Superviz25-SQL

High-Quality Dataset to Empower Unsupervised SQL Injection Detection Systems

Grégor QUETEL - Télécom Paris
Eric ALATA - LAAS CNRS
Pierre François GIMENEZ - Inria
Thomas ROBERT - Télécom Paris
Laurent PAUTET - Télécom Paris

3 March 2025

FRANCE 2030
PROGRAMME DE RECHERCHE
CYBERSÉCURITÉ

TELECOM Paris
INSTITUT POLYTECHNIQUE DE PARIS

Dataset Link

TELECOM Paris

IP PARIS

Motivation

Dataset Link

Motivation

SQL Injection Attack

⚠️ Emblematic application-level attack
🐛 Caused by faulty user-input validation
⏳ Still occurs 25 years after its discovery [1], [2], [3]

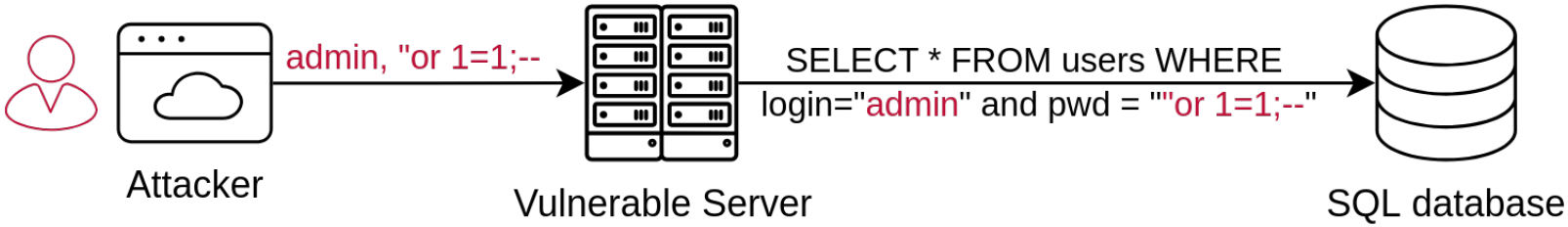


Figure 1: Exploitation of vulnerable login authentication through malicious SQL code injection

Dataset Link

Motivation

SQL Injection Detection Systems

Two components:

1. Collection mechanism
2. Decision engine

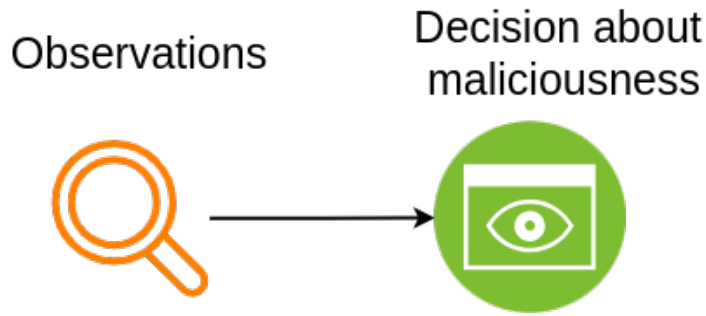


Figure 2: Intrusion Detection components

Grégor QUETEL - Télécom Paris, Eric ALATA - LAAS CNRS, Pierre François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 3 March 2025

Dataset Link

Motivation

SQL Injection Detection Systems

Two components:

