**LABO 18**

**HTTP(S)**

1. Open de capture file met als naam ***Labo18.pcapng*** (cfr Leho > Wireshark-files) en zorg ervoor dat enkel HTTP-pakketten getoond worden.
2. Filter deze capture zodanig dat enkel HTTP-pakketten getoond worden en beantwoord onderstaande vragen over het tweede http-pakket:

* Welk nummer heeft dit pakket?

220

* Is dit een http-request of een http-response pakket?

http-request

* Noteer hieronder de gebruikte http-client socket

178.117.177.241:49823

* Noteer hieronder de gebruikte http-server socket

193.191.136.205:80

* Noteer hieronder de URL van de opgevraagde website

www.howest.be

* Noteer hieronder de naam (volledig pad) van het opgevraagde document op die website

/shadowbox.css

1. Selecteer de TCP-stream waartoe het tweede HTTP-pakket behoort en beantwoord onderstaande vragen over die TCP-stream:

Rechstklikken op het tweede pakket > Follow TCP-stream en dan nieuwe venster sluiten

* Wat is het streamnummer van de TCP-stream waartoe dit pakket behoort?

21

* Hoeveel segmenten bevat deze stream?

25

* Noteer hieronder de nummers van de segmenten die deze TCP-stream geïnitialiseerd hebben.

152 – 171 – 172

1. Sinds HTTP versie 1.1 kan één enkele TCP-sessie (stream) meer dan één HTTP-transactie (bestaande uit een HTTP-request en een bijhorende HTTP-response) bevatten. Dit is bv. het geval voor de HTTP-sessie uit de vorige opdracht.

Om aan te tonen dat de TCP-sessie uit opdracht 3 uit meerdere HTTP-transacties bestaat, dien je de in vorige opdracht ingestelde filter verder te verfijnen, zodat er enkel nog rekening gehouden wordt met HTTP-transacties.

Welke filter dien je hiervoor te gebruiken? Pas deze filter toe!

tcp.stream eq 21 and http

Hoeveel HTTP-transacties bevat deze TCP-sessie volgens deze nieuwe filter?

**Aantal HTTP-transacties =** 6

1. Merk op dat je elke HTTP-transactie kan volgen via het extra venster dat geopend wordt als je rechts klikt op een HTTP-pakket en kiest voor Follow > HTTP stream.

Doe dit voor het eerste HTTP-pakket in de pakketten die je met de vorige opdracht gefilterd hebt en noteer hieronder welke kleuren gebruikt worden in dit extra venster:

**Kleur voor de HTTP-requests =** bruin

**Kleur voor de HTTP-responses =** blauw

Door goed te kijken in dit extra venster kan je nagaan welke webserver er bij deze transactie gecontacteerd werd: Nginx, IIS of Apache.

**Webserver =** IIS

1. Beantwoord onderstaande vragen over het object dat in frame nr 572 via http bij een webserver opgevraagd werd.

* Welke naam heeft het bestand waarin dit object zich bevindt? (enkel de naam, niet het volledig pad)

diploma.jpg

* Om welk soort object gaat het hier dus?

Afbeelding

Omdat het op te halen object niet uit leesbare tekst bestaat, kan je dit niet bekijken via Follow HTTP Stream. Wil je het object toch kunnen bekijken, dan moet je eerst in het details-venster van je pakket opsporen in welke frame de response te vinden is. Noteer hieronder het nummer van die frame.

**Framenummer met response =** 602

Door op de onderstreepte link met dit nummer te dubbelklikken, ga je meteen naar het aangeduide framenummer. Doen!

Je kan nu het opgevraagde object downloaden door in het details-venster van deze “response-frame” rechts te klikken op de laatste regel (met als titel JPEG File Interchange Format) en te kiezen voor de optie “Export packet bytes”. Geef als naam voor het te bewaren bestand de originele naam uit de capture.

Open daarna dit bewaarde bestand.

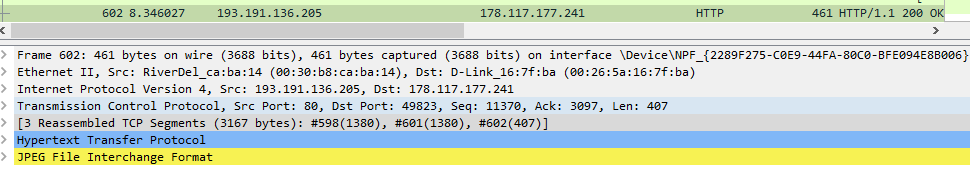
Plak hieronder een screenshot van deze afbeelding.



1. Een object dat via http opgevraagd wordt, is meestal te groot om via één enkele frame verstuurd te worden. Het framenummer met de response dat je in de vorige opdracht moest invullen, is dan meestal ook niet het enige nummer dat response-bytes bevat.

In het details-venster van de “response-frame” kan je zien welke frames response-bytes bevatten. Noteer deze hieronder.

**Framenummers met response-bytes =** 598, 601 en 602



Conclusie: het response-frame waarnaar Wireshark in de http-request verwijst, is altijd het laatste frame van alle frames met response-bytes.

Merk op dat Wireshark achter elk framenummer met response-bytes, de grootte (in bytes) van deze frames noteert. Waarom zijn deze niet allemaal even groot?

De eerste response-frames hebben de maximale grootte; enkel de laatste zal meestal minder groot zijn

1. In de opdracht 6 werd uitgelegd hoe je objecten kan bekijken die tijdens één enkele TCP-sessie werden gedownload. In deze Wireshark-sessie werden er echter verschillende TCP-sessies geopend om objecten via HTTP te downloaden.

Om een overzicht te krijgen van alle objecten die tijdens een capturesessie van een webserver gedownload werden, kan je in Wireshark gebruik maken van het File-menu en vervolgens kiezen voor “*Export Objects”* en daarna *HTTP*.

Maak van deze methode gebruik om de afbeelding met als naam ***wegenwerken.jpg*** op je laptop te bewaren. Plak een screenshot van deze afbeelding hieronder.



1. Pas een filterexpressie toe om alle HTTP GET-requests op te vragen die tijdens deze capturesessie verstuurd werden.

http.request.method eq GET (of http.request.method == GET)

Hoeveel zo’n requests zijn er dus (niet zelf tellen)?

40

Noteer hieronder de filterexpressie om alle HTTP-responses te vinden die geen OK-code hebben.

Tip: zoek de OK-code op in de PPT - Hfdst 15 - deel 1

http.response and not (http.response.code == 200)

Hoeveel zo’n responses zijn er dus?

5

1. Hoeveel segmenten in de capture file maken gebruik van HTTPS?

Tip: filter hiervoor op het protocol waarvan HTTPS gebruikmaakt (niet de oude naam gebruiken).

102

1. Volg de TLS-stream waartoe frame nr 26 behoort.

Wat wordt er in het geopende extra venster getoond? Waarom is dat zo?

Leeg venster omdat alles geëncrypteerd is…

Sluit dit extra venster en keer dus terug naar de ingestelde filter. Wat is het streamnummer van de TCP-sessie waartoe deze TLS-sessie behoort?

7

Geef het nummer van de frame waarin de gecontacteerde HTTPS-server zijn SSL-certificaat naar de client stuurt.

32