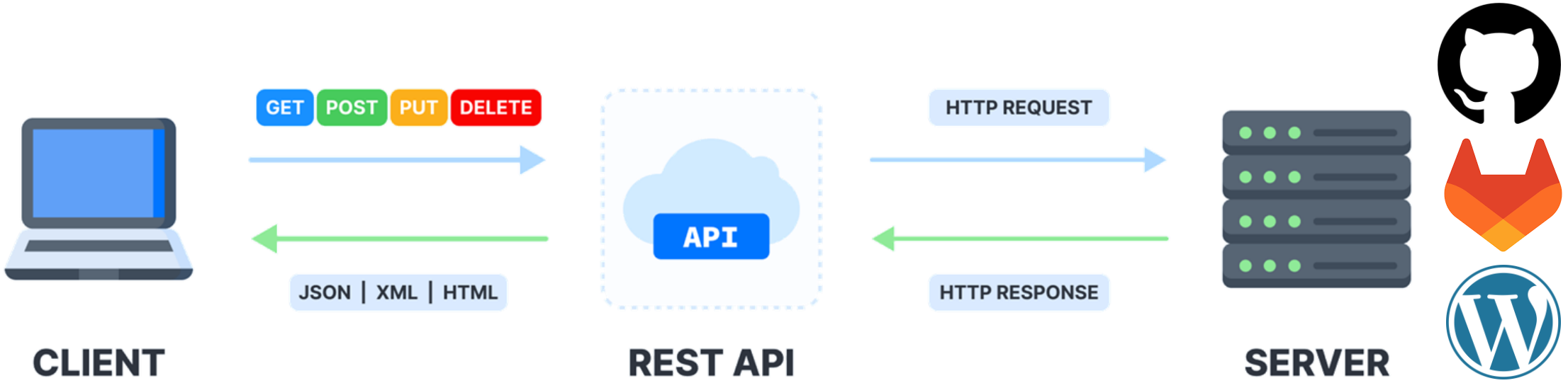
MINER: A Hybrid Data-Driven Approach for REST API Fuzzing

Chenyang Lyu[1] **Jiacheng Xu**[1] Shouling Ji[1] Xuhong Zhang[1] Qinying Wang[1]
Binbin Zhao[2] Gaoning Pan[1] Wei Cao[3] Peng Chen[3] Raheem Beyah[2]

[1]Zhejiang University [2]Georgia Institute of Technology [3]Ant Group

REST API

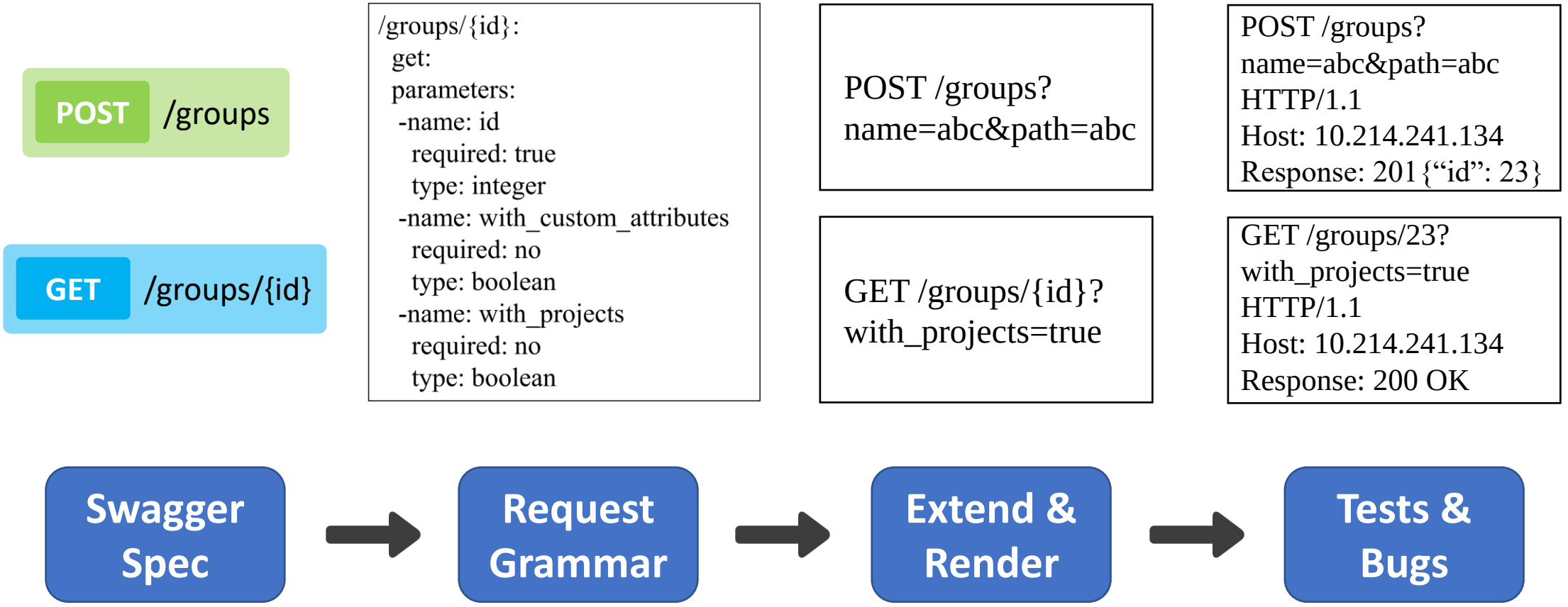
REST API is popular in cloud service but not secure.