1. Collection mechanism
2. Decision engine

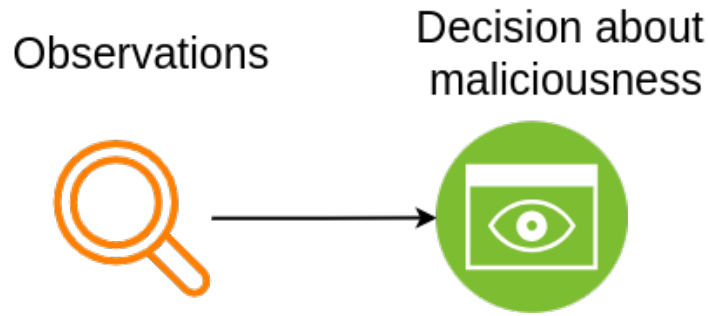


Figure 2: Intrusion Detection components

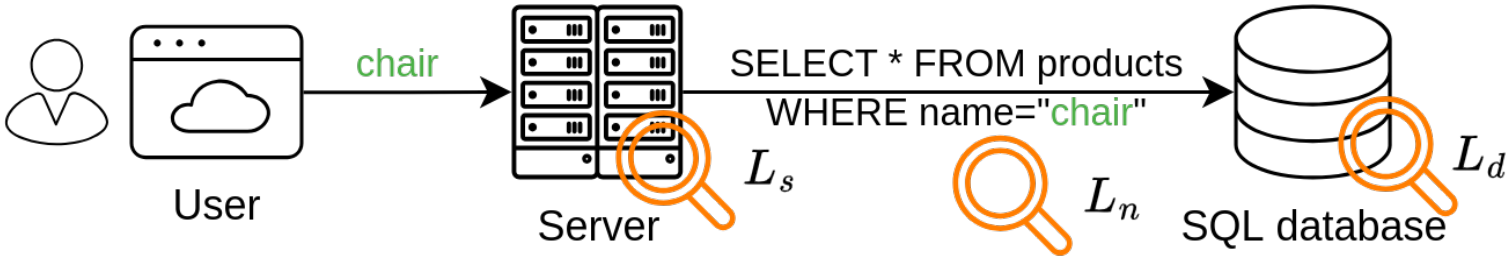


Figure 3: Usual collection mechanisms locations: L_s (server-side), L_n (network-side) and L_d (database-side)

Grégor QUETEL - Télécom Paris, Eric ALATA - LAAS CNRS, Pierre François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 3 March 2025

Dataset Link

Existing SQL Injection Detection Datasets

SQL Injection Detection Systems need to be evaluated and compared (i.e., **we need datasets**):

Kaggle SQL Injection Dataset [4]

📄 No generation documentation
🔤 String escaping errors break SQL parsing
🔀 Observation point is inconsistent: mixes L_s and L_d

WAF-A-MoLE [5]

📉 Low diversity
🔤 String escaping errors break SQL parsing
⚠️ Contains label inconsistencies

Dataset Link

Problem Statement

Existing datasets are not suited, for the evaluation of SQL Injection Detection Systems. We need a **novel, high-quality dataset.**

Design choices

- ⚙️ Is **application-specific**
- 👁️ Designed for **unsupervised settings**
- 🔍 Provides both server-side (L_s) and database-side (L_d) observations
- 🤖 Is reproducible: no human factor in attack generation

Grégor QUETEL - Télécom Paris, Eric ALATA - LAAS CNRS, Pierre François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 3 March 2025

Dataset Link

Related Work

How to characterize dataset quality ?

Dataset Link

IDS Dataset Quality

Three quality dimensions: Diversity, Realism and Documentation of generation process [6], [7], [8]. We derived criteria:

AtkDiv	The attacker should use varied attacks
WLDiv	The intended workload should be heterogeneous
GenDoc	Dataset documentation should entail the entire generation process (and not only the attacks)
WLDoc	The intended workload (i.e., benign queries) should be described
Artefacts	Experimental artefacts should be transparently described
Labels	Labelling should be error-free
WLReal	The intended workload should be realistic in terms of statement types, user-input values and query structure complexity
AtkReal	The attacker should use realistic attacks

Dataset Link

IDS Dataset Quality (ii)

Another dimension: Dataset Benchmarking capabilities (derived from [7]):

Unbal	The test dataset should not be balanced (as it could lead to over-optimistic precision and F1 scores)
FineLab	The labels should be fine-grained: instead of benign or malicious labels, the attack types should be described as well as whether the attack was successful or not
Size	The dataset must be of enough size for novel machine-learning techniques that require considerable training data

