Übungsblatt 04 – Routing & Traffic Analysis

Aufgabe A – TCP: 3-Way-Handshake

Nachdem sie sich bereits theoretisch mit dem 3-Way-Handshake auseinandergesetzt haben, sollen sie nun schauen, ob der TCP-Handshake tatsächlich wie theoretisch beschrieben arbeitet.

1. Überlegen sie sich eine Anfragen an eine Website (dies sollte TCP nutzen, etwa durch ein HTTP-Request!), die sie noch nicht von der VM aus getätigt haben.
2. Starten sie Wireshark, richten sie Interface und Protokoll-Type ein. Filtern sie nur auf eine speziellen Request!
3. Lösen sie den Handshake durch aufrufen der Website (oder Ressource) aus, während Wireshark den Netzverkehr mitschneidet.
4. Analysieren sie den 3-Way-Handshake!
5. Zum Vergleich: Analysieren Sie ihren Mitschnitt mit folgender Aufzeichnung: https://wiki.wireshark.org/TCP_3_way_handshaking?action=AttachFile&do=view&target=3-way+handshake.pcap

Aufgabe B – ICMP

Da die Befehle *ping* und *traceroute ICMP* nutzen, sollen Sie mit Wireshark solche Request mitverfolgen.

1. Setzen sie alle notwendigen Parameter um Wireshark mitlaufen zu lassen, sodass Sie die ICMP-Nachrichten mitverfolgen können.
2. Pingen sie einen Rechner mit seinem Namen an (bspw.: mi.fu-berlin.de).
3. Ping auf eine IP-Adresse (bspw.: 160.45.117.199).
4. Ping auf die IP-Adresse Ihres Routers.
   Hinweis: Sie können diese durch *ip r* oder *route* in Erfahrung bringen.

```
ip r
default via XXX.XXX.XXX dev DEVICE proto dhcp src YOU.RIP.ADD metric VALUE
#or
route -n
Destination  Gateway  Genmask  Flags  Metric  Ref  Use  Iface
0.0.0.0  XXX.XXX.XXX  0.0.0.0  UG  VALUE  0  0  DEVICE
```

5. Ping auf meine eigene IP-Adresse.
6. Ping auf die Loopback-Adresse.
7. Starten sie eine Routenverfolgung via *traceroute* auf eine beliebige Adresse. Verfolgen sie dabei den Ausgabe auf der Konsole als auch in Wireshark (Filtern Sie in Wireshark entsprechend). Spiegeln sich die Einträge in Wireshark mit denen auf der Kommandozeile?
8. Erläutern sie die Ergebnisse ihrer vorigen Aufgabe. Wie funktioniert *traceroute* und wie hängt dies mit *ICMP* zusammen?

Aufgabe C – Routing & Traceroute

Nachdem sie recherchiert haben, wie *traceroute* arbeitet, welche Kritik an Traceroute geäußert wurde und wie diese mit dem Tool Paris-Traceroute abgestellt wurden, sollen beide Tools hier kurz erprobt werden.

1. Überlegen sie sich zunächst anhand Ihrer Recherche was *traceroute* in etwa ausgeben müsste, wenn sie auf der VM eine Route von einem Rechner A zu einem Rechner B verfolgen würden. Wobei beide Rechner zu unterschiedlichen LANs gehören.
2. Nutzen Sie anschließend *traceroute* um sich die Router zwischen zwei VMs anzeigen zu lassen. Stimmen Ihre theoretische Überlegungen mit denen von *traceroute* überein? Falls nicht, sollten Sie analysieren woran dies liegen könnte.
3. Vergleichen Sie die Ausgaben von *traceroute* und *paris-traceroute* für folgende IP-Adressen:
   a) 41.231.21.44
   b) 91.198.174.192
   c) 37.220.21.130
   d) 80.239.142.229

   Hinweis: Für *paris-traceroute* sollten Sie den „exhaustive algorithm“ Nutzen (in machen Versionen als Parameter: `-na exhaustive`)
4. Analysieren sie anschließend die Ausgabe beider Tools.
5. Warum wurde Ihnen eine Liste von IP-Adressen genannt anstelle von Domainnamen? Nennen Sie mindestens zwei Gründe!

