USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Exposing the Guardrails: Reverse-Engineering and Jailbreaking Safety Filters in DALL·E Text-to-Image Pipelines

Corban Villa, *New York University Abu Dhabi;* Shujaat Mirza, *New York University;* Christina Pöpper, *New York University Abu Dhabi*

https://www.usenix.org/conference/usenixsecurity25/presentation/villa

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: Exposing the Guardrails: Reverse-Engineering and Jailbreaking Safety Filters in DALL·E Text-to-Image Pipelines

Corban Villa
New York University Abu Dhabi

Shujaat Mirza
New York University

Christina Pöpper
New York University Abu Dhabi

A Artifact Appendix

A.1 Abstract

This artifact includes all of the required code and dataset to validate and reproduce results of the paper: *Exposing the Guardrails: Reverse-Engineering and Jailbreaking Safety Filters in DALL·E Text-to-Image Pipelines*. This full dataset, which thoroughly evaluates DALL·E 2/3 systems across thousands of requests, is fully available to elicit queries and analyze (with harmful prompts redacted but available upon request). We additionally include scripts to perform a scaled-down experiment to allow evaluators to independently assess the reproducibility of our results.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Our artifact has been carefully constructed to prevent unintended exposure to potentially harmful or disturbing material inherent to the nature of this work. For instance, we redact image prompts in our public dataset–replacing them with SHA256 hashes. (Unredacted prompts are available upon request.) Artifact evaluation scripts are entirely automated and do not require evaluators to interact with any potentially harmful material. Images produced are stored directly in the database to prevent inadvertent exposure.

We do not cause harm to the T2I systems as we are using them as a regular user within the existing rate limits. Furthermore, our methodology is consistent with previous peer-reviewed work that evaluates T2I systems under adversarial conditions [1, 2].

A.2.2 How to access

Artifact: The dataset and code is permanently available at: https://doi.org/10.5281/zenodo.14735417.

GitHub: The code and datasets above are additionally available on GitHub.

A.2.3 Hardware dependencies

A commodity computer:

- 2+ CPU cores
- 4GB+ memory
- 10GB+ disk storage

A.2.4 Software dependencies

Operating System Requirements: Evaluation scripts are validated with `Ubuntu 22.04`. Compatibility with other operating systems may vary.

Software Package Requirements: `docker` and `docker compose` (install guide). Validated with version `27.3.1`.

Privileged Access Requirements: A few of our configuration scripts require the user have sudo access in order to manage container data. For instance, the `postgres` data is owned by root within the container, and thus requires sudo privileges to remove the data after use. If this is a concern, we recommend utilizing a virtual machine for experiments.

A.2.5 Benchmarks

Adversarial prompt datasets originate from Qu et al., Rando et al., and Yang et al. These prompts do not need to be downloaded by evaluators.

A.3 Set-up

A.3.1 Installation

Install the latest version of docker and add your current user to the `docker` group:

- `curl -L https://get.docker.io | bash`
- `sudo usermod -aG docker $USER`
- Logout and log back in.

Clone and enter the GitHub repository:

- `git clone https://github.com/corbanvilla/T2I-Attacks-USENIX-2025.git`
- `cd T2I-Attacks-USENIX-2025`

A.3.2 Basic Test

Execute: `./scripts/validate-install.sh`

```
Command:
> ./scripts/validate-install.sh

Expected Output:
...
Docker compose build completed successfully
All validation checks passed!
```

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): We reverse-engineer the black-box cascading safety guardrails in DALL·E models using a novel time-based side-channel, providing insights into a multi-stage filtering process (§5). This is replicated in experiment (E1), described in §5.1, which identifies the presence of multiple DALL·E 2 filters through statistical significance testing. Experiment results also mirror those in Fig. 2.

(C2): We synthesize key takeaways for T2I system security by juxtaposing safety mechanisms present in DALL·E 2 and DALL·E 3, notably the incorporation of an LLM-based implicit filter in DALL·E 3 (§4.4) to soften harmful prompts, quantified through experiments that analyze prompt toxicity before and after revisions (E2).

(C3): We introduce novel jailbreaking attacks specific to T2I models, namely T2I negation and low-resource-language attacks, which exploit the limitations of safety filters in handling negated phrases and less common languages (§6). We experimentally evaluate the success rates of jailbreaking low-resource languages (E3).

