Dokumentation Rechnernetze Labor TI5

Albert Dorn, Jan Helber, Alexander Irro

Diese Dokumentation beschreibt den Laborversuch der Vorlesung 'Rechnernetze 2' im Semester TI5 unter der Leitung von Prof. Steiper Powered by LaTeX
Generiert am 6. September 2006 um 16:06

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

Tabellenverzeichnis

1 Versuchseinführung

Diese Dokumenation wurde im Rahmen der Vorlesung Rechnernetze 2 im 5. Semester des Studiengangs Technische Informatik an der FH Ulm erstellt. Sie beschreibt den kompletten Laborversuch in dem eine „Screened Subnet“-Firewall mit verschiedenen Diensten aufgebaut und konfiguriert wird. Die Dokumentation wurde so erstellt, dass der Leser, ohne große Vorkenntnisse, den Laborversuch nachstellen kann. Im Rahmen dieses Labors sollte folgender Versuchsaufbau realisiert werden:

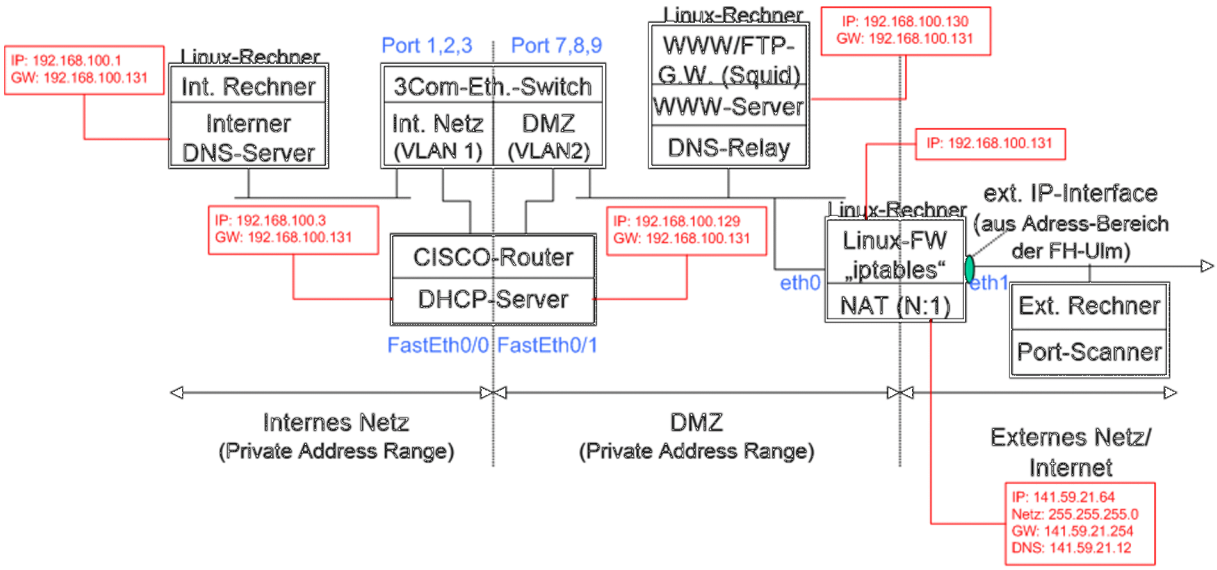

Abbildung 1: Aufbau der Netzwerkstruktur mit allen notwendigen IP-Konfigurationen

Auf den folgenden Seiten wird der Versuch Stück für Stück aufgebaut und erläutert, dabei werden jeweils die Aufgabenstellungen der ausgehändigten Versuchsunterlage zitiert und im Anschluss die Lösung der Fragestellung und deren Erklärung aufgeführt.

2 Grundkonfiguration des Netzwerks

2.1 Aufgabe 2: Netzwerkadressierungsplan

Erstellen Sie einen IP-Adressierungsplan für das Gesamt-Netzwerk. Besorgen Sie sich die Parameter für die DNS (Domain Name System)- Konfiguration von Rechnern der FH-Ulm im Netzwerk-Labor (Bedenken Sie bei der Durchführung des Versuchs, dass ein nicht funktionierender DNS-Dienst äußerst hinderlich für die Funktion der Endgeräte ist; Versuchen Sie also so lange wie möglich, den externen DNS-Dienst der FH zu nutzen und erst am Schluss den eigenen DNS-Dienst zu integrieren).

2.1.1 Lösung

Der IP-Adressraum wird in zwei Bereiche aufgeteilt:

IP-Range 192.168.100.1-127 = Internes Netz:
Enthält einen DNS-Server der die IP 192.168.100.1 bekommt und der Cisco-Router auf dem Port 'FastEthernet0/0' die IP 192.168.100.3. Auf dem Router ist später auch eine DHCP-Server aktiv der Adressen im Bereich von 192.168.100.4 - 192.168.100.127 vergibt.

IP-Range 192.168.100.128-254 = DMZ:
Das 'FastEthernet0/1'-Interface des Cisco Routers erhält die IP 192.168.100.129, der Proxyserver (Squid) erhält die 192.168.100.130 und der Gateway mit der iptables-Firewall die 192.168.100.131.

