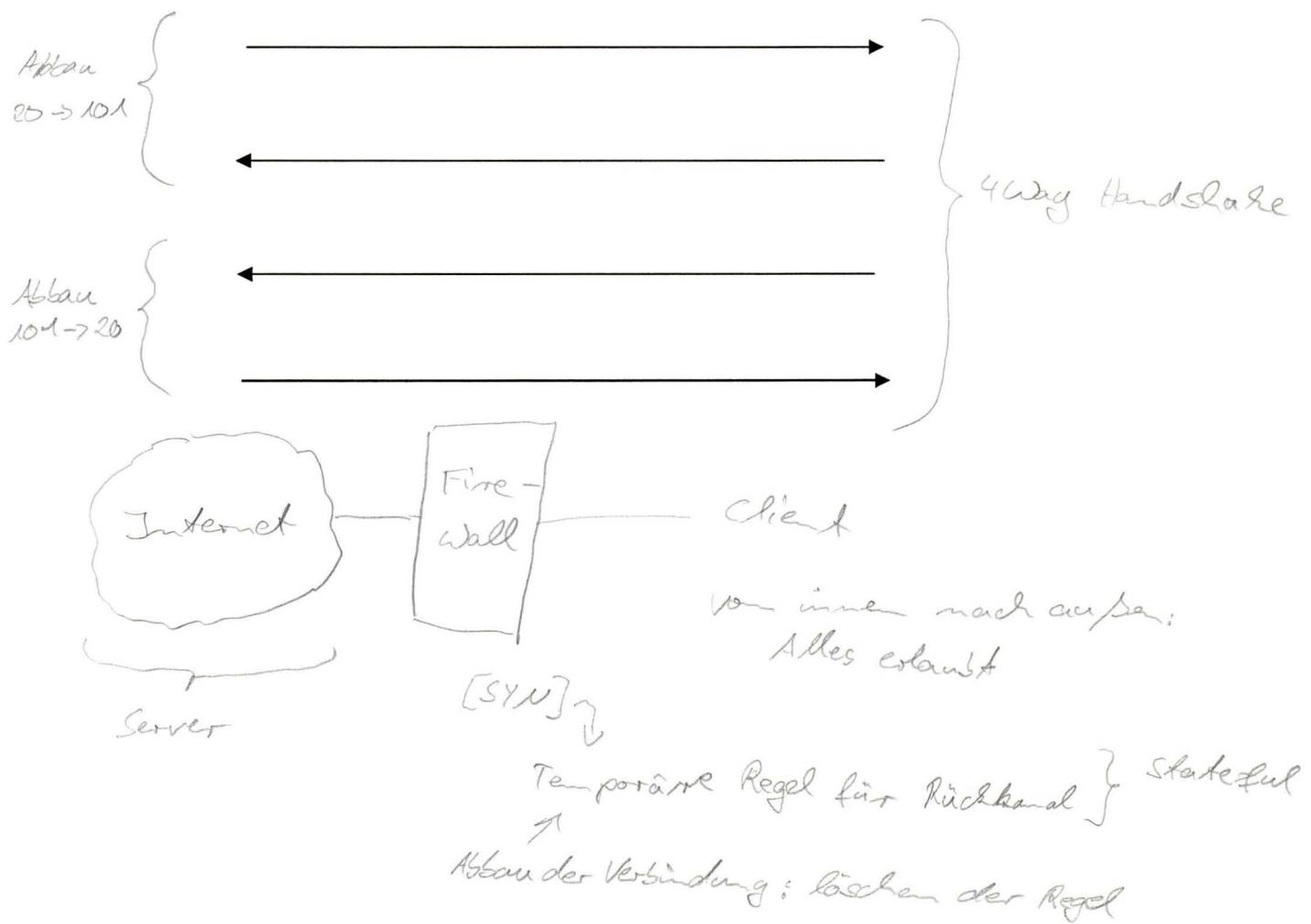
wenn der Buffer nicht geleert wird.

## Verbindungsabbau

Source	Destination	TCP Ports	Flags	SEQ#	ACK#	Length Data	Window Size
141.69.100.20	141.69.100.101	1978>23	fin, ack	388	593	0	65484
141.69.100.101	141.69.100.20	23>1978	ack	593	389	0	5815
141.69.100.101	141.69.100.20	23>1978	fin, ack	593	389	0	5815
141.69.100.20	141.69.100.101	1978>23	ack	389	594	0	65484

141.69.100.20  
Port 1978

141.69.100.101  
Port 23

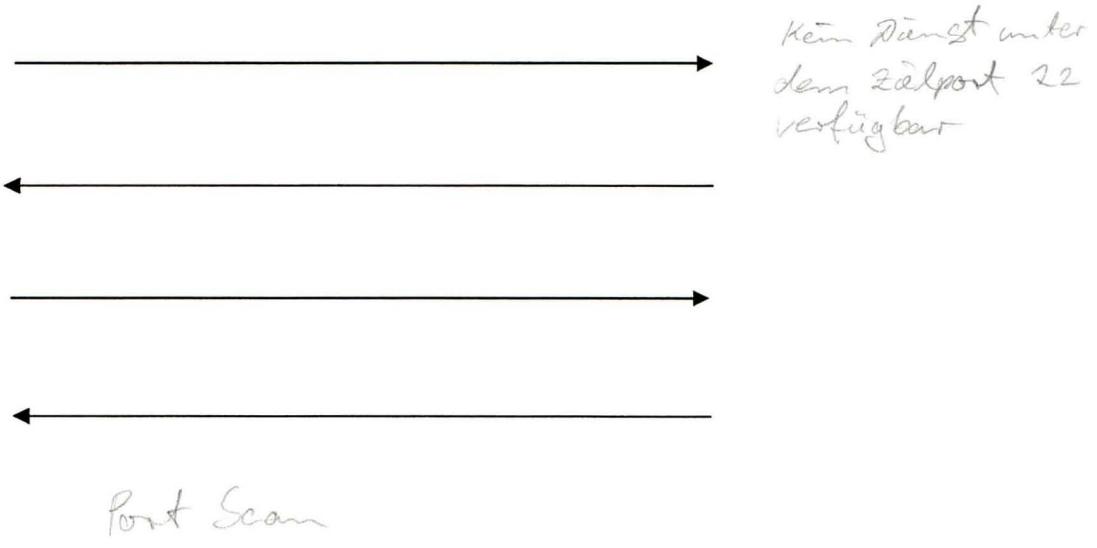


## Rücksetzen einer Verbindung

Source	Destination	TCP Ports	Flags	SEQ#	ACK#	Length Data	Window Size
141.69.100.20	141.69.100.101	3251>22	syn	685	ungültig	0	65535
141.69.100.101	141.69.100.20	22>3251	rst,ack	0	686	0	0
141.69.100.20	141.69.100.101	3251>22	syn	685	ungültig	0	65535
141.69.100.101	141.69.100.20	22>3251	rst,ack	0	686	0	0

141.69.100.20  
Port 1978

141.69.100.101  
Port 23

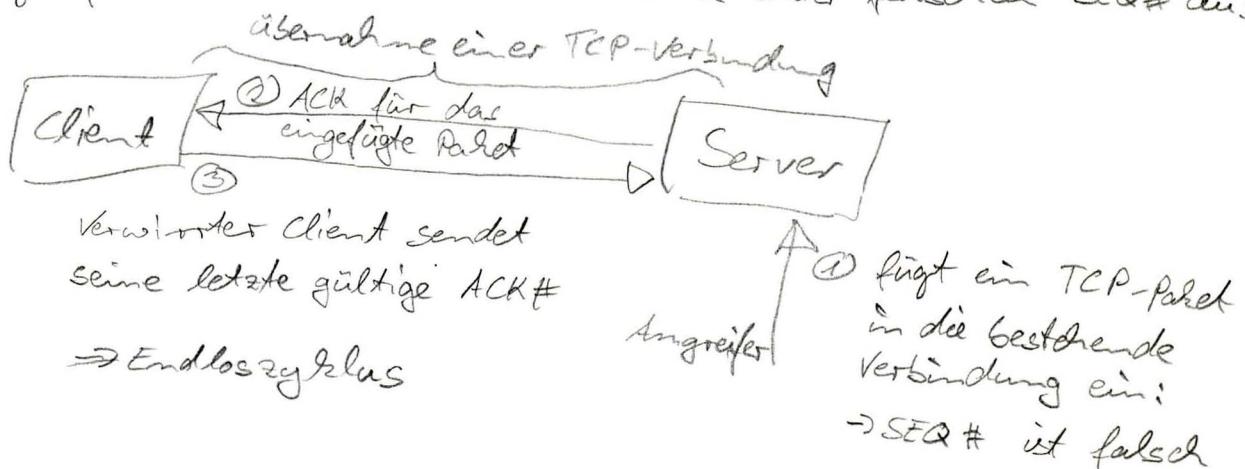


## Gestörte TCP - Protokollabläufe

### TCP ACK-Storm

Im Verlauf eines TCP Hijackings kann es zu TCP ACK-Storms kommen.

→ Angreifer löst ihn durch das Senden einer falschen SEQ# aus.



## SYN- Flooding

SYN -Flooding kann verwendet werden um (Distributed) Denial-of-Service -Attacken durchzuführen. Für DDoS -Attacken werden fernsteuerbare Rechner verwendet, die vorher durch einen Wurm oder Trojaner mit Botnet-SW infiziert worden sind.

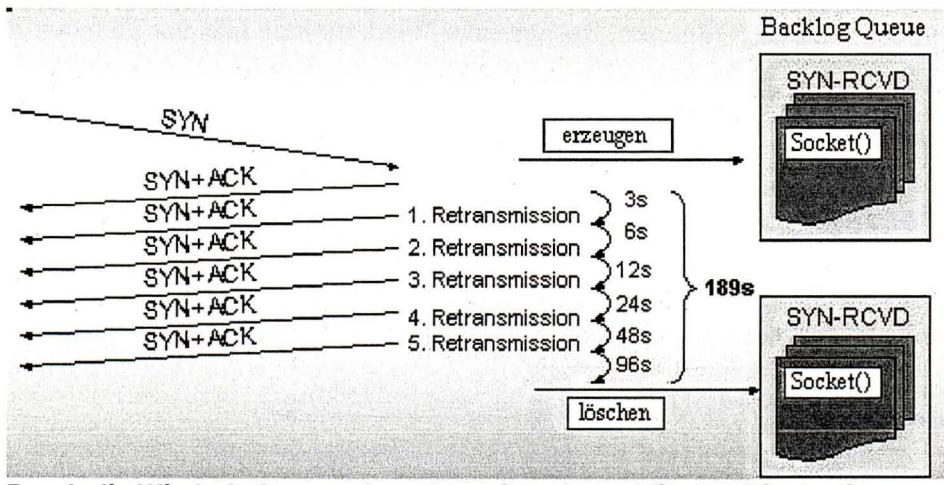
Wikipedia:

Unter einem **Botnet** oder **Bot-Netz** (die Kurzform von *Roboter-Netzwerk*) versteht man ein fernsteuerbares Netzwerk (im Internet) von PCs, das aus untereinander kommunizierenden Bots besteht. Diese Kontrolle wird durch Würmer bzw. Trojanische Pferde erreicht, die den Computer infizieren und dann auf Anweisungen warten. Diese Netzwerke können für Spam-Verbreitung, Denial-of-Service-Attacken und weitere illegale Aktionen verwendet werden, zum Teil ohne dass die betroffenen PC-Nutzer etwas davon erfahren.

