USENIX WOOT Conference on Offensive Technologies (WOOT '26) - Artifact Evaluation Guidline

Accepted Paper Title:

"You Have Been LaTeXpOsEd: A Large-Scale Systematic Analysis of Information Leakage in Preprint Archives Using Large Language Models"

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Artifact Evaluation Guidline

1 Introduction / Artifact abstract

The artifact belongs to the following paper: **"You Have Been LaTeXpOsEd: A Large-Scale Systematic Analysis of Information Leakage in Preprint Archives Using Large Language Models."**

In this research, we present the first large-scale security audit of the *arXiv* preprint repository. Our study analyzes over 1.2 TB of data collected from more than 100,000 arXiv submissions, leveraging the capabilities of Large Language Models (LLMs) **to identify potential security risks such as embedded secrets in source files and comments**.

Reproducing the full set of experimental results reported in the paper—particularly those involving the complete dataset of 100,000 submissions—is computationally expensive and time-consuming. This challenge is further amplified by the reliance on high-demand LLMs, which incur significant GPU requirements and API usage costs.

Despite these limitations, **we release a new dataset, LLM-SecDB**, which can serve as a benchmark for evaluating state-of-the-art LLM capabilities in secret detection. In addition, we provide a reproducible methodology using real arXiv submissions to demonstrate how the reported results can be reproduced. For simplicity, we do not re-download or process the entire 1.2TB dataset; instead, we use a predefined subset of articles to show that the methodology and scripts function correctly, enabling independent reviewers to reproduce the same results.

taTo this end, we provide two complementary demonstrations:

- **LLM-SecDB:** We release the `LLM-SecDB` benchmark, a curated dataset designed to evaluate the capability of LLMs in detecting secrets embedded within comments and textual artifacts. Using this benchmark, we demonstrate how to reproduce the key results reported in the paper for the LLMs evaluated in our study. Running the benchmark against the evaluated models should yield results consistent with those presented in the paper (Table 3 results).
- **Finding secrets in Arxiv submission uisng LLMs:** Reproducing and reanalyzing the entire 1.2TB dataset of 100,000 arXiv submissions is computationally expensive and highly time-consuming due to the substantial GPU resources required for large-scale LLM inference. To address this challenge, we selected a predefined subset of arXiv articles and provide all scripts and methodological details necessary to reproduce the results on this subset, demonstrating that our LLM-based extraction methodology functions as intended. Specifically, we provide a practical end-to-end reproduction pipeline using a predefined set of arXiv papers. This example serves as a proof of concept, illustrating how LLM-based analysis can uncover exposed credentials and thereby validating the effectiveness of our methodology.

2 Environmental setup

For both experiments, we use OpenRouter API keys, allowing the experiments to be conducted without requiring local GPU resources. **We provide working oppenrouter.ai API keys** with a $20 credit balance **to the reviewers**, which is sufficient to reproduce the experiments.

OPENROUTER API KEY:

`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`

2.1 Reproducing LLM-SecDB Benchmark Results

One of the core components of our research is the new `LLM-SecDB` dataset. The dataset is available through multiple sources, including the project GitHub repository and Zenodo.

- **Zenodo link:** `https://doi.org/10.5281/zenodo.20035860`
- **Github project page:** `https://github.com/LaTeXpOsEd`

Although both GitHub and Zenodo contain the same benchmark, **we recommend using the Zenodo DOI to download the dataset, as research artifacts should remain permanently available**.

The **Zenodo link** contains three files:

- `README.md`
- `LLM-SecDB.json`
- `LLM_SecDB_google_colab.ipynb`

After downloading all three files from ZENODO, open `LLM_SecDB_google_colab.ipynb` in Google Colab and upload `LLM-SecDB.json` to the default directory. The Google Colab notebook contains the artifact abstract, relevant documentation, and useful examples explaining how the benchmark works.

