Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks

Trust More, Serverless

SysTor'2019

Stefan Brenner, June 3rd, 2019

Technische Universität Braunschweig, Institute of Operating Systems and Computer Networks

Cloud Popularity Impacted by Security Issues

- Increasing popularity of clouds
- Cloud security challenges
  → Hinder cloud adoption
- Vision: Trusted cloud
  - Enables currently impossible use cases
  - Usage of trusted execution technology

Usage of Trusted Execution Technology

- Creation of a Trusted Execution Environment (TEE)
  - Goal: Small sensitive compartments inside TEE
- Holistic approach (legacy applications)
  - ✗ Large Trusted Computing Base (TCB)
- Application partitioning (tailored)
  - ✗ High porting effort

Software Design: Monolithic $\neq$ Modern

- Modern **modular** architectures
  e.g. micro services, functions
  - Small independent components
  - Clearly defined interfaces
  - Selective scalability
  - Simpler and independent development

Software Design: Monolithic $\neq$ Modern

- Modern **modular** architectures
  e.g. micro services, functions
  - Small in
  - Clearly d
  - Selective
  - Simpler and independent development

Trusted FaaS

Trusted serverless or Function-as-a-Service (FaaS) cloud!

Trust More, Serverless

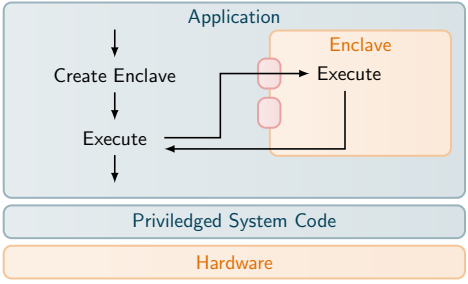
- **Background**
  - Intel SGX
  - Serverless Computing
- **Design & Implementation**
- **Evaluation**
- **Conclusion**

Trust More, Serverless

- **Background**
  - Intel SGX
  - Serverless Computing
- Design & Implementation
- Evaluation
- Conclusion

Intel Software Guard Extensions

- Intel Software Guard Extensions (SGX)
  - CPU instruction set extension for trusted execution
  - “Secure enclaves” inside user processes
  - Transparent memory encryption (with integrity)
  - Remote Attestation via Intel Attestation Service



Serverless and FaaS

- **Evolution of cloud computing**
  1. Infrastructure-as-a-Service (IaaS)
  2. Platform-as-a-Service (PaaS)
  3. **Function-as-a-Service (FaaS)**
     - Single standalone functions → **Lambdas**
     - Fine-grained accounting, no idle cost
     - Most maintenance done by provider

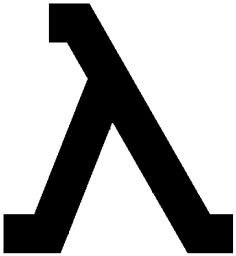
Trust More, Serverless

- Background
  - Intel SGX
  - Serverless Computing
- **Design & Implementation**
- Evaluation
- Conclusion

Secure Serverless Computing

Platform Vision

- **Basic Properties**
  - Lambda inside enclave
  - Parallel (competing) Lambda execution
  - Resource efficiency
  - Transparent Lambda attestation
- **Challenges:**
  - Selection of suitable programming language and Lambda library support
  - Design of a secure and efficient Lambda execution platform
  - Transparent remote attestation of Lambdas



Secure Serverless Computing

Programming Language & Runtime

	TCB	Isolation	Sharing
Native			
Multiple Enclaves		Process	
Single Enclave		Native Sandbox	

Secure Serverless Computing

Programming Language & Runtime

	TCB	Isolation	Sharing
Native			
Multiple Enclaves		Process	
Single Enclave		Native Sandbox	

✗ Native: sandbox?

Secure Serverless Computing

Programming Language & Runtime

✗ Native: sandbox?

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython: large TCB

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython: large TCB
- → JavaScript

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍
JavaScript			
MuJS	👍	Context	👍
Duktape	👍	Context	👍
Google V8	👎	V8 Isolate	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython: large TCB
- → JavaScript
  - ✗ MuJS: language support

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍
JavaScript			
MuJS	👍	Context	👍
Duktape	👍	Context	👍
Google V8	👎	V8 Isolate	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython: large TCB
- → JavaScript
  - ✗ MuJS: language support
  - ✓ Duktape: lean TCB

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍
JavaScript			
MuJS	👍	Context	👍
Duktape	👍	Context	👍
Google V8	👎	V8 Isolate	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython: large TCB
- → JavaScript
  - ✗ MuJS: language support
  - ✓ Duktape: lean TCB
  - ✓ Google V8: high performance

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
Interpreted			
CPython	👎	Sub Interpr.	👍
JavaScript			
MuJS	👍	Context	👍
Duktape	👍	Context	👍
Google V8	👎	V8 Isolate	👍

Secure Serverless Computing

Programming Language & Runtime

- ✗ Native: sandbox?
- ✗ CPython:
- → JavaScript
  - ✗ MuJS: language support
  - ✓ Duktape: lean TCB
  - ✓ Google V8: high performance

Selected Variants:

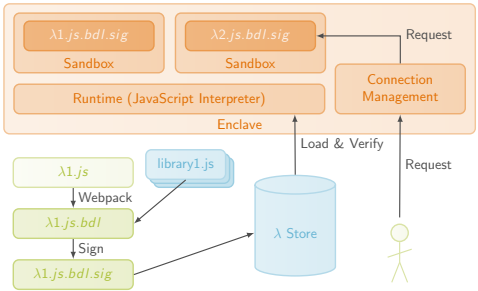
Pure JavaScript Lambdas on Duktape and Google V8.