Beim SYN-Flooding versucht ein Angreifer die Systemressourcen des attackierten Servers zu blockieren. Dazu versendet er SYN-Pakete, die dem Server den Aufbauwunsch einer TCP-Verbindung signalisieren. Der Server legt daraufhin Datenstrukturen für diese Verbindung an (Bufferplatz ...) und sendet an den Client ein SYN/ACK Paket. Damit ist diese Verbindung halb offen. Im Normalfall sendet daraufhin der Client ein abschließendes ACK-Paket und beendet damit den bidirektionalen TCP- Verbindungsauftbau. Beim SYN- Flooding wird dieses Paket jedoch nicht gesendet. Damit hängt der Server in der halboffenen Verbindung. Er wiederholt das SYN/ACK-Paket (Retransmission) in der Annahme, dass das erste SYN/ACK Paket verloren gegangen ist. Verbindungsanfragen werden von Servern für jeden Dienst in der entsprechenden **Backlog-Queue** abgelegt. Ist dieser Puffer gefüllt, so werden keine neuen Verbindungen mehr angenommen. Damit ist der Dienst, den der Server anbietet, nicht mehr erreichbar.

Heise Security

Bis der Server einen einmal angelegten Eintrag in der Backlog-Queue wieder löscht, weil er keine Antwort bekommt, können mehrere Minuten vergehen. Nach einem ersten Timeout, typischerweise nach 3 Sekunden, nimmt der Server an, sein SYN/ACK-Paket sei verloren gegangen und schickt es erneut auf die Reise. Dieser Vorgang wiederholt sich mit immer längeren Timeouts mehrfach (Linux: 5 Retransmissions). Auf einem Standard-Linux-System bietet die Backlog-Queue Platz für 256 solcher halboffenen Verbindungen. Der Angreifer hat also mehr als genug Zeit, diese mit seinen Einträgen zu füllen.



Durch die Wiederholungen dauert es oft mehrere Minuten bis der Server einen Eintrag in der Backlog-Queue wieder löscht

In den meisten Fällen verwendet der Angreifer gefälschte Absenderadressen für seine SYN-Anfragen. Damit bekommt er die Antwortpakete des Servers zwar nicht zu sehen, aber da er ohnehin nicht vorhat, ihren Erhalt zu bestätigen, stört ihn das nicht. Durch geschicktes Verteilen der Adressen kann er ein Filtern der Angriffspakete verhindern. Denn ein vorgeschalteter Router kann die gefälschten Pakete nicht von echten Verbindungswünschen unterscheiden.

Benutzt ein Rechner eine der für den Angriff genutzten Absenderadressen, erhält er vom angegriffenen Server plötzlich ein Bestätigungs-Paket (SYN/ACK), das er nicht angefordert hatte. Er reagiert auf das offensichtliche Missverständnis mit einer Aufforderung die Verbindung zu verwerfen (Reset, RST). Das führt dazu, dass der Server seinen Backlog-Eintrag löscht. Um das zu vermeiden, verwenden Syn-Flooder bevorzugt Adressen, die zum Zeitpunkt des Angriffs nicht belegt sind. So bekommt der Server keine Antwort auf sein SYN/ACK-Paket und wiederholt die volle Prozedur von Warten und erneutem Senden mehrfach, bevor er den Backlog-Eintrag freigibt.

Backlog - Queue voll  
↓

### Gegenmaßnahme: SYN-Cookies

SYN-Cookies dienen als Rückfallstrategie wenn die Backlog-Queue voll ist. Die Clients merken nichts von dieser Vorgehendweise eines Servers. Die Server speichern in diesem Fall keine Informationen zur Verbindung. Sie senden die Informationen als Cookie an den Client zurück. Die TCP-Sequenznummer dient als Träger für diese Cookies. Der Server erstellt aus Quell- und Zielports, den zugehörigen IP-Adressen und einem Geheimnis einen MD5-Hash und sendet ihn als Cookie an den Client. Kommt das dritte Paket eines Verbindungsaufbaus -ein ACK-Paket- vom Client zurück, so enthält es das zuvor vom Server gesendete Cookie in der Acknowledgement-Nummer. Der Server bildet erneut den MD5-Hash über die im Paket enthaltenen Adressen und Ports zusammen mit dem Geheimnis. Stimmt dieser Wert mit der ACK-Nummer überein, stellt der Server die Verbindung her.

### Gegenmaßnahme: TCP- Parameter ändern

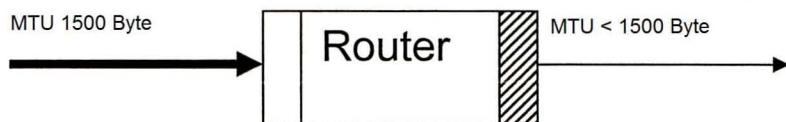
Als eine Gegenmaßnahme kann man die Größe der Backlog-Queue erhöhen. Ferner kann man die Anzahl der Versuche SYN/ACK zu senden herabsetzen. Bei 5 Wiederholungen ergibt sich

bei Linux ein Timeout von ca. 3 Minuten. Bei nur einer Wiederholung fällt das Timeout auf 9 Sekunden.

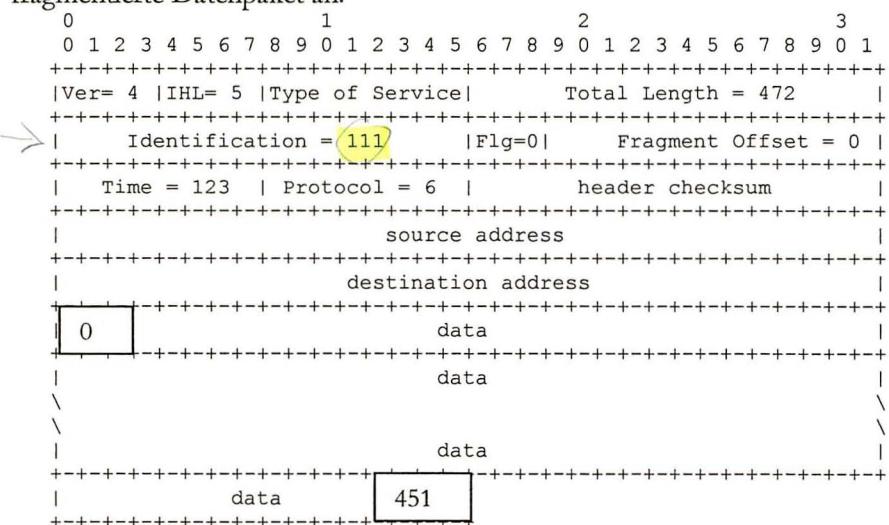
## Angriffe auf IP

### IP Fragmentierung

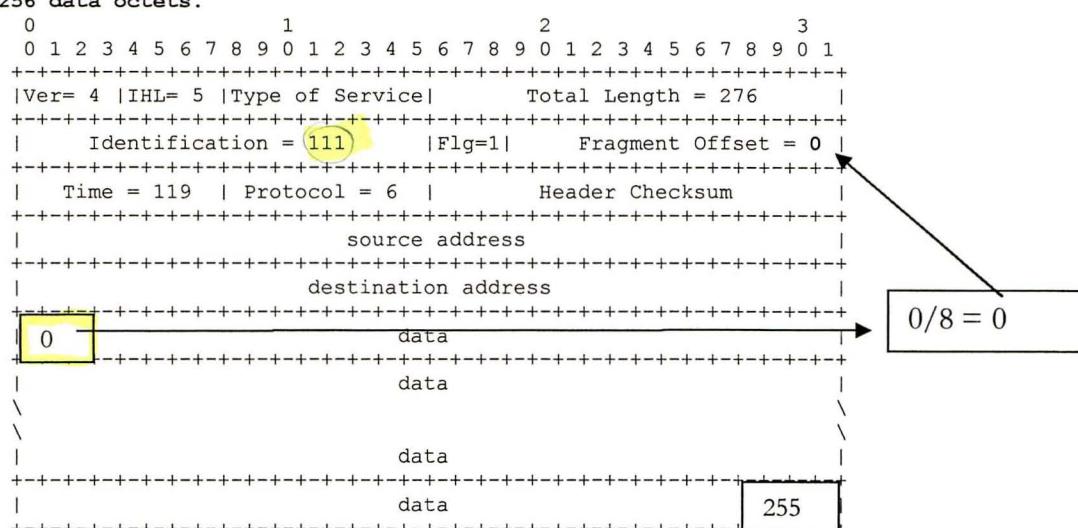
Fragmentierung wird benötigt, wenn ein Router ein empfangenes Datenpaket in kleinere Pakete aufteilen muss, weil die Technologie, über die das Paket weiter versendet werden muss, nur eine kleinere MTU unterstützt, als die Technologie mit der das Datenpaket empfangen wurde.



