Wireshark

Everything we do on a TCP/IP network can be monitored. For these exercises, we are going to try and monitor our own network traffic and then afterwards, navigate through the data that we have recorded.

Capturing data

1) Open Wireshark and begin capturing network traffic on your network interface.
2) Navigate to the website: http://test.dk
3) Stop your Wireshark recording.

Now we are ready to start doing some analysis.

Filter packages

When we just look at the recording, we will see there is a lot of different information. Getting an overview of what we see can be very hard as it is right here. Therefore, we need to filter the information.

1) Make a filter that only shows TCP packages that are using port 80 (HTTP)
2) Try to change the filter, so it only shows packages from your IP.

Extracting files

Since we have recorded the traffic of our visit to the website http://test.dk – We have all the information that was provided by that site.

1) There was an image on the website – Find it in the recording and save if on your computer. (from the recording of course)

Follow the stream

When it comes to communication between different devices over a computer network, there is a lot of different packages send between the endpoints. So when we do a recording, we will capture all packages for a transaction. Wireshark has the option to show the content of all the packages that is connected to a single transaction.

Recording

1) Remove the previous recording, so we don't have and of the old information.
2) Start a new recording on your network interface.
3) Navigate to the websites: https://www.eal.dk then http://im2b.dk and then http://hackertyper.com
4) Stop your Wireshark recording

Filter and TCP streams

1) First filter out the packages that goes to the website http://im2b.dk
2) Right click on one of TCP packages and select the "Follow TCP Stream" from the menu.
3) Now you should be seeing a window with the communication between your device and the webserver. This is all the packages put together.

Plain vs Cipher-text

So fare, we have only looked at HTTP traffic. But since we are recording all the traffic on the network interface, we can also analyze communication for other protocols.

FTP (plaintext)

A lot of people are still using FTP (File Transfer Protocol) – This is a very bad idea, since all the communication is in plaintext (Use FTPs instead).

1) In the following video, they show how easy it is to get the username and password from a Wireshark recording, when communicating with a FTP server - https://www.youtube.com/watch?v=qdo6XtFrEvo
2) If you have access to a FTP server (for instance a VM/WebApp on Azure) – try to do a similar experiment.

Tor

Tor uses onion routing, where is access a lot of different access points (middle nodes and end nodes) before ending up at the address that you requested. It can access the “normal” http(s) protocols, but it can also access .onion sites. This is what is called “the dark web” (or “Deep web”).

Here is a list of different well known .onions:
https://www.deepweb-sites.com/top-50-dark-web-onion-domains-pagerank

Use the Tor browser to access a couple of these site (Free of choice).