Homework #2

Website Fingerprinting (6 points)

Deadline: 31/03/2022

Assignments

- You are requested to implement and test a website fingerprinting approach able to classify what website is visited by a user observing only encrypted HTTPS traffic

- For simplicity you can focus only on the first 10 most popular news websites (no login page):

"https://www.indiatimes.com"

"https://www.washingtonpost.com"

"https://www.ndtv.com"

"https://www.cnbc.com"

"https://www.timesofindia.com"

"https://www.express.co.uk"

"https://www.rt.com"

"https://www.news18.com"

"https://www.nypost.com"

"https://www.abc.net.au"

"https://www.bbc.co.uk"

"https://www.msn.com"

"https://www.cnn.com"

"https://www.news.google.com"

"https://www.dailymail.co.uk"

"https://www.nytimes.com"

"https://www.theguardian.com"

"https://www.foxnews.com"

"https://www.finance.yahoo.com"

"https://www.news.yahoo.com"

Assignments (2 points)

1. Construct the dataset
   - Visit each website for 10 times, capturing the packets exchanged with your client in specific .pcap files
   - Use your preferred approach:
     - Bash script invoking tcpdump + curl 10 times for every website?
     - pyshark.sniff_continuously?
     - others?
   - Hint 1: convert .pcap files into .csv with tshark for easier management with DataFrames (not mandatory, but helpful)
   - Hint 2: use a clever capture filter (avoid capturing traffic which is not part of the HTTPS exchange…) Use DNS queries or other information (e.g., think about the traffic exchanged in a Colab VM…)

Assignment (2 point)

2. Extract **biflow** features from each capture (both uplink and downlink)
    - Num packets up/down
    - Total bytes up/down
    - Min/max/mean/std packet size up/down
    - Min/max/mean/std IAT up/down

3. Create a dataset DataFrame where each row corresponds to a capture file (make sure to append the ground truth information to it!)

Assignment (2 points)

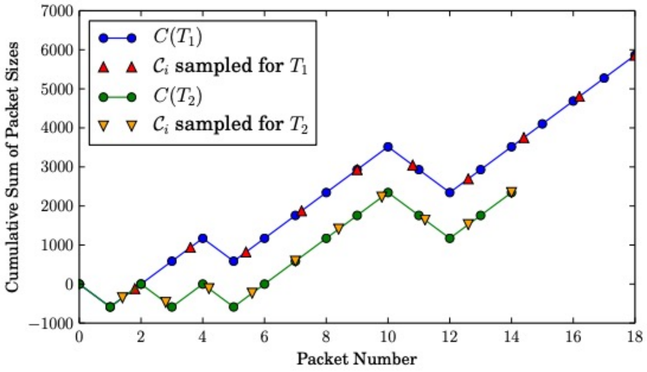
4. Split the dataset in train (70%) and test (30%) set

5. Evaluate the performance (accuracy and confusion matrix) of a k-NN approach. Plot the accuracy-vs-K relation on a figure, for k = 1..10

6. Create a new test set, visiting the same web sites (3 times) after some time (e.g., 1 day). Evaluate the performance obtained using the old training set and comment the results.

Bonus point!

- Some works in the literature propose to use the following feature for fingeprinting:
  - Look at the trace of packet sizes exchanged in the client-server exchange (p1,p2…,pn), removing TCP ACKS
  - p > 0 indicates an incoming packet, p < 0 an outgoing packet
  - Produce a cumulative trace C, where C(1) = p1, C(2) = p1+p2, C(3) = p1+p2+p3
  - Sample the piecewise linear interpolant of C at M equidistant points



A. Panchenko et al. “Website Fingerprinting at Internet Scale” NDSS 2016

Bonus point!

- Implement the approach with M = 20 and evaluate the performance with a k-NN classifier

- Compare the results with the previous approach