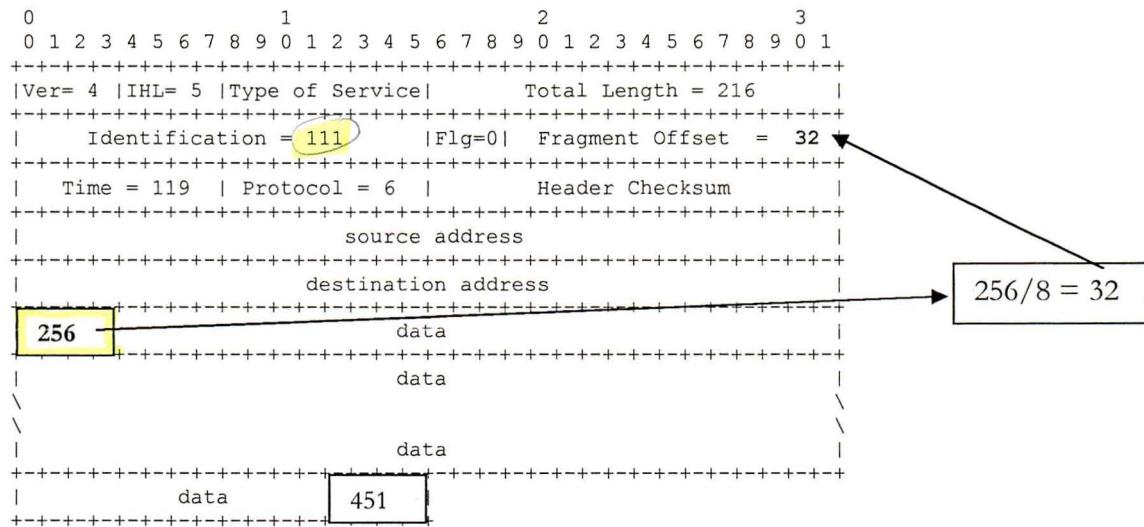
In diesem Beispiel betrachten wir einen Router, der zwei verschiedene Netzwerkkarten besitzt. Die linke Netzwerkkarte kann IP- Pakete mit einer Größe von 1500 Bytes(Ethernet) transportieren, die rechte Netzwerkkarte kann hingegen nur IP- Pakete mit einer maximalen Größe von 280 Bytes transportieren. Der Router empfängt im Beispiel nun ein IP- Paket mit einer Anzahl von 472 IP- Bytes. Dieses Paket muss er in kleinere Datenpakete aufteilen, damit er sie über sein rechtes Interface weiterversenden kann. Die IP- Nutzbytes werden durchnummieriert von 0 bis 451, 20 Bytes entfallen auf den IP- Header. Der Fragment Offset gibt den durch 8 dividierten Offset des 1. Bytes des Fragmentes, in Bezug auf das nicht fragmentierte Datenpaket an.



Now the first fragment that results from splitting the datagram after 256 data octets.



And the second fragment.



## Angriffe, die die Fragmentierung nutzen

Für die Policy einer Firewall gilt häufig, dass aus dem Intranet Clients zu beliebigen Servern im Internet TCP-Verbindungen aufbauen dürfen. Clients aus dem Internet dürfen hingegen nur zu ausgewählten Servern des Intranet eine Verbindung aufbauen. Für Firewalls ist es also wichtig zu erkennen, wer die betreffende Verbindung aufbaut. Sie erkennen den Verbindungsaufbau am ersten Datenpaket einer TCP-Verbindung. Bei diesem Paket ist das SYN-Flag auf 1 und das ACK-Flag auf 0 gesetzt. Bei allen folgenden Paketen ist das ACK-Flag immer auf 1 gesetzt. Die Grundidee dieser Angriffe besteht darin, dass wenn das erste Datenpaket einer Verbindung die Firewall passiert, dass diese Firewall dann alle weiteren Pakete dieser Verbindung ebenfalls durchlässt. Beim ersten Datenpaket muss also verhindert werden, dass das ACK-Flag den Wert 0 annimmt.

Bei der ersten Version dieser Attacke ist das 1. Fragment so klein, dass es nicht die TCP-Flags enthält. Bei der 2. Version dieses Angriffstyps überschneiden sich die ersten beiden Fragmente in dem Bereich in dem sich die TCP-Flags befinden. Beim ersten Fragment ist das ACK-Flag auf 1 gesetzt, beim 2. Fragment ist es auf 0 gesetzt. Das 2. Fragment überschreibt beim 1. Fragment auch den Bereich in dem sich die Flags befinden. Fragmente werden erst beim Empfänger wieder zusammengebaut.

## Tiny Fragments

### 1. Fragment

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Ver= 4  IHL= 5  Type of Service	Total Length = 28		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Identification = 111   Flg=1	Fragment Offset = 0		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Time = 123   Protocol = 6	header checksum		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
source address			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
destination address			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
<b>more fragments</b>			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Source Port	Destination Port		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Sequence Number			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+

### 2. Fragment

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	2	3	
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
Ver= 4  IHL= 5  Type of Service	Total Length = 472		
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
Identification = 111   Flg=0	Fragment Offset = 1		
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
Time = 123   Protocol = 6	header checksum		
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
source address			
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
destination address			
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
<b>Acknowledgment Number</b>			
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
Data	U A P R S F		
Offset  Reserved  R C S S Y I	Window		
	G K H T N N		
	0     1		
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
Checksum	Urgent Pointer		
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
/ Options	/ Padding		/
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+
/ data			/
+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+	+-----+-----+-----+-----+-----+-----+-----+-----+

## *Overlapping Fragments*

## 1. Fragment

```

          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Ver= 4 |IHL= 5 |Type of Service|           Total Length = 472 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Identification = 111 |Flg=1|      Fragment Offset = 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Time = 123 | Protocol = 6 |           header checksum |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           source address |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           destination address |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           Source Port |           Destination Port |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           Sequence Number |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           Acknowledgment Number |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Data |           |U|A|R|S|F|
| Offset| Reserved |R|C|S|S|Y|I|           Window
|       |           |G|K|H|T|N|N|
|       |           | |1| | |0| |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           Checksum |           Urgent Pointer |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           Options |           / Padding |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|           data |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

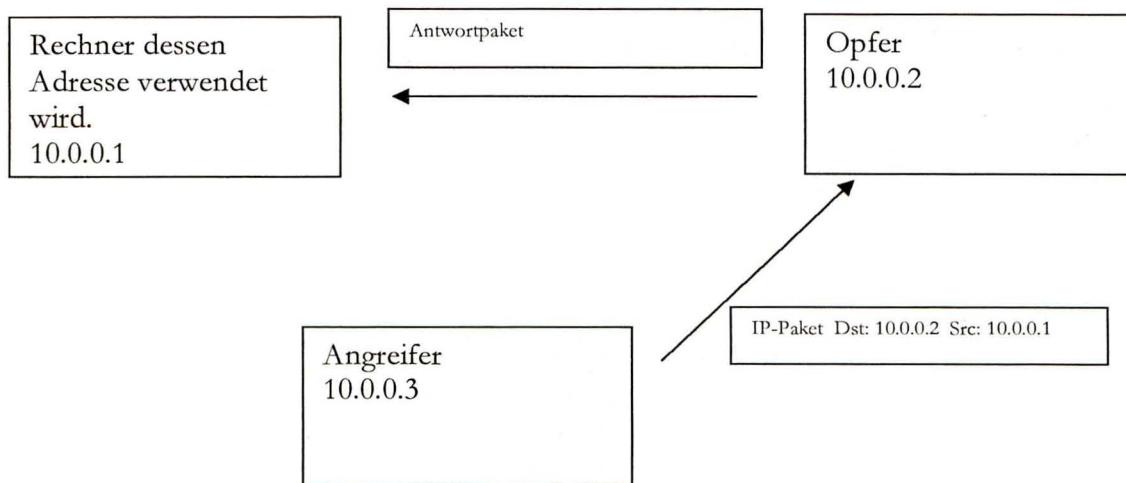
## 2. Fragment

## IP Spoofing

Bei diesen Angriffen verwendet der Angreifer die IP-Adresse eines anderen Rechners. Das Ziel dieses Angriffs ist entweder die Herkunft des Angriffs zu verschleiern, oder durch die Nutzung einer fremden Adresse die Zugriffsschranken auf den angegriffenen Rechnern zu umgehen. Linux TCP-Wrapper oder der Windows IIS (Internet Information Server) verwenden z.B. Access Lists, die auf IP-Adressen basieren.

## Blind Spoofing

Diese Art der Angriffe wird häufig auch als "Blind Spoofing" bezeichnet weil, die Antwortpakete auf Grund des Routings nicht beim Angreifer ankommen.



Der erste bekannte Angriff dieser Art (Blind Spoofing) gegen Tsutomu Shimomura wurde 1994 beschrieben. Shimomura verwendete einen TCP-Wrapper um sein UNIX-System vor unberechtigten Zugriffen zu schützen. Aber der Angreifer konnte einen erfolgreichen Einbruch starten, weil er die Sequenznummer erriet, die in den Antwortpaketen gesendet wurden. Mit diesem Angriff konnte die Konfiguration des UNIX-Systems geändert werden.

*Sequence number attacks sind sehr viel unwahrscheinlicher geworden weil OS-Hersteller die Verfahren geändert haben mit denen Sequence numbers erzeugt werden. Die überholte alte Vorgehensweise addierte einen konstanten Wert, um die nächste Sequence number einer neuen Verbindung zu erhalten.*

Die Verwendung der Absender-IP-Adresse für eine Zugriffskontrolle ist nicht sicher (IP-Spoofing).

Typische Programme: rexec, rlogin, r...

Heute: SSH

SNMP/SNMP2 verwenden immer noch die IP-Adressen als Zugriffskontrolle  
(SNMP = Simple Network Management Protocol)

### Ablauf eines IP-Spoofing-Angriffs

- \* Bestimmung der IP-Adresse, die beim Opfer Vertrauen besitzt (10.0.0.1)
- \* Diese Maschine (10.0.0.1) durch DoS lahmlegen
- \* Raten der Sequenznummer des Opfers (Rückpakte z.B. TCP-Quittungen werden nicht empfangen)
- \* z.B. rsh echo ++ >> /rhosts

## Schutzmaßnahmen basierend auf IP-Adressen

### xinetd/inetd

Extended inetd

Diese Programme werden auch als Superserver bezeichnet. Sie sollen bei Servern die Ressourcen schonen indem Sie nur dann Dienstprogramme starten, falls ein Verbindungswunsch vorliegt. Die andere nicht Ressourcen sparende Vorgehensweise startet Dienst beim Hochfahren des Servers (Vorteil kurze Antwortzeiten). Die Verbindungswünsche werden vom inetd or xinetd entgegengenommen, diese starten dann den angefragten Dienst.

### xinetd

Xinetd erweitert die Möglichkeiten von inetd :

- access control
- erweitertes Logging
- Zeitbeschränkungen für Dienste
- Begrenzung der Anzahl der Server-Prozesse
- Umleiter von TCP-Verbindungen (-> Rechner / Port)
- Binden von Diensten auf bestimmte Interfaces
- verwendeten Arbeitsspeicher begrenzen
- CPU-Zeit beschränken

man xinetd.conf

