Static Analysis of Serverless Applications: Recent Results

Cumberland Lodge CDT Showcase Event

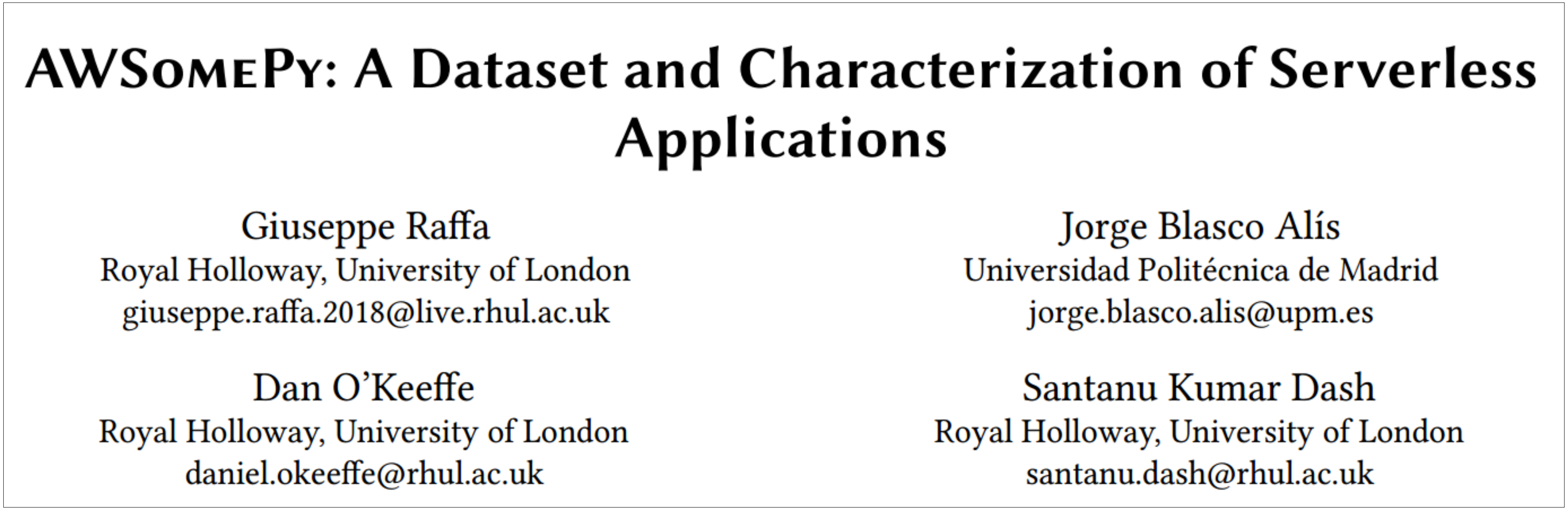
Author: Giuseppe Raffa

Supervisors: J. Blasco, D. O’Keeffe, S. K. Dash

Date: 18th April 2024

ROYAL
HOLLOWAY
UNIVERSITY
Of LONDON

AWSomePy Dataset Paper



AWSomePy: A Dataset and Characterization of Serverless Applications

Giuseppe Raffa
Royal Holloway, University of London
giuseppe.raffa.2018@live.rhul.ac.uk

Jorge Blasco Alís
Universidad Politécnica de Madrid
jorge.blasco.alis@upm.es

Dan O'Keeffe
Royal Holloway, University of London
daniel.okeeffe@rhul.ac.uk

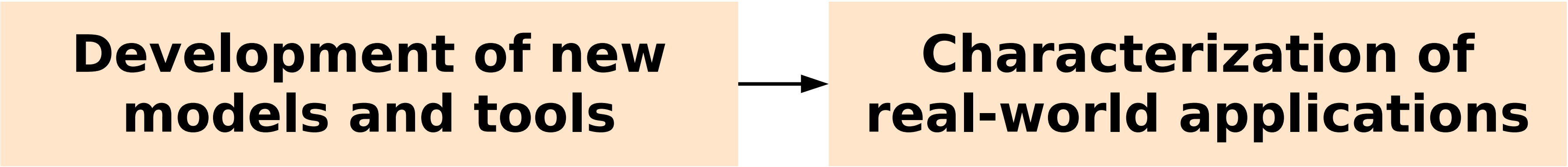
Santanu Kumar Dash
Royal Holloway, University of London
santanu.dash@rhul.ac.uk