2.1.1 Reproducing Table 3 result form the paper

In `LLM_SecDB_google_colab.ipynb`, update the `API_KEY` variable with the API key we provided (STEP 1), then execute the cell (STEP 2).

STEP 2

STEP 1

```
#To run the LLM-SecDB benchmark, simply execute this cell and review the res

import os, json, re, time, csv
from typing import List, Dict, Any, Optional, Tuple
from collections import Counter
from concurrent.futures import ThreadPoolExecutor, as_completed
import threading
from tqdm import tqdm
import pandas as pd  # kept for parity with original; not strictly required
from openai import OpenAI
from openai import APIError, RateLimitError, InternalServerError, BadRequest
# ========= CONFIG=========
API_KEY        = "<YOUR OPENROUTER.AI API-KEY HERE>"
MODEL_NAME     = "qwen/qwen-2.5-72b-instruct" # Any openrouter.ai model
INPUT_PATH     = "LLM-SecDB.json"             # LLM-SecDB.json (upload it f
OUTPUT_BASENAME = "./results/"                # output directory
TEMPERATURE    = 0.0
MAX_RETRIES    = 4
SLEEP_BACKOFF  = 2.0
SHOW_COMMENTS  = True                         # True to include raw comment
TEST_MODE_LIMIT = None                        # e.g., 10 for a quick smoke
MAX_WORKERS    = 3                            # thread count for parallel r
COUNT_NONE_AS_HIT = True                      # treat none/none as a hit in
BASE_URL = "https://openrouter.ai/api/v1"
# =====================================
os.makedirs(os.path.dirname(OUTPUT_BASENAME), exist_ok=True)
```

A key advantage of Google Colab is that no package installation or environment variable configuration is required, as all necessary dependencies are already available by default. This enables a clean and straightforward reproduction setup. By default, the script uses the Qwen2.5-72B model. After running the script in Google Colab, it will produce the evaluation results of the Qwen2.5-72B model on our `LLM-SecDB` benchmark:

```
            *(rec.get(c, "") if not isinstance(rec.get(c, ""), list) else ",
            exact
        ])
except Exception as e:
    print(e)

[*] API key is valid
[*] Model 'qwen/qwen-2.5-72b-instruct' is valid and reachable
[*] LLM-SecDB benchmark started

[*] EMA (Exact-match accuracy : 0.7567
[*] ONE (At least one correct): 273 / 300 (91.0%)
False-positive records (pred≠∅ & gt=∅): 1
False-negative records (gt≠∅ & pred=∅): 25

By-label counts (GT vs Pred, TP/FP/FN and precision/recall):
 - conflict             GT=  38  Pred=  42  TP=  21  FP=  21  FN=  17  P=0.50 R=0.55
 - credentials          GT=  50  Pred=  50  TP=  49  FP=   1  FN=   1  P=0.98 R=0.98
 - network_identifiers  GT=  81  Pred=  77  TP=  71  FP=   6  FN=  10  P=0.92 R=0.88
 - peerreview           GT=  35  Pred=  41  TP=  35  FP=   6  FN=   0  P=0.85 R=1.00
 - pii                  GT=  42  Pred=  52  TP=  41  FP=  11  FN=   1  P=0.79 R=0.98
```

One can observe that the results for Qwen2.5-72B achieve an EMA score of 75.67% and a ONE score of 91.0%. In the results, we can also observe the TP, FP, and FN values (True Positives, False Positives, and False Negatives) for each category, such as credentials, PII, and others. In the paper, we evaluated multiple LLMs using the LLMSec-DB benchmark in order to identify the best-performing models in terms of quality-to-price ratio. These results are reported in Table 3 of the paper. Here, we present the first 5 rows from Table 3 of our paper.

