ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

USENIX Security '25 Artifact Appendix: No Way to Sign Out? Unpacking Non-Compliance with Google Play's App Account Deletion Requirements

Jingwen Yan[†], Song Liao[‡], Jin Ma[†], Mohammed Aldeen[†], Salish Kumar[‡], Long Cheng[†]
[†]Clemson University, {jingwey, jin7, mshujaa, lcheng2}@clemson.edu
[‡]Texas Tech University, {song.liao, salikuma}@ttu.edu

A Artifact Appendix

A.1 Abstract

In this work, we developed a tool named DELETETRACKER to semi-automatically detect various account deletion issues as defined in the paper. DELETETRACKER comprises four modules: Account Deletion Link Crawler, Account Deletion Link Analyzer, Account Deletion Method Classifier, and Account Deletion Validator. The Account Deletion Link Crawler automatically collects account deletion information from both the Google Play mobile app and website, preparing for subsequent analysis. Next, the Account Deletion Link Analyzer examines the availability, cross-platform inconsistency, accessibility, and similarity of account deletion links. The Account Deletion Method Classifier detects content issues, including missing account deletion method, alternative account deletion method, missing web-based account deletion method, and missing in-app path account deletion method. Finally, the Account Deletion Validator verifies whether users can successfully delete their accounts using the methods provided in the apps' account deletion links. Since the Account Deletion Verifier is performed manually, we do not include this module in the artifact. This artifact is intended to demonstrate the functionality of the first three modules of DELETETRACKER, as described in Sections 4.1, 4.2, and 4.3 of our paper.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Our artifact does not execute destructive actions, modify system settings, or disable any security mechanisms. However, we would like to note the following considerations:

- **Crawler Rate Limiting:** When using the Account Deletion Link Crawler, evaluators should avoid sending an excessive number of requests in a short period of time. High-frequency scraping may trigger rate limiting or temporary access blocks by Google.
- **OpenAI API Key Usage:** If evaluators use the provided OpenAI API key for the Account Deletion Method Classifier, care must be taken not to expose or misuse the key. The key is intended for demonstration purposes only.

A.2.2 How to access

The artifact is stored in the Zenodo and the DOI link is https://doi.org/10.5281/zenodo.15610829. To obtain the artifact, please download the files using the URL provided above.

A.2.3 Hardware dependencies

While our artifact does not require specialized hardware, we recommend running the Account Deletion Link Crawler on a system with a quad-core processor, 16 GB RAM, and an SSD to ensure smooth and efficient performance when using the Android Emulator and Appium.

A.2.4 Software dependencies

The artifact requires a system running macOS, which was used as our test environment. We have not tested the artifact on other operating systems and therefore do not claim cross-platform compatibility.

The following software components are required:

- **Docker**: Used to ensure consistent environment setup and execution for part of the provided code.
- **Android Studio**: Required for running the Android Emulator.
- **Appium**: Required to automate interactions with the Android Emulator and Google Play Store.
- **Google Account**: A Google account is required to log in to the emulator's Google Play Store during execution.
- **OpenAI API Key**: Required to run the GPT-based account deletion method classifier. A valid key has been provided through the HotCRP system.

A.2.5 Benchmarks

We have included all necessary data in the `/data` and `/artifact/result` directories. The `/data` directory contains 50 apps used in the pilot study, 863 apps used during the development of DeleteTracker, and 100 apps used for end-to-end performance evaluation. The `/artifact/result` directory includes 10 apps used to test the mobile crawler and 100 apps used to evaluate all other modules.

A.3 Set-up

A.3.1 Installation

To install the required dependencies and set up the artifact environment, please follow the steps below:

1. Download the repository from `https://doi.org/10.5281/zenodo.15610829` and enter the artifact directory.

2. Install Python dependencies: pip install -r requirements.txt

3. Install spaCy language model: python -m spacy download en_core_web_sm

4. Install Docker: Docker can be downloaded and installed from `https://www.docker.com/`.

5. Install Android Studio and set up an emulator (Android Studio can be downloaded and installed from `https://developer.android.com/`):

   - Open Android Studio and go to "More Actions > Virtual Device Manager".
   - Create and launch a virtual device (e.g., Pixel 9) with Google Play support.
   - Log into Google Play using a valid Google account.
   - Ensure the emulator has Chrome installed.