Um zwei getrennte Netze zu erhalten wird der 3Com-Switch in zwei virtuelle LANs aufgeteilt die jeweils drei Ports beinhalten:

- Internes Netz: Port 1,2,3
- DMZ: Port 7,8,9

Jeder Rechner bekommt als Gateway den 192.168.100.131, natürlich bis auf den Gateway-Rechner selbst. Zum aktuellen Zeitpunkt soll der DNS des FH-Netzes verwendet werden. D.h. auf jedem Rechner wird zunächst als DNS die 141.59.21.12 eingetragen.

2.2 Aufgabe 3: Konfigurieren der Netzwerkgeräte

Erstellen Sie die Grundkonfigurationen für den CISCO-Router und den 3Com-Switch über deren RS232-Konsoleanschlüsse.

2.2.1 3Com-Switch

Der Switch wird zuerst über den 'Console'-Port and die serielle Schnittstelle des PCs angeschlossen und mit der Linux-Anwendung 'minicom' konfiguriert. Dazu wird Minicom vom Linux-Terminal aus gestartet:

```
minicom -s
```

Im Programm selbst wird zuerst unter dem Menü 'Serial Port Setup' zuerst der korrekte serielle Anschluss (serial device) gewählt werden, z.B. '/dev/ttyS0' und als Verbindungseinstellung '9600 8N1' mit deaktivierter Hardwareflusskontrolle. Durch drücken von [ENTER] gelangt man dann zurück zum Hauptmenü, speichert die Koniguration ab und wählt anschließend 'Exit' um die Verbindung zum Switch aufzubauen. Nach kurzer Zeit verlangt der Switch einen Login. Hier wählt man 'admin' ohne Passwort. Wenn alles richtig läuft erscheint dann das Hauptmenü. Um sicherzustellen, dass der Switch keine alten Konfigurationen enthält wird Reset ausgeführt. Dazu muß folgendes eingegeben werden:

```
system
initialize
```

Der Switch wird danach eine Weile beschäftigt sein und sich schließlich wieder mit dem Login-Bildschirm zurückmelden. Anschließend müssen zwei VLANs erzeugt werden, ein 'default'-LAN existiert bereits standardmäßig und kann ignoriert werden. Geben Sie an der Konsole also folgendes ein:

```
bridge                  //Wechsel ins Bridge-Menü
vlan                    //Wechsel ins VLAN-Menü
create                  //Neues VLAN erzeugen
[ID: 2]
[Local-ID: 2]
[Name: VLAN intern]     //Symbolischer Name
create                  //Neues VLAN erzeugen
[ID: 3]
[Local-ID: 3]
[Name: VLAN DMZ]        //Symbolischer Name
addPort                 //Ports hinzufügen
[ID: 2]                 //VLAN 2
[Port: 1]               //Port #1
addPort                 //Ports hinzufügen
[ID: 2]                 //VLAN 2
[Port: 2]               //Port #2
addPort                 //Ports hinzufügen
[ID: 2]                 //VLAN 2
[Port: 3]               //Port #3
addPort                 //Ports hinzufügen
[ID: 3]                 //VLAN 3
[Port: 1]               //Port #1
[ID: 3]                 //VLAN 3
[Port: 2]               //Port #2
[ID: 3]                 //VLAN 3
[Port: 3]               //Port #3
```

Nachdem die VLANs eingerichtet wurden ist die Konfiguration des Switch abgeschlossen. Um zu testen ob Sie alles korrekt konfiguriert haben geben Sie im 'bridge'-Menü Folgendes ein:

```
summary
all
```

Tip: Um in eine höhere Menüebene zu wechseln drücken Sie in der Konsole [ESC].

Drücken Sie anschließend [STRG] + [A], dann [Z] um ins Hauptmenü von Minicom zu gelangen und wählen Sie 'Exit from Minicom (X)'.

2.2.2 Cisco Router

Starten Sie Minicom per Linux-Terminal, ändern Sie die seriellen Einstellungen auf 9600 8N1 und verbinden Sie zum Router. Geben Sie für sämtliche Benutzernamen 'router' oder 'x' und Passwörter 'x' ein. Sobald Sie auf der Konsole sind, führen Sie zuerst einen Reset durch:

```
enable        //Wechseln in den ENABLE-Modus
system        //Wechseln zum 'system'-Menü
setup         //Initialisierungsprozess starten (folgenen Sie den Bildschirmanweisungen)
```

Nachdem der Router neu gestartet ist (dauert eine Weile) konfigurieren Sie dann die zwei Interfaces. Dabei soll Interface 'FastEthernet0/0' dem internen Netz und 'FastEthernet0/1' der DMZ zugeordnet werden. Geben Sie also folgendes ein:

```
enable                                    //ENABLE-Modus starten
configure terminal                        //Configure-Modus starten
interface FastEthernet0/0                 //FastEthernet0/0 auswählen
ip address 192.168.100.3 255.255.255.128  //IP 192.168.100.3, Subnetz: 255.255.255.128
no shutdown                               //Interface einschalten
interface FastEthernet0/1
ip address 192.168.100.129 255.255.255.128
no shutdown
```

Drücken Sie anschließend [STRG]+[Z] um zurück zum ENABLE-Modus zu kommen. Der Router gibt dann einige Status-Meldungen aus. Um zu sehen ob die Konfiguration erfolgreich war geben Sie folgendes ein um den Status der Interfaces anzuzeigen:

```
configure
show interfaces FastEthernet0/0
show interfaces FastEthernet0/1
```