Table 3 from the Paper (first 5 row only)

				CRED			PII			NETID			Costs	
Model	**EMA**	**ONE**	**O**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**MT**	**ALL**
GPT-5	82.7%	91.7%	✗	50	3	0	42	16	2	38	9	4	$1.25	$840
Claude Sonnet 4	81.0%	87.7%	✗	50	0	0	44	10	0	39	8	3	$3.00	$2016
Claude 3.7 Sonnet	80.3%	92.0%	✗	50	0	0	43	21	1	37	12	4	$3.00	$2016
Qwen-2.5 72B	80.0%	91.0%	✔	50	2	0	44	11	0	38	11	3	$0.07	$47
Llama-3.1 405B	80.0%	87.7%	✔	49	2	1	44	11	0	36	10	5	$0.80	$538

The reproduced results can be compared against the reported values in the paper. A small variation of around ±7–8% between runs is expected, even when using a temperature of 0.0. This is because OpenRouter may route requests to different backend servers with different configuration, and some models use Mixture-of-Experts architectures that can produce non-deterministic outputs. However, significantly larger deviations than this range should not be expected. We also note that, for very small models, hallucinations and random guessing may lead to different values; however, this should not occur for larger models. In our Google Colab run, we evaluated Qwen2.5-72B. For comparison, our reproduced EMA accuracy was 75.67% in the Google Colab run, which falls within the expected deviation range, while the ONE score (91.0%) exactly matches the value reported in the paper. In our experiments, the ONE metric (i.e., at least one correct label) is even more important than exact matching, because if the model flags at least one relevant problem, we are still interested in those cases to avoid missing any potential issues. **NOTE:** While we encourage reviewers to perform additional experiments with different models, please note that selecting very expensive models will quickly exhaust the provided $20 API credit associated with OpenRouter.

2.2 Extraction of Sensitive Credentials from arXiv Submissions

The next demonstration focuses on the methodology for using LLMs to extract secrets from arXiv submissions. Since downloading all 100,000 arXiv submissions and rerunning LLMs on them would be extremely expensive, we adopted an alternative approach to demonstrate that our methodology is reproducible. We published the "ArXiv Secret Extraction" tool on Zenodo at the following link: `https://doi.org/10.5281/zenodo.20058989`. The tool can be used to analyze standalone arXiv submissions for potential secret and credential exposure. The **Zenodo link** contains two files: `README.md` and `Arxiv_secret_extraction.ipynb`. In the paper, we selected Qwen2.5-72B, as we demonstrated in the previous section that it can achieve very good results while maintaining a highly favorable price-to-utility ratio. Therefore, the Qwen model will also be used in this demonstration.

After downloading all two files from ZENODO, open `Arxiv_secret_extraction.ipynb` in Google Colab. In `Arxiv_secret_extraction.ipynb`, update the `API_KEY` variable with the API key we provided (STEP 1), then execute the cell (STEP 2).

```
STEP 2
# ========= CONFIG=========
ARXIV_ARTICLE_URL = "https://arxiv.org/abs/2510.03761" #Change for any arxiv RL
API_KEY           = '<YOUR OPENROUTER API-KEY HERE>'  STEP 1
MODEL_NAME        = "qwen/qwen-2.5-72b-instruct" # Any openrouter.ai model
BASE_URL          = "https://openrouter.ai/api/v1"
# ========= CONFIG=========

import os, json,re, gzip, shutil, tarfile,zipfile, requests
from pathlib import Path
from openai import OpenAI
from openai import APIError, RateLimitError, InternalServerError, BadRequestErr

def validate_api_and_model(api_key, base_url, model_name):
    try:
        client = OpenAI(api_key=api_key,base_url=base_url)

        # tiny test request
        response = client.chat.completions.create(model=model_name, messages=[{

        print(f"[*] API key is valid")
        print(f"[*] Model '{model_name}' is valid and reachable")
        return True
```