```

#
# Sample configuration file for xinetd
#
defaults
{
    log_type      = FILE /var/log/service.log
    log_on_success = PID
    log_on_failure = HOST
    only_from     = 128.138.193.0 128.138.204.0
    only_from     = 128.138.252.1
    instances     = 10
    disabled      = rstatd
}

#
# Note 1: the protocol attribute is not required
# Note 2: the instances attribute overrides the default
#
service login
{
    socket_type   = stream
    protocol      = tcp
    wait          = no
    user          = root
    server        = /usr/etc/in.rlogind
    instances     = UNLIMITED
}

#
# Note 1: the instances attribute overrides the default
# Note 2: the log_on_success flags are augmented
#
service shell
{
    socket_type   = stream
    wait          = no
    user          = root
    instances     = UNLIMITED
    server        = /usr/etc/in.rshd
    log_on_success += HOST
    = default + ...
}

service ftp
{
    socket_type   = stream
    wait          = no
    nice          = 10
    user          = root
    server        = /usr/etc/in.ftp
    server_args   = -l
    instances     = 4
    log_on_success += DURATION HOST USERID
    access_times  = 2:00-9:00 12:00-24:00
}

# Limit telnet sessions to 8 Mbytes of memory and a total
# 20 CPU seconds for child processes.
service telnet
{
    socket_type   = stream
    wait          = no
    nice          = 10
    user          = root
    server        = /usr/etc/in.telnetd
    rlimit_as     = 8M
    rlimit_cpu    = 20
}

#
# This entry and the next one specify internal services. Since
# this is the same service using a different socket type, the
# id attribute is used to uniquely identify each entry
#
service echo
{
    id            = echo-stream
    type          = INTERNAL
}

```

```
        socket_type      = stream
        user            = root
        wait           = no
    }

    service echo
    {
        id              = echo-dgram
        type            = INTERNAL
        socket_type     = dgram
        user            = root
        wait           = no
    }
#
# Sample RPC service
#
service rstatd
{
    type            = RPC
    socket_type     = dgram
    protocol        = udp
    server          = /usr/etc/rpc.rstatd
    wait            = yes
    user            = root
    rpc_version     = 2-4
    env             = LD_LIBRARY_PATH=/etc/securelib
}

#
# Sample unlisted service
#
service unlisted
{
    type            = UNLISTED
    socket_type     = stream
    protocol        = tcp
    wait           = no
    server          = /home/user/some_server
    port            = 20020
}
```

← nur lokal benutzen

} eigener Server

## inetd

inetd beruht die Sicherheitsmechanismen des Unix-Netz nicht.

Zooplankton - 2-3 metres

## TCP-Wrapper

#### Inhalt der Datei inetd.conf:

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard           stream  tcp      nowait  root    internal
#discard           dgram   udp      wait    root    internal
#daytime           stream  tcp      nowait  root    internal
#time              stream  tcp      nowait  root    internal
telnet            stream  tcp      nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
```

 Dienstname  
 Socket type  
etc/services

Bearbeitung zu starken  
Unterschieden Prozess, ruft  
Dient lange Zeitverzögerungen hervor

## TCP-Wrapper

Ein TCP- Wrapper soll Dienste schützen, indem Zugriffe nur von bestimmten IP-Adressen oder IP- Adressbereichen erfolgen und er soll einkommende Dienstanforderungen überwachen (log).

**Tcpd** ist ein TCP-Wrapper. Die Zugriffskontrolle für Dienste, die durch den Wrapper geschützt werden sollen, erfolgt über die Dateien hosts.allow und hosts.deny.

```
hosts.allow:
#
ALL: 141.69.100.
ALL: 127.0.0.1
```

```
hosts.deny:
#
ALL: ALL
```

### ACCESS CONTROL FILES

The access control software consults two files. The search stops at the first match:

- Access will be granted when a (daemon,client) pair matches an entry in the /etc/hosts.allow file.
- Otherwise, access will be denied when a (daemon,client) pair matches an entry in the /etc/hosts.deny file.
- Otherwise, access will be granted.

A non-existing access control file is treated as if it were an empty file. Thus, access control can be turned off by providing no access control files.

### ACCESS CONTROL RULES

Each access control file consists of zero or more lines of text. These lines are processed in order of appearance. The search terminates when a match is found.

- A newline character is ignored when it is preceded by a backslash character. This permits you to break up long lines so that they are easier to edit.
- Blank lines or lines that begin with a # character are ignored. This permits you to insert comments and whitespace so that the tables are easier to read.
- All other lines should satisfy the following format, things between [] being optional:

```
daemon_list : client_list [ : shell_command ]
```

daemon\_list is a list of one or more daemon process names (argv[0] values) or server port numbers or wildcards (see below).

client\_list is a list of one or more host names, host addresses, patterns or wildcards (see below) that will be matched against the client host name or address.

Mit **tcpdmatch** kann man die Zugriffsregeln testen, die in hosts.allow und hosts.deny abgelegt sind.

```
VMatch1:/etc# tcpdmatch in.telnetd 141.69.1.1
client: address 141.69.1.1
server: process in.telnetd
matched: /etc/hosts.deny line 20
access: denied

VMatch1:/etc# tcpdmatch in.telnetd 141.69.100.1
client: address 141.69.100.1
server: process in.telnetd
matched: /etc/hosts.allow line 14
access: granted
```

Was ist ein **TCP-Wrapper**? → Zugriffsbegrenzung aufgrund von IP-Adressen.

inetd wartet selbstverstndlich auf Verbindungsanwnde.

Inhalt der Datei `inetd.conf`:

`/etc/services ... port/main xx`

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard          stream  tcp    nowait  root   internal
#discard          dgram   udp    wait    root   internal
#daytime          stream  tcp    nowait  root   internal
#time             stream  tcp    nowait  root   internal
telnet            stream  tcp    nowait  telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

Annotations:

- xx: Bezeichnung für den Dienst
- vertndungsorientierter Socket: Beschreibung des Socketypräzess
- es werden so viele Verbindungs-Wünsche akzeptiert, wie im Wert für das Backlog-Antritt angesetzt ist: Anzahl der Anträge, die TCPd akzeptieren kann
- Benutzer unter dem der Prozess läuft: Benutzer, unter dem der Prozess läuft
- Pfad des zu startenden Programms: Pfad des zu startenden Programms
- Parameter des den Programms übergeben wird: Parameter, die dem Programm übergeben werden

**Tcpd** ist ein TCP-Wrapper. Die Zugriffskontrolle erfolgt über die Dateien `hosts.allow` und `hosts.deny`.

```
hosts.allow:
#
# ALL: 141.69.100.
ALL: 127.0.0.1
```

erlaubte Dienste

```
hosts.deny:
#
# ALL: ALL
```

Mit **tcpdmatch** kann man die Zugriffsregeln testen, die in `hosts.allow` und `hosts.deny` abgelegt sind.

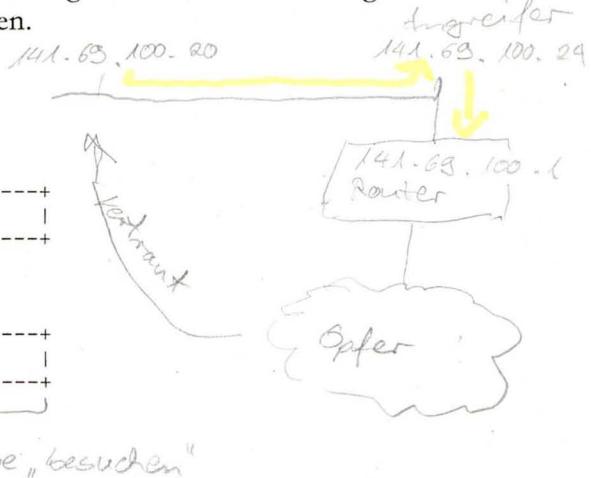
```
VMetch1:/etc# tcpdmatch in.telnetd 141.69.1.1
client: address 141.69.1.1
server: process in.telnetd
matched: /etc/hosts.deny line 20
access: denied
```

```
VMetch1:/etc# tcpdmatch in.telnetd 141.69.100.1
client: address 141.69.100.1
server: process in.telnetd
matched: /etc/hosts.allow line 14
access: granted
```

## Verwendung der Source Route Option

→ Der Angreifer hört die Antwortpakete ab

Beim Blind Spoofing kommen die Antwortpakete beim Angreifer nicht an. Die Angreifer versuchen daher die IP-Option Source Route zu nutzen.



### IP-Source Route Optionen

#### Loose Source and Record Route

```
+-----+-----+-----+-----+-----+
|10000011| length | pointer| route data |
+-----+-----+-----+-----+
Type=131
```

#### Strict Source and Record Route

```
+-----+-----+-----+-----+
|10001001| length | pointer| route data |
+-----+-----+-----+-----+
Type=137
```

### Ping kann Source Routes vorgeben

Eine "Loose Source Rout" oder eine "Strict Source Route" kann mit Hilfe des Ping-Kommandos angegeben werden. Im folgenden Beispiel wird eine LooseSource Route vorgegeben.

**ping -j 141.69.100.24 141.69.100.1 141.69.1.1**

Wireshark screenshot showing two ICMP Echo Request frames. Both frames have a Source of 141.69.100.20 and a Destination of 141.69.100.24. The second frame's details pane shows a loose source route option:

- version: 4
- Header length: 32 bytes
- Type of service: 0x00 (None)
- Total Length: 72
- Identification: 0x9919 (39193)
- Flags: 0x00
- Fragment offset: 0
- Time to live: 128
- Protocol: ICMP (0x01)
- Header checksum: 0xa759 [correct]
- Source: 141.69.100.20 (141.69.100.20)
- Destination: 141.69.100.24 (141.69.100.24)
- options: (12 bytes)
  - Loose source route (11 bytes)
    - Pointer: 4
    - 141.69.100.1 ← (current)
    - 141.69.1.1

**echo 1 >> /proc/sys/net/ipv4/ip\_forward**

- Die Angabe der Route wird
- auch für den Rückweg benötigt

Die **Antwortpakete** auf ein "Loose Source Route"- Paket nehmen ebenfalls den vorgegebenen Pfad in der umgekehrten Reihenfolge.