REST API Fuzzing

Zhejiang University

RESTler[1], the first stateful REST API fuzzer.



[1] Atlidakis, Vaggelis, Patrice Godefroid, and Marina Polishchuk. "Restler: Stateful rest api fuzzing." *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019.

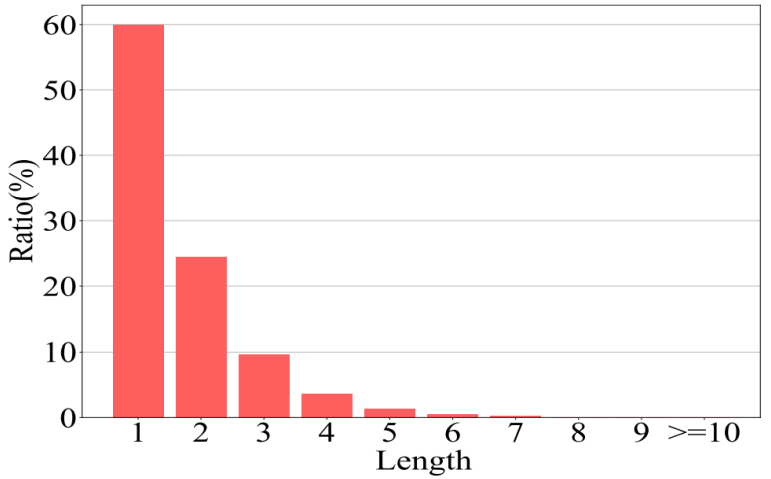
Insight

Zhejiang University

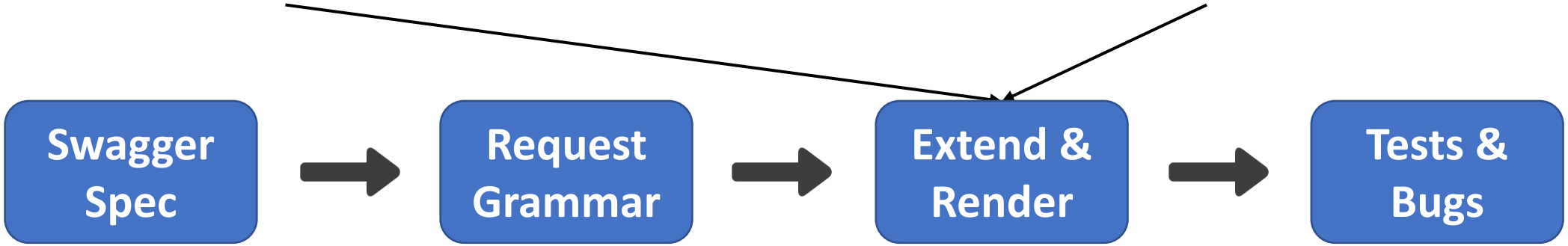
- Parameters in a REST API request have explicit/implicit relations.
- Requests could be easily rejected irrespective of relations.
  - *updated_before* and order in GitLab *Project* API are compulsory.
  - *import_url* and *initialize_with_readme* in GitLab *Project* API are contradicting.
- Some parameter combinations could lead to server errors.
  - The co-existence of *requirements_access_level*, *auto_cancel_pending_pipelines* and *initialize_with_readme* in GitLab *Group* API could raise a server error.

