



USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

Websites' Global Privacy Control Compliance at Scale and over Time

Katherine Hausladen, Oliver Wang, and Sophie Eng, *Wesleyan University*;
Jocelyn Wang, *Princeton University*; Francisca Wijaya, Matthew May,
and Sebastian Zimmeck, *Wesleyan University*

<https://www.usenix.org/conference/usenixsecurity25/presentation/hausladen>

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.



USENIX Security '25 Artifact Appendix: Websites' Global Privacy Control Compliance at Scale and over Time

Katherine Hausladen¹, Ebuka Akubilo¹, Matthew May¹, Francisca Wijaya¹, Jocelyn Wang², Oliver Wang¹, Sophie Eng¹, and Sebastian Zimmeck¹

¹*Department of Mathematics and Computer Science, Wesleyan University*

²*Department of Computer Science, Princeton University*

A Artifact Appendix

A.1 Abstract

The California Consumer Privacy Act (CCPA) and its regulations give California residents the right to opt out of the sale or sharing of their personal information. Similar privacy rights exist in other jurisdictions as well. These rights can be exercised via Global Privacy Control (GPC) [9]. GPC is a binary privacy preference signal that can be implemented in browsers, browser extensions, mobile operating systems, and other software. GPC signals are sent via HTTP headers or can be set via a JavaScript DOM property. When a first-party site receives a GPC signal, it must propagate users' opt out preferences to integrated third parties, such as ad networks. This propagation is done using privacy strings, such as the Global Privacy Platform (GPP) String [5]. To evaluate websites' compliance with the CCPA opt out right expressed via GPC, we implemented our main artifact: a GPC web crawler based on the Selenium WebDriver [8] and a browser extension for analyzing privacy string propagation. We based our browser extension on the OptMeowt extension in analysis mode [6], as described and implemented by Zimmeck et al. [10]. Our browser extension identifies sites' GPC compliance by sending GPC signals to sites and detecting the presence of privacy strings, their values, and value changes in response to receiving GPC signals. In addition to our GPC web crawler we also provide a Python script for decoding GPP Strings, crawl data, and privacy policy analysis results, the latter of which we used to identify selling and sharing of personal information from first to third parties.

A.2 Description & Requirements

The artifact requires a consumer-grade computer with Docker [3] and Python [7] (version 3.10 or later) installed.

A.2.1 Security, privacy, and ethical concerns

None.

A.2.2 How to access

All artifacts are available in our Zenodo repository [4].

A.2.3 Hardware dependencies

While there are no particular hardware dependencies, we tested the functionality described here using a MacBook with macOS operating system.

A.2.4 Software dependencies

We use Docker [3] for the GPC web crawler and Python [7] for the GPP privacy string decoding.

A.2.5 Benchmarks

None

A.3 Set-up

1. Create a Docker account [2].
2. Authenticate to Docker Hub [1].
3. Install Python [7].

A.3.1 Installation

- Download and unzip *gpc-web-crawler.zip* from the Zenodo repository.
- Download and unzip *cmp_api_python.zip* from the Zenodo repository.

A.3.2 Basic Test

GPC Web Crawler Test

1. Open a new terminal session.
2. Navigate to the *gpc-web-crawler-main* directory. This directory was created by unzipping *gpc-web-crawler.zip*.

- From within that directory run `make custom`.
- Once completed, the results from the crawl are saved to the `./crawl_results` directory. The directory the results are saved to is prefixed with "CUSTOMCRAWL" and suffixed with the timestamp you started the crawl with. Verify that there is an entry for `yelp.com` in each of the `analysis.json` and `well-known-data.csv` files.

Python Installation Check

- Open your terminal.
- Run the following command:

```
python3 --version
```

- If the command returns a version earlier than 3.10 or indicates Python is not installed, download and install Python 3.10 or later from the official website: <https://www.python.org/downloads/>.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *We identify sites' compliance with the CCPA opt out right when receiving GPC signals at scale and over time.*

A.4.2 Experiments

(E1): *[~10 compute-minutes]: The main experiment is running the GPC web crawler on a batch of sites. The result of the experiment is a dataset of 10 sites' privacy string behavior when receiving GPC signals.*

Preparation: It is assumed that you have followed the set-up instructions (§A.3).