Tip: Es ist zu empfehlen nach der erfolgreichen Konfiguration des Routers dessen Einstellungen per TFTP zu sichern. Schließen Sie dazu einen beliebigen Rechner per CROSS-Link Kabel an eines der Interfaces an und geben Sie dem Rechner eine IP-Adresse innerhalb der für dieses Interface gültigen IP-Range und starten Sie den TFTP Server. Geben Sie dann an der Minicom-Konsole ein:

```
copy running-config tftp://[IP-Adresse]/config
```

Bestätigen Sie alle darauffolgenden Aufforderungen. Umgekehrt können Sie auch eine Konfiguration auf den Router laden in dem Sie mindestens ein Interface per Hand konfigurieren und anschließend im ENABLE-Modus folgendes eingeben:

```
copy tftp://[IP-Adresse]/config running-config
```

2.3 Aufgabe 4: Statische Integration der drei Rechner

Integrieren Sie die drei Rechner (zunächst mit statischer IP-Konfiguration) für das interne Netzwerk und die DMZ

2.3.1 DNS-Rechner

Gemäß der o.g. Definition erhält der DNS-Server die IP 192.168.100.1 und der Gateway 192.168.100.131. Geben Sie dazu am Linux-Terminal des betroffenen Rechners folgendes ein:

```
ifconfig eth0 192.168.100.1 netmask 255.255.255.0                  //Lege IP-Adresse mit Subnetz fest
ifconfig eth0 up                                                   //eth0 starten
route add -net 0.0.0.0 netmask 0.0.0.0 gw 192.168.100.131 dev eth0 //Lege Gateway fest
route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0        //Festlegung des Netzes
```

In der Datei /etc/hosts sollte am Ende folgende Zeile eingetragen werden (falls sie dort noch nicht steht): "192.168.100.1 dns.de dns"

Tip: Ob alles richtig eingestellt wurde kann bei jedem Rechner mit folgenden Kommandos überprüft werden:

```
route -F        //Zeige alle Routing-Einträge
ifconfig eth0   //Zeige Konfiguration von eth0 an
```

2.3.2 Proxy/WWW-Rechner

Gemäß der o.g. Definition erhält der Proxy-Server die IP 192.168.100.130 und der Gateway 192.168.100.131. Geben Sie dazu am Linux-Terminal des betroffenen Rechners folgendes ein:

```
ifconfig eth0 192.168.100.130 netmask 255.255.255.0                //Lege IP-Adresse mit Subnetz fest
ifconfig eth0 up                                                   //eth0 starten
route add -net 0.0.0.0 netmask 0.0.0.0 dev eth0 gw 192.168.100.131 //Lege Gateway fest
route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0        //Festlegung des Netzes
```

In der Datei /etc/hosts sollte am Ende folgende Zeile eingetragen werden (falls sie dort noch nicht steht): "192.168.100.130 squid.de squid"

2.3.3 Firewall-Rechner

Gemäß der o.g. Definition erhält der Firewall-Rechner die IP 192.168.100.131 an eth0 (obere Buchse). Für eth1 (untere Buchse) wurde die IP 141.59.21.64 und der Gateway 141.59.21.254 festgelegt. Geben Sie dazu am Linux-Terminal des betroffenen Rechners folgendes ein:

```
ifconfig eth0 192.168.100.131 netmask 255.255.255.0                //Lege IP-Adresse mit Subnetz für eth0 fest
ifconfig eth1 141.59.21.64 netmask 255.255.255.0                   //Lege IP-Adresse mit Subnetz für eth1 fest
ifconfig eth0 up                                                   //eth0 starten
ifconfig eth1 up                                                   //eth1 starten
route add -net 0.0.0.0 netmask 0.0.0.0 dev eth1 gw 141.59.21.254   //Lege Gateway fest
route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0        //Festlegung des Netzes
```

In der Datei /etc/hosts sollte am Ende folgende Zeile eingetragen werden (falls sie dort noch nicht steht): "192.168.100.131 firewall.de firewall"

Für den Fall das es nicht sofort funktioniert probieren Sie die Devices neu zu starten:

```
ifconfig eth0 down
ifconfig eth0 up
```

Wenn auch das nicht hilft starten Sie den Rechner neu und führen Sie den Device-Neustart erneut durch.

2.4 Aufgabe 5: Konfiguration der iptables-Firewall

Erstellen Sie eine Grundkonfiguration der Firewall-Software ('iptables') auf dem Firewall-Rechner, sodass dieser IP-Masquerading ausführt, aber noch keine Paketfilterung durchführt.

Durch Ausführen folgenden Shell-Scriptes wird dem Firewallrechner seine feste IP (192.168.100.131) und die beiden Routen eingetragen. Außerdem wird das Routing aktiviert und die iptables-Einträge für das Forwarding/Masquerading ausgeführt.

