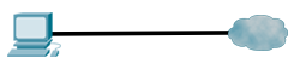
# Opgave: Labo Wireshark

## Topologie



## Doelstellingen

Na dit labo moet je in staat zijn om:

* Vanuit Wireshark het verloop van een UDP-communicatie te volgen
* Vanuit Wireshark het verloop van een TCP-sessie te analyseren
* Vanuit Wireshark het verloop van een TCP-sessie te volgen

## Scenario - deel 1: analyse van een DNS-lookup via UDP

* Sluit je PC aan op het schoolnetwerk en stel het IP(v4)-adres en de DNS-server in op ‘automatisch’.
* Noteer volgende basisgegevens voor jouw PC:

|  |  |
| --- | --- |
| MAC-adres: | 255.255.255.0 |
| IP-adres: | 192.168.0.103 |
| Default Gateway: | 192.168.0.1 |
| DNS-server: | 192.168.0.1 |

* Open een command line venster en typ als commando: **ipconfig /flushdns**
* Start Wireshark op en start een capture voor de Ethernet interface
* start je browser en surf naar <http://www.ipv6-test.com>
* Stop de capture in Wireshark zodra de pagina geladen is (en sluit ook je browser)
* Stel als display filter ‘dns’ in.
* Als je meer dan 2 frames ziet, zoek dan een frame waarin je de tekst ‘www.ipv6-test.com’ en de vermelding ‘A’ (niet: ‘AAAA’) vindt, klik daarop met de rechtermuisknop, selecteer ‘follow UDP-stream’ en sluit het pop-up venster met de ASCII-voorstelling van de communicatie.
* Selecteer het eerste frame uit de communicatie en noteer volgende informatie:

|  |  |
| --- | --- |
| destination MAC adres: | 50:d4:f7:fd:f2:82 |
| source MAC-adres: | d0:c0:bf:42:27:87 |
| destination IP-adres: | 192.168.0.1 |
| source IP-adres: | 192.168.0.103 |
| UDP source port: | 50331 |
| UDP destination port: | 53 |

* Verifieer dat beide source-adressen overeenkomen met je eigen PC, en dat het destination IP-adres overeenkomt met dat van de DNS-server die is ingesteld voor je PC.
* Bekijk en noteer de inhoud van de velden in het DNS-gedeelte.   
  Zoek op Internet naar de betekenis van deze velden (tip: google ‘dns request format’)

|  |
| --- |
| dns-query-format-5 |

* Selecteer het tweede frame uit de communicatie en noteer volgende informatie:

|  |  |
| --- | --- |
| destination MAC adres: | d0:c0:bf:42:27:87 |
| source MAC-adres: | 50:d4:f7:fd:f2:82 |
| destination IP-adres: | 192.168.0.103 |
| source IP-adres: | 192.168.0.1 |
| UDP source port: | 53 |
| UDP destination port: | 50331 |

* Bekijk specifiek de inhoud van het DNS-veld en daarin het antwoord dat de DNS-server gegeven heeft.
* Noteer het IP-adres dat je terug kreeg voor www.ipv6-test.com:

|  |  |
| --- | --- |
| IPv4-adres www.ipv6-test.com: | 51.75.78.103 |

* Stel de display filter opnieuw in op dns en selecteer nu het eerste frame waarin je ‘www.ipv6-test.com ‘ en ‘AAAA’ ziet staan, klik daarop met de rechtermuisknop, selecteer ‘follow UDP-stream’ en sluit het pop-up venster met de ASCII-voorstelling van de communicatie.
* Bekijk opnieuw beide frames en noteer volgende informatie voor het antwoord-frame:

|  |  |
| --- | --- |
| destination MAC adres: | d0:c0:bf:42:27:87 |
| source MAC-adres: | 50:d4:f7:fd:f2:82 |
| destination IP-adres: | 192.168.0.103 |
| source IP-adres: | 192.168.0.1 |
| UDP source port: | 53 |
| UDP destination port: | 50331 |
| IPv6-adres www.ipv6-test.com: | ns1.t0x.net: type AAAA, class IN, addr 2001:41d0:701:1100::29c8 |

* Sluit Wireshark NIET af: we gebruiken dezelfde data voor deel 2 van deze opgave.