After running the cell, the results clearly demonstrate that the provided arXiv submission does not reveal any secrets within the LaTeX source, which is the expected outcome. By default, this Google Colab notebook is configured to use our "LaTeXpOsEd" arXiv submission. For this particular submission, no sensitive information or secrets should be present, and the tool correctly confirms this.

```
[*] API key is valid
[*] Model 'qwen/qwen-2.5-72b-instruct' is valid and reachable
[*] Trying: https://arxiv.org/e-print/2510.03761
[*] Downloaded source
[*] Detected tar archive
[*] Kept 1 .tex files
[*] Final directory: 2510.03761
[*] Extracted 15 comments

[*] Found Labels:

[*] Evidence Excerpts:

[No sensitive excerpts found]
```

There is also one important configuration parameter. The `ARXIV_ARTICLE_URL` variable, which specifies the arXiv submission to be downloaded. In this demonstration, we do not use a large collection of arXiv submissions from an Amazon S3 bucket, instead, a reviewer or evaluator can provide any arXiv URL for testing purposes.

2.2.1 Real secrets extraction demonstration

Now it is time to change the `ARXIV_ARTICLE_URL` variable which contains valid secrets..

> **Ethical and Responsible Disclosure Notice:** The following demonstration includes a real arXiv submission identified during our empirical evaluation. To demonstrate the effectiveness of our methodology, we provide an arXiv submission that clearly highlights the strengths and practical capabilities of the proposed approach. **We respectfully request that reviewers and evaluators do not disclose, redistribute, publicize, or misuse the referenced submission or any contained information. These details do not appear directly in the paper and are only reported in aggregated form and explicit credentials are intentionally omitted from the paper. This example is included solely for scientific evaluation and reproducibility purposes.**

The following arXiv submission demonstrates a realistic case in which our LLM-based methodology identifies an arXiv paper as a potential security risk. To evaluate this example, set the `ARXIV_ARTICLE_URL` variable to: `https://arxiv.org/abs/2306.14210`. Run the cell again, and the follwoing result can be seen:

```
[*] Trying: https://arxiv.org/e-print/2306.14210
[*] Downloaded source
[*] Detected tar archive
[*] Kept 1 .tex files
[*] Final directory: 2306.14210
[*] Extracted 108 comments

[*] Found Labels:
credentials, pii

[*] Evidence Excerpts:

----CREDENTIALS EXCERPT----
$Id:locomotion.hmd@gmail.com
pass- IIIT@789
github - locomotionhmd
password - IIIT@789@serc
----CREDENTIALS EXCERPT----

----PII EXCERPT----
$Id:locomotion.hmd@gmail.com
$Id:anonymousauthor362@gmail.com
----PII EXCERPT----
```

The Qwen-2.5-72B model identified a particularly **sensitive section within the LaTeX source containing GitHub and Gmail credentials**. To verify this behavior, one may compare the compiled PDF with the original LaTeX source and confirm that these credentials **DO NOT APPEAR** in the final PDF intended for public release. Instead, they are embedded as hidden comments within the LaTeX source for the authors' internal use only.

Surprisingly, we found that this is a relatively common practice: credentials associated with GitHub repositories linked to the research project are often embedded as hidden content within the LaTeX source. This clearly demonstrates the significant security risks posed by exposing raw source files.

2.2.2 No hallucination

Finally, we demonstrate another important aspect of the methodology: reducing hallucinations and false positives. These issues can be particularly problematic when analyzing arXiv papers related to passwords or authentication, where numerous example credentials and weak passwords may naturally appear throughout the text. In such cases, LLMs must correctly understand the context and distinguish illustrative examples from genuinely exposed credentials. Consider the following two arXiv submissions that focus on passwords and authentication-related topics:

- `https://arxiv.org/abs/2604.19410`: *Understanding Password Preferences, Memorability, and Security through a Human-Centered Lens*
- `https://arxiv.org/abs/2212.08796` *A Survey on Password Guessing*