Um solche Angriffe abzuwehren verwirft fast jede Firewall Pakete die diese Optionen enthalten. Wrapper und viele OS können ebenfalls source-routed Pakete blockieren.

Die meisten Firewalls verwerfen IP- Pakete aus dem Internet, die als IP-Quelladresse eine Intranetadresse verwenden, um IP- Spoofing zumindest aus dem Internet zu verhindern.

## Routingtabellen manipulieren

Damit ein Angreifer die Antwortpakete erhält, kann er versuchen die Routingtabellen der beteiligten Router zu verändern. Dies kann er zum Beispiel durch gefälschte Pakete eines Routingprotokolls realisieren.

## Bewertung der IP-Spoofing Attacken

- Source routing und Manipulation der Routingtabellen:  
Schwer zu nutzen } warum?
- Blind Spoofing:  
wendet sich gegen Dienste (rlogin; rsh; rcp; ...)  
die die IP-Adresse des Client als  
Zugriffskontrolle verwendet }  
+ - Dienste
- R-Dienste durch ssh (Putty) Remote clients abgelöst

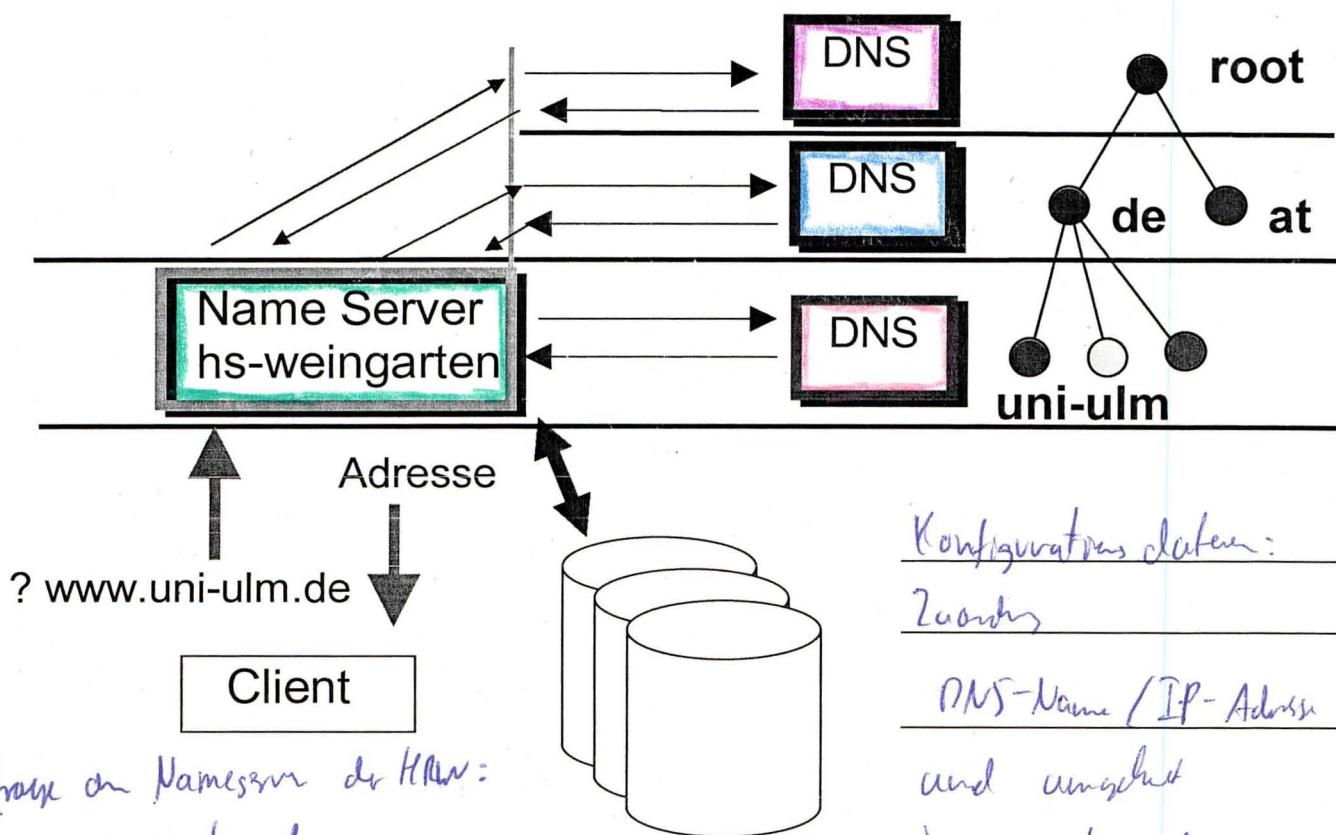
## Angriffe auf die Namensauflösung

### Reguläre DNS-Anfragen

Erhält ein „Domain Name Server“ (DNS) eine Anfrage von einem „Name Resolver Process“, prüft er, ob der gesuchte Name in der Domain liegt, für die er zuständig ist.

- Falls dem so ist, wird der Name in eine Adresse übersetzt.
- Wenn er nicht für den Namen zuständig ist, prüft er, welche Art der Interaktion angefragt wurde.
  - a) vollständige Übersetzung (recursive resolution) In diesem Fall nimmt der befragte Nameserver Verbindung mit dem Nameserver auf, der die Anfrage beantworten kann.
  - b) (nonrecursive resolution) DNS kann den Namen nicht übersetzen: Er liefert eine Fehlermeldung zurück.

DNS-Antwort  
aus Cache:  
Nicht autorisiert



Anfrage an Nameserver der HAW:

www.uni-ulm.de.

- www.uni-ulm.de IP-Adresse im DNS-Cache?

ja: Antwort aus Cache → nicht autorisiert

nein: Frage an root-Name-Servi

gib mir DE-NS

Frage an DE-NS

gib mir uni-ulm-NS

Frage an uni-ulm-NS

gib mir www.uni-ulm.de

Konfigurationsdaten:

Zurück

DNS-Name / IP-Adresse

und umgekehrt

hs-weingarten.de

69.161.192.194

CLASS B 192.168.0.0

## Angriffe auf die Namensauflösung

### DNS ID- Fälschung

Im Folgenden sind eine DNS-Anfrage und deren Antwort aufgelistet. In der Anfrage wird eine ID zur Kennzeichnung der Anfrage gesetzt. Diese ID muss in der Antwort ebenfalls verwendet werden.

```
Name: orion2.fhwgt.de
Address: 141.69.46.109
Aliases: www.fhwgt.de

> www.fhwgt.de.
Server: rz-sun1.fh-weingarten.de
Address: 141.69.1.1
-----
SendRequest(), len 30
HEADER:
    opcode = QUERY, id = 6, rcode = NOERROR
    header flags: query, want recursion
    questions = 1, answers = 0, authority records = 0, additional = 0

QUESTIONS:
    www.fhwgt.de, type = A, class = IN
-----
Got answer (234 bytes):
HEADER:
    opcode = QUERY, id = 6, rcode = NOERROR
    header flags: response, auth. answer, want recursion, recursion avail.
    questions = 1, answers = 2, authority records = 4, additional = 4

QUESTIONS:
    www.fhwgt.de, type = A, class = IN
ANSWERS:
-> www.fhwgt.de
    type = CNAME, class = IN, dlen = 9
    canonical name = orion2.fhwgt.de
    ttl = 604800 (7 days)
-> orion2.fhwgt.de
    type = A, class = IN, dlen = 4
    internet address = 141.69.46.109
    ttl = 604800 (7 days)
AUTHORITY RECORDS:
.
.
.
```

#### Bedeutung der Header Flags:

Query: Anfrage an einen NS      response: Antwort von einem NS

Auth. Answer: Antwort kommt nicht aus dem Cache

Want recursion: Server soll rekursiv andere DNS-Server befragen.

Recursion available: NS- kann rekursiv fragen

Rcode: Fehlercode

questions = 1, answers = 2, authority records = 4, additional = 4

Ziel eines DNS-Angriffs ist:

Den Cache eines DNS-Servers zu einem Domain-Namen eine falsche IP-Adresse ablegen.

Dieser Eintrag bleibt für die Dauer die der TTL-Wert im Schl.-angibt im Cache.

Wird der gefälschte DNS-Server zu diesem Domain-Namen gefragt, liefert er den falschen Eintrag aus dem Cache.

Das DNS-Protokoll verwendet UDP für Lookups. Damit eine Antwort akzeptiert wird muss die ID (DNS-Header) von Frage und Antwort übereinstimmen.

DNS dessen Cache vergiftet werden soll

A  
falsche Antwort  
ID  
www.postbank.de  
IP von Postbank.de

Um eine geeignete ID zu ermitteln, führt ein Angreifer die folgenden Aktionen durch.  
Im lokalen Netz muss man dazu die DNS-Anfragen belauschen. Befindet sich der anzugreifende Rechner in einem anderen Subnetz ist es komplizierter.

## Voruntersuchungen zum Verhalten von Namensauflösungen

- 1.) Lassen Sie sich den DNS-Cache von XP anzeigen! (ipconfig /displaydns)
- 2.) Löschen Sie bei XP den DNS-Cache des Rechners! (ipconfig /flushdns)
- 3.) Lassen Sie sich den DNS-Cache nach dem Löschen erneut anzeigen. Welche Einträge hat er jetzt?

4.)

Rufen Sie nslookup auf.

Geben Sie "set d2" ein.

Vergewissern Sie sich, dass als DNS-Server der Server 141.69.1.1 gesetzt ist.

Geben Sie mehrfach www.uni-ulm.de ein.

a.) Erhalten Sie eine autorisierte Antwort?

b.) Welche TTL-Werte werden Ihnen für die IP-Adresse von www.uni-ulm.de angezeigt?

Führen Sie mehrere Anfragen hintereinander durch!

windows 32\etc\hosts

Geben Sie nun mehrfach www.hs-weingarten.de ein.

c.) Erhalten Sie eine autorisierte Antwort?

d.) Welche TTL-Werte werden Ihnen für die IP-Adresse von www.hs-weingarten.de angezeigt?

- 5.) Untersuchen Sie wie bei Ubuntu und Windows XP die IDs bei DNS-Anfragen bei 2 aufeinander folgenden Anfragen gesetzt werden! Verwenden Sie bei Windows nslookup und bei ubuntu dig mit der Option +qr.

## Schlussfolgerungen

6.)

a.) Was speichert ein Nameserver in seinem Namecache?

b.) Woran kann man bei einer Antwort von einem Nameserver erkennen, dass es sich um eine Antwort aus dem Cache handelt?

c.) Wie kann man den Namecache eines Nameservers vergiften?

d.) Was begünstigt den Angriff auf einen Nameserver?

