

# Labreport #4

Patrick Eickhoff, Alexander Timmermann

## Aufgabe 1

2. Die Surfing-VM erhält eine IP-Adresse aus dem Subnetz 192.168.254.0/24, das Standard-Gateway ist 192.168.254.1 und der konfigurierte Nameserver ist mit 192.168.99.1 adressiert.
3. Die Routing-VM besitzt mehrere NICs (*Network Interface Card*). Unter `eth0` ist das Labor-Netz angebunden, das über das Gateway 172.16.137.2 angesprochen werden kann. Unter `eth1` ist das sog. host-only network angeschlossen, über das sich die Surfing-VM verbindet.
4. Bevor die Surfing-VM eine Verbindung herstellen konnte, mussten wir die in der Einleitung aufgeführten Schritte, d.h. Entfernen einiger udev-Regeln, durchführen.

## 2 Aufgabe 2

2. Um den Anforderungen der Aufgabe gerecht zu werden, müssen wir mehrere Optionen von `tcpdump` kombinieren, sodass wir schlussendlich folgenden Befehl erhalten:

```
1 root@routingvm# tcpdump -p -i eth1 -n "port 53 and host 192.168.254.44"
```

- `-p` deaktiviert den promiscuous mode, da er von der Virtualisierungsumgebung nicht bereitgestellt werden kann.
- `-i eth0` beschränkt die Aufzeichnung auf das Interface `eth1`, über das die Surfing-VM angeschlossen ist
- `-n` deaktiviert die Auflösung von IP-Adressen zu einem Hostnamen.
- `port 53` filtert nur nach Paketen, die über Port 53, der DNS zugeordnet ist, gesendet bzw. empfangen werden.
- `host 192.168.254.44` beschränkt auf Pakete, deren Ziel oder Ausgang mit der IP der Surfing-VM übereinstimmt.

Die Nachrichten, die uns angezeigt werden, protokollieren den DNS-Verkehr, an dem die Surfing-VM beteiligt ist, d.h. sowohl Anfragen als auch Antworten.

Das Format der Antworten kann wie folgt beschrieben werden:

```
timestamp | source ip | destination ip | dns transaction id | answer records/NS  
records/additional records | answer
```

- `timestamp` ist die Zeit und das Datum, an dem das Paket erfasst wurde
- `source ip` ist die Quell-IP-Adresse des Pakets
- `destination ip` ist die Ziel-IP des Pakets
- `dns transaction id` ist eine eindeutige ID, die Bestandteil des DNS-Protokolls ist und mit der Transaktionen identifiziert werden
- `answer records` ist die Anzahl an DNS records, die als Antwort zur Verfügung gestellt werden
- `NS records` ist die Anzahl an DNS records, die die zuständigen Nameserver angeben
- `additional records` ist die Anzahl an DNS records, die zusätzlich zur Verfügung stehen.
- `answer` ist dann schließlich die Antwort, die der Nameserver zurückliefert, d.h. entweder eine IP-Adresse (A/AAAA record), ein weiterer Hostname (CNAME record) oder weitere record typen.

3. Abgewandelt von 2. verwenden wir hier folgenden Befehl:

```
1 root@routingvm# tcpdump -p -i eth1 -n "(port 80 or port 443) and src host 192.168.254.44"
```

Wir spezifizieren hier zwei Ports, die "verodert" werden, und legen fest, dass die **Quell-IP** die der Surfing-VM sein muss.

4. Zur Ausgabe der Payload fügen wir den Parameter `-A` hinzu. Dadurch wird uns der Payload der Pakete in ASCII konvertiert ausgegeben.

Bisher werden Pakete jedoch nur verkürzt ausgegeben. Dies rührt daher, dass `tcpdump` als voreingestellte Paketlänge, die sog. *snap length* 68 bytes verwendet. Wir fügen also als Parameter `-s 1514` hinzu. Damit schneiden wir die kompletten 1500 bytes des Pakets plus 14 bytes Layer 2 Header mit.

5. HTTP Basic Authentication über unverschlüsseltes HTTP ist inhärent unsicher, was wir auch sehen wenn wir den Traffic mitschneiden. Im **Authorization-Header** werden die Anmeldedaten im Format `username:password` als base64 codiert übertragen und lassen sich sehr leicht mit

```
1 root@routingvm# echo -n "YWxpY2U6c2VocmdlaGVpbQ==" | base64 -d
```

decodieren.

### 3 Aufgabe 3

1. Zur Demonstration schneiden wir alle HTTP-Interaktionen mit **heise.de** mit. Die Seite wurde gewählt, da sie nur HTTP unterstützt und HTTPS mit **urlsnarf** nicht mitzuschneiden ist.

Wir starten **urlsnarf** mit dem Befehl

```
1 root@routingvm# urlsnarf -n -i eth1 ".*heise\.de"
```

Die Parameter sind dabei weitestgehend deckungsgleich mit denen von **tcpdump**. Der letzte Parameter ist ein regulärer Ausdruck, der spezifiziert welche URLs mitgeschnitten werden sollen.

Der Output erfolgt dabei im *Common Log Format*, wie es z.B. auch **nginx** benutzt.

2. Mit Hilfe von **dsniff** können wir uns das Passwort aus **2.5** direkt anzeigen lassen. Dazu starten wir **dsniff** wie folgt:

```
1 root@routingvm# dsniff -n -i eth1 'port 80'
```

Auch hier sind die Parameter wieder deckungsgleich mit **tcpdump**. Rufen wir nun die Testseite auf und authentifizieren uns mit **alice:sehrgeheim**, so zeigt uns **dsniff** diese als Output an.

### 4 Aufgabe 4

2. Ein Capture-Filter spezifiziert, welche Pakete tatsächlich mitgeschnitten werden, während ein Display-Filter alle Pakete mitschneidet, jedoch die Anzeige der Pakete filtert.
4. Schlauerweise schneiden wir nur Traffic mit, der auf dem Interface **eth1** gesendet bzw. empfangen wird, da über dieses Interface die Surfing-VM angebunden ist.

Unabhängig vom Interface können wir auch einen Capture-Filter einrichten, der den Mitschnitt auf die IP-Adresse der Surfing-VM beschränkt.