https://dl.acm.org/doi/abs/10.1145/3592533.3592811

https://doi.org/10.5281/zenodo.7838076

Introduction

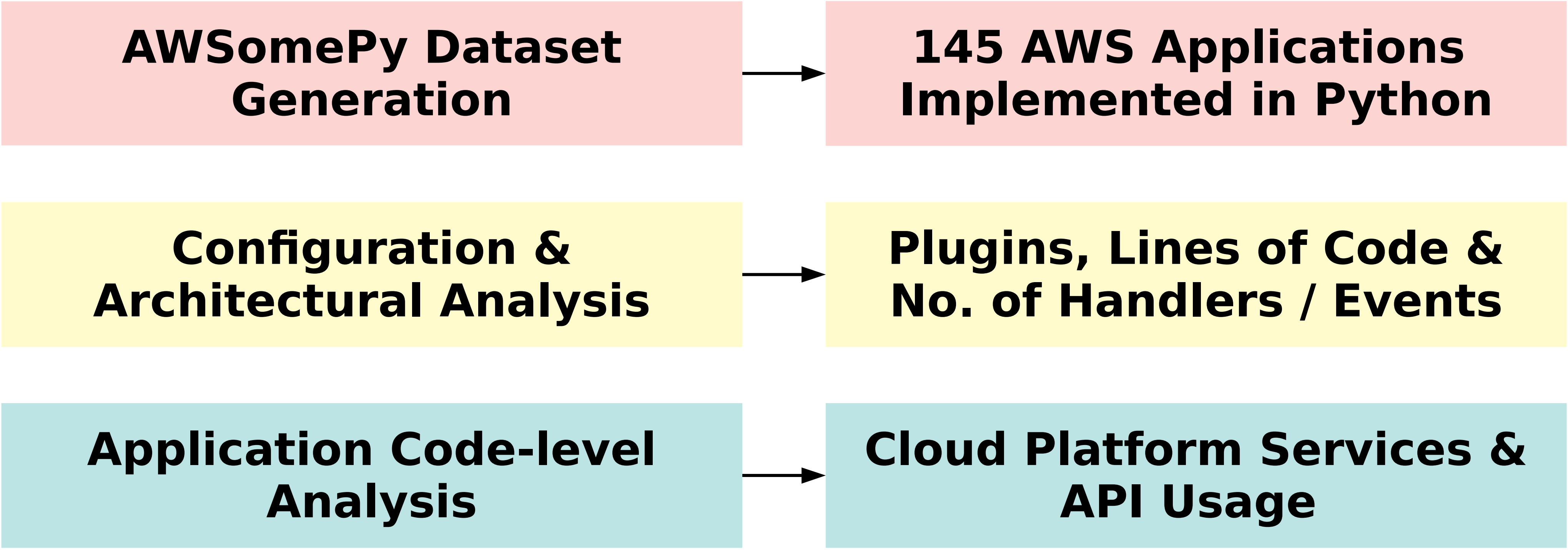
- **Serverless paradigm challenges**
  - Performance
  - Traceability
  - Security
- **Static and dynamic analysis**
  - Variety of sources and events
  - Existing analysis frameworks not optimized
  - Models / approximations needed for static analysis



Development of new models and tools → **Characterization of real-world applications**