6. Install node: brew install node.

7. Install and start Appium. Appium can be downloaded and installed from `https://appium.io/docs/en/2.2/quickstart/install/`. After installation, go to the terminal and start Appium by using the command `appium`.

8. Install the driver: appium driver install uiautomator2.

9. ADB Setup and Emulator Verification: To run the mobile crawler or interact with Android emulators, you must ensure that adb (Android Debug Bridge) is installed and properly configured in your environment.

   - If adb is not recognized in your terminal (zsh: command not found: adb), add the following lines to your /.zshrc: `exportANDROID_HOME=$HOME/Library/Android/sdk;exportPATH=$ANDROID_HOME/emulator:$ANDROID_HOME/tools:$ANDROID_HOME/tools/bin:$ANDROID_HOME/platform-tools:$PATH.`
   - Reload the configuration: source /.zshrc.
   - Verify ADB Connection: adb devices

10. Set OpenAI API key: Use the key provided through HotCRP and replace the API key in Line 13 of artifact/3_AccountDeletionMethodClassifier/GPT_classifier.py.

After completing these steps, the evaluator should be able to execute the core functionality described in Sections 4.1–4.3 of the paper.

A.3.2 Basic Test

To verify that all required components are correctly installed and working, we provide a simple functionality test. This test focuses on the Mobile Account Deletion Link Crawler module, as all other modules execute quickly and do not require additional setup. This test verifies the integration of Appium, the Android Emulator, and access to the Google Play Store. We provide a list of 10 apps in the `/artifact/result` directory for testing. Evaluators can run the crawler on these apps using the following file: `artifact/1_AccountDeletionLinkCrawler/1_mobile_crawler.py`. The script will attempt to retrieve account deletion-related links from Google Play for the given apps. The output will be saved to `artifact/result/10_test_app_link.csv`. To assist evaluators, we also provide an example video demonstrating how to run the crawler and visualize the step-by-step process for these 10 apps.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): DeleteTracker can automatically crawl and collect account deletion links from Google Play mobile app and webpage using both mobile and web crawlers. This is proven by the experiments (E1).

(C2): DeleteTracker can identify multiple types of link-related issues, including inaccessible, inconsistent, and duplicate account deletion links across platforms. This is proven by the experiments (E2).

(C3): DeleteTracker can classify the account deletion method using a combination of NLP and GPT-based analysis. This is proven by the experiments (E3).

A.4.2 Experiments

(E1): *[Account Deletion Link Crawler] [30 human-minutes]*
Preparation: Please refer to Section A.3.1 to prepare virtual device setup and start appium Server.
Execution: Enter `1_AccountDeletionLinkCrawler`. First, run `1_mobile_crawler.py`, then run `2_web_crawler.py`.
Results: After both scripts have been executed, check the updated CSV file in the `artifact/result` directory. The mobile crawler will add two new columns: `delete_account_url` and `manage_data_url`. The web crawler will add one new column: `Data Deletion Info`.

(E2): *[Account Deletion Link Analyzer] [10 human-minutes]*
Preparation: Please refer to Section A.3.1 to prepare required Python dependencies.
Execution: Run all the scripts in the `2_AccountDeletionLinkAnalyzer` directory (the execution order does not matter).
Results: After all scripts have been executed, check the updated CSV file in the `artifact/result` directory. Three new columns will be generated in the result CSV file: `delete_account_url_available`, `inconsistency`, and `domain`, corresponding to different types of link issues.

(E3): *[Account Deletion Method Classifier] [30 human-minutes]*
Preparation: Please refer to Section A.3.1 to prepare required Python dependencies.
Execution: For `3_AccountDeletionMethodClassifier`, please first run `web2html.py` to convert URLs into HTML files. Then, enter the `HtmlToPlaintext-master` folder and follow the `README` instructions to build and run the Docker container using `./build.sh` and `./run.sh`. The remaining scripts have no specific execution order and can be run independently after preprocessing is complete.
Results: After `web2html.py` and the HtmlToPlaintext tool have been executed, please check the `artifact/3_AccountDeletionMethodClassifier/HtmlToPlaintext-master/ext` directory to ensure it contains both HTML files and corresponding plaintext files. After running all the remaining classification scripts, six new columns will be generated in the result CSV file: `contains_delete`, `url_contains_delete`, `has_user_input_form`, `has_delete_button`, `GPT_Classification`, and `StepbyStep`, representing a different classification result.

A.5 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/`.