## Ablauf eines DNS-Angriffs (DNS cache poisoning)

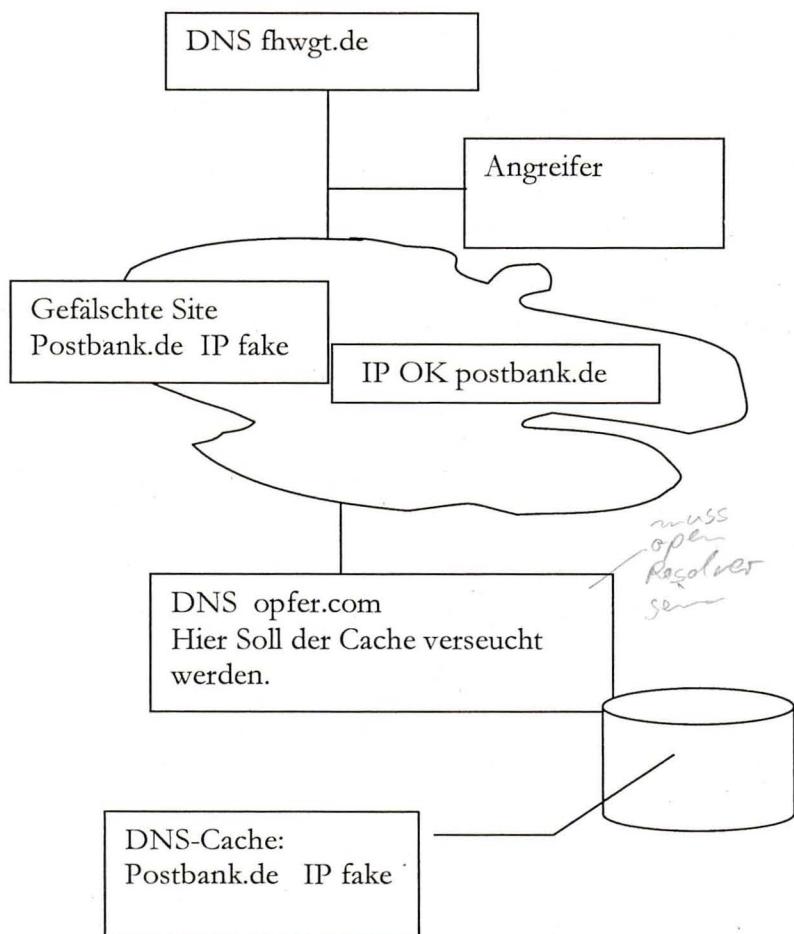
Der DNS opfer.com wird nach der IP-Adresse www.fhwgt.de befragt. Wie verhält er sich?

a.) Adresse von www.fhwgt.de im DNS-Cache:

*Antwort aus Cache*

b.) Adresse von www.fhwgt.de nicht im DNS-Cache:

① Es befragt den Name-Server fhwgt.de nach der Adresse www.fhwgt.de  
Der Angreifer sendet die Fragen vom DNS opfer.com um Beim DNS opfer.com wird der DNS-cache vergiftet.



- Antwort aus Cache*
- ① Es befragt den Name-Server fhwgt.de nach der Adresse www.fhwgt.de  
Der Angreifer sendet die Fragen vom DNS opfer.com um Beim DNS opfer.com wird der DNS-cache vergiftet.
  - ② Angreifer fragt DNS opfer.com nach der IP-Adresse von www.postbank.de
  - ③ Angreifer sendet an opfer.com falsche Antwort zurück
  - ④ opfer.com trägt falsche IP-Adresse im Cache ein
- muss open Resolver sein*

## Ermittlung des Zeitbedarfs für ein erfolgreiches DNS-Spoofing

Hubert & van Mook Expires June 18, 2009  
 Internet-Draft DNS resilience against forged answers December 2008:

The probability of spoofing a resolver is equal to the amount of fake packets that arrive within the window of opportunity, divided by the size of the problem space.

When the resolver has 'D' multiple identical outstanding queries, each fake packet has a proportionally higher chance of matching any of these queries. This assumption only holds for small values of 'D'.

In symbols, if the probability of being spoofed is denoted as  $P_s$ :

$$P_s = \frac{D * F}{N * P * I}$$

It is more useful to reason not in terms of aggregate packets but to convert to packet rate, which can easily be converted to bandwidth if needed.

If the Window of opportunity length is 'W' and the attacker can send 'R' packets per second, the number of fake packets 'F' that are candidates to be accepted is:

$$F = R * W \rightarrow P_s = \frac{D * R * W}{N * P * I}$$

Finally, to calculate the combined chance ' $P_{cs}$ ' of spoofing over a chosen time period 'T', it should be realised that the attacker has a new window of opportunity each time the TTL 'TTL' of the target domain expires. This means that the number of attempts 'A' is equal to ' $T / TTL$ '.

To calculate the combined chance of at least one success, the following formula holds:

$$P_{cs} = 1 - (1 - P_s)^A = 1 - (1 - \frac{D * R * W}{N * P * I})^{(T / TTL)}$$

When common numbers (as listed above) for D, W, N, P and I are inserted, this formula reduces to:

$$P_{cs} = 1 - (1 - \frac{R}{1638400})^{(T / TTL)}$$

From this formula it can be seen that, if the nameserver implementation is unchanged, only raising the TTL offers protection. Raising N, the number of authoritative nameservers, is not feasible beyond a small number.

For the degenerate case of a zero-second TTL, a window of opportunity opens for each query sent, making the effective TTL equal to 'W' above, the response time of the authoritative server.

This last case also holds for spoofing techniques which do not rely on TTL expiry, but use repeated and changing queries.

## Zeitbedarf bei klassischem DNS-Cache poisoning

Angegriffen werden soll der caching DNS ns1.target.com. Es soll sein Eintrag im Name-Cache für www.example.com vergiftet werden.

```
h4l@b4by10n:~$ dig -t ns example.com
;; QUESTION SECTION:
;example.com. IN NS
;; ANSWER SECTION:
example.com. 172800 IN NS dns1.example.com.
example.com. 172800 IN NS dns2.example.com.
;; ADDITIONAL SECTION:
dns2.example.com. 172800 IN A 192.168.100.2
dns1.example.com. 172800 IN A 192.168.100.1
```

```
;; ANSWER SECTION:
www.example.com. 86400 IN CNAME example.com.
example.com. 86400 IN A 192.168.100.99
```

Die Wahrscheinlichkeit für eine richtig geratene ID beträgt:

$$P_{CS} = 1 - \left[ 1 - \frac{D * R * W}{N * P * I} \right]^{(T / TTL)}$$

I: Number distinct IDs available (maximum 65536)

P: Number of ports used (maximum around 64000, but often 1)

N: Number of authoritative nameservers for a domain

R: Number of packets sent per second by the attacker

W: Window of opportunity, in seconds. Bounded by the response time of the authoritative servers (often 0.1s)

D: Average number of identical outstanding questions of a resolver (typically 1)

T: Time period in which the attack occurs

TTL: Time to live of the legitimate resource record

Tag	P-CS in %
1	1,1
3w	29,1
265	98,5

### Zeitbedarf bei DNS-Cache poisoning mit Node Re-Delegation

Anfrage zu Objekten die nicht im Cache liegen können. Diese Objekte gehören anscheinend zu einer höherliegenden Zone in der DNS-Hierarchie.

Gefragt wird `DNS-Opfer.com`



- Query 1: IN A ASJSDHASASSMXOEUWIUEUXMID.www.example.com
- Query 2: IN A HFUEFNDKHUEHJDHKJAFDHKJD.www.example.com
- ...

Spoofed response.: `www.example.com IN NS ns1.attacker.com`

$$P_{CS} = 1 - \left[ 1 - \frac{1 * 15000 * 0.1}{2 * 1 * 65535} \right]^T$$

- 1 minute : 49.7 %
- 2 minutes: 74.3 %
- 5 minutes: 96.7 %
- 10 minutes: 99.9 %

## Einträge in der /etc/hosts -Datei fälschen

Die Einträge in der Datei /etc/hosts haben Vorrang vor den DNS-Anfragen an einen Nameserver. Bei Windows findet man diese Datei unter:

C:\WINDOWS\system32\drivers\etc\hosts

```
# Copyright (c) 1993-1999 Microsoft Corp.  
#  
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP  
# für Windows 2000 verwendet wird.  
#  
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.  
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-  
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen  
# Hostnamen stehen.  
# Die IP-Adresse und der Hostname müssen durch mindestens ein  
# Leerzeichen getrennt sein.  
#  
# Zusätzliche Kommentare (so wie in dieser Datei) können in  
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,  
# aber müssen mit dem Zeichen '#' eingegeben werden.  
#  
# Zum Beispiel:  
#  
#      102.54.94.97      rhino.acme.com      # Quellserver  
#      38.25.63.10      x.acme.com          # x-Clienthost  
  
127.0.0.1      localhost  
141.69.100.200  toshi
```

## SNMP

Mark  
V1,2,3

Das Simple Network Management Protocol wird für die Verwaltung von Netzkomponenten eingesetzt. Der Zugriff auf Netzkomponenten kann auf IP-Adressen / VLANs eingeschränkt werden. Als eine Art Passwort werden die Community Strings verwendet, diese gehen je nach SNMP-Version unverschlüsselt über das Netz. Es gibt zwei community strings:

Zugriff auf Netzkomponenten nur über Management-VLAN, IP-basierte Zugriffsregeln ebenfalls denkbar.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Foxconn_7c:d5:25	Broadcast	ARP	who has 192.168.0.117? Tell 192.168.0.114
2	0.000869	HewlettP_d4:ce:a4	Foxconn_7c:d5:25	ARP	192.168.0.117 is at 00:30:6e:d4:ce:a4
3	0.000880	192.168.0.114	192.168.0.117	SNMP	get-request
4	0.004486	192.168.0.117	192.168.0.114	SNMP	get-response

```

# Frame 3 (92 bytes on wire, 92 bytes captured)
# Ethernet II, Src: Foxconn_7c:d5:25 (00:15:58:7c:d5:25), Dst: HewlettP_d4:ce:a4 (00:30:6e:d4:ce:a4)
# Internet Protocol, Src: 192.168.0.114 (192.168.0.114), Dst: 192.168.0.117 (192.168.0.117)
# User Datagram Protocol, Src Port: 1253 (1253), Dst Port: 161 (161)
# Simple Network Management Protocol
version: version-1 (0)
community: harryHirsch
data: get-request (0)
get-request:
request-id: 1297390207
error-status: noError (0)
error-index: 0
variable-bindings: 1 item
item:
name: 1.3.6.1.2.1.2.1.5.1 (IF-MIB::ifSpeed.1)
valueType: unspecified (1)

