The Security Lottery: Measuring Client-Side Web Security Inconsistencies

Sebastian Roth, *CISPA Helmholtz Center for Information Security;* Stefano Calzavara, *Università Ca' Foscari Venezia;* Moritz Wilhelm, *CISPA Helmholtz Center for Information Security;* Alvise Rabitti, *Università Ca' Foscari Venezia;* Ben Stock, *CISPA Helmholtz Center for Information Security*

https://www.usenix.org/conference/usenixsecurity22/presentation/roth

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

A Artifact Appendix

A.1 Abstract

Our artifact is the code used for our data collection and the analysis of the collected data. For the results of our paper, we focus on the data of our first crawl (2nd January 2022). We executed the scripts on a machine with 4 Intel(R) Xeon(R) Platinum 8160 CPUs (192 Cores total), 1.5 TB of RAM, and a one Gbps network connection. This execution took less than two days but will take longer if a machine has less CPU Power or a slower network connection. We provide a docker-compose config file to execute our pipeline inside a docker container regarding the software requirements. In this case, any machine that can build and spawn docker containers should work. We also provide files, installation scripts, and requirements.txt files for direct execution on a Linux system. In the appendix of our paper, we also listed other crawls and their overlaps between the first and follow-up crawls (Paper Appendix B). The results of the follow-up crawls, and thus also the results of the artifact execution, highlight that our results can be confirmed over multiple crawls. The expected results of the execution of our artifact is a table similar to Table 4 of our paper. Notably, the results might vary slightly (see Paper Appendix B), especially now the numbers might be lower due to our notification campaign (see Paper Section 6.4 Disclosure).

A.2 Artifact check-list (meta-information)

- **Data set:** Tranco List from 01-01-2022
- **Hardware:** In general no restrictions, but depending on the CPU Power and Network speed it will take longer.
- **Security, privacy, and ethical concerns:** Crawl process might put load on the crawled servers.
- **Metrics:** Number of sites that have inconsistent security configurations.
- **Output:** The script will print on console output individual numbers as well as a latex table similar to the paper's Table 4.
- **Experiments:** Results might vary slightly (see Paper Appendix B), especially now due to our notification campaign.
- **How much disk space required (approximately)?:** ~80GB
- **How much time is needed to prepare workflow (approximately)?:** Docker Setup: 5-10 min; Manual Setup: 10-15 min.
- **How much time is needed to complete experiments (approximately)?:** Highly depends on CPU/Network speed, for us the crawl took less than two days.
- **Publicly available (explicitly provide evolving version reference)?:** GitHub Repository incl. version history[1]
- **Code licenses (if publicly available)?:** AGPL-3.0 license
- **Archived (explicitly provide DOI or stable reference)?:** Stable reference to Git Commit[2].

A.3 Description

A.3.1 How to access

We made our pipeline publicly available via GitHub[1]. The stable reference for the submitted version is the commit where we incorporated the feedback of our reviewers[2].

A.3.2 Hardware dependencies

In general no restrictions, except ~80GB free space and enough CPU Power and RAM to perform HTTP requests. We executed the scripts on a machine with 4 Intel(R) Xeon(R) Platinum 8160 CPUs (192 Cores total), 1.5 TB of RAM, and a one Gbps network connection. The execution took us less than two days but will take longer if a machine has less CPU Power or a slower network connection.

A.3.3 Software dependencies

We provide a docker-compose config file to execute our pipeline inside a docker container regarding the software requirements. In this case, any machine that can build and spawn docker containers should work. We also provide files, installation scripts, and requirements.txt files for direct execution on a Linux system. The installation script will install `python3`, `pip`, `curl`, `tor`, `openvpn`, `psmisc`, `wget`, and `fping`. For the execution of the python scripts the *requirements.txt* contains `requests`, `tldextract`, `psycopg2`, `beautifulsoup4`, `lxml`, and `pysocks`.

A.3.4 Data sets

As a list of sites where we want to investigate if they have inconsistent behavior we used the Tranco List of Top sites from 01-01-2022 (see `scripts/xvwn_20220101.csv` in our repository).

A.3.5 Models

N/A

A.3.6 Security, privacy, and ethical concerns

Not too many of the crawls should be conducted in parallel because it might put load on the crawled Web sites, which might interfere with their availability or response speed.