Insight

Gitlab API	Validation
/projects	72.01%
/groups	65.01%
/commits	53.52%



- **Low-quality** Requests
- **Short** Sequences



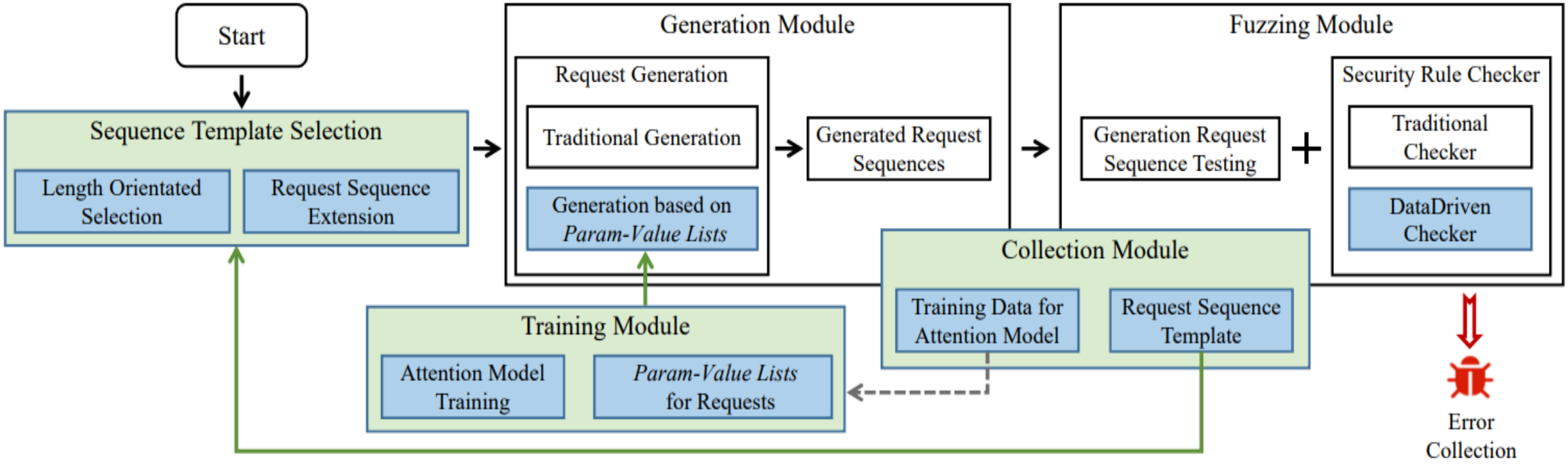
How to automatically enhance testcases to pass syntax/sematic check and test deep logic?

Design

Overview of MINER

Zhejiang University

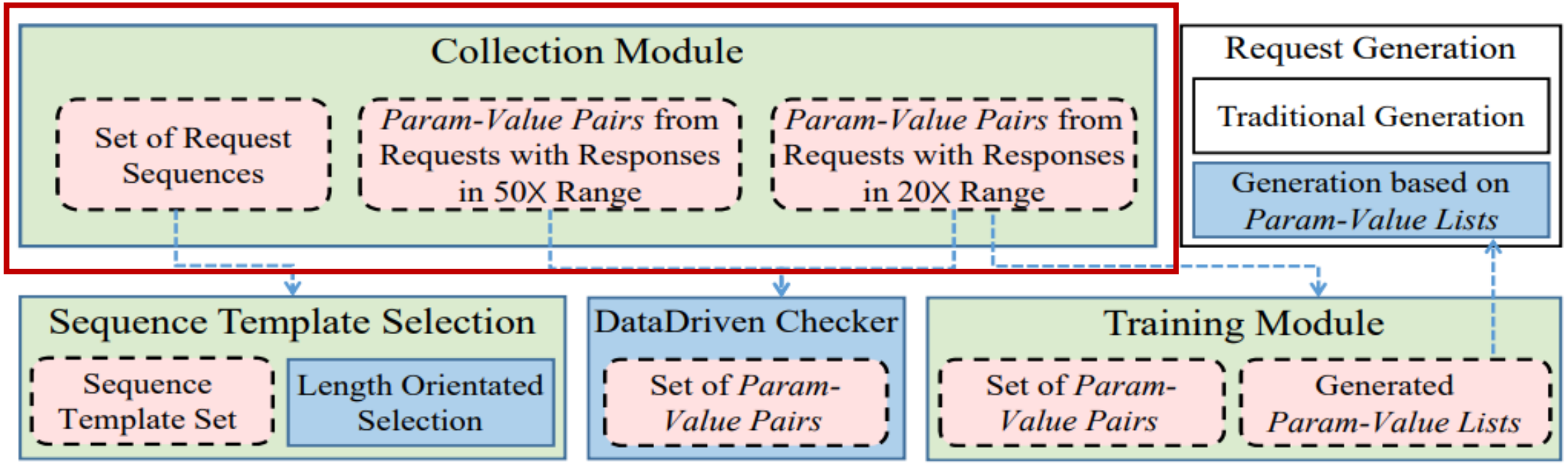
MINER: a hybrid data-driven framework to fuzz REST APIs in cloud services and discover the corresponding logic vulnerabilities.



