

Rechnernetze Aufgabe 3 Auswertung

Triebe, Marian
`marian.triebe@haw-hamburg.de`

Kirstein, Katja
`katja.kirstein@haw-hamburg.de`

January 12, 2015

Contents

1 Routingtabellen

Das erstellen der Routingtabellen verlief wie geplant, jedoch stellte sich die Frage, ob die Rechner (R1-R7) aus den internen Netzen mit R8 kommunizieren dürfen. Wir nehmen an, dass R8 einen Rechner aus dem Internet darstellen soll, da R8 auch nicht im Topologie-Plan gelistet ist. Die Rechner aus den internen Netzen erreichen R8 somit über die default Routings ihres jeweiligen Routers.

Kritisch waren die Routingtabellen für R6 sowie R2, da darauf geachtet werden musste, dass das 172.16.12.0/24 Netz nur über das VPN Netzwerk (172.16.15.0/24) erreichbar sein darf. Somit ist gewährleistet, dass die SSH Verbindung von R7 in die DMZ (R3) nicht über das Internet geroutet wird.

Das korrekte Routing über die einzelnen Knoten wurde mit *tracert* sowie *ping* geprüft.

2 Portscan DMZ (ohne Firewall)

Der Portscan ergab, dass auf R3 die folgenden Dienste aktiv sind:

- FTP (21)
- SSH (22)
- HTTP (80)

Dies entspricht dem erwarteten Verhalten, da R3 einen SSH-daemon, FTP sowie HTTP bereitstellt. Der Portscan wurde durch den Befehl *nmap -P0 -p1-99 r3_0* angestartet und hat die Port-range 1 bis 99 gescannt. Gestartet wurde *nmap* von R1.

3 Einstellen der Policy

Die default Policy der Ketten *INPUT*, *OUTPUT*, *FORWARD* wurden auf *DROP* gestellt (Listing 1).

```
1 ## Default Policy DROP
2 iptables -P INPUT DROP
3 iptables -P OUTPUT DROP
4 iptables -P FORWARD DROP
```

Listing 1: Default Policy

Dies hatte zur Folge, dass *nmap* keine freien Ports mehr scannen konnte, außerdem war es nicht mehr möglich einen anderen Rechner per *ping* zu erreichen. Generell wurden alle Pakete automatisch verworfen, da es keine Ausnahmeregel gab.

4 Ping erlauben

Teil der Aufgabe ist es, innerhalb der internen Netzwerke sowie von internen Netzwerken in das Internet zu pinggen. Es sollen jedoch keine Pings aus dem

Internet an PCs in internen Netzwerken weitergeleitet werden.
Es ist somit notwendig bei jedem PC in den internen Netzwerken eingehende ICMP Pakete zu erlauben (Listing 2).

```
1 ## Ping senden
2 iptables -A INPUT -p icmp --icmp-type echo-request -
  j ACCEPT
3 iptables -A OUTPUT -p icmp --icmp-type echo-request -
  j ACCEPT
4 ## Ping antwort
5 iptables -A INPUT -p icmp --icmp-type echo-reply -j
  ACCEPT
6 iptables -A OUTPUT -p icmp --icmp-type echo-reply -j
  ACCEPT
```

Listing 2: ICMP eingehend/ausgehend

Die Rechner R2, R4 sowie R6 benötigen zusätzlich die Erlaubnis Pakete die nur weitergeleitet werden sollen zu akzeptieren (Listing 3).

```
1 ## Ping senden
2 iptables -A FORWARD -p icmp --icmp-type echo-request -
  j ACCEPT
3 ## Ping antwort
4 iptables -A FORWARD -p icmp --icmp-type echo-reply -j
  ACCEPT
```

Listing 3: ICMP Weiterleitung

Das filtern von ICMP Paketen von fremden Netzwerken (bspw. aus dem Internet) geschieht bei R5. Hierzu haben wir die *-i* Option von *iptables* verwendet. Diese Option bezieht das Interface auf welchem das Paket empfangen wurde mit ein (Listing 4). Zum filtern fremder Pakete wurde es einfach nicht erlaubt Pakete vom Typ *echo-request* von Interface *eth2* freizugeben, das hat zur folge, dass die Pakete korrekt gefiltert werden.

```
1 ## Filter fuer Internes Netz / Internet Pings
2 iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-
  reply -j ACCEPT
3 iptables -A FORWARD -i eth0 -p icmp --icmp-type echo-
  request -j ACCEPT
4 iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-
  reply -j ACCEPT
5 iptables -A FORWARD -i eth1 -p icmp --icmp-type echo-
  request -j ACCEPT
6 iptables -A FORWARD -i eth2 -p icmp --icmp-type echo-
  reply -j ACCEPT
```