Dataset Link

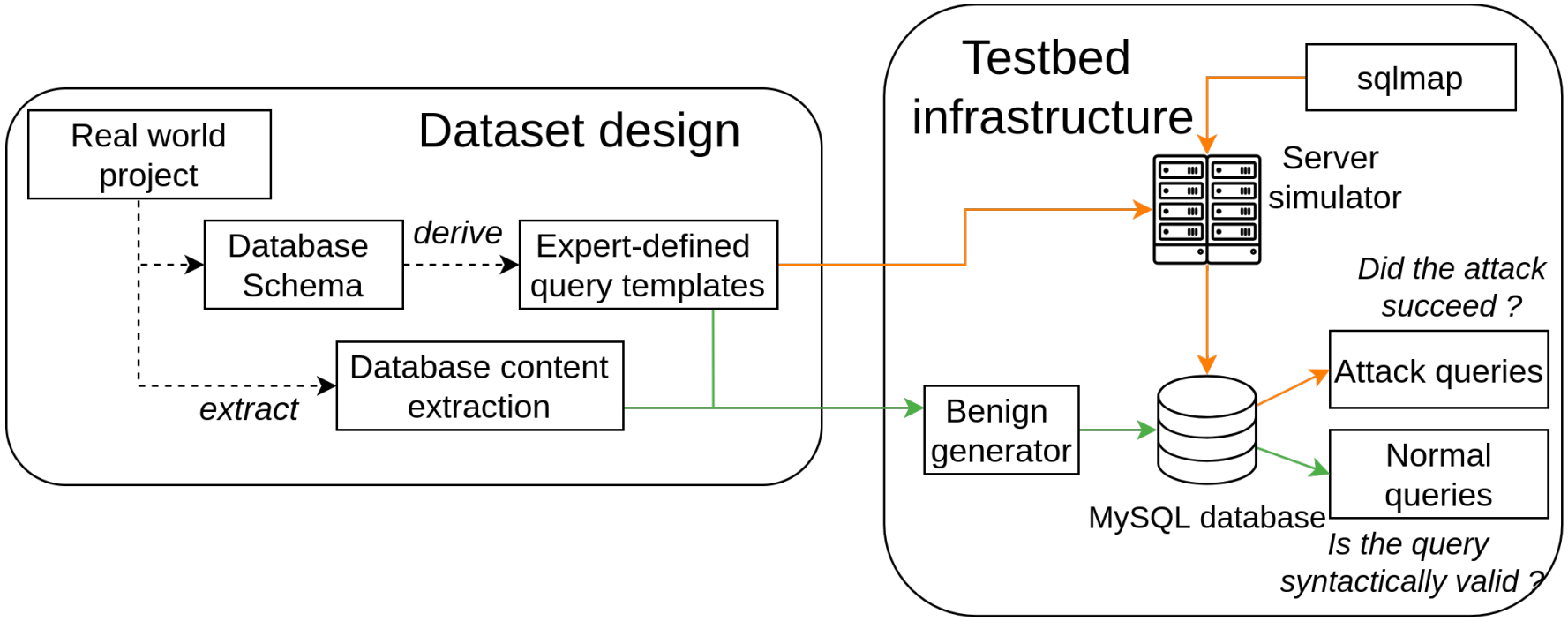
Approach

How to generate a dataset following these criteria ?

Dataset Link

Approach

Template-based generation



Grégor QUETEL - Télécom Paris, Eric ALATA - LAAS CNRS, Pierre François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 3 March 2025

Dataset Link

Approach

Generating Samples from Templates

- Based on the OurAirports (https://ourairports.com/) open-source project.
- 62 different templates of queries were designed
  - Various field types, tables, joins, function calls...

```
SELECT r.name AS region_name, COUNT(a.id) AS airport_count FROM airport
a JOIN regions r ON a.iso_region = r.code WHERE a.iso_country =
{countries_code} GROUP BY r.name ORDER BY r.name ASC;
```

Listing 1: Template *airport-S5* used in Superviz25-SQL. The query retrieves the number of airports in each region of a given country code.