Research Objective & Outline

- **Objective**
  - Identification of key trends in serverless applications



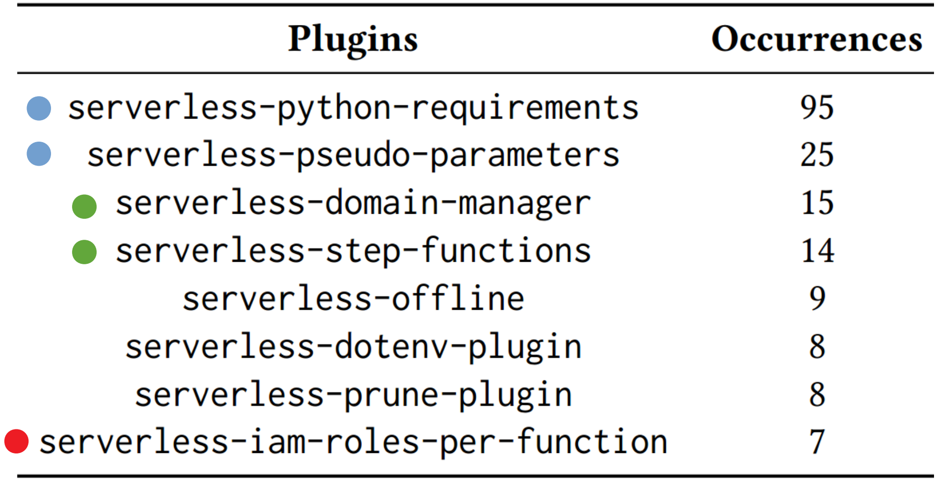
AWSomePy Dataset Generation → **145 AWS Applications Implemented in Python**

Configuration & Architectural Analysis → **Plugins, Lines of Code & No. of Handlers / Events**

Application Code-level Analysis → **Cloud Platform Services & API Usage**

Config. & Architectural Analysis

- **Plugin analysis**
  - Specified in infrastructure code file (YAML)
  - 44 plugins in total
- **Results**
  - 1st & 2nd => configuration
  - 3rd & 4th => functionality