Even though these articles contain numerous password examples and extensively discuss passwords as part of their core topic, our methodology should not generate false positives—or at most only a very small number of them—purely due to such contextual content. Running our tool on these two arXiv submissions, for example, revealed no hidden sensitive excerpts or exposed credentials.

```
[*] API key is valid
[*] Model 'qwen/qwen-2.5-72b-instruct' is valid and reachable
[*] Trying: https://arxiv.org/e-print/2604.19410
[*] Downloaded source
[*] Detected tar archive
[*] Kept 1 .tex files
[*] Final directory: 2604.19410
[*] Extracted 112 comments

[*] Found Labels:


[*] Evidence Excerpts:

[No sensitive excerpts found]
```

```
[*] API key is valid
[*] Model 'qwen/qwen-2.5-72b-instruct' is valid and reachable
[*] Trying: https://arxiv.org/e-print/2212.08796
[*] Downloaded source
[*] Detected tar archive
[*] Kept 9 .tex files
[*] Final directory: 2212.08796
[*] Extracted 1286 comments

[*] Found Labels:


[*] Evidence Excerpts:

[No sensitive excerpts found]
```

3 Overview of the Methodology Used in the Paper

While this guideline demonstrates the methodology, in the paper we did not manually download arXiv submissions one by one. In accordance with arXiv's terms of use, we avoided issuing large numbers of direct requests and instead used the official Amazon S3 bucket to download approximately 1.2 TB of arXiv submissions. **All related scripts are also available on the GitHub page**. A similar automated approach was then applied for analysis, and any flagged or potentially sensitive submissions were manually reviewed to validate the methodology.

Surprisingly, we found very few false positives, most of which were related to conflicts of interest or peer review discussions. In terms of credential and secret detection, however, the methodology proved to be highly effective. The full pipeline can be seen in the following Image (For further details, check the paper).

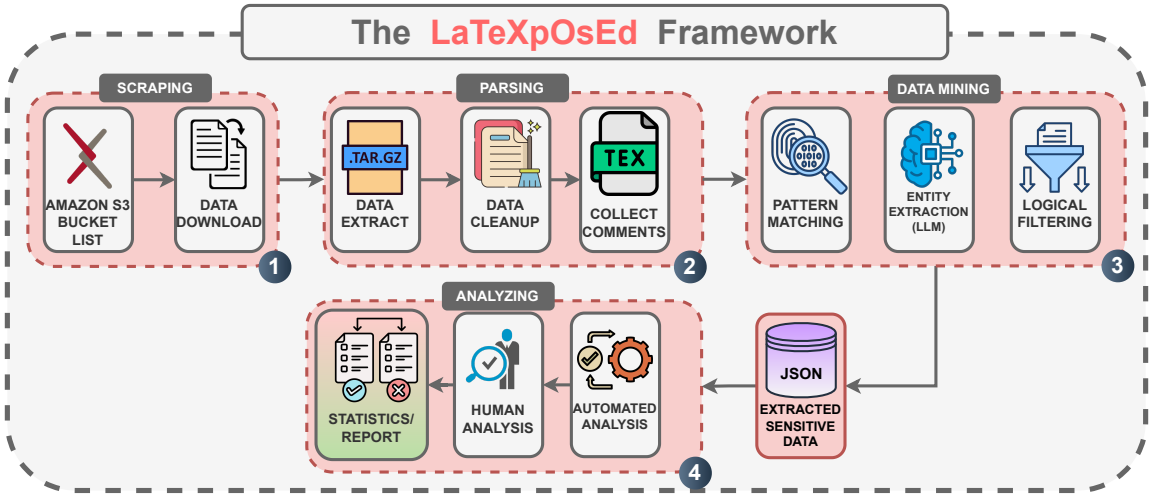

Figure 1: The LaTeXpOsEd framework methodology used in the paper

Just to ensure consistency, we will add an "Availability" section to the camera-ready paper:

Availability

The LaTeXpOsEd project is publicly available on its GitHub project page: `https://github.com/LaTeXpOsEd`. Additionally, both the tools and the LLM-SecDB benchmark are archived on Zenodo at the following links:

- LLM-SecDB benchmark: `https://doi.org/10.5281/zenodo.20035860`
- Arxiv secret detection tool: `https://doi.org/10.5281/zenodo.20058989`