Listing 4: ICMP filter

5 SSH in die DMZ

Es ist nur von R1 sowie R7 möglich eine SSH Verbindung in die DMZ (zu R3) aufzubauen. Dazu musste R3 so konfiguriert werden, dass eingehende SSH Verbindungen akzeptiert werden (Listing 5). Es werden nur Verbindungen von R1 und R7 zugelassen, das wird durch die *-s ip* Option von *iptables* sichergestellt.

```
1 ## R1
2 iptables -A INPUT -p tcp -s 172.16.11.1 --sport
   513:65535 --dport 22 -m state --state NEW,
   ESTABLISHED -j ACCEPT
3 ## R7
4 iptables -A INPUT -p tcp -s 172.16.14.2 --sport
   513:65535 --dport 22 -m state --state NEW,
   ESTABLISHED -j ACCEPT
```

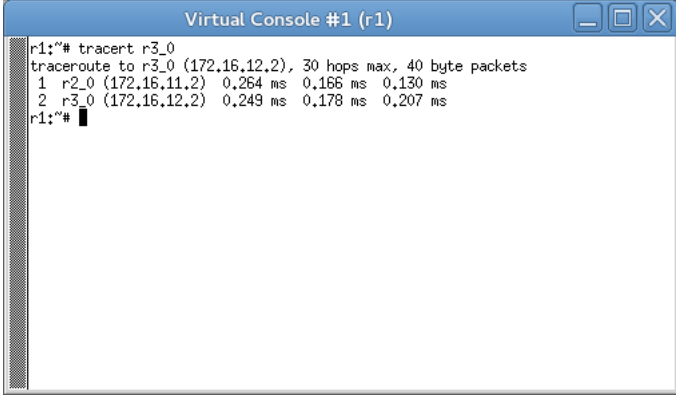
Listing 5: SSH eingehend/ausgehend

Desweiteren musste die Weiterleitung von SSH Paketen erlaubt werden. Dies musste auf R2 sowie R6 konfiguriert werden (Listing 6). Die Konfiguration von R6 ist Identisch mit der von R2, jedoch kann die Regel zum weiterleiten der Pakete von R1 nach R3 entfallen, da Pakete von R1 nicht über R6 geroutet werden.

```
1 ## R1
2 iptables -A FORWARD -p tcp -s 172.16.11.1 -d
   172.16.12.2 --sport 513:65535 --dport 22 -m state
   --state NEW,ESTABLISHED -j ACCEPT
3 iptables -A FORWARD -p tcp -s 172.16.12.2 -d
   172.16.11.1 --sport 22 --dport 513:65535 -m state
   --state ESTABLISHED -j ACCEPT
4 ## R7
5 iptables -A FORWARD -p tcp -s 172.16.14.2 -d
   172.16.12.2 --sport 513:65535 --dport 22 -m state
   --state NEW,ESTABLISHED -j ACCEPT
6 iptables -A FORWARD -p tcp -s 172.16.12.2 -d
   172.16.14.2 --sport 22 --dport 513:65535 -m state
   --state ESTABLISHED -j ACCEPT
```

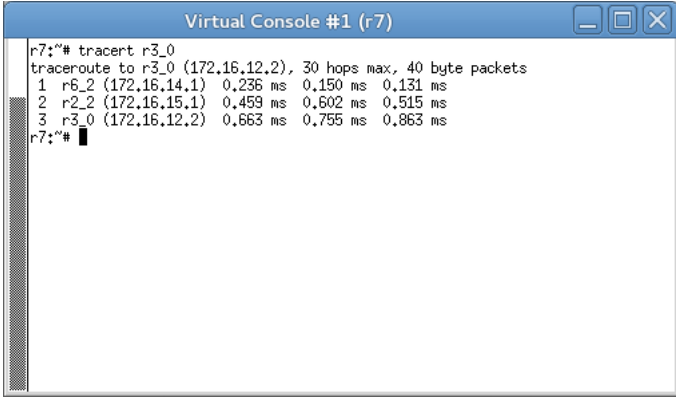
Listing 6: SSH Weiterleitung (Konfiguration von R2)