Listing 1: FIREWALL-simple

```
1  echo 'Statische IP (192.168.100.131) wird eingetragen'
2  killall dhclient
3  ifconfig eth0 192.168.100.131 netmask 255.255.255.0
4  #obere Buchse
5
6  ifconfig eth1 141.59.21.64 netmask 255.255.255.0
7  #untere Buchse
8
9  ifconfig eth0 up
10 ifconfig eth1 up
11
12 route add -net 0.0.0.0 netmask 0.0.0.0 dev eth1 gw 141.59.21.254
13 route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0
14
15 echo 'Routing wird aktiviert'
16 echo 1 > /proc/sys/net/ipv4/ip_forward
17
18 echo
19 echo
20 echo "Firewall stoppen"
21 service iptables stop
22
23 echo
24 echo
25 echo "iptables alle Einträge löschen"
26
27 echo
28 echo
29 echo 'iptables: nat wird gelöscht'
30 iptables -F -t nat
31
32 echo 'iptables: filter wird gelöscht'
33 iptables -F -t filter
34
35 echo 'iptables: nat und filter werden aufgelistet (sollten leer sein)'
36 iptables -L -t nat
37 iptables -L -t filter
38
39 echo
40 echo
41 echo "Firewall starten"
42 service iptables start
43
44 echo
45 echo
46 echo "Addiere neue Einträge für MASQUERADING"
47 #Anfragen aus dem internen Netz nach draußen auf Port 80 zulassen
48 iptables -A FORWARD -p tcp -i eth0 -o eth1 --dport 80 -s 192.168.100.0/24 -d 0.0.0.0/0 -j ACCEPT
49 #Anfragen von draußen (Port 80) in das interne Netz zulassen
50 iptables -A FORWARD -p tcp -i eth1 -o eth0 --sport 80 -s 0.0.0.0/0 -d 192.168.100.0/24 -j ACCEPT
51 #MASQUERADING
52 iptables -A POSTROUTING -s 192.168.100.0/24 -t nat -o eth1 -j MASQUERADE
```

In der Datei /etc/hosts sollte am Ende folgende Zeile eingetragen werden (falls sie dort

noch nicht steht): "192.168.100.131 firewall.de firewall"

2.5 Aufgabe 6: Ausführliche Netzwerkfunktionsprüfung

Überprüfen Sie die grundsätzliche Funktion des so entstandenen Gesamtnetzes (Kann jeder Rechner in jedem Netz von jedem anderen Netz aus erreicht werden? Funktioniert DNS noch für interne Rechner?). Erst dann fahren Sie mit der weiteren Konfiguration der einzelnen Komponenten fort.

Von jedem Rechner im internen Netzt folgende Befehle testen:

Listing 2: Ping

```
1 ping 192.168.100.1
2 ping 192.168.100.3
3 ping 192.168.100.129
4 ping 192.168.100.130
5 ping 141.59.21.12
6 ping 192.168.100.131
7 ping gmx.de
```

Es sollte jedesmal 0% Verlust angezeigt werden, sobald man das Pingen mit STRG+C abbricht. Außerdem sollte es von jedem Rechner aus möglich sein mit einem Browser eine beliebige Seite im Internet zu erreichen.

3 Der DHCP-Server des Cisco Routers

3.1 Aufgabe 7: Konfiguration des DHCP-Servers

Der CISCO-Router soll einen 'Dynamic Host Configuration Protocol (DHCP)'-Service für Rechner des internen Netzes und der DMZ liefern:

1. Sowohl im internen Netz, als auch in der DMZ sollen neben IP-Adressen auch Default-Gateway-Adressen, DNS-Server-Adressen und DNS-Domainnamen durch DHCP vergeben werden.
2. Im internen Netz sollen die IP-Adressen für 'Nicht-Server-Rechner' dynamisch aus einem Adress-Pool vergeben werden. Ein vorher unbekannten Rechner (z.B. ihre eigenes Notebook), der mit einem Switch-Port des internen Netzes verbunden wird, soll also eine funktionsfähige IP-Konfiguration bekommen. Ein später betriebener DNS-Server-Rechner soll anhand seiner MAC-Adresse immer die gleiche IP-Adresse zugewiesen bekommen.
3. Der Linux-Rechner (Application-level Gateway) in der DMZ soll seine IP-Adresse statisch anhand der MAC-Adresse zugeordnet bekommen.
4. Das interne Interface des Firewall-Rechners wird nicht per DHCP konfiguriert.

3.1.1 Starten des DHCP-Dienstes

Bevor der DHCP-Dienst genutzt werden kann muß erst ein IP-Pool festgelegt werden aus dem ein Rechner eine IP erhalten kann. In unserem Fall wäre das 192.168.100.0/24, also 192.168.100.1-127. Loggen Sie sich dazu zunächst per Minicom auf dem Router ein, wechseln Sie in den ENABLE-Modus und geben Sie Folgendes ein.

```
configure terminal                  //Wechsel in den Configure-Modus
service dhcp                        //DHCP-Dienst starten
ip dhcp pool 192.168.100.0/24       //Festlegen des IP-Pools
network 192.168.100.0 255.255.255.0 //Netz festlegen
default-router 192.168.100.131      //Gateway festlegen
lease 0 0 1                         //Lease-Time auf 1 Minute stellen
exit                                //Menü verlassen
```

Zur Überprüfung ob die Konfiguration erfolgreich war schließen Sie einen beliebigen Rechner mit aktivierter DHCP-Adresskonfiguration an das Interne VLAN an und überprüfen Sie ob Sie eine IP in der gültigen IP-Range von 192.168.100.1-127 bekommen haben:

```
ipconfig -all                       //Bei Windows-Systemen
ifconfig eth0                       //Bei Linux-Systemen
```