	TCB	Isolation	Sharing
Native			
Multiple Enclaves	👍	Process	👎
Single Enclave	👎	Native Sandbox	👍
			👍
JavaScript			
MuJS	👍	Context	👍
Duktape	👍	Context	👍
Google V8	👎	V8 Isolate	👍

Secure Serverless Computing

Architecture

- **JavaScript Runtime** in enclave
  - Lightweight JavaScript interpreter: Duktape
  - Additional: Fast but large Google V8
  - Lambdas executed in interpreter sandbox
- Secure Lambdas:
  - **Signed** Lambda **bundles**
  - **Load and verify** on demand

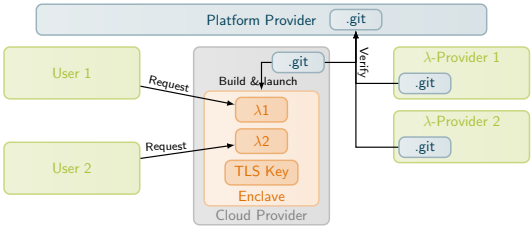


Secure Serverless Computing

Trust Model

- How to establish trust into Lambdas?
  1. Signed Lambda is loaded
  2. Attester verifies enclave
  3. Attester verifies Lambda based on its signature
  4. Attester uploads TLS key

  $\Rightarrow$ Implicit attestation on every request



Trust More, Serverless

- Background
  - Intel SGX
  - Serverless Computing
- Design & Implementation
- **Evaluation**
- Conclusion

Evaluation Methodology and Trusted Computing Base

- Methodology
  - Clients issue TLS-encrypted requests to trusted Lambda platform
  - TCB, throughput and enclave memory footprint measurement

Evaluation Methodology and Trusted Computing Base

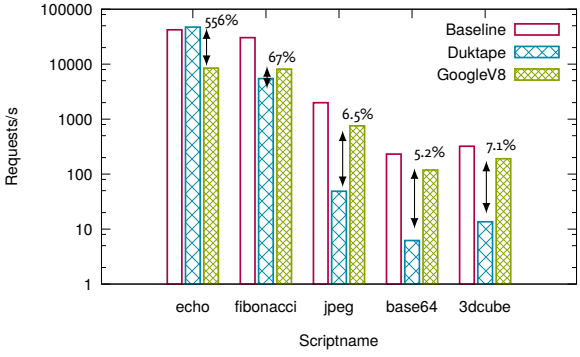
- Methodology
  - Clients issue TLS-encrypted requests to trusted Lambda platform
  - TCB, throughput and enclave memory footprint measurement
- Trusted Computing Base
  - Google V8 TCB 7× larger than Duktape

	Duktape	V8
Interpreter	185,392	1,308,702
Environment	214,156	17,193,624
Platform	1,529	1,002
Sum	401,077	18,503,328

Secure Serverless Computing

Performance

- Low overhead of secure Duktape (echo)
- Secure Google V8 almost 16× faster than secure Duktape
- Secure Google V8 ≈50% of baseline
  Secure Duktape only ≈ 3%

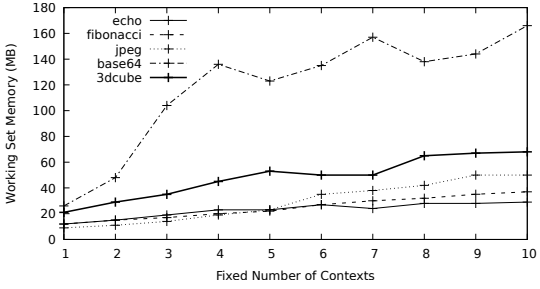


(base64 and 3dcube are part of the JetStream JavaScript benchmark suite)

Secure Serverless Computing

Memory Footprint

- No excessive SGX paging due to lean memory footprint
- Secure Duktape ≈ 38% lower memory footprint than secure Google V8



Secure Google V8 memory footprint

Trust More, Serverless

- Background
  - Intel SGX
  - Serverless Computing
- Design & Implementation
- Evaluation
- **Conclusion**

Conclusion

- Secure Lambda execution platform based on Intel SGX
- Execution of pure JavaScript Lambda inside SGX enclave
- Secure Duktape is much slower than secure Google V8

  ...but requires significantly less memory

  ...and comprises a much smaller TCB

  $\Rightarrow$ A price tag for transparent security in the FaaS cloud!

$\Rightarrow$ This project was funded by Intel in the *TFaaS project*!