**Network Exercises – part 1**

Brug Wireshark til at monitorere netværks trafikken. Til nogle opgaver bruger vi en node/Express server som kilde til requests.

**Important:** Unless otherwise stated, use the Network Setting **NAT** in Virtual Box for these exercises.

Getting Started

Setup the server as sketched below (on your local OS, NOT on KALI):

* Create a folder somewhere on your system called networkdemo.

In this folder, do the following:

Type:

npm init -y

npm install express

* Create a file called app.js, and paste in the following code:

const express = require("express");  
const app = express();  
app.use(express.static('public'));  
const PORT = 5551;  
app.listen(PORT,()=>console.log(`Server started: ${PORT}`));

* Create a subfolder called public and use Google to find and download a [large picture](https://upload.wikimedia.org/wikipedia/commons/d/d5/OverWhelmingly_Large_Telescope.jpg) (>500Kb)

Now start the server and verify that you can access it on:

http:/localhost:5551   → Should provide a simple 404 error

http:/localhost:5551/**nameOfImage**   → Should fetch the image

OPGAVE 1

**Monitoring HTTP requests and the TCP/IP protocol Stack**

1. **Find din IP-adresse**

Der findes både en IP-adresse med 10.xxx og 192.xxx. Begge kan bruges til denne opgave.

Find IP-adresse:”*ifconfig*”.

1. Start virtual box og åbn Kali Linux
2. Åbn en shell og send en ping til din fundne IP-adresse, for at bekræfte adgang til denne.

I terminal på Kali Linux skriv: **ping + IP**

1. Åbn Fire Fox I Kali Linux og skriv **IP + port**, men tryk ikke ENTER.
2. Åbn Wireshark og vælg eth0 samt skriv **host + IP** i capture filter box**.**
3. Start capture.
4. Tryk **ENTER** I Fire Fox.

**Capture filter box**: http

* This should provide you with two HTTP packages, the Request and the Response → Verify this via the Source and Destination fields

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

I response findes:

**Payload**

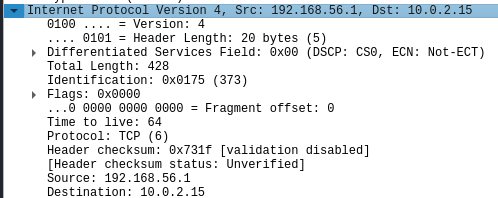
Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

**Port numre**



**IP-adresse**

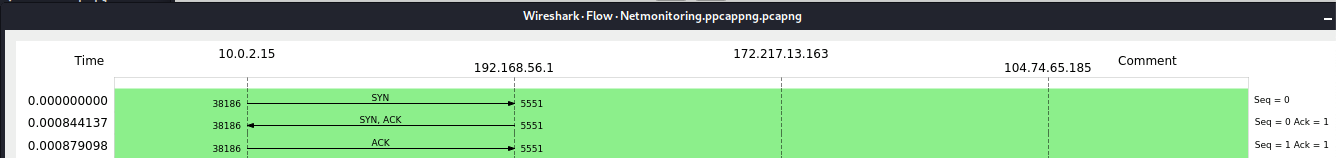


**MAC adresse**



**Identify the initial TCP Three-Way Handshake (and the close down Handshake)**

* 1. **Capture filter box:** tcp
  2. Vælg i menuen: **Statistics → Flow Graph**
  3. Vælg derefter Flow Type: **TCP Flows**
  4. Identificer Three-Way Handshake (SYN, SYN ACK, ACK)