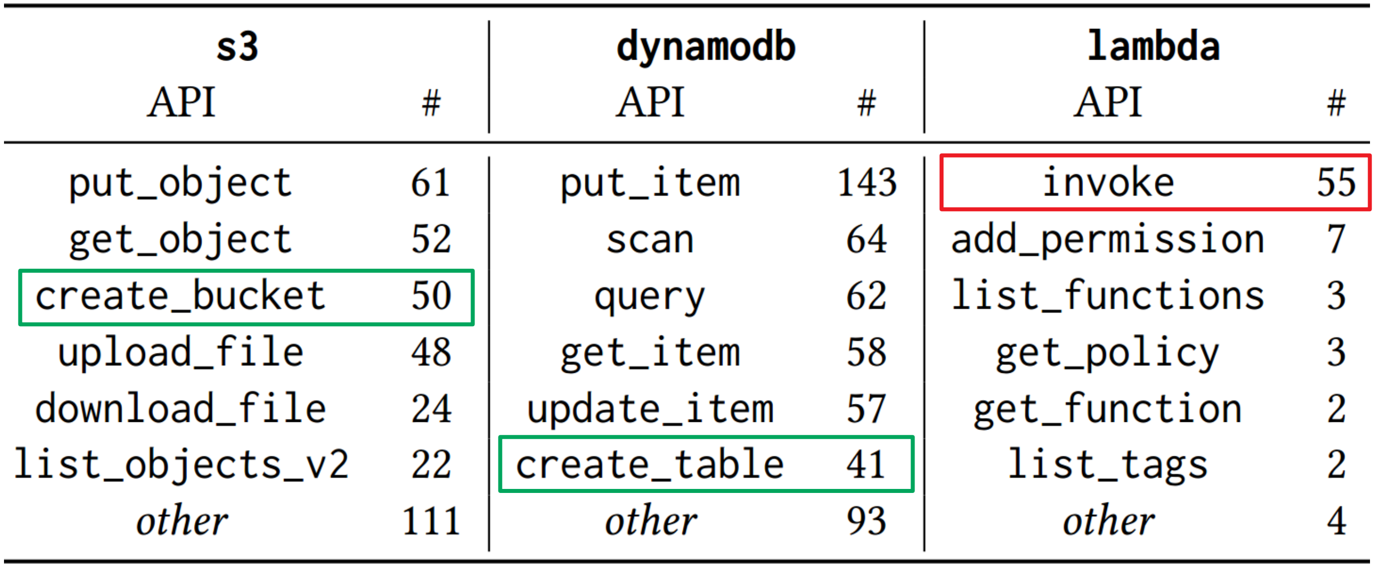


Plugins	Occurrences
🔵 serverless-python-requirements	95
🔵 serverless-pseudo-parameters	25
🟢 serverless-domain-manager	15
🟢 serverless-step-functions	14
serverless-offline	9
serverless-dotenv-plugin	8
serverless-prune-plugin	8
🔴 serverless-iam-roles-per-function	7

Developers are not configuring permissions in a granular fashion

Application Code-level Analysis

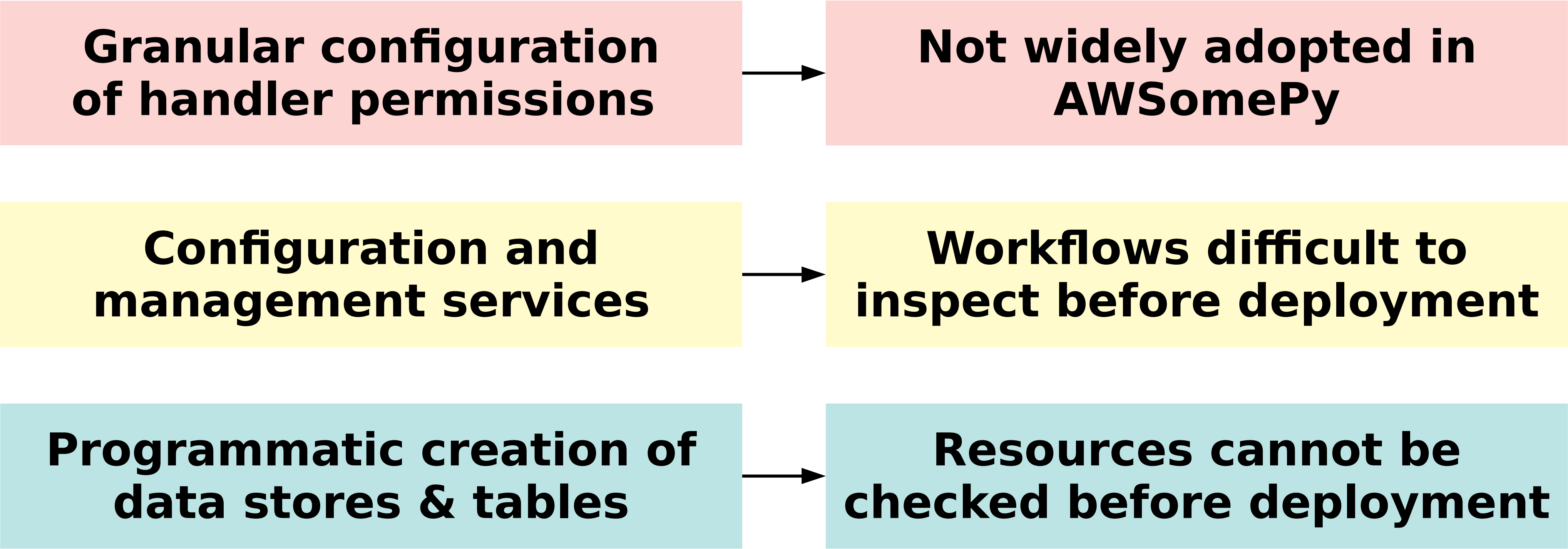
- **Cloud APIs**
  - Programmatic creation of buckets & tables
    - Resources cannot be checked via infrastructure code analysis
  - Use of invoke API to trigger handler execution
    - Added workflows not easily detectable via static analysis