A.4.2 Experiments

Experiment Data Collection: [30-45 compute-minutes]: To run all experiments (E1-E3) and collect the required dataset for analysis, run the following commands. Experiments are estimated to require 30-45 minutes to complete. Please allow the experiments to *run continuously and uninterrupted to completion* to minimize the potential for noise variation in the DALL·E system performance.

The script will prompt evaluators to provide three key configuration secrets, which are included via a HotCRP comment: **1)** An OpenAI API key, **2)** an OpenAI organization key, and **3)** a prompt dataset decryption key.

Execute: `./scripts/artifact-evaluation.sh`

Results: Experiments are validated using Jupyter Notebook analysis scripts. Connect to the pre-configured Jupyter notebook server at: http://127.0.0.1:8888/lab (The default security token is: `T2I-USENIX-rubpGTNAzgd4tgk7npb`)

(E1): [DALL·E 2 Blocklist Probing] [5 human-minutes]:

```
Command:
> ./scripts/artifact-evaluation.sh

Expected Output:
...
All experiments completed
Artifact evaluation completed
```

Description: This experiment demonstrates the effectiveness of our DALL·E 2 blocklist probing attack, which allows an attacker to detect independent filter systems by their respective response times (§5.1).
Dataset: This experiment uses 5 pairs of blocklisted and mutated words, repeated 5 times each, for a total of 50 requests to DALL·E 2.
Success Criteria: The experiment demonstrates the statistical significance between rejection timings.
Results: To validate our blocklist probing side-channel attacks:

1. Open `src/artifact_evaluation/1-timing-side-channel.ipynb` in the Jupyter notebook.
2. Select `Run All Cells` from the `Run` menu.
3. Validate the final cell demonstrates statistical significance.

(E2): *[Quantifying Prompt Softening] [5 human-minutes]:*
Description: This experiment quantifies the effect of the LLM prompt revision prompt as an implicit filtering mechanism in DALL·E 3 using two key metrics: Toxicity Absolute Change and Toxicity Theme Similarity (§3.3).
Dataset: This experiment uses 5 prompts in 4 languages (Hawaiian, Lao, Nepali, Sinhala) for a total of 20 requests to DALL·E 3. Prompts and revised prompts are evaluated for toxicity using the OpenAI Moderation API.
Success Criteria: The experiment demonstrates that the prompt revision model "softens" the toxicity of harmful prompts to various degrees of success under multilingual inputs.
Results: To validate our prompt softening claims:

1. Open `src/artifact_evaluation/2-prompt-softening.ipynb` in the Jupyter notebook.
2. Select `Run All Cells` from the `Run` menu.
3. Validate the final cell demonstrates various toxicity changes and similarity decreases, with the magnitude of change dependent on the respective language.

(E3): [Low-Resource Languages] [5 human-minutes]:
Description: This experiment demonstrates that low-resource languages can be used as an effective mechanism to bypass DALL·E safety filters (§6.1).
Dataset: This experiment uses 5 prompts in 5 languages (English, Nepali, Sinhala, Lao, Hawaiian) for a total of 25 requests to DALL·E 3.

Success Criteria: Languages with fewer pages in the Common Crawl dataset frequently feature higher prompt acceptance rates.

Results: To validate our low-resource language attacks:

1. Open `src/artifact_evaluation/3-low-resource-lang-attack.ipynb` in the Jupyter notebook.
2. Select `Run All Cells` from the `Run` menu.
3. Validate the final cell depicts various language bypass rates.

A.4.3 Cleanup Procedure

To shutdown and remove the deployed containers and their respective images, run the following script. After executing, you may additionally wish to remove (`rm -rf`) the repository. Execute: `./scripts/cleanup.sh`

A.5 Notes on Reusability

We encourage future research to leverage our comprehensive datasets. The full experiment dataset is published under `datasets/postgres/redacted.sql.zip`. The `README.md` documentation describes how to interact with the dataset more comprehensively. Moreover, `src/examples.ipynb` details a number of ways to query the database, including how to use specific functions such as cosine similarity or vector norms.

While specifics of the dataset, such as average or median response times may vary as DALL·E 2/3 are continuously updated by OpenAI, we expect the core findings to maintain relevancy (e.g., the statistical distributions of response times). **Exploring Key Findings:** All tables and figures in this work can be produced using notebook files located under: `src/paper`.

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/.