3.1.2 Vergabe von statischen IPs per DHCP

Bei manchen Diensten innerhalb eines Netzwerkes ist es empfehlenswert, dass ein Rechner immer dieselbe IP-Adresse zugewiesen bekommt. In unserem Falle wären das der Proxy-server und der DNS-Server (die Firewall bleibt weiterhin statisch). Um sicherzustellen, ob es sich um den richtigen Computer handelt wird die MAC-Adresse des betroffenen Computers im DHCP-Server hinterlegt und einer IP-Adresse zugewiesen. Wechseln Sie zunächst in den ENABLE-Modus und beginnen Sie mit der Konfiguration:

```
configure                              //Wechsel in den Configure-Modus
ip dhcp pool 192.168.100.0/25          //Festlegen des IP-Pools
host 192.168.100.130 255.255.255.0     //IP-Adresse des Proxy-Servers
hardware-address 00.13D4.D72B.F9       //MAC-Adresse des Proxy-Servers
client-name proxy                      //Symbolischer Name
default-router 192.168.100.131         //Gateway festlegen
end                                    //Konfiguration abschließen
ip dhcp pool 192.168.100.0/25          //Festlegen des IP-Pools
host 192.168.100.1 255.255.255.0       //IP-Adresse des DNS-Servers
hardware-address 00.13D4.D72B.EF       //MAC-Adresse des DNS-Servers
client-name dns                        //Symbolischer Name
end                                    //Konfiguration abschließen
```

Tip: Windows und Linux senden eine unterschiedliche MAC-Kennung. Bei Windows ist dies der sog. Media-Type bei Linux die Hardware-Adresse. Der Media-Type wird durch ein vorangestelltes 01 in der MAC-Adresse identifiziert. Im obigen Skript wird davon ausgegangen, dass es sich um Linux-Rechner handelt (Kommando 'hardware-address'). Wenn beispielsweise der Proxy-Server ein Windows-Rechner wäre, so müsste die Zeile folgendermaßen lauten:

```
(...)
client-identifier 0100.13D4.D72B.F9
(...)
```

3.1.3 DHCP-Autokonfiguration bei DNS- und Proxy-Server aktivieren

Da ein DHCP-Dienst dem betroffenen Rechner keine IP-Adresse aufzwingen kann, muß dieser so konfiguriert werden, dass er zuerst einen DHCP-Request sendet um dann die Konfiguration zu erhalten. Geben Sie daher bitte bei den beiden betroffenen Computern folgende Kommandos ein:

```
dhclient
ifconfig eth0 down
ifconfig eth0 up
```

Tip: Beim DHCP-Server führt oftmals die sog. Lease-Time zu Problemen. Standardmäßig ist diese nämlich auf 24 Stunden eingestellt. Lease-Time ist diejenige Zeit, für die eine dynamisch vergebene IP-Adresse nach der Trennung der Verbindung für eine bestimmte MAC-Adresse reserviert bleibt. Erst nach Ablauf der Lease-Time steht Sie dem DHCP-Server wieder zur Verfügung und kann an andere Rechner vergeben werden. Wir haben bereits darauf geachtet, dass die Lease-Time auf eine Minute gestellt wird, dennoch kann es sein, dass es eine Weile dauert bis eine IP-Adresse wieder freigegeben wird. Für den Fall, dass die Probleme bestehen bleiben starten Sie den DHCP-Dienst auf dem Router neu:

```
configure terminal                  //Wechsel in den Configure-Modus
no service dhcp                     //DHCP-Dienst beenden
service dhcp                        //DHCP-Dienst starten
end                                 //Configure-Modus verlassen
```

4 Integration eines Application-Gateways in die DMZ

4.1 Aufgabe 8: Einrichtung eines Proxy-Servers (Squid)

In der DMZ soll der Linux-Rechner als WWW-Proxy-Server agieren. Dazu wird das Software-Paket SQUID eingesetzt. Der WWW-Zugriff interner Rechner auf externe WWW-Server im Internet soll nur über dieses Gateway möglich sein. Drei Alternativen sollen getestet werden:

1. Squid-Server läuft auf Firewall-Rechner, also transparente Konfiguration:
   Versuchen Sie einen 'Transparenten Proxy-Server' auf dem Firewall-Rechner zu installieren. Damit ist es möglich, dass Anfragen eines internen WWW-Browsers ohne lokale Proxy-Konfiguration (also nur durch entsprechende Konfiguration von 'iptables' auf dem Firewall-Rechner) immer an den Proxy-Server weitergeleitet werden. Wenn es funktioniert hat, stoppen Sie wieder den Squid- Server auf dem Firewall-Rechner und kommentieren den erforderlichen iptables-Eintrag in der Firewall-Konfiguration aus.

2. Squid-Server läuft auf separatem Rechner der DMZ, jedoch eine transparente Konfiguration:
   Bauen Sie jetzt die eigentliche Squid-Konfiguration gemäß Abbildung auf. Der Squid-Server wird jetzt auf dem Linux-Rechner in der DMZ gestartet. WWW-Browser auf internen Rechnern werden so konfiguriert, dass alle Anfragen auf den Proxy-Server in der DMZ gerichtet werden. Um Nutzer von nicht korrekt konfigurierten WWW-Browsern entsprechend zu informieren, soll auf dem Firewall- Rechner ein WWW-Server installiert sein, der als einzige Seite eine entsprechende Warnung mit Konfigurationshinweisen liefert. Alle Anfragen von falsch konfigurierten internen WWW-Clients müssen dazu an diesen WWW-Server weitergeleitet werden (durch iptables- Konfiguration). Kommentieren Sie nach dem erfolgreichen Test wieder die entsprechenden Zeilen der Firewall-Konfiguration aus.