## Scenario - deel 2: analyse van een TCP-connectie

Hierbij gaan we in detail een HTTP-sessie (over TCP) bekijken die gebruikt werd om de pagina [www.ipv6-test.com](http://www.ipv6-test.com) op te vragen.

* Stel in de display-filter bovenaan ‘http’ in.
* Selecteer een http-pakket met als destination IP adres het adres dat je voor www.ipv6-test.com terugkreeg van de DNS-server, klik daarop met de rechtermuisknop, selecteer ‘follow TCP-stream’.
* In het popup-venster Zie je de ASCII-voorstelling van de hele HTTP-sessie (rood = pakket vanwege de client/browser, blauw = antwoord vanwege de server).
* Bekijk eventjes de HTTP-headers. Noteer volgende informatie:

|  |  |
| --- | --- |
| ‘User agent’ van de client: | User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36 |
| Besturingssysteem server: |  |
| Gebruikte webserver: |  |

* Om verder de TCP-sessie in detail te volgen, sluit je het pop-up venster.
* Selecteer het eerste frame, en vouw in het middelste deel van het scherm het ‘TCP’-gedeelte open, en daarin het ‘Flags’ gedeelte.  
  Noteer volgende informatie:

|  |  |
| --- | --- |
| source IP-adres: | 192.168.0.103 |
| destination IP-adres: | 51.75.78.103 |
| source port: | 50921 |
| type v/d source port:: |  |
| destination port: | 80 |
| type van de destination port: |  |
| welke vlaggen zijn geset: | 0x00000002 |
| relative sequence number: | 1192866334 (1) |

* Selecteer het tweede frame en vouw ook hier in het middelste deel van het scherm het ‘TCP’-gedeelte open, en daarin het ‘Flags’ gedeelte.  
  Noteer volgende informatie:

|  |  |
| --- | --- |
| source IP-adres: | 51.75.78.103 |
| destination IP-adres: | 192.168.0.103 |
| source port: | 80 |
| destination port: | 50921 |
| welke vlaggen zijn geset: | SYN ACK |
| relative sequence number: | 2673265120 (1) |
| relative acknowledgement nr: | 1192866335 (1) |
| window size: | 29200 |

* selecteer het derde frame en noteer volgende informatie:

|  |  |
| --- | --- |
| welke vlaggen zijn geset: | ACK |
| relative sequence number: | 1192866335(1) |
| relative acknowledgement nr: | 2673265121(1) |
| window size: | 516 |

Na dit derde pakket is de TCP-connectie opgezet en kan HTTP als toepassingsprotocol hierover werken:

* Selecteer het vierde frame, verifieer dat het hier over een HTTP-pakket gaat en noteer volgende informatie:

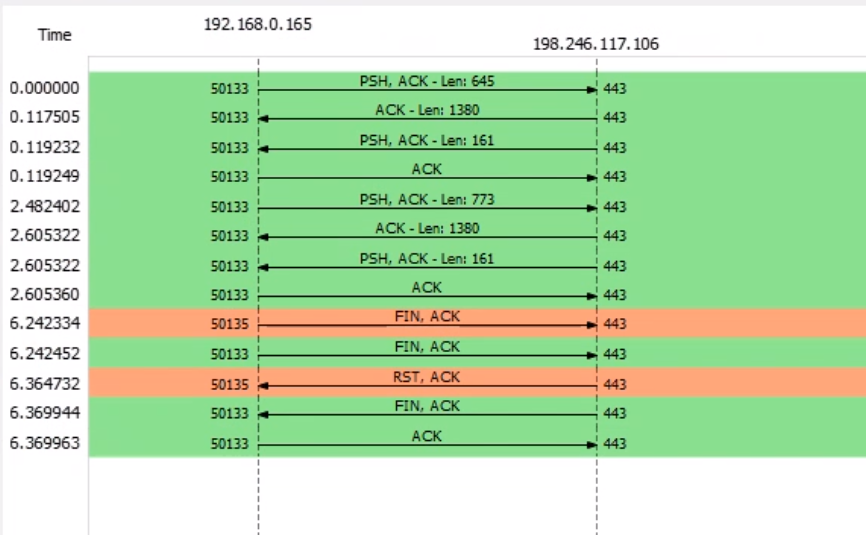
|  |  |
| --- | --- |
| welke vlaggen zijn geset: | ACK |
| relative sequence number: | 2673265121(1) |
| relative acknowledgement nr: | 1192866810(476) |

