



NWI-IMC074 Online Tracking and Privacy - Assignment 1

 Grade weight: **3.5/10** of the final grade
 Due: **5 October** 2025 (23:59, Nijmegen time)

Objectives:

- Capture and export HTTP requests and responses from website visits as HAR files.
- Write a Python script to analyze HAR data to extract privacy- and tracking-related statistics and information.

Part 1: Capture and export HTTP traffic

Capture HTTP traffic as HAR files following the instructions below. The HAR files you export will be used in the analysis parts (Part 2 & 3).

1. Create a [new Chrome/Chromium profile](#).
2. Open the DevTools/Network panel and enable [exporting HAR with sensitive headers](#).
3. Enable the Preserve log option to retain all requests made during a session.
4. Go to a website from the list given below; accept all cookies/data processing, dismiss other potential dialogs such as permission to send notifications, location access, and newsletter signup.
5. Click on a link on the homepage.
6. Scroll down until the bottom of the inner page.
7. Save all HTTP requests/responses as HAR to a file named *[registrable domain].har* (e.g., *ru.nl.har*, or [tue.nl.har](#)). Make sure to choose export HAR **with sensitive data** so Cookie and Set-Cookie headers are included.
8. Repeat steps 4-7 for a second website you select from the list below.

Websites
<ul style="list-style-type: none">• https://vu.nl/en• https://www.uva.nl/en• https://www.universiteitleiden.nl/en• https://www.tilburguniversity.edu/• https://www.tue.nl/en• https://www.ru.nl/en

Part 2: Analyze individual HAR files

Create a standalone Python script named based on your student number; such as s012345.py. Write a function (e.g. analyze_har), which takes a HAR path as input and returns a results dictionary with the following fields:

- 1) num_reqs: Integer; number of requests.
- 2) num_responses: Integer; number of responses.
- 3) num_redirections: Integer; number of redirections.
- 4) num_cross_origin_redirections: Integer; number of redirections that redirect to a URL with a different eTLD+1 than the original URL (e.g. abc.com -> def.com)
- 5) num_requests_w_cookies: Integer; number of requests with a non-empty Cookie header
- 6) num_responses_w_cookies: Integer; number of responses with a non-empty Set-Cookie header
- 7) third_party_domains: List[String]; list of distinct third-party domains (eTLD+1) contacted, excluding requests for which no response is captured.
- 8) potential_tracking_cookies: List[Tuple]; list of distinct (cookie name, cookie value, request_domain) tuples of cookies with attribute SameSite=None. Process request Cookie headers to obtain this list (avoid using the already extracted cookies field of the HAR file). Use the registrable domain extracted from the request's URL; not the cookie's domain attribute.
- 9) third_party_entities: List[String]; list of distinct entity (i.e. company/organization) **displayName**'s that own the contacted request URL domains (based on [DuckDuckGo's domain -> entity map](#)). Ignore the domains that are not included in the entity map. Use the field.
- 10) non_get_request_origins: List[Tuple]; list of distinct [origins](#) to which one or more requests with HTTP methods other than GET (e.g., POST, OPTIONS, ...) were observed. You can assume port number 443 for https connections, and port 80 for http connections.



The script should also serialize the result dict as a JSON file, keeping the same name as HAR, except the file extension (e.g. ru.nl.har -> ru.nl.json)

Part 3: Combine the results

Write a method that processes the output of analyze_har for each of the exported HAR files, and returns the following. The result should be serialized to a file named combined.json :

1. common_third_party_domains: List[String]; list of third-party domains that were contacted in both HAR files. Based on the intersection of third_party_domains fields computed in Part 2.
2. common_third_party_entities: List[String]; list of third-party entities that were contacted in both HAR files. Based on the intersection of third_party_entities lists computed in Part 2.
3. common_cookies: List[Tuple]; list of cookies with the same (cookie name, cookie value, request_domain) tuple observed on both sites. Based on the intersection of potential_tracking_cookies lists computed in Part 2.

Tips and Requirements:

- All code should be written in a single Python file, named based on your student number s012345.py.
- Running “python s012345.py” should re-generate three JSON files (e.g., ru.nl.json, tue.nl.json, combined.json) with the same content as the submitted JSONs (i.e., the results should be reproducible). Thus, your script should process the HAR files you’ve exported in Part 1, when given no command line arguments. HAR files and the script should be in the same folder, named based on your student number (e.g., s012345).
- If two HAR files are provided as command line input: “python s012345.py HAR_1_PATH HAR_2_PATH”, the script should generate the correct JSON outputs (3 files) for the given HAR files. This mode will be used to test if your script generates accurate and complete results. Your script should be able to handle HAR files of arbitrary websites, not just the two that you’ve chosen in Part 1.
- You can use publicly available Python packages, except [haralyzer](#) or similar HAR parsing libraries. You should list these dependencies in a [requirements.txt file](#).
- Your code should not make any calls to online APIs. It should work offline.
- You can print log messages from your code, but you don’t have to.
- Your code should work with Python 3
- Unless specified, “domain” means eTLD+1 (aka., *registrable domain*)
- Comment your code when what you do is not obvious
- DRY: Don’t Repeat Yourself. Break your code into reusable small functions
- Avoid deep code indentations
- Use meaningful variable and function names
 -  good: request_domain, response_headers, get_entity_by_request_url
 -  not good: foo, bar, tmp, do_stuff
- The number of requests and responses may be equal for your visits. They differ when requests are blocked or cancelled due to different reasons, and the response headers and content is completely empty.

Submission format:

Zip the s012345 folder, naming it after your student number (e.g. s012345.zip). Upload the zip file on Brightspace, under Assignment 1. The zip file should contain the following files:

- site_1.nl.har (e.g. ru.nl.har)
- site_1.nl.json
- site_2.nl.har
- site_2.nl.json
- combined.json
- s012345.py
- requirements.txt: Python packages required to run your script
- domain_map.json: The domain-entity map you downloaded and used

Relevant learning objectives:

- *understand key online tracking mechanisms;*
- *audit websites and mobile apps to identify tracking and data collection practices;*

Help:

- You can ask your questions on Brightspace, if anything is unclear:
<https://brightspace.ru.nl/d2l/le/573292/discussions/topics/137056/View>
- More clarifications will be shared in the lectures. Follow the lectures to stay up to date.

🍀 Good luck! 🍀