Execution: Run the experiment as follows:

- Open a new terminal session.
- Navigate to the `gpc-web-crawler-main` directory that you created when unzipping `gpc-web-crawler.zip`.
- Ensure no Docker containers are running by executing the command `make stop`.
- Run `make start`.
- When prompted to select a batch (0-8), select 0.
- Once completed, the result are saved to the `./crawl_results` directory. The directory the results are saved to is prefixed with "CRAWLSET0" and suffixed with the timestamp you started the crawl with.

Results: Verify the contents of `CRAWLSET0.analysis.json` and `well-known-data.csv` should contain the privacy string analysis results for 10 sites. An error-logging subdirectory will also be created, recording extra data collected during crawl failures. In that subdirectory, there will be a file named `error-logging.json`, a JSON file associating domains with the exception that caused the crawler to fail. If possible, a screenshot of the site will also be taken and saved in this subdirectory.

(E2): *[~5 developer minutes]: The second experiment consists of decoding GPP Strings.*

Preparation: It is assumed that you have followed the set-up instructions (§A.3).

Execution: Run the experiment as follows:

- Open a new terminal session.
- Extract the file `cmp_api_python.zip`.
- Navigate into the resulting `cmp_api_python` directory:

```
cd cmp_api_python
```

- Run the decoding script with Python:

```
python3 decode.py
```

- When prompted, enter the GPP string `DBABBg~BUUAAAGA.YA`.

Results: Verify the output matches the following object:

```
{ "uscav1": {
  "Version": 1,
  "SaleOptOutNotice": 1,
  "SharingOptOutNotice": 1,
  "SensitiveDataLimitUseNotice": 0,
  "SaleOptOut": 1,
  "SharingOptOut": 1,
  "SensitiveDataProcessing": [0, 0,
0, 0, 0, 0, 0, 0, 0],
  "KnownChildSensitiveDataConsents":
[0, 0],
  "PersonalDataConsents": 1,
  "MspaCoveredTransaction": 2,
  "MspaOptOutOptionMode": 0,
  "MspaServiceProviderMode": 0,
  "GpcSegmentType": 1,
  "Gpc": True,
}
}
```

A.5 Notes on Reusability

You can use your own set of sites to crawl with the GPC web crawler and GPP string to decode with the GPP String decoder.

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/usenixsec2025/>.

References

- [1] Docker Inc. Authenticate to Docker Hub with web-based login. <https://docs.docker.com/reference/cli/docker/login/#authenticate-to-docker-hub-with-web-based-login>. Accessed: April 3, 2025.
- [2] Docker Inc. Create your account. <https://app.docker.com/signup>. Accessed: April 3, 2025.
- [3] Docker Inc. Install Docker Engine. <https://docs.docker.com/engine/install/>. Accessed: April 3, 2025.
- [4] K. Hausladen, E. Akubilo, M. May, F. Wijaya, J. Wang, O. Wang, S. Eng, and S. Zimmeck. GPC Web Crawler. <https://doi.org/10.5281/zenodo.14729170>. Accessed: April 3, 2025.
- [5] IAB Tech Lab. Global-Privacy-Platform. <https://github.com/InteractiveAdvertisingBureau/Global-Privacy-Platform>. Accessed: April 3, 2025.
- [6] privacy-tech-lab. OptMeowt. <https://github.com/privacy-tech-lab/gpc-optmeowt/tree/v3.0.0-paper>. Accessed: April 3, 2025.
- [7] Python Software Foundation. Python. <https://www.python.org/>. Accessed: April 3, 2025.
- [8] Software Freedom Conservancy. WebDriver. <https://www.selenium.dev/documentation/webdriver/>. Accessed: April 3, 2025.
- [9] S. Zimmeck, P. Snyder, J. Brookman, and A. Zuckerscharff. Global Privacy Control (GPC). <https://w3c.github.io/gpc/>. Accessed: April 3, 2025.
- [10] S. Zimmeck, O. Wang, K. Alicki, J. Wang, and S. Eng. Usability and Enforceability of Global Privacy Control. In *23rd Privacy Enhancing Technologies Symposium (PETS 2023)*, pages 265–281, Lausanne, Switzerland, July 2023. PETS.