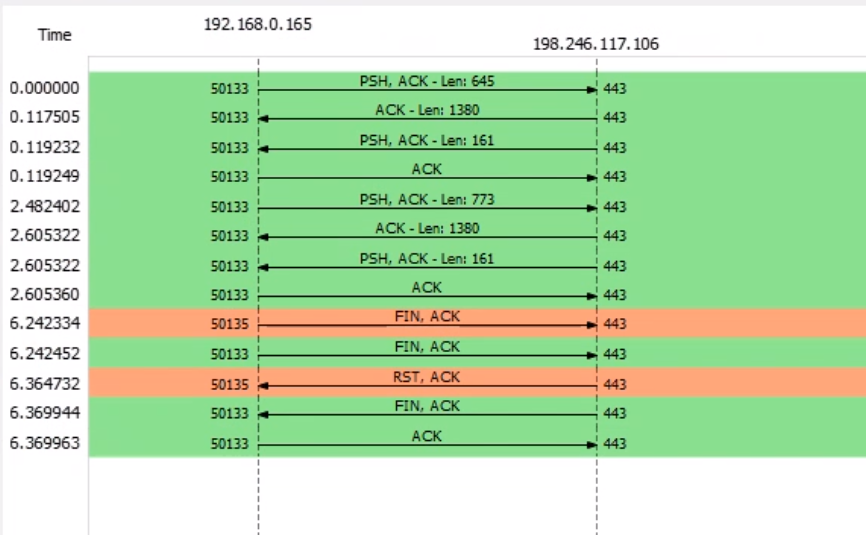
* Bekijk ook eventjes ter informatie de inhoud van het HTTP-gedeelte van dat pakket.
* Bekijk de volgende frames en leg het verband tussen relatieve sequence nummers, relatieve acknowledgement nummers en de lengte van (de data-inhoud van) de pakketten.
* Bekijk in de laatste frames van de verbinding hoe de TCP-connectie beëindigd wordt (let in het bijzonder op de FIN-vlag) en controleer of dit gebeurt zoals het protocol voorschrijft (eventueel op te zoeken via Internet).

**Opmerking**: Bij het volgen van de ‘TCP-stream’ kon je de volledige http-communicatie volgen. Behalve de headers is de inhoud echter niet rechtstreeks leesbaar. Dit komt omdat de gegevens uit de HTTP-stream gecomprimeerd zijn (zie in de headers: ‘Content-encoding: gzip’). Dit is dus geen encryptie! Je kan Wireshark vragen die content voor jou uit te pakken en zichtbaar te maken: selecteer daarvoor bij het eerste HTTP segment ‘Follow HTTP Stream’. Dan zal je de HTML, CSS en Javascript te zien krijgen waaruit de site is opgebouwd.

## Scenario - deel 3: analyse van een FTP-connectie

* ping naar ftp.cdc.gov (deze ping zal een timeout genereren) en noteer het IP-adres dat daarbij gebruikt wordt.
* start met Wireshark een capture van je netwerkverkeer, maar stel nu bij de capture options volgend **capture filter** in: ‘host aa.bb.cc.dd’ waarbij aa.bb.cc.dd het IP-adres is dat je hierboven gevonden hebt (eventueel kan je ook een **display filter** gebruiken ‘ip.addr == a.b.c.d).  
  Wireshark zal nu enkel het verkeer onderscheppen (of weergeven bij display filter) tussen jouw PC en de server op het gevonden IP-adres.
* Open je browser en ga naar volgend adres: ‘ftp://ftp.cdc.gov’
* Je krijgt dan een directory listing. Klik op ‘Readme’.
* Sluit je browser.
* Bekijk de verschillende frames in Wireshark.
* Probeer een schematische voorstelling te maken van de verschillende connecties die er gebeuren en bekijk de afzonderlijke TCP-streams. Bekijk ook het verband tussen die verschillende TCP-connecties. Zoek eventueel informatie op over de werking van het FTP-protocol.





* Voor gevorderden :) : wat is de betekenis van het ‘PASV’ commando dat de browser geeft en van het antwoord dat de server daarop geeft?