CSC 533: Privacy in the Digital Age (Fall 2023)
Home Assignment #3
Assigned: Friday, Sept 29, 2023, Due: Thursday, Oct. 19, 2023

Instruction: Completed homework should be typed (e.g., using LaTeX or word document) or hand-written clearly and scanned and uploaded into Moodle. You can discuss about how to use certain tools for data collection and analysis, but no collaboration is permitted to solve the problems.

1. **Learning objective:** How browser fingerprints can be extracted and comparing browsers.
   Visit the following browser fingerprinting site https://amiunique.org/ first from a Firefox Browser and then from a Tor browser (install Firefox and Tor browser if you don't have them).
   a. What are the major differences that you see for the **two browser fingerprints**? (compare the fingerprints generated for the two browsers). Include a screenshot of the fingerprints in your answer. **[points 15]**
   b. What is the basic takeaway in terms of tracking capability while using the two browsers? **[points 5]**

2. **Learning objective:** Determining the third-party requests while visiting a website. And blocking requests based on well-known tracking domains.

   You are given the HAR files (HTTP Archive format) for the following sites: *macys.com* and *cnn.com*. HAR file stores all the http requests made while loading a website. You can parse HAR files using the following parser https://gist.github.com/tomatohater/8853161 (a simple HAR parser is also provide inside the zip file).

   a. List the number of ***unique*** **third-party domains** that are loaded while visiting these two sites? **[points 20]**

      Third-party domains are domain other than the domain you are explicitly visiting (e.g., a request going to *htttps://xyz.com* while visiting *htttps://abc.com* will be considered as third party). You can use ***get_fld*** python API (https://pypi.org/project/tld/) to retrieve the top-level effective domains to determine if a request is a third-party request (if the effective top-level domain of the requesting URL doesn't match with the site you are visiting, you can assume it is a third-party request for this analysis).

      Hint: usage **get_fld("http://www.google.idontexist", fail_silently=True)**

   b. List the *unique* third-party domains that appear on **both** *macys.com* and *cnn.com* using the list of third-party domains derived from part 'a'? **[points 10]**

   c. Now let us assume we are going to do third-party domain-based **filtering**. Disconnect is a company that blocks trackers based on a filter list (https://github.com/disconnectme/disconnect-tracking-protection/blob/master/services.json). This list contains different categories of trackers and provides the list of domains owned by the tracking companies (the filter list provided in the zip file). **For example**, the following domains (**facebook.com, fbcdn.net, instagram.com, messenger.com**) are owned by Facebook and all contents from these **four domains** will be blocked.

      Determine the **number of requests** that would be blocked if you were to use Disconnect (first extract the list of all unique domains from the Disconnect.json file under all categories; next for each request URL while visiting *macys.com* and *cnn.com* compare them with the filter list to see if they would be blocked or not). Report how many of the URL requests would be blocked while visiting the two websites. **[points 20]**

      Your report should be formatted in the following manner.

Website	# of requests blocked
Cnn.com	
Macys.com	

      You need to upload the code used for generating the result (with proper README), but write down the answers in the pdf.

CSC 533: Privacy in the Digital Age (Fall 2023)
Home Assignment #3
Assigned: Friday, Sept 29, 2023, Due: Thursday, Oct. 19, 2023

3. **Learning objective:** Writing and testing Adblock plus rules.
In this question we are going to write AdBlock Plus filtering rules and cross-check them with the requests generated while visiting *cnn.com* (using the HAR file provided in question 2) to determine how many of the requests would be blocked. **adblockparser** is a python package for working with AdBlock Plus filter rules. It can parse AdBlock Plus filters and match URLs against them (https://github.com/scrapinghub/adblockparser). Using the *cnn.com* HAR file from question 2, write a code that will list the number of requests that would be blocked for the following filter rules.

   a. Block any request containing 'cookiesync?' string **[points 10]**
   b. Block any image (e.g., jpg, gif etc.) loading from `scorecardresearch.com` **[points 10]**
   c. Block any script loading from doubleclick.net **[points 10]**

   Remember to pass **all** the right **options** when checking if a URL should be blocked. For example, while visiting *example.com* if you see a HTTP request for *sample.com* then this request is a **third-party** request. Also, the HAR file contains the **content type** (like image or script) for each HTTP request. This information should also be passed as options when checking whether an URL should be blocked or not (see the example shown below).

```
Look at examples provided here -- https://github.com/scrapinghub/adblockparser

Examples of passing different options:
rules.should_block("http://ads.example.com/notbanner", {'script': True, third-party: True})

Script, image, domain, third-party options are most important
```

   Your report should be formatted in the following manner.

Rule	# of HTTP requests blocked

You need to upload the code used for generating the result (with proper **README**), but write down the answers in the pdf.

Do the following:

1. The first step is to define the blocking rules. (this is similar to the rules we covered in the class).

2. Second step is to pass the rules to the adblock object that will decide whether a URL should be blocked or not

3. Third step you iteratively go through all the URLs you encountered while visiting *cnn.com* and call the **.should_block** function of the adblock object your created in step 2 to decide if the URL request should be blocked or not. One additional thing you need to pass to the **.should_block function** along with the URL is the different attributes of the URL request like whether its content is an image or script or whether it is a 3rd-party requests. The HAR parser provides you with such details

Submission:

You have to submit three files:

1. Merge all the written parts into a single pdf file named <your unity id>_HW3.**pdf**.
2. Rename the program file you used for as <your unity id>_HW3_QX.extension (e.g., .c/.cpp/.java/.py).
3. Add a README file regarding how to run your code.

Zip all files into <your unity id>_HW3.zip and submit the zip file on Moodle.