SoK: State of the Krawlers – Evaluating the Effectiveness of Crawling Algorithms for Web Security Measurements

Aleksei Stafeev and Giancarlo Pellegrino,
CISPA Helmholtz Center for Information Security

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

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

USENIX Security '24 Artifact Appendix: SoK: State of the Krawlers – Evaluating the Effectiveness of Crawling Algorithms for Web Security Measurements

Aleksei Stafeev
CISPA Helmholtz Center
for Information Security

Giancarlo Pellegrino
CISPA Helmholtz Center
for Information Security

A Artifact Appendix

A.1 Abstract

Starting from the past 12 years of the top security, web measurement, and software engineering literature, we categorize and decompose in building blocks crawling techniques and methodologic choices. We share the results of this survey. We then reimplement and patch crawling techniques and integrate them into Arachnarium, a framework for comparative evaluations, which we use to run one of the most comprehensive experimental evaluations against nine real and two benchmark web applications and top 10K CrUX websites to assess the performance and adequacy of algorithms across three metrics (code, link, and JavaScript source coverage). We open-source our framework. In addition, our results show a complex relationship between experiment parameters, the study's domain, and the available computing resources, where no single best-performing crawler configuration exists. We believe that the collected data still holds more insights. We release the data to encourage researchers to use it in new studies.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

In Arachnarium we use official Docker images and application source codes.

Our scripts are available in human-readable text format which allows manual inspection in case of concerns.

A.2.2 How to access

The artifact is available at the following GitHub links.

Arachnarium:
https://github.com/pixelindigo/arachnarium/tree/sec24

Experiments data: https://github.com/pixelindigo/state-of-the-krawlers/tree/sec24

A.2.3 Hardware dependencies

Our artifact is best evaluated on a x86_64 CPU. Arm processors (Apple Silicon in particular) could have issues running some of the web applications as there are no native arm docker images for these.

A.2.4 Software dependencies

Arachnarium and all scripts support systems on modern GNU/Linux with Python 3, Docker, and docker-compose installed.

We tested our tool on Ubuntu 22.04.

Docker can be set up with the following commands:

Install docker packages and python venv

```
sudo apt update && sudo apt install -y
docker.io docker-compose python3-virtualenv
```

Add your user to docker group

```
sudo usermod -aG docker user
```

Enable docker service

```
sudo systemctl enable docker
```

Reboot system and ensure that your user is in docker group

`id` command output should contain `docker`

A.2.5 Benchmarks

None

A.3 Set-up

First, the environment should be prepared as outlined in A.2.4.

A.3.1 Installation

Clone the repository with experiment data

```
git clone https://github.com/pixelindigo/
state-of-the-krawlers.git -b sec24
```

Clone Arachnarium repository

`git clone https://github.com/pixelindigo/arachnarium.git -b sec24 && cd arachnarium`

Create and activate a virtualenv

`virtualenv .env && source .env/bin/activate`

Install Arachnarium

`pip install .`

A.3.2 Basic Test

`arachnarium run examples/crawlers/simplecrawler examples/apps/simplewebapp -t 1 http://web/` command should produce an experiment in `experiments/simplewebapp/simplecrawler/*/` folder. No errors should have been logged in the standard output.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): Arachnarium can be used to perform experiments as in Section 4.1 as demonstrated in (E1).

A.4.2 Experiments

(E1): [30 human-minutes + 10 minutes + 20GB disk]: Arachnarium works with real crawlers and web applications.
How to: Run the experiments
Preparation: Make sure you are in the virtual python environment with Arachnarium installed and navigated to the `state-of-the-krawlers` directory.
Execution: Run `arachnarium batch -w 4 arachnarium_examples/tools.yml`.
Results: After 5 minutes the command should produce results in `experiments/hotcrp/<crawler>/*/` folder. No errors should have been logged in the standard output. The experiments should contain `coverage/` folder with coverage files.

A.5 Notes on Reusability

The Arachnarium can be extended with additional crawler and application modules. To achieve that the user would need to create a docker-compose file with a similar structure to the ones provided in the artifact.

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/.