3. Squid-Server läuft auf separatem Rechner der DMZ, jedoch mit transparenter Konfiguration:
   Es soll jetzt wieder ohne spezielle Proxy-Konfiguration des internen WWW-Clients möglich sein, dass alle WWW-Anfragen über den Firewall-Rechner abgefangen und an den Squid-Server auf dem Linux-Rechner der DMZ weitergereicht werden.

4. Modifizieren Sie jetzt ihr iptables-Konfigurationsscript so, dass beim Starten der Firewall eingegeben werden kann, welche Konfiguration verwendet werden soll. Konfiguration 3 soll immer verwendet werden, falls der Anwender nichts anderes eingibt.

5. Optional Aufgabe (falls Zeit bleibt): Testen Sie im Script auch, ob ein Squid-Server auf der Firewall-Maschine oder auf dem DMZ-Rechner aktiv ist und geben Sie entsprechende Warnmeldungen aus, die dem Anwender sagen, was er noch zu tun hat, um den Proxy-Serverdienst richtig nutzen zu können!

Um zu Beginn eines Laborversuches alle Systeme möglichst schnell wieder im vorherigen Zustand zu haben um dadurch viel Zeit für neue Versuche/Konfigurationen zur Verfügung zu haben, wurden von uns auf jede Rechner Shell-Skripte gespeichert.

Folgendes Shell-Scrip fragt den User ob er die statische IP (192.168.100.130) für den Squid-Rechner eintragen oder vom DHCP beziehen soll und konfiguriert daraufhin den Squid-Rechner:

Listing 3: STARTUP-SQUID.SH auf (192.168.100.130)

```
a=n
echo -n 'statische IP anstatt DHCP holen? (j/n): '
read a
if [ "$a" != "n" ]
  then
    echo 'Statische IP (192.168.100.130) wird eingetragen'
    killall dhclient
    ifconfig eth0 192.168.100.130 netmask 255.255.255.0
    route add -net 0.0.0.0 netmask 0.0.0.0 dev eth0 gw 192.168.100.131
    route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0
fi
echo 'Interface eth0 aktivieren'
ifconfig eth0 up

echo 'DNS eintragen (141.59.21.254)'
echo 'nameserver 141.59.21.254' > /etc/resolv.conf

echo 'squid wird gestartet'
/etc/init.d/squid start

echo 'Apache wird gestartet'
httpd
```

Folgendes Shell-Scrip stellt dem User verschiedene Fragen und konfiguriert daraufhin den Firewall-Rechner. Es stehen folgende Möglichkeiten zur Auswahl:

- Statische IP (192.168.100.131) für Firewall-Rechner eintragen oder vom DHCP beziehen
- Trasparenter Proxy (Squid mit Prerouting) oder beim Client den Proxy manuell eintragen?
- Falls manuell: Standart-Seite auf Client ausgeben, wenn Proxy nicht eingetragen ist?

Listing 4: STARTUP-FIREWALL.SH auf (192.168.100.131)

```
a=n
b=n
c=n
d=n
echo -n 'statische IP anstatt DHCP holen? (j/n): '
read c
echo -n 'Mit Squid-Prerouting (Transparent)? (j/n): '
read a
if [ "$a" == "n" ]
  then
    echo -n 'Hinweis auf falsche Konfig & Apache starten? (j/n): '
    read b
fi
echo -n 'Anfragen aus WWW an Port 80 an Squid weiterleiten? (j/n): '
read d
```

```
16
17 if [ "$c" != "n" ]
18   then
19     echo
20     echo
21     echo 'Statische IP (192.168.100.131) wird eingetragen'
22     killall dhclient
23     ifconfig eth0 192.168.100.131 netmask 255.255.255.0
24     #obere Buchse
25
26     ifconfig eth1 141.59.21.64 netmask 255.255.255.0
27     #untere Buchse
28
29     ifconfig eth0 up
30     ifconfig eth1 up
31
32     route add -net 0.0.0.0 netmask 0.0.0.0 dev eth1 gw 141.59.21.254
33     route add -net 192.168.100.0 netmask 255.255.255.0 dev eth0
34 fi
35
36 echo 'Routing wird aktiviert'
37 echo 1 > /proc/sys/net/ipv4/ip_forward
38 echo
39 echo
40 echo "Firewall stoppen"
41 service iptables stop
42 echo
43 echo
44 echo "iptables alle Einträge löschen"
45 echo
46 echo
47 echo 'iptables: nat wird gelöscht'
48 iptables -F -t nat
49 echo 'iptables: filter wird gelöscht'
50 iptables -F -t filter
51 echo 'iptables: nat und filter werden aufgelistet (sollten leer sein)'
52 iptables -L -t nat
53 iptables -L -t filter
54 echo
55 echo
56 echo "Firewall starten"
57 service iptables start
58 echo
59 echo
60 echo "Addiere neue Einträge für MASQUERADING"
61 #Anfragen aus dem internen Netz nach draußen auf Port 80 zulassen
62 iptables -A FORWARD -p tcp -i eth0 -o eth1 --dport 80 -s 192.168.100.0/24 -d 0.0.0.0/0 -j ACCEPT
63 #Anfragen von draußen (Port 80) in das interne Netz zulassen
64 iptables -A FORWARD -p tcp -i eth1 -o eth0 --sport 80 -s 0.0.0.0/0 -d 192.168.100.0/24 -j ACCEPT
65 #MASQUERADING
66 iptables -A POSTROUTING -s 192.168.100.0/24 -t nat -o eth1 -j MASQUERADE
67
68 if [ "$a" != "n" ]
69   then
70     echo
71     echo
72     echo "Addiere Squid-Prerouting (Transparent)"
73     iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT -s ! 192.168.100.130 --to 192.168.1
74   else echo Kein Squid-Prerouting (manuelle Konfiguration bei den Clients)
75 fi
76
77 echo
78 echo
79 echo beende Apache
80 killall httpd
81
82 echo
83 echo
84 if [ "$b" != "n" ]
85   then
```

```
86      echo "Starte Apache"
87      /usr/sbin/apachectl -k start
88      echo "iptables eintrag (für falsch konfigurierte Clients)"
89      iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT -s ! 192.168.100.130 --to 192.168.1
90    else
91      echo Apache wird nicht gestartet
92  fi
93
94  if [ "$d" != "n" ]
95    then
96      echo
97      echo
98      echo 'Anfragen aus WWW an Port 80 werden an Linux-Rechner (Squid) weiterleiten'
99      iptables -t nat -A PREROUTING -i eth1 -p TCP --dport 80 -j DNAT --to-destination 192.168.100.130
100 fi
```