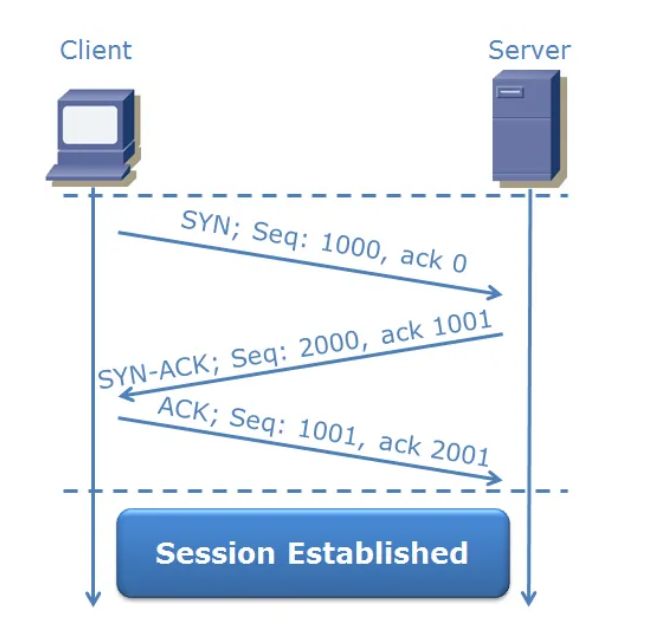
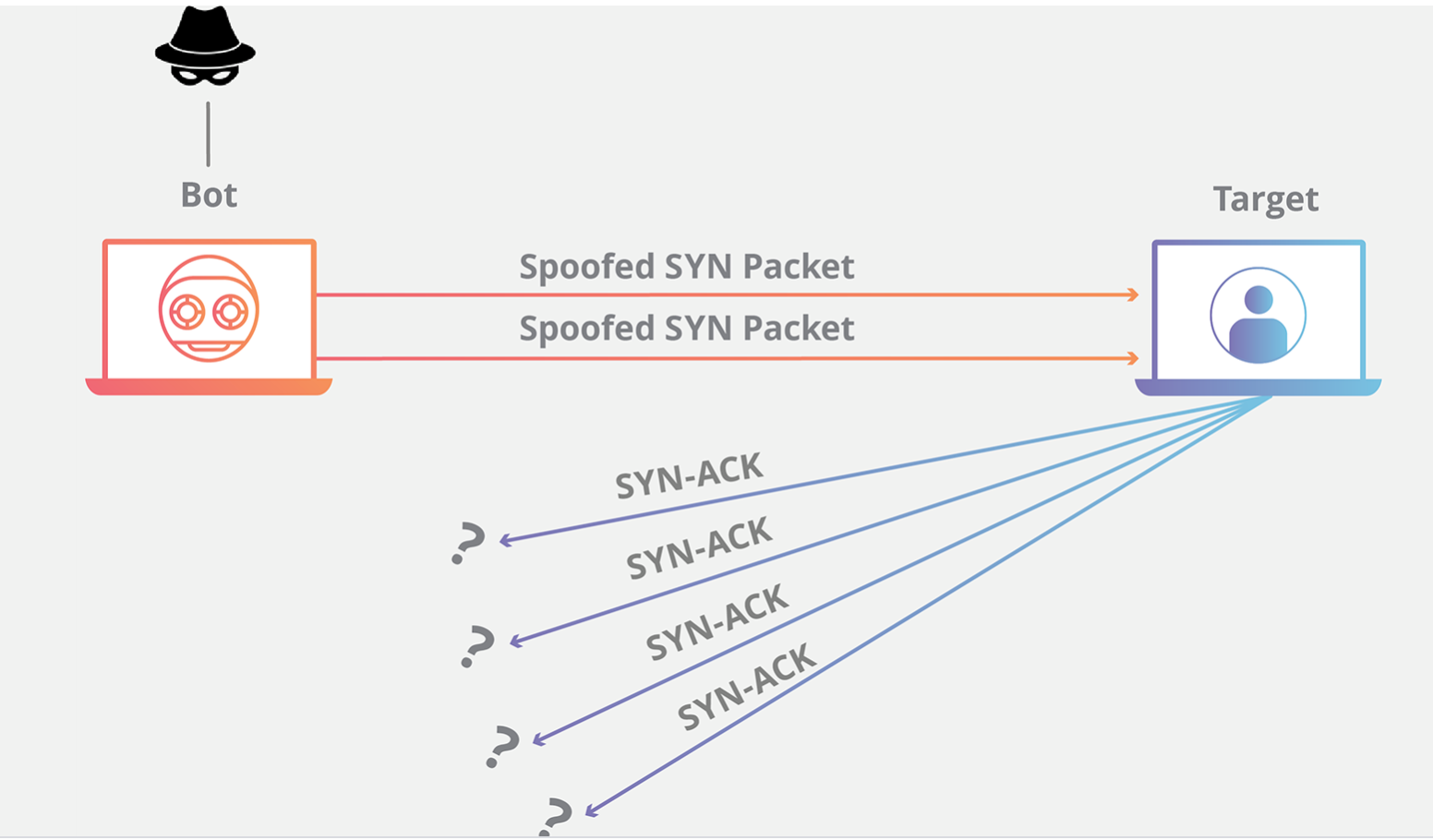
**Hvorfor er denne del af TCP relevant for en hacker?**

Et ”three-way handshake” er en metode der bruge s i TCP/IP netværk til at skabe forbindelse mellem en localhost/klient og en server.