Collection Module

¤ Requests Collection

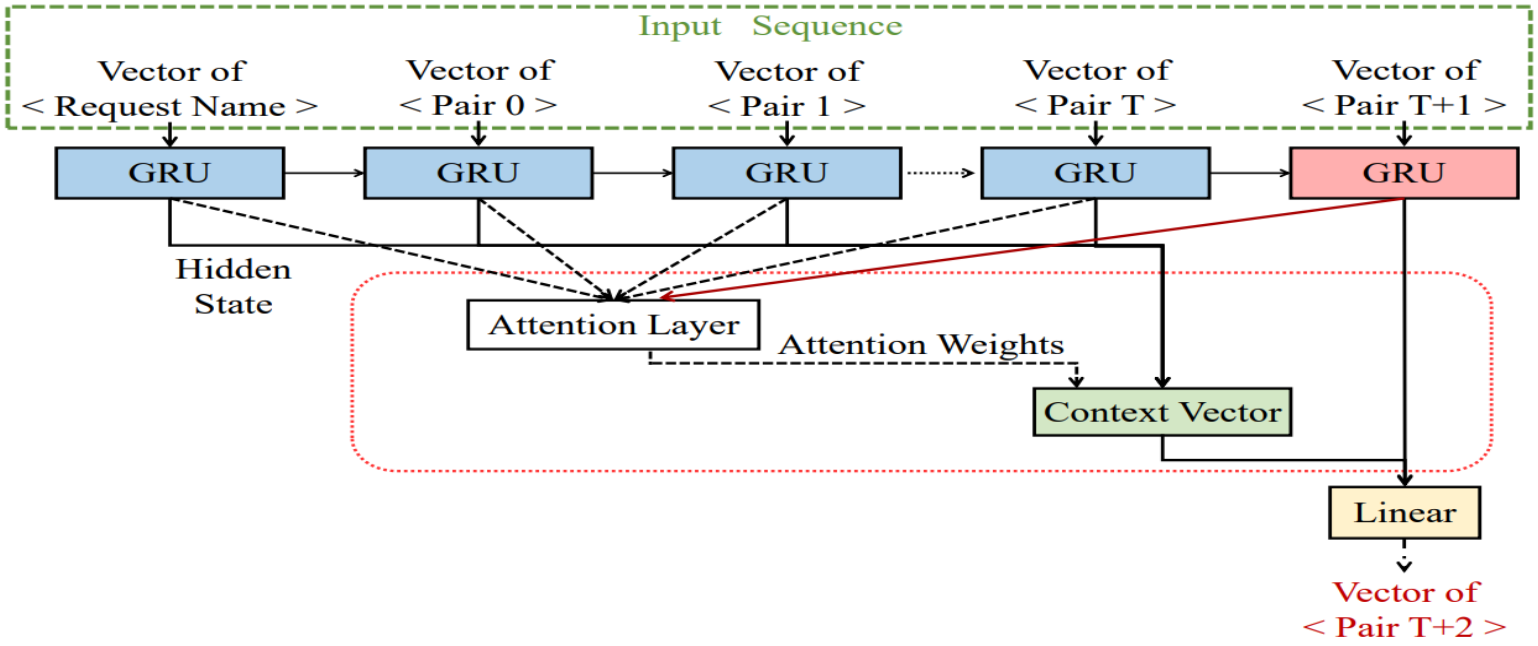
- Filter the requests through corresponding responses.
- Collect effective request sequences (20X, 50X).
- Collect param-value pairs from effective requests (20X, 50X).