Dataset Link

Generating Samples from Templates (ii)

Benign

- Generated using real values from OurAirports dumps
- Realistic workload distribution [9] : 70% `SELECT`, 10% `UPDATE`, 10% `INSERT`, 9% `DELETE`, 1% admin
- Syntactic validity verification

```
SELECT r.name AS region_name, COUNT(a.id) AS airport_count FROM airport
a JOIN regions r ON a.iso_region = r.code WHERE a.iso_country =
"FI" GROUP BY r.name ORDER BY r.name ASC;
```

Listing 2: Benign query found in Superviz25-SQL generated from template *airport-S5* and "FI" – extracted from the project dump.

Dataset Link

Generating Samples from Templates (iii)

Attacks

- Simulated SQL injection attacks using **sqlmap**
- Extensively used the options of the tool: techniques, mutation operators, initialization code to foster diversity and realism
- Options values provided as metadata

```
SELECT r.name AS region_name, COUNT(a.id) AS airport_count FROM airport
a JOIN regions r ON a.iso_region = r.code WHERE a.iso_country =
"MO'bGPxtt<'">kgTNMB" GROUP BY r.name ORDER BY r.name ASC;
```

Listing 3: Attack query found in Superviz25-SQL generated by the invocation of `sqlmap` on template *airport-S5* .

Dataset Link

Resulting Dataset

Superviz25-SQL

- Training set of 335,306 benign queries;
- Test set of 3,017,390 benign queries / 336,281 malicious queries (90:10)

Field	Description
full query	The full statement, as observed in L_d
label	Benign (0) or attack (1)
user inputs	The user input without the template, as observed in L_s
tamper method	The sqlmap tamper-script used for this sample
attack technique	The technique used by `sqlmap`, either `boolean`, `error`, `inline`, `stacked`, `time`, `union` or `insider`
split	Proposed split for the dataset, either `train` or `test`

Table 1: Subset of the fields characterizing Superviz25-SQL samples

Dataset Link

TELECOM Paris

IP PARIS

Evaluation

- What is the diversity of Superviz25-SQL ?
- Is Superviz25-SQL of enough size ?

Dataset Link

Evaluation

Dataset Diversity

The variability of the data across specific dimensions (semantic, syntactic, lexical, SQL dialect).

- **Vocabulary Size**: The number of tokens in the vocabulary constructed from all samples in each dataset
- **Number of Unique Parse Trees**: The number of unique parse trees in each dataset
- **Semantic Space Dispersion** [10] : the average pairwise similarity between queries in the semantic space of a Sentence-BERT model

Dataset Link

Evaluation

Dataset Diversity Evaluation

Dataset	Sample Type	Number of Samples	Vocabulary Size	Unique Parse Trees	Semantic Space Dispersion
Superviz25-SQL	Normal	3,352,696	**459,215**	**58,891**	**0.01197**
WAF-A-MoLE		345,199	332,827	15,131	0.00623
Kaggle		19,537	14,471	515	0.01156
Superviz25-SQL	Attack	335,192	**118,267**	4,970	**0.01103**
WAF-A-MoLE		393,345	12,669	**7,336**	0.00991
Kaggle		11,382	10,805	242	0.01099

Table 2: Diversity Metrics Across **Superviz25-SQL**, **WAF-A-MoLE**, and **Kaggle** datasets.

Dataset Link

Sufficient Size Evaluation

Is Superviz25-SQL of sufficient size to evaluate deep-learning-based decision engines ?

- We evaluated the detection performance on a training set with twice as many samples
- No significant performance change is observed; a performance plateau is reached; the training set is of sufficient size

Dataset	AUPRC	ROCAUC
Superviz25-SQL	0.9576	0.9884
Big-Superviz25-SQL	0.9260	0.9881
Performance Difference	–0.0316	–0.0003

Table 3: Secure-BERT [11] + Autoencoder Performance on Superviz25-SQL vs. Big- Superviz25-SQL

Dataset Link

Baselines Study

What would be the classical experimental evaluation protocol on our dataset ?