s3		**dynamodb**		**lambda**	
API	#	API	#	API	#
put_object	61	put_item	143	invoke	55
get_object	52	scan	64	add_permission	7
create_bucket	50	query	62	list_functions	3
upload_file	48	get_item	58	get_policy	3
download_file	24	update_item	57	get_function	2
list_objects_v2	22	create_table	41	list_tags	2
other	111	*other*	93	*other*	4

Conclusion

- **Key takeaways**
  - All security-related

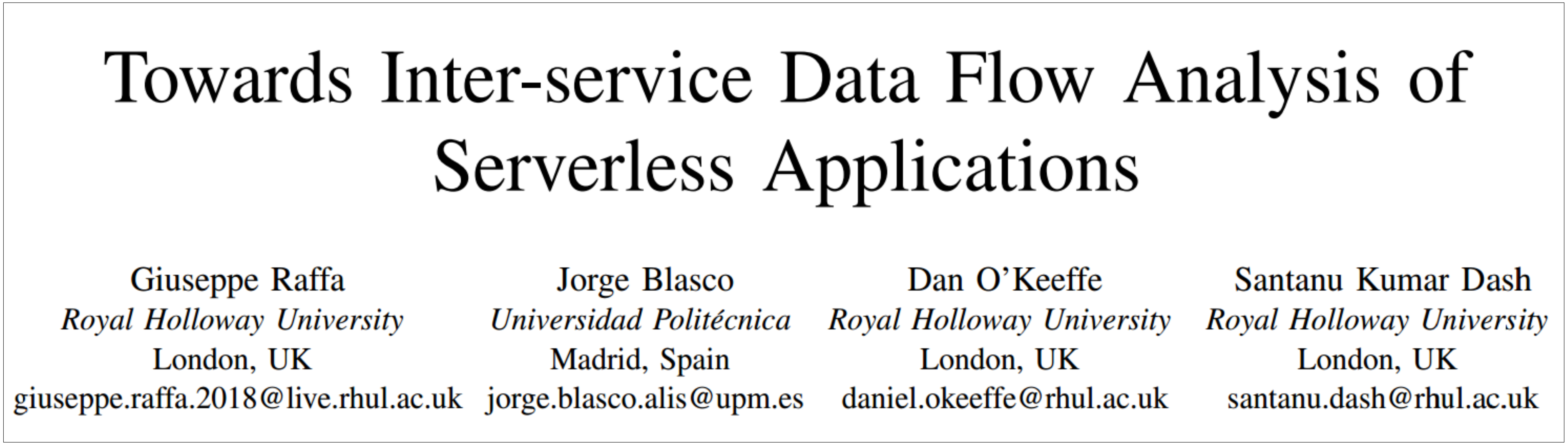


Granular configuration of handler permissions → **Not widely adopted in AWSomePy**

Configuration and management services → **Workflows difficult to inspect before deployment**

Programmatic creation of data stores & tables → **Resources cannot be checked before deployment**

Microbenchmarks Paper



Towards Inter-service Data Flow Analysis of Serverless Applications

Giuseppe Raffa
Royal Holloway University
London, UK
giuseppe.raffa.2018@live.rhul.ac.uk