Training Module

¤ Attention Model Training

- Learn the implicit relations among the parameters from history.
- Transform param-value pair generation into text generation.
- Embed the param-value pair lists into vectors and train the model.

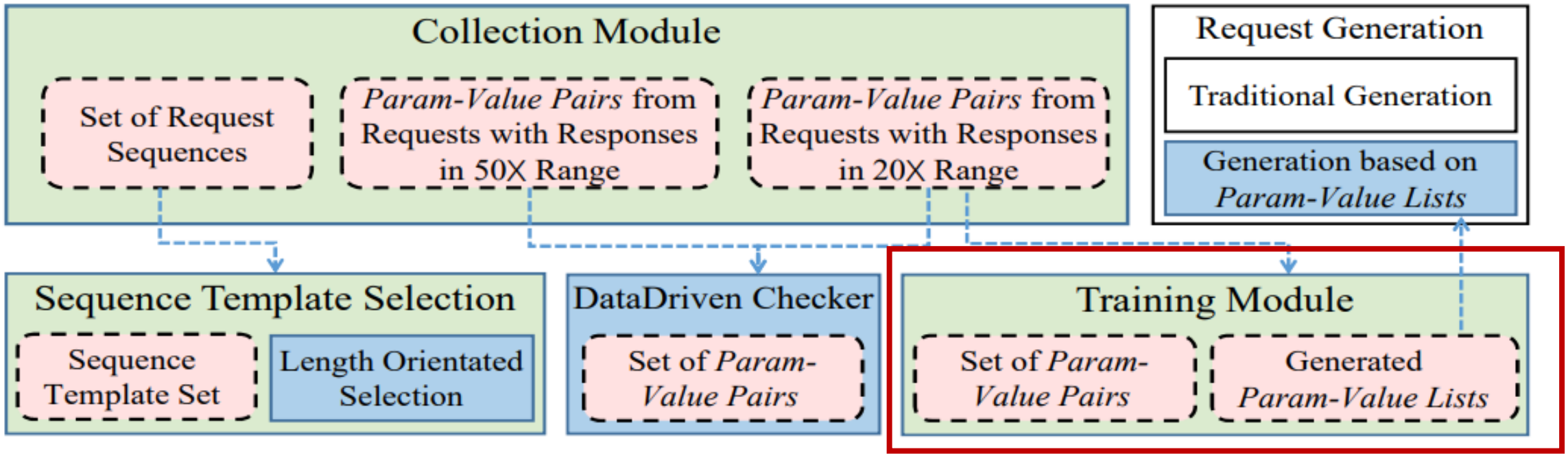


Training Module

Zhejiang University

¤ Attention Model Usage

- **Produce a set of lists of param-value pairs in which pairs are interrelated.**
- **The predicted pairs serve as candidates when MINER mutates requests.**
- **Training module is invoked periodically to update the model.**



DataDriven Checker & Sequence Template Selection

Zhejiang University

¤ DataDriven Checker

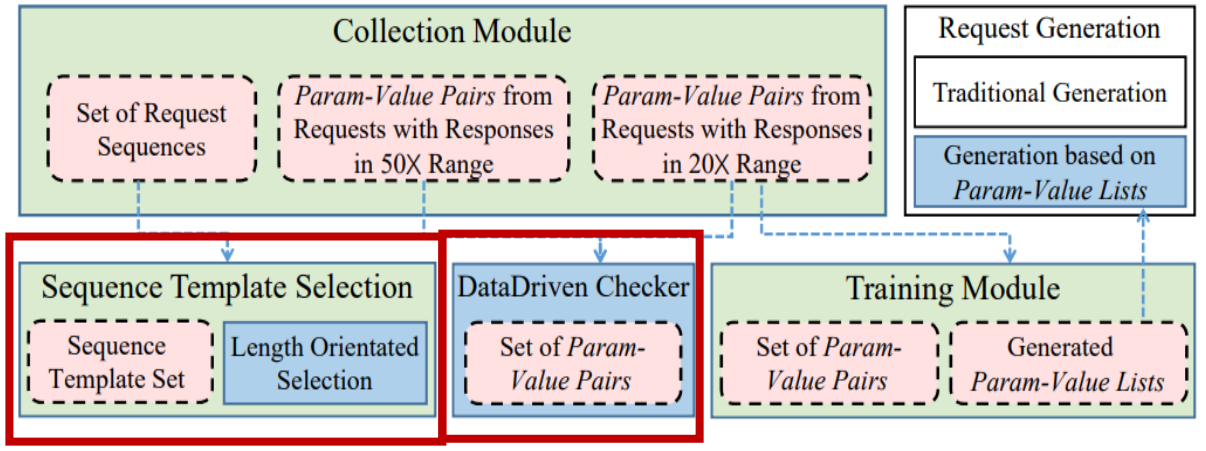
- Target at the undefined parameter violation.
- The parameters are sampled from the collected pairs.
- Inject undefined parameters into the requests.
- Bug oracle: whether the mutate requests lead to error code.