Dataset Link

Case Study

We illustrate the proposed evaluation protocol using 9 Pipelines (feature extraction + detection mechanism):

🔍 Feature extraction

- CountVectorizer
- Manually selected features specific to SQL [12] (further denoted as Li)
- Secure-BERT [11], a Sentence-BERT model trained on cybersecurity data

🚨 Novelty detectors

- Local Outlier Factor
- One-class SVM
- Autoencoder

Dataset Link

Baselines Study

Area Under Curves

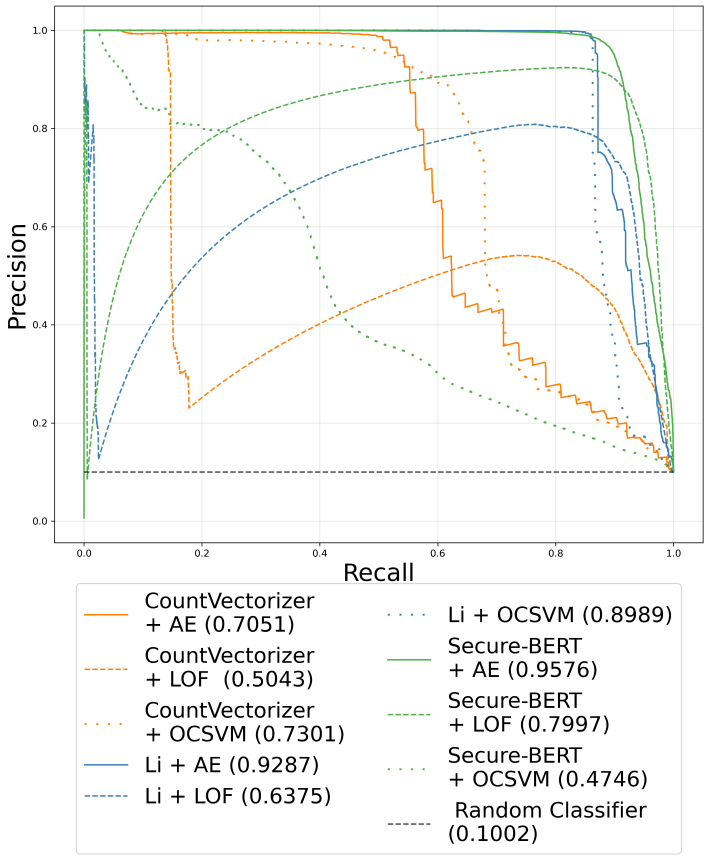


Figure 5: Baselines AUPRC

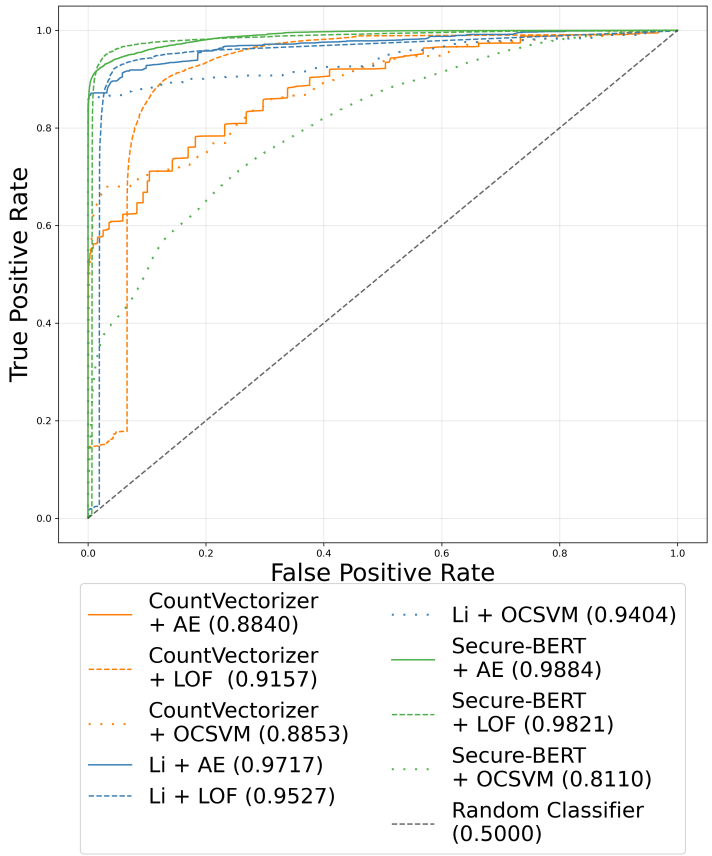


Figure 6: Baselines AUROC

Dataset Link

Baselines Study

Recall per attack technique

The metadata provided by the dataset allows fine-grained analysis.
Such as Recall scores per attack technique :