Aufgabe C - Bestimmung des physischen Rechners zu einer IP-Adresse – ARP

Sie haben bereits theoretisch recherchiert, wie die Zuordnung von physischer Adresse zu einer IP-Adresse vonstatten geht. Im Folgenden sollen sie herausfinden, ob die Auflösung von IP-Adresse auf physische Adresse wirklich analog zu ihren theoretischen Recherchen abläuft.

1. Finden sie mithilfe Wiresharks heraus, wie die Adressauflösung funktioniert.
   a) Leeren sie zunächst den ARP-Cache.
   b) Pingen sie nun einen Rechner an, den Sie vorhin noch nicht „angepingt“ haben. Die dafür ausgetauschten Pakete (und wahrscheinlich einige mehr) werden „gesnifft“.
   c) Beenden sie das Mitschneiden des Netzwerksverkehrs und setzen sie als Filtern die MAC-Adresse ihres Adapters.
   d) Versuchen sie über den Mitschnitt herauszufinden, wie die Bestimmung des zugehörigen Netzadapters und die MAC-Adresse erfolgt.

   1)

2. Damit ihr Rechner nicht jedes mal eine Auflösung veranlassen muss, werden die ARP-Informationen lokal in einem Cache zwischengespeichert („cached“).
   a) Lassen sie sich Ihren aktuellen ARP-Cache anzeigen. Welche Informationen können sie diesem entnehmen?
   b) Schauen sie kurz nach, wie lange der ARP-Cache Einträge vorhält.
   c) Lassen sie zwei VMs die IP-Adressen tauschen. Dies sollte möglichst schnell umgesetzt werden!
   d) Versuchen sie nun durch eine dritte VM eine „alte“ IP-Adresse zu erreichen. Werden die Daten an den richtigen Knoten übermittelt?
   e) Verfolgen sie die Datenübermittlung per Wireshark mit.

Aufgabe D - Packet Analysis

Im Moodle-Kurs liegt eine Zip-Datei `network_packets.zip`. Diese enthält verschiedene Dateien die sie auf verschiedene Arten in Wireshark öffnen können. Sie sollen diese Pakete analysieren. Teilweise sind in diesen Paketen Passwörter und Zugangsdaten zu finden, in einigen Fällen können ganze Nachrichten oder Geräteinformationen gefunden werden.

1. Die Datei `ftp.pcap` ist eine FTP-Session mit Passwort Authentifizierung. Finden sie das Paket sowie Passwort.

2. Die Datei `telnet.pcap` ist eine Telnet-Session mit Passwort Authentifizierung. Finden sie das Paket sowie Passwort.

3. Die Datei `raw_ethernet_frame` ist ein Ethernet Frame in Hex-Format. Das heißt, sie müssen einen Hex-Dump auswerten. Finden Sie heraus, was im Ethernet-Frame enthalten ist.

4. Die Datei `twitter.pcap` ist eine Twitter-Session welche die Authentifizierung enthält. Finden sie heraus, wie diese umgesetzt wurde und finden sie das Passwort.

5. Die Datei `bt.bin` für die Authentisierung von Bluetooth-Geräten gedacht. Im wesentlichen benötigen sie die MAC-Adresse des Gerätes und den Gerätenamen. Beides ist in der Datei enthalten. Die Authentisierung erfolgt die Hashing mit SHA1 (eine kryptografische Hashfunktion [1]). Finden sie das Tupel aus MAC-Adresse und Gerätenamen heraus und lassen Sie die SHA1 Funktion darüber laufen.

```
1 echo "XXXXXXXXXXXXXXXXXXX" | sha1sum
```

[1] Mehr dazu demnächst. SHA1 gilt seit Jahren als unsicher!