5 Der DNS-Service

5.1 Aufgabe 9: Interner DNS-Service

Ein Linux-Rechner soll im internen Netz als DNS-Server für die zugehörige DNS-Domain 'intern.de' dienen (DNS-Server und DNS-Client können auf gleichem Rechner laufen). Der interne DNS-Server antwortet nur auf Anfragen interner Rechner und dient dort als Primary Domain Server. Alle Anfragen auf DNS-Namen außerhalb der internen DNS-Domain sollen an einen DNS-Relay-Server in der DMZ weitergeleitet werden.

Änderungen in der /etc/named.conf zur Namensauflösung des internen Netzes sowie der angabe der Weiterleitung bei unbekannten Adressen.

Listing 5: DNS /etc/named.conf

```
1  options {
2          directory "/var/named/";
3          forwarders {192.168.100.130; }; //Weiterleitung zum DNS-Relay im Squid
4          forward only;
5  };
6
7  .
8  .
9  .
10
11 zone "100.168.192.in-addr.arpa" {
12         type master;
13         file "100.168.192.in-addr.arpa.zone";
14 };
15
16 zone "intern.de" {
17         type master;
18         file "intern.de.zone";
19 };
```

Unsere internen Namen der Rechner werden im Verzeichnis /var/named/intern.de.zone aufgelöst. Indem man dort folgendes einträgt:

Listing 6: DNS /var/named/intern.de.zone

```
1  $TTL      86400
2  @         IN      SOA     @   root.localhost {
3                                      2;                      //serial
4                                      28800;                  //refresh
```

```
5                                       7200;                   //retry
6                                       604800;                 //expire
7                                       86400;                  //TTL
8                               }
9
10 @            IN      NS      intern
11
12 dns          IN      A       192.168.100.1
13 squid        IN      A       192.168.100.130
14 firewall     IN      A       192.168.100.131
```

5.2 Aufgabe 10: Interner DNS-Relay

In der DMZ soll der Linux-Rechner als DNS-Relay-Server agieren (d.h. er leitet lediglich DNS-Anfragen des internen DNS-Servers aus dem internen Netz an einen DNS-Server der FH-Ulm weiter und gibt die Antworten an den internen DNS-Server zurück).

Eine anfrage die nicht aufgelöst werden kann wird an den DNS-Relay im DNZ weitergeleitet. Die anfrage wird dann mit Hilfe eines DNS-Forwarder an eine externe DNS weitergeleitet.

Angabe der möglichen DNS-Servern, wo angefragt werden kann müssen in der Datei /etc/named.conf auf dem Squid Rechner eingetragen werden

Listing 7: SQUID /etc/named.conf

```
1 options {
2         directory "/var/named/";
3         forwarders {
4                 141.59.40.12;
5                 141.59.41.206;
6                 141.59.41.207};
7         forward only;
8 };
```

6 Der WWW-Server in der DMZ

6.1 Aufgabe 11: WWW-Server installieren und konfigurieren

Auf dem Linux-Rechner der DMZ soll ein WWW-Server aktiviert werden, der vom Internet aus erreichbar ist! (Da in der DMZ private IP-Adressen verwendet werden, muss der Firewall-Rechner entsprechend konfiguriert werden!)