Model	Boolean	Error	Union	Stacked	Time	Inline	Insider
CountVectorizer + AE	53.37%	25.61%	37.20%	6.57%	14.52%	1.87%	0.00%
Li + AE	91.33%	79.80%	93.37%	65.97%	59.61%	74.30%	79.61%
Li + OCSVM	91.49%	77.04%	93.77%	67.12%	61.98%	66.74%	73.74%
Secure-BERT + AE	92.30%	78.14%	83.14%	63.47%	62.11%	75.95%	99.27%

Table 4: Most performance baselines recall scores per attack technique

Dataset Link

TELECOM Paris

IP PARIS

Conclusion

Dataset Link

Summary

- Superviz25-SQL is :
  - an **Unsupervised** SQL Injection Detection Dataset
  - designed with a focus on **Realism**, **Diversity**, proper **Documentation** and **Benchmarking**
- We evaluated its **Diversity** and **Size**
- We provided the **recommended evaluation protocol** on our dataset and the performance of baselines

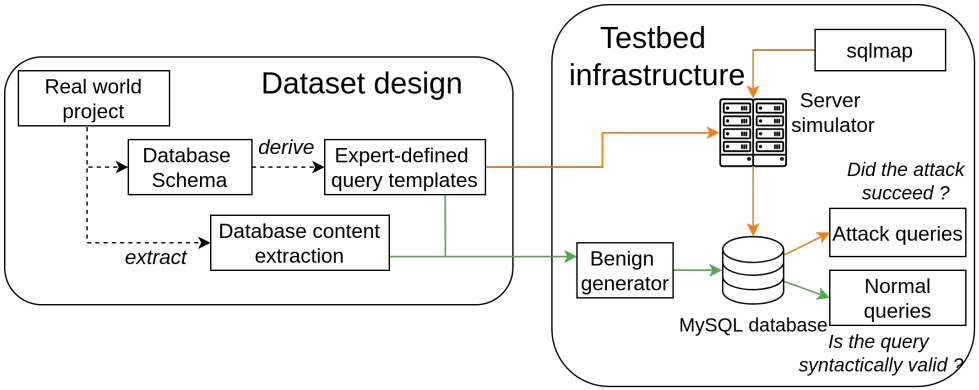
Dataset	Kaggle	Waf-A-Mole	Superviz25-SQL
GenDoc	❌	✅	✅
WLDoc	❌	✅	✅
Artefacts	❌	❌	✅
Labels	❌	❌	⚠️
WLReal	❌	❌	✅
AtkReal	❓	✅	✅
WLDiv	✅	✅	✅
AtkDiv	❓	✅	✅
Unbal	❌	❌	✅
FineLab	❌	❌	✅
Size	❓	❓	✅

Conclusion

Thanks

Dataset Link

Dataset	Kaggle	Waf-A-Mole	Superviz25-SQL
GenDoc	❌	✅	✅
WLDoc	❌	✅	✅
Artefacts	❌	❌	✅
Labels	❌	❌	⚠️
WLReal	❌	❌	✅
AtkReal	❓	✅	✅
WLDiv	✅	✅	✅
AtkDiv	❓	✅	✅
Unbal	❌	❌	✅
FineLab	❌	❌	✅
Size	❓	❓	✅



This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The views reflected herein do not necessarily reflect the opinion of the French government.

- Dataset available at: https://zenodo.org/records/17086037
- Replication code available at: https://github.com/gquetel/sqlia-dataset-generator

References

Dataset Link

[1] J. Forristal, "NT Web Technology Vulnerabilities." Dec. 1998.