A.4 Installation

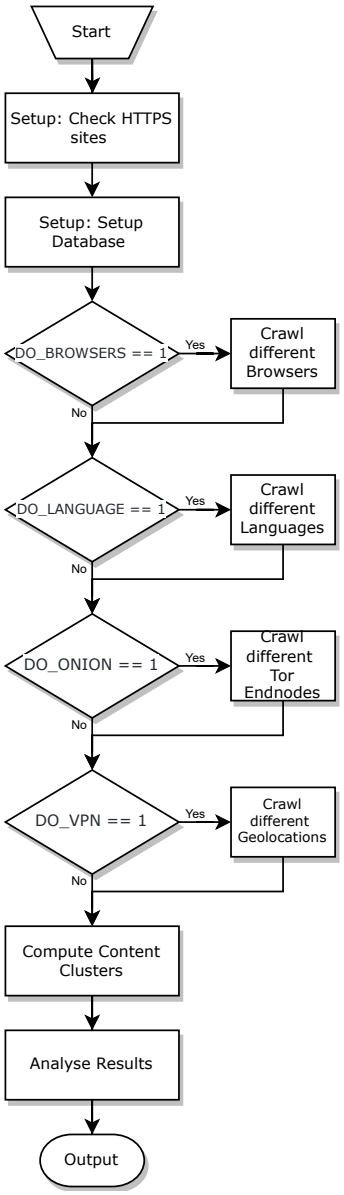
For the **docker** way you need to install the `docker`, `docker-compose`, and if you want to also crawl VPNs the `openvpn` package. Afterwards you can configure the crawl by editing the `docker-compose.yaml` file according to the GitHub README.md. Then you can build the docker via executing `docker-compose build` in the root folder of our repository. For **manual** setup you can execute `install.sh` which installs all

[1] https://github.com/cispa/the-security-lottery

[2] https://github.com/cispa/the-security-lottery/tree/66cc012fe7603e1758dde68fe9eec2d23542968e

dependencies. Then you should also set your environment variables to the values that are specified within the `environment` part of the `docker-compose.yaml` file. Afterwards you can execute the pipeline by executing `start.sh` or following the steps specified in the Repo's README.md.

A.5 Experiment workflow



A.6 Evaluation and expected results

Our measurement shows that a significant fraction of the Top Web sites suffer from different types of client-side security inconsistencies. Remarkably, the inconsistencies of 194 sites can be attributed to specific client characteristics, which identify weak spots in the security configuration, while the inconsistencies of 127 other sites can be attributed to non-deterministic factors, which may nevertheless be exploitable by an attacker.

Detailed numbers for the detected intra-test and inter-test inconsistencies for each factor and security mechanism are presented in Table 4 of the paper. Here we also present the numbers with and without page similarity for HSTS to highlight the impact of this choice on the measurement. The numbers in this table are also the output of our main analysis script (`scripts/sql_table.py`), which is executed as the last part of our pipeline. Furthermore details about the specific inconsistencies are saved into the `/data` directory as JSON files. Notably, the current numbers might be lower due to our notification campaign (see Paper Section 6.4 Disclosure), where we notified the affected parties about the problem and got responses that the issue had been fixed. Also, due to the nature of some of the inconsistencies, especially the intra-test inconsistencies, the number in general, varies as we depicted in Appendix B of the paper.

A.7 Experiment customization

One can specify which of the crawls should be performed by changing the corresponding DO_<BROWSER|LANGUAGE|ONION|VPN> values to 1=enabled or 0=disabled. Notably, when you want to execute the crawls individually, you need to set SKIP_SETUP to 1 after the first crawl, such that the database is not cleared. Also, note that the VPN crawl requires valid hidemyass.com credentials.

A.8 Notes

The results might vary slightly from ours due to the nature of inconsistencies (see Paper Appendix B). Especially now, the numbers might be lower due to our notification campaign (see Paper Section 6.4 Disclosure), where we notified the affected parties about the problem and got responses that the issue had been fixed.

Also, note that the VPN crawl requires valid hidemyass.com credentials in order to work.

A.9 Version

Based on the LaTeX template for Artifact Evaluation V20220119.