Automated Large-Scale Analysis of Cookie Notice Compliance

Ahmed Bouhoula, Karel Kubicek, Amit Zac, Carlos Cotrini, and David Basin, *ETH Zurich*

https://www.usenix.org/conference/usenixsecurity24/presentation/bouhoula

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 33rd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Artifact Appendices to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

USENIX Security '24 Artifact Appendix: Automated Large-Scale Analysis of Cookie Notice Compliance

Ahmed Bouhoula
ETH Zurich

Karel Kubicek
ETH Zurich

Amit Zac
ETH Zurich

Carlos Cotrini
ETH Zurich

David Basin
ETH Zurich

A Artifact Appendix

A.1 Abstract

This artifact contains the source code and instructions needed to reproduce our results. In particular, we provide experiments to reproduce both our crawls and the evaluation results of our machine learning models.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

We ignore `robots.txt`, which we believe is meant for search engines and not research crawlers. As per the ethics committee at our institution, we do not require ethics approval for our crawling study since it does not involve human subjects. Moreover, our crawl does not harm website owners, since the computational costs are negligible and our findings are published in aggregated form without exposing individual websites. Regarding the legal aspect, we considered different legal regimes and concluded that our research does not violate laws such as fraud, trespass, or breach of contract since our intentions are to carry out good-faith privacy research. We have not identified any security or privacy concerns for website owners stemming from our crawls.

Although our artifact requires root privileges for Docker to be executed, there are no security or privacy risks to the host machine.

A.2.2 How to access

The artifact is available on GitHub: `https://github.com/bouhoula/alsacnc/releases/tag/v1.0.3`

A.2.3 Hardware dependencies

The artifact can run on any machine with an `x86-64` architecture CPU and the necessary software dependencies. We used a machine with a 16-core CPU (AMD 5950X), 64GB RAM, and an RTX 3080 Ti GPU. Machines with different performance may require to adjust the `--num_browsers` parameter, which indicates the number of parallel browsers used by the crawler. As a rule of thumb, modern CPUs can handle two browsers per physical core.

We recommend running the artifact with 50 GB of free storage on the root partition.

While a GPU is not mandatory, it significantly speeds up the training times of the BERT models in experiment (E2). To use a GPU, please adapt the `docker/docker-compose.yaml` according to the instructions under the `classifier` and `ml-eval` services.

A.2.4 Software dependencies

The artifact requires Docker and Docker Compose. As for the host machine, we tested various Linux distributions without issues.

The crawler uses the ports 5000, 5001, and 5432. Please make sure that these ports are available or adapt `docker/docker-compose.yaml` to use different ports.

A.2.5 Benchmarks

Benchmarks do not apply to our artifact. However, the machine learning models are trained on the following datasets:

- A dataset of 400 cookie notices annotated by Santos et al. [45]. This dataset is not included as it is not public. However, it is available upon request by contacting the authors.
- A dataset of interactive elements created and annotated by us, which is included in the GitHub repository.
- The dataset of labeled cookies created by Bollinger et al. [4], available at `https://zenodo.org/records/5838646`.

Our crawler visits a list of websites generated using the Chrome User Experience Report (*CrUX*) [10]. By default, the crawler uses the same list as we used in our study, which is included in the repository. When you wish to modify the CrUX list generation using the parameters in `config/experiment_config.yaml`, you must provide an API key to download the new list. Details about setting up the API key are included in the repository's README.

A.3 Set-up

A.3.1 Installation

Clone the GitHub repository and install Docker and Docker Compose. To run Docker without root privileges, add your user to the docker group `sudo usermod -a -G docker $USER`. Exit your current terminal session and start a new one for the changes to take effect.

A.3.2 Basic Test

Run `./run.sh --test`. If this command does not encounter errors, the Docker environment was successfully built. A typical failure can be caused by insufficient storage.

This commands also starts the LibreTranslate container, which has to download several models upon creation and remains unhealthy until the download is complete. Please wait until the container is healthy before running the crawl. You can monitor this by using the `docker ps` command.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): We conduct the largest case study to date on cookie purpose compliance, covering 97k websites. We report on potential GDPR violations and dark patterns. For instance, we find that 65.4% of websites do not respect users' negative consent. Figure 4 includes our findings regarding other violations and dark patterns. Our sample is heterogeneous and removes the selection bias specific to a subset of websites implementing certain CMPs. In particular, as shown by Section 6 and Figure 7, restricting the analysis to websites implementing certain types of CMPs leads to biased observations.

A.4.2 Other Claims

(C2): We made other claims related to the performance of our machine learning models as described in Section 4.

A.4.3 Experiments

(E1): *[Crawl] [30 human-minutes + 1 compute-hour for each 300 websites + 50GB disk + 10 Mbit/s network I/O on average]*: This experiment involves crawling (a subset of) the 97k websites covered in our paper. Note that, since the full crawl may take several days, the number of crawled websites can be restricted using the `--num_domains` argument. For example, `--num_domains 1000` randomly samples 1000 websites to crawl from our list.

How to: Once the environment is set up as described in Section A.3, run the following command and adapt its arguments if needed: `./run.sh --crawler --num_domains 1000 --num_browsers 30`. Once the Docker container finishes the execution, get the corresponding experiment ID by running: `./run.sh --ls`. Then, process the crawl results by: `./run.sh --predictor --experiment_id <experiment_id>`. Finally, the following command displays a summary of the crawl results: `./run.sh --summary --experiment_id <experiment_id>`.

Results: The last command displays a summary of the crawl results. In particular, this includes various violations and dark pattern ratios included in our paper. Please note that the results may vary, since they depend on the following factors.

Time: The current prevalence of violations and dark patterns may have evolved since the May 2023 crawl described in the paper.

Location: Usage of cookies and cookie consent depends on jurisdictions, so performing the crawl from a different country could lead to different results. In particular, if you use an IP address from outside the EU, one may encounter much fewer cookie notices than we observed in the crawl from Germany.Note that it is possible to use a custom proxy for the crawl, which you can specify in `config/experiment_config.yaml`.

Variance: Crawling a smaller sample of websites may lead to results that differ from those obtained from all 97k websites.

Further data: The rank-based analysis conducted in our paper requires a separate crawl with the `--maximum_rank` parameter set to 500000. An API key is required to download CrUX domains. Details about setting up the API key are included in the repository's README.

(E2): *[ML evaluation] [15 human-minutes + 2.5 compute-hours + 50GB disk]*: This experiment involves reproducing the evaluation of the machine learning models described in Section 4. Note that some randomness aspects were not properly fixed in our experiments, and fixing these issues leads to negligibly different scores than what we reported in Section 4. The evaluation only covers Sections 4.2 and 4.3, since results from Section 4.1 require a dataset which is available upon request by contacting Santos et al. [45].

How to: Run `./run.sh --ml-eval`. This will first run the evaluation of the declared purpose and interactive elements models if their respective datasets are available in the `data/` directory. You can move the dataset files to a different location to skip this part of the evaluation. Next, the script runs the evaluation of the cookie classification model.

Results: The evaluation of each of the declared purpose and interactive elements models shows logs for individual cross-validation folds that can be ignored, followed by the cross-validation results. The evaluation of the cookie classification model shows results that can be compared to those included in Table 1.

A.5 Notes on Reusability

If you intend to re-use our artifact for your own research, you might want to update the version of OpenWPM used by our crawler as this allows you to use a more recent version of Firefox. Although this should improve the results, some OpenWPM updates include significant changes that require some effort to integrate properly. You should also refer to `config/experiment_config.yaml` for customization options.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2024/.