Labreport #5

Patrick Eickhoff, Alexander Timmermann

Aufgabe 1: Netzwerkeinstellungen

2.
- Client-VM

  IP-Adresse: `192.168.254.44`

  Gateway: `192.168.254.2`

  Nameserver: `10.1.1.1`
- Router-VM

  IP-Adresse eth0: `172.16.137.222`

  IP-Adresse eth1: `192.168.254.2`
- Server-VM

  IP-Adresse: `172.16.137.144`

Aufgabe 2: Absichern eines Einzelplatzrechners mit iptables

1. Auf der Client-VM sind keine iptables-Regeln vorhanden, die man löschen könnte. Zum Löschen könnte man sonst folgende Befehle benutzen:

```
1 sudo iptables -F              # flush chains in 'filter' table
2 sudo iptables -t nat -F       # flush chains in 'nat' table
3 sudo iptables -t mangle -F    # flush chains in 'mangle' table
4 sudo ipables -X               # delete custom chains
```

Mit

```
1 sudo apt-get update
2 sudo apt-get install openssh-server
```

installieren wir den OpenSSH Server.

2. Um das Surfen auf Webseiten zu erlauben, müssen wir den Datenverkehr über die Ports 80 (HTTP), 443 (HTTPS) und 53 (DNS) der RouterVM erlauben:

```
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
 iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Desweiteren wollen wir sowohl als ICMP-Nachrichten senden und empfangen, als auch SSH-Verbindungen (Port 22) aufbauen können:

```
iptables -A INPUT -p icmp-j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A OUTPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Letzendlich wollen wir jeglichen anderen Traffic unterbinden:

```
iptables -A INPUT -j REJECT
iptables -A OUTPUT -j REJECT
```

3.
- Die SSH-Verbidung von der CLientVM auf die RouterVM (`ssh user@192.168.254.2`) wird verweigert ("refused"), während die Verbindung von RouterVM auf ClientVM (`ssh user@192.168.254.1`) problemlos funktioniert.
- Per `nc -l 5555` setzen wir einen Server auf der CLientVM auf. Wenn wir diesen jedoch von der RouterVm mit `nc 192.168.254.2 5555` ansprechen wollen, wird die Verbindung verweigert ("refused").
- Wenn wir statt REJECT DROP für unsere Firewall verwenden, bekommen wir bei einem Verbindungsversuch keine Refused-Nachricht mehr zurück. Da die Firewall das Packet einfach ignoriert.

4. Mithilfe dynamischer Regeln können wir einfach definieren, dass ein- und ausgehende Packete, die zu bereits etablierten Verbindungen gehören (ESTABLISHED,RELATED), automatisch akzeptiert werden:

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Die restlichen Regeln definieren sich dann wie folgt:

```
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p icmp-j ACCEPT
iptables -A INPUT -j REJECT
```

Dynamische Regeln sind sehr angenehm, da sie erlauben Packete abhängig von ihrem Zustand zu behandeln. So werden deutlich weniger Regeln benötigt, um die Kommunikation bereits aufgebauter Verbindungen zu erlauben.