Når serveren modtager en SYN pakke sender den en SYN,ACK tilbage og venter på en ACK. Det kan være relevant for en hacker, der kan lave et SYN FLOOT attack, hvilket vil sige at hackeren sender en stor mængde af SYN pakker til serveren. Serveren vil således besvare hver af forbindelses requests med en SYN og ACK, og reserverer derefter en port åben og klar til at modtage en ACK. Mens serveren venter på den sidste ACK, som aldrig bliver modtaget, fortsætter hackeren med at sende flere SYN requests. Hver gang serveren modtager et SYN request vil den reservere en ny port til forbindelse i en vis mængde tid. Når alle portene så er optaget, kan serveren ikke længere fungere normalt. Det er en type af Denail Of Service Attack (DoS)

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

OPGAVE 2

**TCP in action for “large documents”**

1. Du skal ændre URL I FireFox, således at den sender et request til billedet, man placerede på serveren (billed.jpg)
2. **Capture filter box:** tcp
3. Vælg i menuen: **Statistics → Packet Lengths**
4. Vælg derefter Display Filter: **tcp**

**Answer the following questions:**

* 1. **Totalt antal af TCP-pakker der sendt:** 459

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

* 1. **Lav to grupper til pakkerne (lille og stor) og identificer formålet af de to grupper af pakker (se på direction)**

Man kan se pakkerne, som deles op i små og store

* De små pakker (40-79) er kvitteringerne der sendes frem og tilbage, for at kvittere for at der er en forbindelse og sendes fra Kali Linux (ACK)
* De store pakker (5120 eller større) udgøre selve dataen (overførslen af payload) og sendes fra den lokale computer (ACK).

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse



OPGAVE 3

**IP and Routers**

Opret en droplet til både Bangladesh og Frankfurt

For at se de(n) ruter pakkerne tager, kan man i Kali Linux skrive:

sudo traceroute -I 159.89.174.41 (IP-adressen for Bangladesh)

Et billede, der indeholder sidder, bord, sort, stor

Automatisk genereret beskrivelse

sudo traceroute -I 167.71.37.53 (IP-adressen for Frankfurt)

Et billede, der indeholder sort, sidder, computer, bord

Automatisk genereret beskrivelse

Traceroute viser den rute pakkerne tager til modtageren. Hver linje definerer en identisk pakke, og pakken dør når den ikke har mere ”time-to-live” og sender en kvittering tilbage til afsenderen.

Når pakken rammer en router, tager den et liv.

\*\*\*: dette definerer at der gik noget galt, f.eks. hvis routeren ikke returnerer en kvittering.

**4) Monitoring DNS Packages**

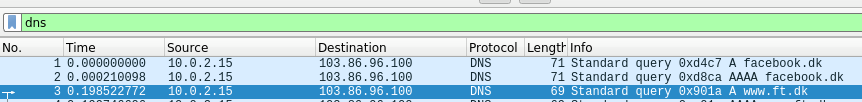
1. Start Wireshark og start capture (**DO THIS FIRST**)
2. Naviger på Firefox til **ft.dk**
3. Stop capture
4. **Display filter:** dns
5. **Hvilken protokol bruger DNS?**

UDP (User datagram protokol)

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

1. **Hvilken IP-adresse er DNS query beskeden sendt til?**



Beskeden er sendt til 103.86.96.100.

Man kan finde den nærmeste DNS server ved at skrive i Kali Linux: sudo cat /etc/resolv.conf

Ved at gøre dette kan vi bekræfte at det er den DNS server der er tættest på.

1. **Hvad navn er brugt til serveren?**

Navnet er: Domain Name Server

1. **Undersøg den retursendte response-pakke der er modtaget og find IP-adressen fra ft.dk**

IP-adressen til ft.dk: 152.115.53.91

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

1. **I Display filter skriv: tcp && ip.dst== IP\_FROM\_DNS\_RESPONSE. Blev IP-adressen, der blev returneret I DNS query-response, brugt i et efterfølgende request?**

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

OPGAVE 5

**Monitoring DHCP Traffic**

Unless hardcoded (which is possible, but not optimal for a dynamic network) then any computer that connects to a network, does not:

* initially, have an IP address to use on that network,
* know the nearest DNS-server for that network,
* know the address of the DHCP-server (which hopefully eventually will provide it with an IP).

The DHCP protocol solves this problem.

**Getting the Wireshark Sample**

1. Åbn en terminal og start en ny Wireshark capture
2. For at frigive din nuværende IP-adresse, skriv I terminalen: **sudo dhclient -r eth0**
3. For at anmode om en ny IP-lease, skriv i terminalen: **sudo dhclient eth0**
4. Stop capture.