¤ Sequence Template Selection

- Prioritize the sequence templates

whose lengths are longer $p = \log_{10}(l+1)$

where l is the sequence length.



Evaluation

Experiment Settings

Zhejiang University

- **Baselines includes RESTler, MINER_PART(without DataDriven Checker)**
- **Each evaluation lasts 48 hours.**
- **We evaluate MINER on GitLab, Bugzilla and WordPress with 11 APIs.**

Cloud Service	REST API	# Request Templates	Description
GitLab 14.1.0-ce.0.	Projects	29	REST API to interact with projects of GitLab
	Groups	13	REST API related to groups of GitLab
	Issues	23	REST API to interact with GitLab issues
	Commits	14	REST API related to commits of GitLab
	Branches	8	REST API to interact with GitLab branches
Bugzilla 5.0.4	Comments	8	REST API to maintain comments in Bugzilla
	Bugs	11	REST API to maintain bug reports in Bugzilla
	Groups	6	REST API to maintain user groups in Bugzilla
WordPress 5.8.1	Categories	5	REST API to maintain categories in WordPress
	Posts	10	REST API to interact with posts in WordPress
	Comments	8	REST API to maintain comments in WordPress

Fuzzing Performance Analysis

Zhejiang University

¤ Results

- MINER_PART and MINER achieve **23%** higher pass rate and covers **7%** more unique request templates than RESTler.
- MINER discovers **15.27** unique errors while MINER_PART only discovers **9.91** and RESTler discovers **7.73** unique errors on average.
- MINER also uniquely finds **5.45** unique errors caused by misuse of undefined parameters using *DataDriven Checker.*

Target	API	# Total Request Templates	RESTler			MINER			MINER_PLUS		
			Pass Rate	# Unique Request Templates	Errors	Pass Rate	# Unique Request Templates	Errors	Pass Rate	# Unique Request Templates	Errors[a]
GitLab	Projects	29	72.01%	22	7	95.77%	26	17	95.78%	26	21 (9)
	Groups	13	65.01%	10	21	92.37%	12	26	92.28%	12	33 (10)
	Issues	23	86.56%	21	7	96.13%	21	7	95.42%	21	15 (6)
	Commits	14	53.52%	12	16	86.65%	12	20	86.70%	12	37 (12)
	Branches	8	81.10%	8	2	89.91%	8	3	89.31%	8	9 (6)
Bugzilla	Comments	8	88.79%	7	8	89.74%	8	8	90.23%	8	8 (0)
	Bugs	11	45.03%	4	7	91.05%	4	7	93.16%	4	14 (6)
	Groups	6	54.88%	4	5	74.01%	5	6	72.85%	5	11 (5)
WordPress	Categories	5	75.33%	4	8	91.96%	4	10	92.64%	4	13 (4)
	Posts	10	94.06%	10	4	95.13%	10	5	95.56%	10	7 (2)
	Comments	8	96.61%	4	0	99.65%	4	0	99.43%	4	0 (0)
Average			73.90%	9.64	7.73	91.12%	10.36	9.91	91.21%	10.36	15.27 (5.45)

[a]The number of unique errors found by MINER_PLUS is presented in two parts: the total number of all the unique errors as shown in front of the parentheses, and the number of unique errors found by the *DataDriven Checker* as shown in the parentheses.