Listing 8: Apache installieren

```
#Zunächst muss der Apache-Daemon heruntergeladen und entpackt werden
wget ftp://ftp.apache.de/mirrors/dev.apache.org/dist/httpd/httpd-2.2.2.tar.gz
tar xfz httpd-2.2.2.tar.gz

#Danach wechselt man in das gerade (durchs entpacken) entstandene Verzeichnis
cd httpd-2.2.2

#Da wir keine besonderen Wünsche für unseren Server haben,
#übergeben wir keine Parameter an configure und sind mit der Standart-Konfiguration zufrieden.
./configure

#Nun sind die sources soweit vorbereitet, dass sie compiliert und installiert werden können.
make
make install

#hier muss normalerweise nichts verändert werden,
#wenn man mit dem pfad /var/www/ für die Webseite zufrieden ist
#Wenn man ''AllowOverride'' auf All setzt,
#kann man einige Einstellungen auch über die .htaccess im Web-Verzeichnis ändern.
vi /etc/httpd/conf/httpd.conf

#Zu guter letzt startet man den Apache-Server als Daemon
#und testet anschließend mit einem Browser http://192.168.100.130 ob er funktioniert.
#Es sollte eine Standart-Testseite angezeigt werden.
/bin/apachectl -k start
lynx http://192.168.100.130
```

Die iptables-Weiterleitungseinträge, damit der Server auch aus dem Internet erreichbar ist, werden unter 4.1 beschrieben.

7 Filter-Regeln

7.1 Aufgabe 12: Konfigurieren der Firewall des Cisco Routers

Erstellen Sie Firewall-Regeln für den CISCO-Router und den Firewall-Rechner (zunächst als verbale Auflistung der Zugriffsregeln, die für das interne Netz, die DMZ und das externe Netz und zwischen diesen Netzen gelten sollen), die nur noch Zugriffe auf die benötigten Dienste zulassen. Konfigurieren Sie die entsprechenden Regeln auf dem CISCO-Router und dem Firewall-Rechner schrittweise und testen Sie dabei immer, ob die oben integrierten Dienste noch funktionieren.

7.2 Access-Lists

Zur Konfiguration der CISCO-Firewall müssen sog. Access-Lists eingegeben werden. Dabei wir bestimmt welcher Port auf welcher IP bei welchem Interface welche Rechte bekommt. In unserem Fall sind die Ports 53 (DNS), 80 (HTTP), 3128 (Sockets) und 68 (DHCP) von Interesse. Verbinden Sie also mit minicom auf den Router und geben Sie Folgendes ein:

Listing 9: Cisco-Firewall

```
1  enable
2  configure
3
4  // 104 Eth 0/0 IN
5  access-list 104 permit udp 192.168.100.0 0.0.0.255 any eq 53
6  access-list 104 permit udp 192.168.100.0 0.0.0.255 any eq 68
7  access-list 104 permit tcp 192.168.100.0 0.0.0.255 any eq 80
8  access-list 104 deny ip any any
9
10 // 103 Eth 0/1 IN
11 access-list 103 permit udp 192.168.100.0 0.0.0.255 any eq 53
12 access-list 103 permit udp 192.168.100.0 0.0.0.255 any eq 68
13 access-list 103 permit tcp any host 192.168.100.130 eq 80
14 access-list 103 permit tcp any host 192.168.100.130 eq 3128
15 access-list 103 permit tcp any host 192.168.100.131 eq 80
16 access-list 103 deny ip any any
17
18 interface FastEthernet0/0
19 ip access-group 104 in
20 exit
21
22 interface FastEthernet0/1
23 ip access-group 103 in
24 exit
```

8 'Lessons learned'

Es ist notwendig jeden Schritt, jede Kleinigkeit zu protokollieren um später nachvollziehen zu können was man überhaupt gemacht hat, denn oft ist es so gewesen, dass einem am Ende des einen Labors viele Dinge klar waren aber eine Woche später sitzt man dann davor und denkt sich 'Wie war das jetzt noch gleich?'. Genau diese Fragestellung kostete jedes Mal aufs Neue erheblich Zeit, so dass der Versuchsaufbau mindestens eine Stunde benötigte. Es ist wirklich von Vorteil soviel Abläufe wie nur möglich zu automatisieren in Form von Bash-Skripten oder Konfigurationsdateien per TFTP. Ein detailliertes Protokoll aller ausgeführten Schritte ist daher für zukünftige Projekte und Labore sicher sehr empfehlenswert vor allem wenn diese über einen längeren Zeitraum andauern.

Ein weiterer wichtiger Punkt ist die Kommunikation und Arbeitsaufteilung innerhalb des Teams. Zu Beginn war es hier öfters so, dass Alle überall ein bisschen etwas gemacht haben und es daher keinen 'Verantwortlichen' für einen Projektteil gab. Je mehr man jedoch mit der Materie vertraut wurde desto besser klappte es auch im Team und jeder hatte nach einer gewissen Vorlaufzeit sein eigenes 'Fachgebiet'.

Notiz am Rande von Jan Helber: Was bin ich froh, dass wir so was nicht fürs Zeitgeschichtliche Archiv machen mussten!

9 Marken

Diese Dokumentation beinhaltet eingetragene Marken oder Marken der jeweiligen Eigentümer.
Einige Firmen- und/oder Produktbezeichnungen in dieser Dokumentation sind Warenzeichen und/oder eingetragene Warenzeichen ihrer jeweiligen Besitzer.

10 Haftungsausschluss

Die Vollständigkeit und verlässlichkeit der in dieser Dokumentation enthaltenen Informationen wurden sorgfältig überprüft. Für die Richtigkeit der Angaben kann ich jedoch keine Gewähr übernehmen und für eventuelle Schäden nicht haften. In keinem Fall bin ich Ihnen gegenüber haftbar für Schäden, die auf die Verwendung oder den anderweitigen Einsatz dieser oder anderer Dokumentationen zurückzuführen sind.