**Filter: dhcp**

DHCP-servere står for tildeling af IP-adresser

**Først frigiv den nuværende IP-adresse:** ”*sudo dhclient -r eth0”*

**Anmod om at få en ny IP-adresse:** ”*sudo dhclient eth0”*

**Investigating the sample**

Et billede, der indeholder computer, skilt, skærm, sidder

Automatisk genereret beskrivelse**Display filter:** DHCP (eller bootp)

Tildeling af en IP-adresse indebærer 4 pakker på DHCP

1. Discover 🡪
2. Offer 🡨
3. Request 🡪
4. ACK (Acknowledgement) 🡨

* **Identificer beskederne relateret til DHCP I figuren**

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

* **Hvilken IP-adresser er destination og source for de to beskeder (Discover og Request) sendes til DHCP-server (husk at den ikke nødvendigvis kender adressen til serveren).**

**Discover**

**Source:** 0.0.0.0 – indikere at der ikke er en IP-adresse

**Destination:** 255.255.255.255 – broadcast address (alle der er forbundet på netværket).

**Request**

**Source:** 0.0.0.0 – indikere at der ikke er en IP-adresse

**Destination:** 255.255.255.255

* **Who is offering a new IP address and what was the offered address?**

DHCP tilbyder en ny IP-adresse, som er 10.0.2.15

* **Accepterede vi den tilbudte IP-adresse?**

Til den tilbudte IP-adresse, sender vi en request tilbage som broadcast, om at vi gerne vil anvende IP-adressen, dette kan anses for en accept. Efterfølgende acceptere DHCP-serveren, vores request.

* Bekræft at dette nu er din nye IP-adresse: ifconfig

Et billede, der indeholder sort, foto, sidder, hvid

Automatisk genereret beskrivelse

6) Monitoring the ARP protocol and cache

The ARP protocol typically maintains a cache of *IP-to-Ethernet address translation pairs* on your computer The arp command is used to view and manipulate the contents of this cache. Since the arp command and the ARP protocol have the same name, it’s easy to confuse them.

But keep in mind that they are different - the arp command is used to view and manipulate the ARP cache contents, while the ARP protocol defines the format and meaning of the messages sent and received, and defines the actions taken on message transmission and receipt

Getting started

1. **Find MAC-adressen på netværket:** sudo ifconfig eth0

MAC-adressen på netværket er: 08:00:27:1f:30:76

1. **Skriv i terminalen: arp -a og find MAC-adressen for DNS-serveren**

MAC-adressen for DNS-serveren er: 52:54:00:12:35:02

Brug de to MAC-adresser til efterfølgende brug.

**Getting the Wireshark Sample**

1. Åbn en terminal og start en ny Wireshark capture.
2. **Capture filter:** arp
3. For at clear cache, skriv i terminalen (en linje):

**ip link set arp off dev eth0 ; ip link set arp on dev eth0**

1. For at bekæfte at cache er tom, skriv I terminalen: **arp -a**
2. Stop capture

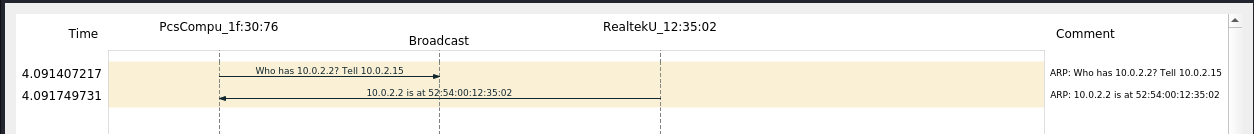
På nedenstående billede kan det ses at cache består af: MAC adressen for DNS-serveren. Dernæst er cache tom.

Et billede, der indeholder sort, foto, luk, rød

Automatisk genereret beskrivelse

**Investigating the sample**

1. Vælg i menuen: **Statistics → Flow Graph**
2. Vælg *“Limit to display filter”*



**Forklar formålet med arp-command, ARP-protokollen og adresseringsmekanismen brugt på netværkslaget.**

* **Arp-kommandoen** gør det muligt at se ARP cache, samt at kunne tilføje og slette indhold.
* **ARP** (Address Resolution Protocol) bruges imellem Ethernet og netværkslaget. ARP konverterer IP-adresser til Ethernet adresser (MAC adresser), og ARP protokollen bruges til at opretholde en cache af oversættelser af IP-adresser til MAC adresser.

ARP-protokollen definerer formatet og betydningen af beskeder sendt og modtaget.

ARP er en internetprotokol, som oversætter IP-adresser til MAC adresser (hardware) på et netkort (cache). Protokollen kan kun bruges mellem enheder på samme fysiske net. Når der første gang skal sendes en IP-pakke fra en maskine til en anden, kender afsender maskinen ikke modtageren hardware adresse, og sender derfor en broadcast meddelelse, med den ønskede IP-adresse til alle andre maskiner, på det aktuelle netværk. Den maskine der har den ønskede IP-adresse, sender sin hardware adresse tilbage og gemmer samtidig afsenderens hardware adresse til senere brug.