Significance of Designs

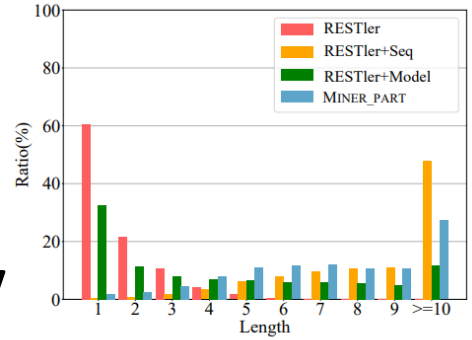
Zhejiang University

¤ Four variations: RESTler+Seq, RESTler+Rec1, RESTler+RecList, RESTler+Model

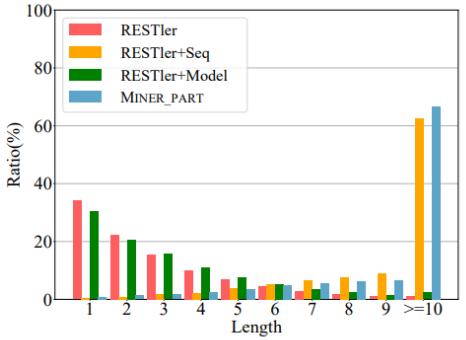
¤ Results

- Both the model and the sequence construction improve pass rate and generate longer sequences.
- The model can improve the ability of error discovery.
- Recording and replaying the history mechanically are low efficiency.

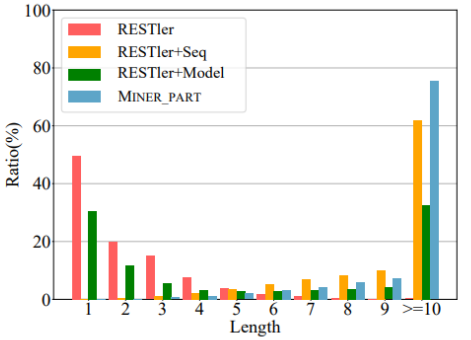
Cloud Service	REST API	RESTler		RESTler+Seq		RESTler+Rec1		RESTler+RecList		RESTler+Model		MINER_PART	
		Pass Rate	Errors	Pass Rate	Errors	Pass Rate	Errors	Pass Rate	Errors	Pass Rate	Errors	Pass Rate	Errors
GitLab	Projects	69.48%	7.00	89.57%	7.50	66.10%	7.50	65.66%	7.25	87.11%	9.00	94.70%	9.75
	Groups	60.28%	16.00	94.67%	19.50	65.22%	19.00	65.56%	17.00	94.53%	19.00	94.66%	19.50
	Issues	86.32%	6.50	89.06%	6.50	88.14%	7.00	90.38%	7.00	91.00%	7.50	94.31%	7.00
Bugzilla	Comments	86.19%	8.00	89.05%	8.00	87.05%	8.00	89.05%	8.00	87.94%	8.25	89.36%	8.00
WordPress	Categories	81.35%	9.00	86.40%	8.75	81.03%	9.00	81.44%	9.00	82.41%	9.50	90.53%	9.50



(a) GitLab Projects API.



(b) Bugzilla Comments API.



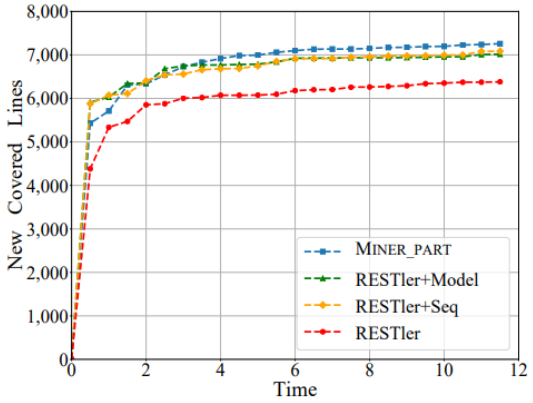
(c) Wordpress Categories API.

Coverage Analysis

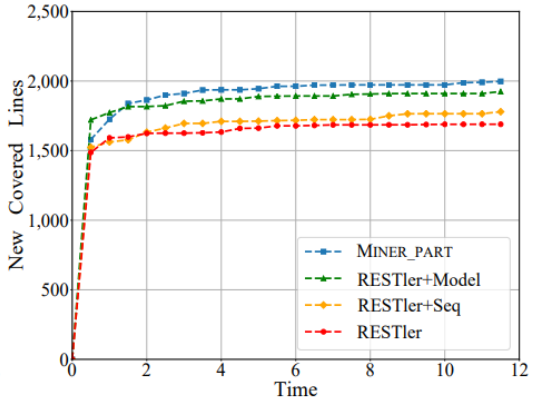
¤ Hook GitLab's source code to trace line coverage using *Coverband*[2].

