This homework consists of three parts, in increasing levels of difficulty. In parts 1 and 2, you will analyze network traffic. In part 3, you will write a script to modify network traffic.

1 I know what you did last minute

Note 1: For this exercise you need to download `pcap1.pcap` and `pcap2.pcap` from com301-pcap-validation.epfl.ch. Wireshark is your friend when it comes to interpreting pcap files, use it! A tutorial for Wireshark can be found here: https://www.varonis.com/blog/how-to-use-wireshark/

Note 2: You have a limited number of attempts to get a token (3 attempts per day). Check that the website you found is in the whitelist, `website_whitelist`, before submitting your answers for Parts 1.1 and 1.2. If you find several websites in the pcaps that are also in the whitelist, submit the first website present in the pcap.

1.1 Coffee Break

The semester ended a few weeks ago. The grades are out on IS-academia for COM-301, but are not visible to the students until the end of the week. You are too impatient to wait till then. Therefore, you decided to hack the account of Morty Smith, the TA who is in charge of grading the students. In the past, you saw Morty several times at this small cafe called Meeseeks. Luckily for you, the owner of the coffee shop, Mr. Meeseeks, is not tech savvy and did not change the default password of the cafe's router. After 5 minutes of trial, you obtained the router password. Now that you own the router, you are able to run a man-in-the-middle attack against Morty to monitor him and discover his secrets.

After finishing his latte, Morty starts browsing the Internet. You can finally capture some of his traffic, and you store it in `pcap1.pcap`. Unexpectedly, Morty is browsing the Internet using HTTP, everything is plain text! Can you find which website he has been visiting? Submit it to the server to get your token (remember Note 2).

Food for Thought: Some of you may have noticed that the HTTP request you got was redirected to the HTTPS version of the website. What is the mechanism behind this redirection and what is its purpose?

1.2 Let's Encrypt

You are not satisfied with the website you found, so you decide to continue monitoring Morty. Unfortunately for you, Morty noticed that the green lock symbol next to his URL was missing and finally remembered what he had learned in COM-301. He does not want to be spied on, so he starts using HTTPS. Can you find another way to find out which website he is visiting in the capture file, `pcap2.pcap`? Submit it to the server to get your token (remember Note 2). Do you see any other connection that is interesting in this capture?

2 Catch Them All!

You are still not satisfied as you couldn't find relevant information to blackmail Morty. However, the monitoring was not that unsuccessful. Thanks to what you discovered in `pcap2.pcap`, you know that Morty tried to visit a weird website belonging to the EPFL. Could it be the secret grading server? You want to continue the hack and find out if you can obtain the credentials for the database containing the grades. Once you have them, what are you going to do? Why not dump all the grades and share them with your friends?

Assume that Morty's connection to the grading server is unencrypted. You decide to deploy a man in the middle attack to steal the credentials. You have a program `hw04`, to simulate this situation. To run `hw04`, we provide binaries for 3 operating systems. Use the Linux binary if you are doing the exercise from within the VM. For your convenience, we provide binaries for Windows or MacOS X if you prefer to work from your main OS, *but the TAs will not provide support if you encounter problems when running these binaries on your main OS*. This program works in two modes:

- `mitm` to perform a man-in-the-middle to steal the credentials. Observe the traffic while running this to see if you can get the credentials.
- `dump-grades` to retrieve the grades from the database using the stolen credentials. You can use this to check that you have obtained the right credentials.

For the `mitm` mode, you need to provide your SCIPER number, and the host and port you found in the previous exercise, as arguments.

```
./hw04 mitm SCIPER hostname port
```

For the `dump-grades`, in addition to the arguments mentioned above, you need to provide the user name (an email address) and the password you found.

```
./hw04 dump-grades SCIPER hostname port email password
```

When you obtain the correct password, use that as your token for submission to the grading server.

3 Let's go Phishing

The TAs have done their job, and the connection to the grading server is now encrypted. You wonder whether you can use other material from the class to steal the credentials. What better way to do this than to make Morty connect to your fake grading server instead of the actual one? In class, you learned about changing DNS records so that the DNS response has the attacker's IP instead of the legitimate one. You decide to try this out such that when Morty sends out a DNS query for the EPFL grading server, you replace the grading server's IP with a fake entry in the DNS response.

For this exercise, please use the Python library **scapy**. Scapy is a library that allows you to perform packet manipulation. With scapy, you can read a pcap file, iterate through the packets in the file, and analyze each packet field.

scapy is already installed in the VM we provided. If you prefer not to work on the VM, you can install scapy with the command (you may need sudo):

```
pip3 install scapy
```

Your goal is to modify the trace in `pcap2.pcap`. Write a script using the skeleton code we provide, `modify_pcap.py`. The script has to take in pcap2.pcap as input, find the DNS responses for the grading server, and **replace the server's IP in the response with your SCIPER**. Write the trace to a new file `pcap2new.pcap` (the file has to contain all the packets as in the original file). Submit pcap2new.pcap at `com301-pcap-validation.epfl.ch`. If the pcap is correct, you will receive a token. We expect you to modify only the DNS records of type 'A' (IPv4). The skeleton code `modify_pcap.py` contains more details.

Food for Thought: Note that this is a simplified version of an actual man-in-the-middle attack, since you are changing a pcap file instead of modifying network traffic on the fly.

Read up about how you could perform an actual attack.

Scapy tips

Scapy can take some time to master, so we're providing a few tips to help you use it.

- Useful scapy commands for debugging: `pkt.summary()` and `pkt.show()`. `pkt.summary()` displays a short description of the packet, `pkt.show()` displays all the fields in the packet.
- Scapy follows a 'layer' system to analyze packets (at each network layer). For example, `pkt[IP]` extracts the IP layer, `pkt[UDP]` extracts the UDP layer, and so on. Once you extract the layer, the *fields* parameter can be used to access different field values in that layer. Fields are stored as dictionaries. For example, `pkt[IP].fields` will give you a dictionary of field names and values at the IP layer.
- To recalculate checksums and lengths, this discussion is useful: https://stackoverflow.com/questions/5953371/how-to-calculate-a-packet-checksum-without-sending-it
- All documentation on Scapy can be found here: https://scapy.readthedocs.io/en/latest/introduction.html.