```

lesen/schreiben lesen → Klartext in V1/2  
Verschlüsselt ab V3

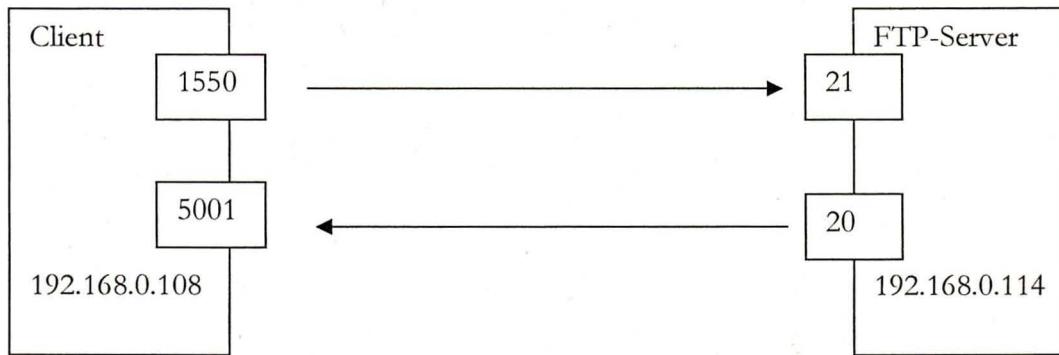
Default-Werte: lesen: public  
(community-string) schreiben: private

## FTP

Verwendet einen Kommandokanal (Port 21) und einen Datenkanal (Port 20).

### active FTP

Der FTP-Server baut die Datenverbindung auf. Über das Port-Kommando teilt der Client mit, auf welchem Port er Verbindungswünsche für die Daten entgegennimmt.



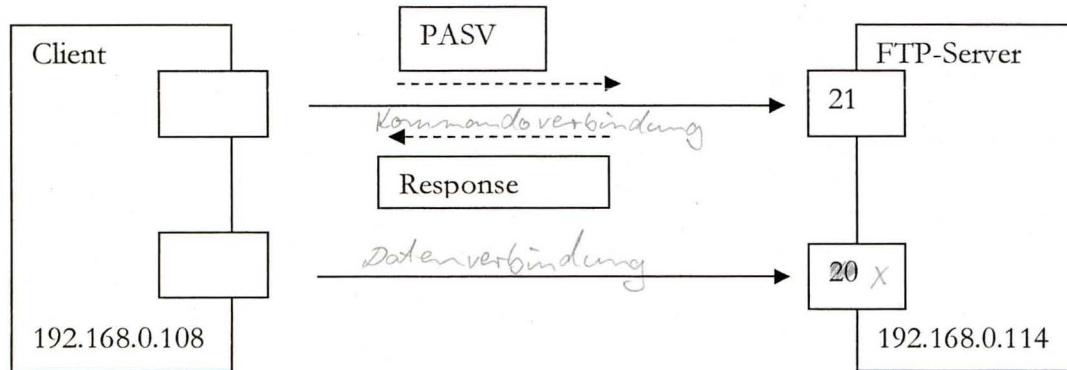
Source	Destination	Protocol	Info
192.168.0.108	192.168.0.114	TCP	1547 > 21 [SYN] Seq=953073269 Len=0 MSS=1460
192.168.0.108	192.168.0.114	TCP	1550 > 21 [SYN] Seq=241737061 Len=0 MSS=1460
Foxconn_7c:d5:25	Broadcast	ARP	who has 192.168.0.108? Tell 192.168.0.114
Ibm_f9:3c:17	Foxconn_7c:d5:25	ARP	192.168.0.108 is at 00:0d:60:f9:3c:17
192.168.0.114	192.168.0.108	TCP	21 > 1550 [SYN, ACK] Seq=2446928106 Ack=241737062 Win=65535 Len=0 MSS
192.168.0.108	192.168.0.114	TCP	1550 > 21 [ACK] Seq=241737062 Ack=2446928107 Win=65535 Len=0
192.168.0.114	192.168.0.108	FTP	Response: 220 3Com 3CDaemon FTP Server Version 2.0
192.168.0.108	192.168.0.114	TCP	1550 > 21 [ACK] Seq=241737062 Ack=2446928149 Win=65493 Len=0
192.168.0.108	192.168.0.114	FTP	Request: USER anonymous
192.168.0.114	192.168.0.108	FTP	Response: 331 User name ok, need password
192.168.0.108	192.168.0.114	TCP	1550 > 21 [ACK] Seq=241737078 Ack=2446928182 Win=65460 Len=0
192.168.0.108	192.168.0.114	FTP	Request: PASS a@b
192.168.0.114	192.168.0.108	FTP	Response: 230 User logged in
192.168.0.108	192.168.0.114	TCP	1550 > 21 [ACK] Seq=241737088 Ack=2446928202 Win=65440 Len=0
192.168.0.108	192.168.0.114	FTP	Request: PORT 192,168,0,108,19,137
192.168.0.114	192.168.0.108	FTP	Response: 200 PORT command successful.
192.168.0.108	192.168.0.114	FTP	Request: NLST
192.168.0.114	192.168.0.108	TCP	20 > 5001 [SYN] Seq=992265723 Len=0 MSS=1460
192.168.0.108	192.168.0.114	TCP	5001 > 20 [SYN, ACK] Seq=3889004831 Ack=992265724 Win=65535 Len=0 MSS
192.168.0.114	192.168.0.108	TCP	20 > 5001 [ACK] Seq=992265724 Ack=3889004832 Win=65535 [TCP CHECKSUM]
192.168.0.114	192.168.0.108	FTP	Response: 150 File status OK ; about to open data connection
192.168.0.114	192.168.0.108	FTP-DATA	FTP Data: 3 bytes
192.168.0.114	192.168.0.108	FTP-DATA	FTP Data: 1460 bytes
192.168.0.114	192.168.0.108	FTP-DATA	FTP Data: 4 bytes
192.168.0.108	192.168.0.114	TCP	5001 > 20 [ACK] Seq=3889004832 Ack=992267191 Win=65535 Len=0

### Aufgabe FTP 1

- Nennen Sie das Kennzeichen des aktiven FTP.
- Wer sendet das Port-Kommando?
- Welche Informationen sind im Port-Kommando enthalten?
- Wie errechnet sich die Portnummer?
- Welche Nachteile ergeben sich bei der Verwendung des aktiven FTP bei den Firewallregeln?

## passive FTP

Der FTP-Server baut die Datenverbindung auf. Über das Port-Kommando teilt der Client mit, auf welchem Port er Verbindungswünsche für die Daten entgegennimmt.



Der Server wählt mit dem Response auf das PASV-Kommando aus, welches Port der Client für den Aufbau der Datenverbindung verwenden soll.

Der Client schaltet den Server in den Modus passiv -FTP

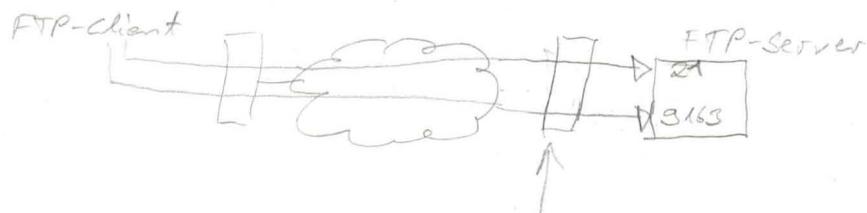
mit dem Passiv - Kommando.

Server sendet als Antwort das Port x unter dem er die Datenverbindung erwartet.

## Aufgabe FTP 2

- a.) Welche Art des FTP wird im folgenden Trace verwendet? *passiv*
- b.) Wer gibt das Port vor, über welches die Daten ausgetauscht werden sollen? *Server*
- c.) Welche Firewallregeln sind hier erforderlich, damit der Datenverkehr durchgelassen wird?