Die Verbindung von R1 zu R3 erfolgt ausschließlich über die internen Netzwerke, wie in Figure 1 zu sehen. Selbiges gilt für Verbindungen von R7 zu R3, in Figure 2 zu sehen.



```
Virtual Console #1 (r1)
r1:~# tracert r3_0
tracert to r3_0 (172.16.12.2), 30 hops max, 40 byte packets
 0  r1_0 (172.16.11.1)  0.264 ms  0.166 ms  0.130 ms
 1  r2_0 (172.16.11.2)  0.264 ms  0.166 ms  0.130 ms
 2  r3_0 (172.16.12.2)  0.249 ms  0.178 ms  0.207 ms
r1:~#
```

Figure 1: tracert zu R3 von R1



```
Virtual Console #1 (r7)
r7:~# tracert r3_0
tracert to r3_0 (172.16.12.2), 30 hops max, 40 byte packets
 0  r7_0 (172.16.14.1)  0.236 ms  0.150 ms  0.131 ms
 1  r6_2 (172.16.14.1)  0.236 ms  0.150 ms  0.131 ms
 2  r2_2 (172.16.15.1)  0.459 ms  0.602 ms  0.515 ms
 3  r3_0 (172.16.12.2)  0.663 ms  0.755 ms  0.863 ms
r7:~#
```

Figure 2: tracert zu R3 von R7

6 Stateful Firewall

Mit Hilfe der Stateful Option von *iptables* ist es möglich Verbindungen die in bestimmten Zuständen sind zu erlauben. Bspw. wenn bei einem TCP Paket das SYN Flag gesetzt ist, oder eine Verbindung bereits aufgebaut wurde (ESTABLISHED). Wir erlauben alle Pakete die bereits zu einer bestehenden Verbindung zugeordnet werden können (Listung 7).

```
1 iptables -A INPUT -m state --state ESTABLISHED,
   RELATED -j ACCEPT
2 iptables -A OUTPUT -m state --state ESTABLISHED,
   RELATED -j ACCEPT
3 iptables -A FORWARD -m state --state ESTABLISHED,
   RELATED -j ACCEPT
```

Listing 7: Stateful ESTABLISHED,RELATED

Außerdem erlauben wir TCP Verbindungen zu Port 20, 21 sowie 80 bei denen das SYN Flag gesetzt wurde. Dies in Verbindung mit den Regeln aus Listing 6 erlaubt HTTP, sowie FTP Verbindungen.

```

1 iptables -A FORWARD -m state --state NEW -p tcp --syn
  --dport 20 -j ACCEPT
2 iptables -A FORWARD -m state --state NEW -p tcp --syn
  --dport 21 -j ACCEPT
3 iptables -A FORWARD -m state --state NEW -p tcp --syn
  --dport 80 -j ACCEPT

```

Listing 8: Stateful NEW, SYN FLAG

Desweiteren mussten auf R3 eingehende Pakete für Port 20, 21 sowie 80 akzeptiert werden. Hierzu haben wir die Regeln aus Listing 7 leicht angepasst. Zum anpassen der Regel wurde das *FORWARD* Keyword durch *INPUT* ersetzt.

Die HTTP Verbindung zu R3 von externen Netzwerken (R8) ist erfolgreich und wurde mit der Hilfe des Programms *links* getestet (Figure 3). Die FTP



Figure 3: HTTP zu R3 von R7 (links http://r3_0)

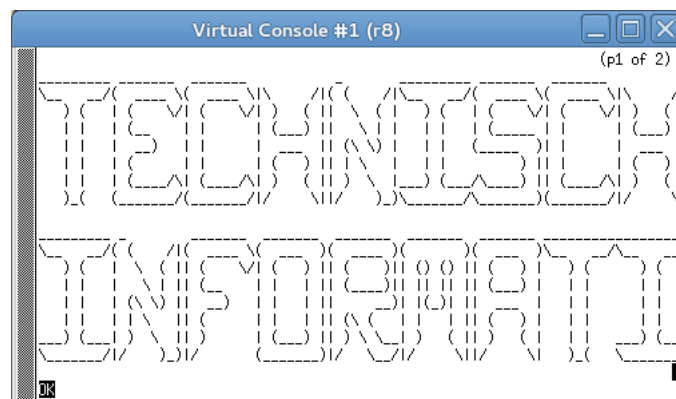


Figure 4: FTP zu R3 von R7 (links ftp://r3_0/demo.txt)

Verbindung zu R3 von externen Netzwerken (R8) ist ebenfalls erfolgreich und wurde mit der Hilfe des Programms *links* getestet (Figure 4). Von internen Hosts (r1 bis r7) war ebenfalls der Zugriff auf die HTTP, FTP Dienste von R3 möglich.