[2] sumaart.com, "Luxury Retail Targeted: Cyberattack on Louis Vuitton Exposes Customer Data." Aug. 2025.

[3] T. H. News, "Hackers Exploit Job Boards, Stealing Millions of Resumes and Personal Data." 2024.

[4] sajid576, "Kaggle - SQL Injection Dataset." 2022.

[5] L. Demetrio, A. Valenza, G. Costa, and G. Lagorio, "WAF-A-MoLE: Evading Web Application Firewalls through Adversarial Machine Learning," ACM, 2020. doi: 10.1145/3341105.3373962.

[6] J. Liu, M. A. Inam, A. Goyal, A. Riddle, K. Westfall, and A. Bates, "What We Talk About When We Talk About Logs: Understanding the Effects of Dataset Quality on Endpoint Threat Detection Research," 2025. doi: 10.1109/SP61157.2025.00112.

[7] R. Flood, G. Engelen, D. Aspinall, and L. Desmet, "Bad Design Smells in Benchmark NIDS Datasets," in *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, Jul. 2024, pp. 658–675. doi: 10.1109/EuroSP60621.2024.00042.

[8] T. Gebru *et al.*, "Datasheets for Datasets," no. arXiv:1803.09010. arXiv, Dec. 2021. doi: 10.48550/arXiv.1803.09010.

[9] D. Anderson, "MODELING AND ANALYSIS OF SQL QUERIES IN PHP SYSTEMS," 2018.

[10] Y. Guo, G. Shang, M. Vazirgiannis, and C. Clavel, "The Curious Decline of Linguistic Diversity: Training Language Models on Synthetic Text," in *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 3589–3604. doi: 10.18653/v1/2024.findings-naacl.228.

[11] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A Domain-Specific Language Model for Cybersecurity," no. arXiv:2204.02685. arXiv, Oct. 2022. doi: 10.48550/arXiv.2204.02685.

[12] Q. Li, W. Li, J. Wang, and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[13] M. I. Khan, S. N. Foley, and B. O'Sullivan, "Database Intrusion Detection Systems (DIDs): Insider Threat Detection via~Behaviour-Based Anomaly Detection Systems - A~Brief Survey of~Concepts and~Approaches," in *Emerging Information Security and Applications*, W. Meng and S. K. Katsikas, Eds., Cham: Springer International Publishing, 2022, pp. 178–197. doi: 10.1007/978-3-030-93956-4_11.

[14] Verizon, "Data Breach Investigations Report," 2025.

Dataset Link

TELECOM Paris

IP PARIS

Appendix

Dataset Link

Appendix

Superviz25-SQL - Sample Metadata

Field	Description
full query	The full statement, as observed in L_d
label	Benign (0) or attack (1)
user inputs	The user input without the template, as observed in L_s
attack stage	Corresponds to `sqlmap`'s attack stage, either `recon` or `exploit`
tamper method	The sqlmap tamper-script used for this sample
attack status	The sqlmap attack campaign status, either `success` or `failure`
statement type	Statement type, either `select`,`delete`, `execute`, `update`, `admin` or `insider`
query template id	The expert-defined template ID from which the sample was generated
attack id	Attack campaign identifier
attack technique	The technique used by `sqlmap`, either `boolean`, `error`, `inline`, `stacked`, `time`, `union` or `insider`
split	Proposed split for the dataset, either `train` or `test`

Table 7: Fields characterizing each Superviz25-SQL sample and their description

Dataset Link

Appendix

Adding Insider Attacks

What would be the detection performance on a **different applicative attack** ?

- **Insider attack:** *action that puts an organization or its resources at risk*. Usually consists of financially motivated data dumps [13], [14].
- Also generated using `sqlmap` campaigns, in direct connection mode and various extraction objectives.

```
SELECT description, type, airport_ident, airport_ref, frequency_mhz, id
FROM dataset.airport_frequencies ORDER BY id
```

Listing 4: Insider attack query generated by `sqlmap` using direct access mode.