# **Traffic analysis with Wireshark**

**Scenario 1: Data Extraction**

We have been provided with pcap file. There is an image that needs to be used as evidence of improper network usage. Our task is to extract image from network traffic.

**Preparation**

**Task 1: Open pre-captured file**

First let’s open pcap file. Start Wireshark and then select on toolbar File -> Open Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie



**Task 2: Filter results**

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

As we can see, there is a lot of traffic. In total, there are around 1000 packets, and less than 20 of them are HTTP packets. Our goal is to extract data from HTTP packets. Let’s filter out noise.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

We can see GET requests and OK response from webserver. We are interested in extracting data contained in packets with OK Response.

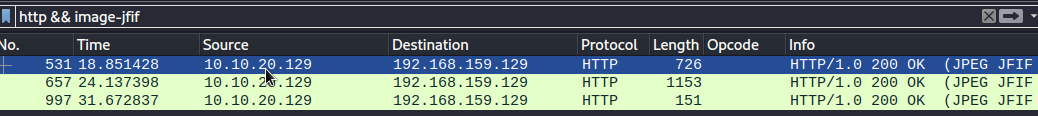
**Task 3: Follow the stream and extract the items found**

Let’s select one of OK response packets and follow their TCP stream Obraz zawierający tekst, oprogramowanie, Czcionka, numer

Opis wygenerowany automatycznie

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

Not only we see TCP handshake and data transfer but there were in total 3 JPG files that must be used for further investigation. If we filter to **http && image-jfif** we can see that those 3 files were transferred. Let’s extract them!

Click File -> Export Objects -> HTTP. And extract these 3 files. This is all for this scenario.

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie**

**Scenario 2: Live Capture**

We are requested that we now capture traffic to determine if anything else is going on from the user's host 10.129.43.4 (my own VM). We will need to start a capture, categorize and filter the data, and extract anything significant to the investigation.

**Preparation**

Let’s connect to host 10.129.43.4 via xfreerdp. We will use this host to generate traffic to capture it in real time. We will even try to extract downloaded images. We are provided with credentials.

* IP == 10.129.43.4
* Username == htb-student
* Password == HTB\_@cademy\_stdnt!

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

We connected and captured some interesting traffic on this host. Let’s analyse it.

**FTP Analysis:** going deeper into captured packets we can see FTP communication. Let’s dive in to them and try to find something interesting. Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

We found that anonymous user performed action of requesting some kind of images. Let’s extract these jpeg files. We can easily determine that 172.16.10.20 is server and 172.16.10.2 is a client.

I followed TCP stream of ftp-data packet correlated with it.

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Now, let’s change “Show and save data as” to “Raw” and save content as original file name. Now let’s open this file to check what it was

**Success!** We extracted raw data from packet, reassembled and extracted cute dog image from live capture.

**Obraz zawierający pies, ssak, Rasa psa, zwierzę domowe

Opis wygenerowany automatycznie**

**HTTP Analysis**

Our job is not over. We saw some HTTP packets as well. Let’s explore.

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

We can determine that 172.16.10.20 is webserver ran by Apache server

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

Most frequent method request was GET request. I found nothing interesting. User didn’t requested any files to extract (lab specified it as file.jpg).

**This concludes this lab exercises. Thanks for reading and I hope you found the information here useful.**

**Source:** https://academy.hackthebox.com/module/81/section/789