Jorge Blasco
Universidad Politécnica
Madrid, Spain
jorge.blasco.alis@upm.es

Dan O'Keeffe
Royal Holloway University
London, UK
daniel.okeeffe@rhul.ac.uk

Santanu Kumar Dash
Royal Holloway University
London, UK
santanu.dash@rhul.ac.uk

SANER 2024 Early Research Achievement (ERA) Track

https://github.com/giusepperaffa/serverless-security-microbenchmarks

Motivation & Challenges

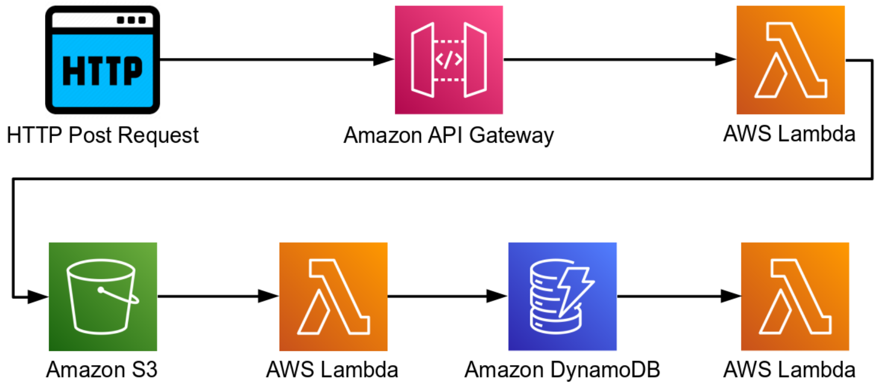
- **Why static data flow analysis?**
  - Most of serverless security tools rely on dynamic analysis
  - Static analysis is an effective supplement
- **What are the challenges?**
  - Information from infrastructure and application code
  - Variety of sources and events
  - Black-box nature of platform services
- **Our work**

Suite of security-oriented microbenchmarks

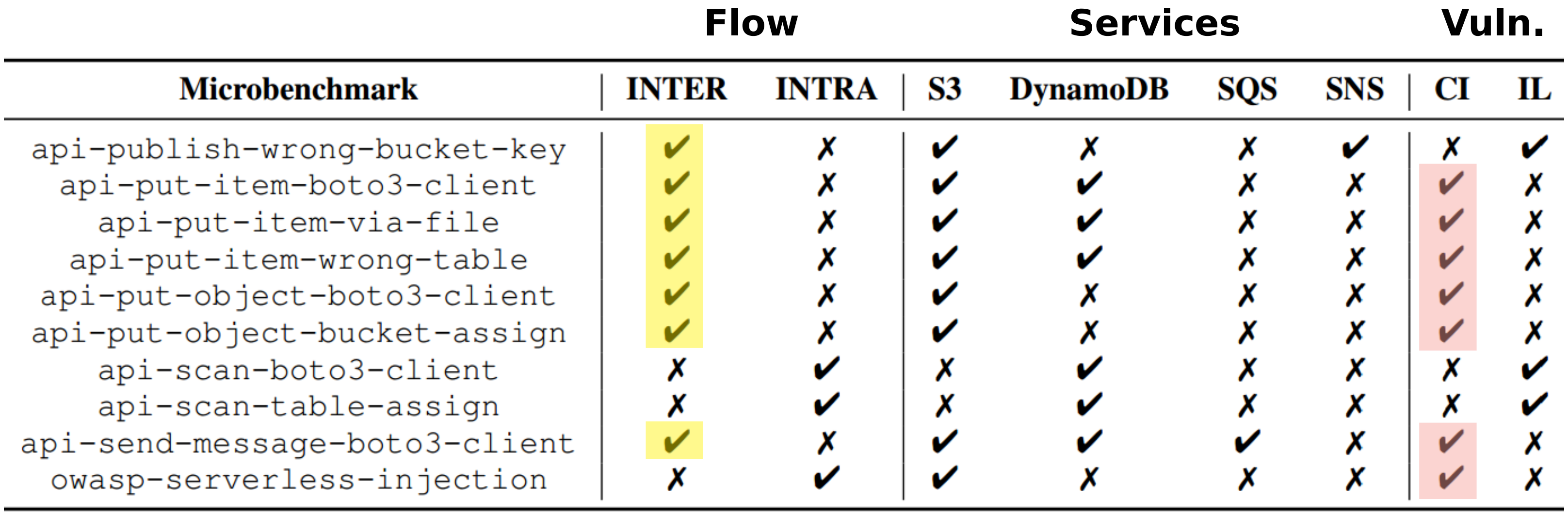
Approach to detecting security-sensitive data flows

