

Measuring the Prevalence of Online Tracking

Overview:

In this project, you will conduct measurements to analyze the impact of using ad blocking extensions on randomly selected 100 websites with advertisements (list and random website selection code provided with the project). Specifically, you will compare the third-party¹ HTTP(S) requests, third-party cookies, and third-party JavaScript API calls in the “ad blocking mode” (ad blocking extension enabled) and the “vanilla mode” (ad blocking extension disabled).

You will use an open-source tool called DuckDuckGo Tracker Radar Collector (<https://github.com/duckduckgo/tracker-radar-collector>) to automatically visit websites and record HTTP(S) requests, cookies, and JavaScript API calls. You will use uBlock Origin extension for ad blocking. Instructions to add uBlock Origin are provided below.

Data collection with DuckDuckGo Tracker Radar Collector:

Vanilla mode:

1. Clone this project locally:
 - a. `git clone https://github.com/duckduckgo/tracker-radar-collector.git`
2. Go to cloned repo:
 - a. `cd tracker-radar-collector`
3. Install all dependencies:
 - a. `npm i`
4. Download the python program (`random_websites.py`) and the websites list (`website_list.txt`) that will extract 100 websites selected at random from the `website_list.txt`
 - a. Replace the placeholder WASHUID with your own and run the program (the purpose of this step is to tailor the random-number generation to each student)
5. Copy the `random_websites.txt` in the cloned repository.
6. Replace the `crawler.js` file provided by Tracker Radar with the `crawler.js` provided with the project. It will run the browser in a window in the default context, instead of the default headless mode in incognito mode.

¹ Note that an HTTP(S) request or a cookie is called third-party if they do not belong to the visited website. For example, if you visit facebook.com then requests to any domain but facebook.com will be considered third-party. Some examples of third-party requests for facebook.com would be google.com and microsoft.com. cdn.facebook.com will not be a third party to facebook.com. If you are using python, you can use <https://pypi.org/project/tldextract/> to extract domain and suffix from a request.

7. Replace `collector/RequestCollector.js` with the `RequestCollector.js` provided with the project.
8. Run the command line tool:
 - a.

```
npm run crawl -- --input-list ./random_websites.txt
--data-collectors 'requests,apis,cookies' --output ./vanilla_data/
--force-overwrite --verbose
```

Ad block mode:

1. Replace the `crawler.js` file provided by Tracker Radar with the `crawler-adblock.js` provided with the project. **Note that you need to copy the `crawler-adblock.js` code in the `crawler.js` file.** It includes the code to integrate browser extension with the crawler.
2. Download uBlock by visiting the following link:
 - a. https://github.com/gorhill/uBlock/releases/download/1.61.0/uBlock0_1.61.0.chromium.zip
3. Unzip the uBlock folder.
4. Provide the path to unzipped folder in `crawler.js` by changing the following line:
 - a.

```
const ext = 'PATH_TO_UNZIPPED_UBLOCK';
```
5. Run the command line tool (*This is different from the command above. We need to specify a different output folder for this*):
 - a.

```
npm run crawl -- --input-list ./random_websites.txt
--data-collectors 'requests,apis,cookies' --output ./adblock_data/
--force-overwrite --verbose
```

You can also use the `--crawlers <number>` to override the default number of concurrent crawlers, which is picked based on the number of CPU cores. We recommend that you do not change the default. If you run into issues related to the number of crawlers, e.g., memory etc, you can decrease the number of crawlers to 2 or 3, or even 1.

Data Collection:

The above commands will crawl the websites specified in the `random_websites.txt`. This will generate a JSON (<https://www.geeksforgeeks.org/read-json-file-using-python/>) file for each of the crawled websites in folders named “vanilla_data” and “adblock_data”. Each JSON file will have the following objects:

1. “requests”, which contains all the requests sent and received by the website, along with their headers.
 - a. HTTP cookies are present in the “set-cookie” response header.

2. “apis”, which contains a large set of JavaScript APIs called by scripts on the website. JavaScript API calls are grouped by the script that calls them.
3. “savedCalls” for some websites, contains the cookies set by different JavaScript files.
4. “cookies” contains the cookies set by JavaScript files.

Tasks:

Task 1) Follow the above instructions to visit 100 websites in the “vanilla mode” and “ad blocking mode”. **[3 points]**

Task 2) Plot the distribution of the number of third-party requests made by 100 websites in the “vanilla mode”. List the top-10 most popular third-party domains. Briefly explain the functionality of these popular third-party domains. Compare the distribution of third-party HTTP(S) requests made by the websites in the “vanilla mode” and “ad blocking mode”. Discuss the differences between the two distributions. **[4 points]**

Task 3) Plot the distribution of the number of third-party cookies (both HTTP and JS) stored by 100 websites in the “vanilla mode”. List the top-10 most popular third-party domains storing cookies on websites. Briefly explain the functionality of these popular third-party domains. Compare the distribution of third-party cookies stored by the websites in the “vanilla mode” and “ad blocking mode”. Discuss the differences between the two distributions. For the top-10 domains, manually analyze the cookies values and see what they contain. **[4 points]**

Task 4) Plot the distribution of the number of JavaScript API calls made by third-party domains on 100 websites in the “vanilla mode”. List the top-10 most popular third-party domains that are making JavaScript API calls. Briefly explain the functionality of these popular third-party domains. Compare the distribution of JavaScript API calls made by third-party domains in the “vanilla mode” and “ad blocking mode”. Discuss the differences between the two distributions. **[4 points]**

Task 5) Plot the distribution of the number of non-standard (custom) HTTP headers (a file of standard HTTP headers is provided with the assignment) on 100 websites in the “vanilla mode”. List the top-10 most popular domains that send custom HTTP headers. Briefly explain the HTTP headers and functionality of these popular domains. Compare the distribution of HTTP headers on websites in the “vanilla mode” and “ad blocking mode”. Discuss the differences between the two distributions. **BONUS [5 points]**

Submission Guidelines:

Submit all code files (e.g. python scripts) with a README.txt (include details on running code files), data files (e.g. JSON files), and report (in PDF) in a single zip file.

Contact TA or me via email or Slack if you have any questions or meet him during office hours.

- 1. You cannot share your code/data with other students in the class.**
- 2. You can take help but you are not allowed to directly copy any code/data from the Internet (including ChatGPT).**
- 3. Late submissions are not allowed unless you have prior permission to do so.**