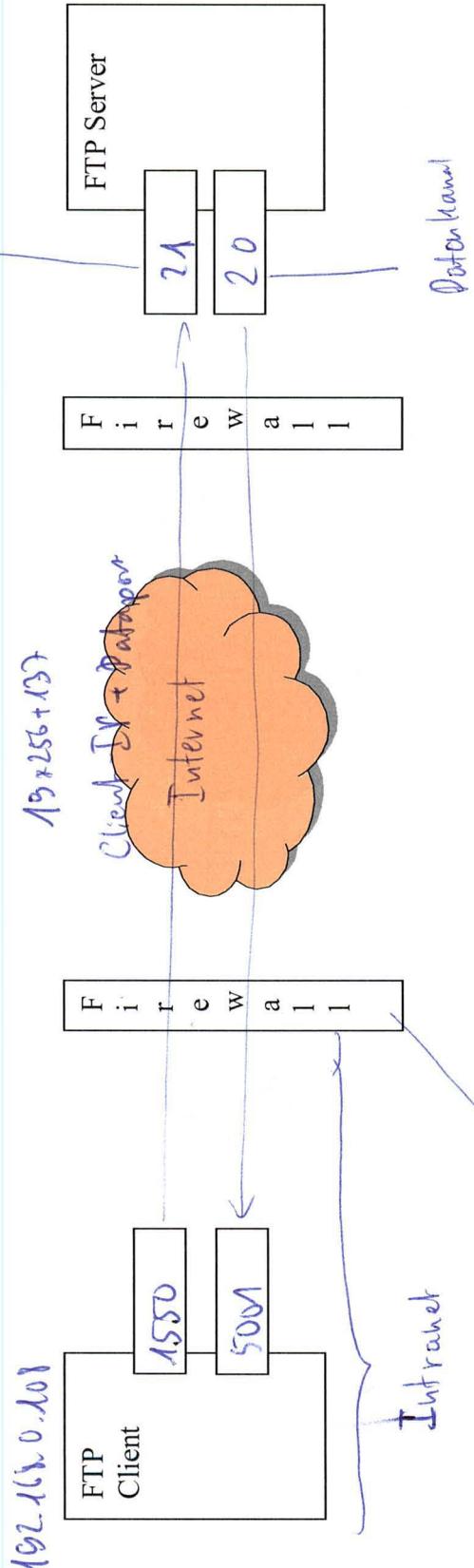
Source	Destination	Protocol	Info
192.168.0.172	212.42.244.90	TCP	58490 > 21 [SYN] Seq=2881355017 Len=0 MSS=1460 TSV=125245 TSER=0 WS=2
212.42.244.90	192.168.0.172	TCP	21 > 58490 [SYN, ACK] Seq=371442428 Ack=2881355018 Win=5792 Len=0 MSS=1380 TSV=1095
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355018 Ack=371442429 Win=1460 Len=0 TSV=125252 TSER=1095
212.42.244.90	192.168.0.172	FTP	Response: 220 (vsFTPD 2.0.4)
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355018 Ack=371442449 Win=1460 Len=0 TSV=125261 TSER=1095
192.168.0.172	212.42.244.90	FTP	Request: USER anonymous
212.42.244.90	192.168.0.172	TCP	21 > 58490 [ACK] Seq=371442449 Ack=2881355034 Win=1448 Len=0 TSV=1095154724 TSER=1095
212.42.244.90	192.168.0.172	FTP	Response: 331 Please specify the password.
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355034 Ack=371442483 Win=1460 Len=0 TSV=126790 TSER=1095
192.168.0.172	212.42.244.90	FTP	Request: PASS aab
212.42.244.90	192.168.0.172	FTP	Response: 230 Login successful.
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355044 Ack=371442506 Win=1460 Len=0 TSV=128065 TSER=1095
192.168.0.172	212.42.244.90	FTP	Request: SYST
212.42.244.90	192.168.0.172	FTP	Response: 215 UNIX Type: L8
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355050 Ack=371442525 Win=1460 Len=0 TSV=128083 TSER=1095
192.168.0.172	212.42.244.90	FTP	Request: PASV
212.42.244.90	192.168.0.172	FTP	Response: 227 Entering Passive Mode (212,42,244,90,35,203) <i>Server port 35-256+203 = 3163</i>
192.168.0.172	212.42.244.90	TCP	58490 > 21 [ACK] Seq=2881355056 Ack=371442575 Win=1460 Len=0 TSV=134364 TSER=1095
192.168.0.172	212.42.244.90	TCP	57402 > 9163 [SYN] Seq=2914885996 Len=0 MSS=1460 TSV=134364 TSER=0 WS=2
212.42.244.90	192.168.0.172	TCP	9163 > 57402 [SYN, ACK] Seq=1643646464 Ack=2914885997 Win=5792 Len=0 MSS=1380 TSV=1095
192.168.0.172	212.42.244.90	TCP	57402 > 9163 [ACK] Seq=2914885997 Ack=1643646465 Win=1460 Len=0 TSV=134371 TSER=1095
192.168.0.172	212.42.244.90	FTP	Request: LIST
212.42.244.90	192.168.0.172	FTP	Response: 150 Here comes the directory listing.
212.42.244.90	192.168.0.172	FTP-DA	FTP Data: 808 bytes
212.42.244.90	192.168.0.172	TCP	9163 > 57402 [FIN, ACK] Seq=164365273 Ack=2914885997 Win=1448 Len=0 TSV=109516231
192.168.0.172	212.42.244.90	TCP	57402 > 9163 [ACK] Seq=2914885997 Ack=164365273 Win=1864 Len=0 TSV=134379 TSER=1095
192.168.0.172	212.42.244.90	TCP	57402 > 9163 [FIN, ACK] Seq=2914885997 Ack=164365274 Win=1864 Len=0 TSV=134380 TSER=1095
212.42.244.90	192.168.0.172	TCP	9163 > 57402 [ACK] Seq=164365274 Ack=2914885998 Win=1448 Len=0 TSV=1095162321 TSER=1095



- Firewall bedacht Passiv-Kommando und Antwort darauf und reagiert automatisch / öffnet Port
- bei statischer Firewall muss ganzer Port-Bereich geöffnet werden

## Active FTP

Homebank - Mary



## Stateful firewall

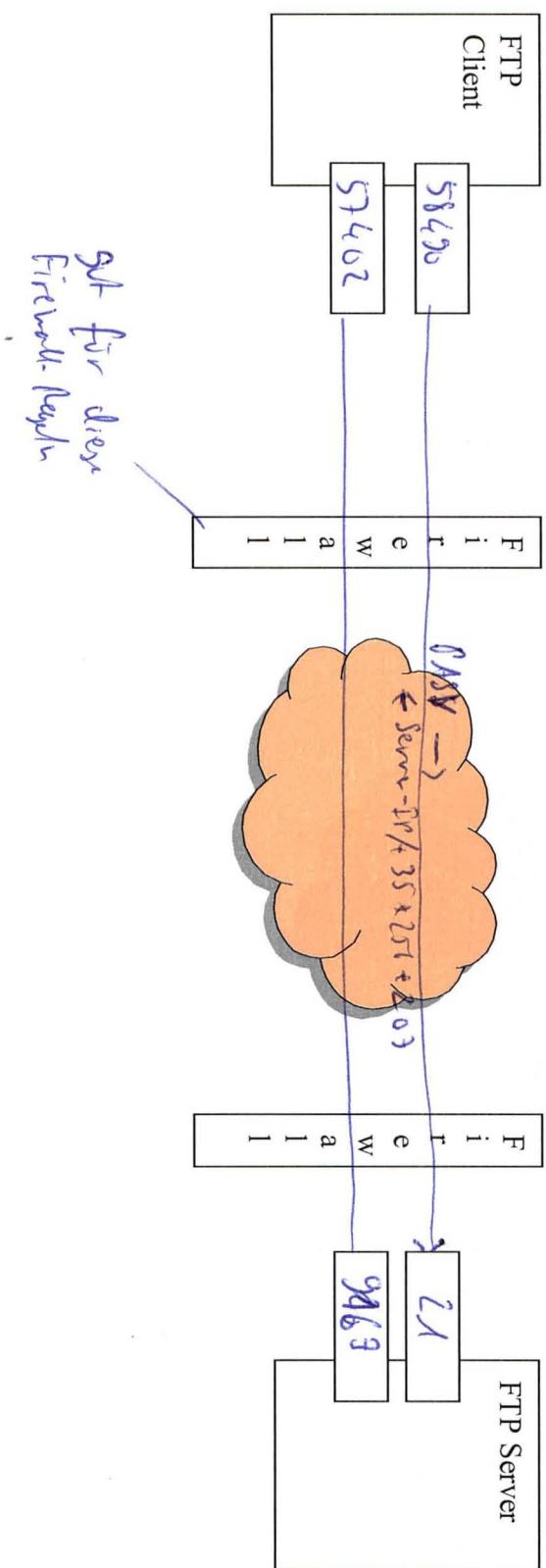
- 1) Verbindungen vom Intranet ins Internet immer zulassen } An Port 21  
Client [SYN]
- 2) Firewall erzeugt Regel für Server-Antwortpaket  
(temporäre Regel)

Für alle FTP-Clients vom alten Server  
im Internet Source-Port 21 auf  
Destination-Port > 1023 zulassen

getaktet

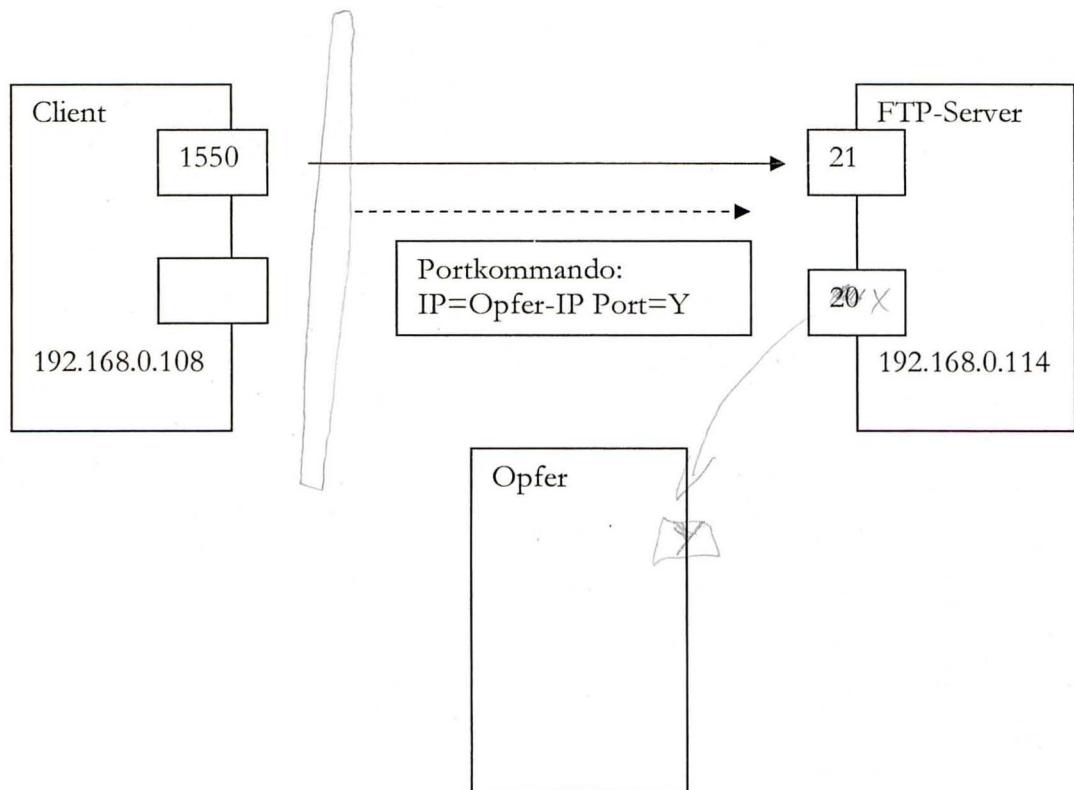
- Aus dem Internet alle DVs auf FTP-Server zulassen auf Port 21
- FTP-Server source-Port 21 an alle DST Port, > 1023 an alle Rechen unlesser
- FTP-Server von Spec-Port 20 an alle Clients  
PORT-Port > 1023 zulassen
- + Antwortpaket

## Passive FTP



## FTP- Bounce Attack

Server arbeitet im Active-FTP-Modus



So kann der Client einen Port-Scan durchführen.  
 Geht nur, wenn der Server eine Fehlermeldung  
 zurückgibt, falls die Verbindung zum Server-Ziel nicht  
 aufgebaut werden kann.