Microbenchmarks Suite

- **Design approach**
  - Code injection and information leakage vulnerabilities
  - AWSomePy dataset characterization



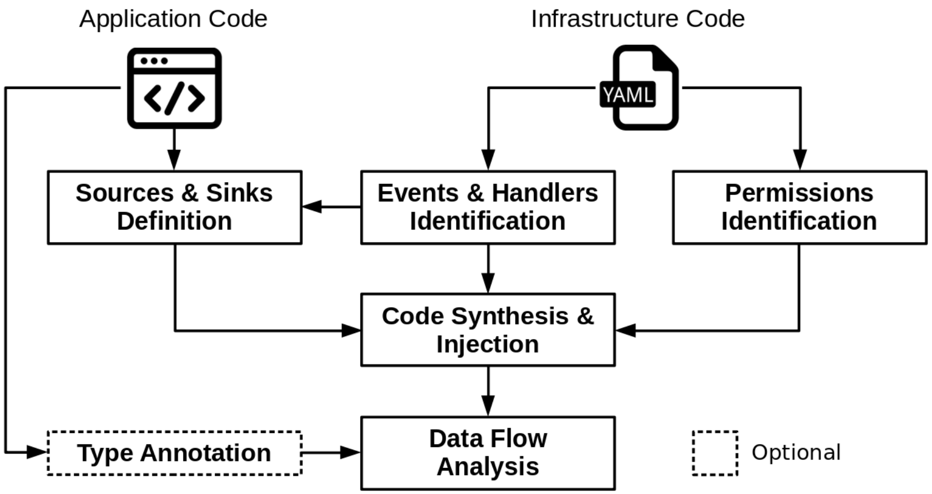
- **Summary**



	Flow		Services				Vuln.	
Microbenchmark	**INTER**	**INTRA**	**S3**	**DynamoDB**	**SQS**	**SNS**	**CI**	**IL**
api-publish-wrong-bucket-key	✓	✗	✓	✗	✗	✓	✗	✓
api-put-item-boto3-client	✓	✗	✓	✓	✗	✗	✓	✗
api-put-item-via-file	✓	✗	✓	✓	✗	✗	✓	✗
api-put-item-wrong-table	✓	✗	✓	✓	✗	✗	✓	✗
api-put-object-boto3-client	✓	✗	✓	✗	✗	✗	✓	✗
api-put-object-bucket-assign	✓	✗	✓	✗	✗	✗	✓	✗
api-scan-boto3-client	✗	✓	✗	✓	✗	✗	✗	✓
api-scan-table-assign	✗	✓	✗	✓	✗	✗	✗	✓
api-send-message-boto3-client	✓	✗	✓	✓	✓	✗	✓	✗
owasp-serverless-injection	✗	✓	✓	✗	✗	✗	✓	✗

Prototype Analysis Framework

- **Analysis approach**
  - Infrastructure and application code processed
  - Code instrumented to obtain synchronous equivalent


Application Code
Infrastructure Code
YAML
Sources & Sinks Definition
Events & Handlers Identification
Permissions Identification
Code Synthesis & Injection
Type Annotation
Data Flow Analysis
Optional

- **Implementation**
  - Code modified semi-automatically
  - Data flow analysis with Pysa
- **Evaluation**

7 true positives

2 false positives

1 false negative

ROYAL
HOLLOWAY
UNIVERSITY
Of LONDON

Conclusion & Future Work

- **Key takeaways**

Security-sensitive data flows

New suite of microbenchmarks

Studied approach is feasible

- **Future work**
  - Fully automated analysis pipeline
  - Improvement of infrastructure code processing
  - Support for higher number of cloud services and APIs

https://github.com/giusepperaffa/serverless-security-microbenchmarks