¤ Results

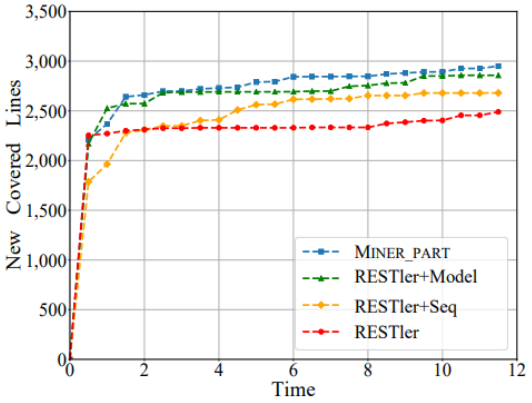
- Both the model and the sequence construction improve the line coverage.
- MINER_PART achieves the most line coverage due to the data-driven designs.



(a) GitLab Projects API.



(b) GitLab Groups API.



(c) GitLab Issues API.

[2] Coverband. https://github.com/danmayer/coverband

Analysis on Reproducing Published Serious Bugs

Zhejiang University

¤ Collect 4 published serious bugs manually as the ground truth.

¤ Results

- MINER can reproduce all the bugs more effectively and efficiently compared to RESTler.
- MINER not only finds relatively shallow bugs, but also specializes in constructing complex sequences.

No.	API	# Request Templates	Minimum Sequence Length To Trigger Bug	RESTler	MINER
1	Projects	3	3	N/A	66.5 mins
2	Projects	4	5	N/A	142.6 mins
3	Groups	1	1	27.6 mins	26.6 mins
4	Groups	1	1	13.7 mins	14.2 mins

Real Bug Analysis

Zhejiang University

¤ Results

- **MINER finds all the bugs discovered by RESTler, and also find 10 extra bugs, including 4 parameter misuse bugs.**
- **MINER could discover bugs effectively, especially the ones require long request sequences and specific parameter combinations due to data-driven designs.**
- **Example: The third request updates a hook with three undefined parameters "`requirements_access_level`", "`auto_cancel_pending_pipelines`" and "`initialize_with_readme`" will raise a server error.**

1. POST /api/v4/projects
2. POST /api/v4/projects/:proj_id/hooks
3. PUT /api/v4/projects/:proj_id?/hooks/:hook_id

1. `201` Created
2. `200` OK
3. `500` Internal Error

Analysis with Different Training Duration

Zhejiang University

¤ Evaluate the impact of different schedules of the Training Module.

¤ Results

- The training overhead increases when using a larger iteration duration.
- The pass rate slightly decreases when using a longer iteration duration.
- MINER's data-driven approaches have minimal extra costs.

Target	1 Hour		2 Hours		3 Hours	
	Overhead	Pass Rate	Overhead	Pass Rate	Overhead	Pass Rate
GitLab Projects API	346.20s	92.78%	380.80s	91.54%	989.30s	89.55%
GitLab Groups API	318.90s	93.49%	386.90s	92.80%	1,002.67s	91.92%
GitLab Issues API	463.40s	90.90%	498.67s	90.67%	1,001.60s	90.23%
GitLab Commits API	29.00s	86.66%	35.60s	84.45%	82.33s	82.56%
GitLab Branches API	16.70s	90.72%	34.40s	87.83%	43.33s	86.78%
Bugzilla Comments API	97.60s	88.89%	117.30s	86.47%	143.70s	85.37%
Bugzilla Bugs API	73.47s	91.33%	136.51s	90.65%	150.80s	89.86%
Bugzilla Groups API	28.52s	71.01%	41.00s	70.75%	55.33s	70.12%
WordPress Categories API	305.93s	91.02%	323.25s	90.84%	402.40s	90.14%
WordPress Posts API	198.59s	93.79%	212.50s	93.87%	384.67s	93.51%
WordPress Comments API	154.84s	99.07%	176.40s	98.86	216.30s	98.79%

Summary

Zhejiang University

¤ A novel data-driven REST API fuzzer named MINER, with 3 designs against low-quality requests, short sequences and parameter misuse.

¤ Evaluate MINER on three cloud services via 11 REST APIs. MINER achieves 23% higher pass rate, find 100% more reproducible errors than RESTler. 17 new bugs are found by MINER(4 from DataDriven Checker).

¤ Conduct comprehensive to demonstrate the outstanding performance of MINER.

Contact: puppet@zju.edu.cn
& stitch@zju.edu.cn