Enforcing Authorization Policies using Transactional Memory Introspection

Arnar Birgisson, Mohan Dhawan, Úlfar Erlingsson, Vinod Ganapathy, and Liviu Iftode

Proc. 15th ACM Conference on Computer and Communications Security (CCS '08), pp. 223-234, 2008.

DOI: 10.1145/1455770.1455800

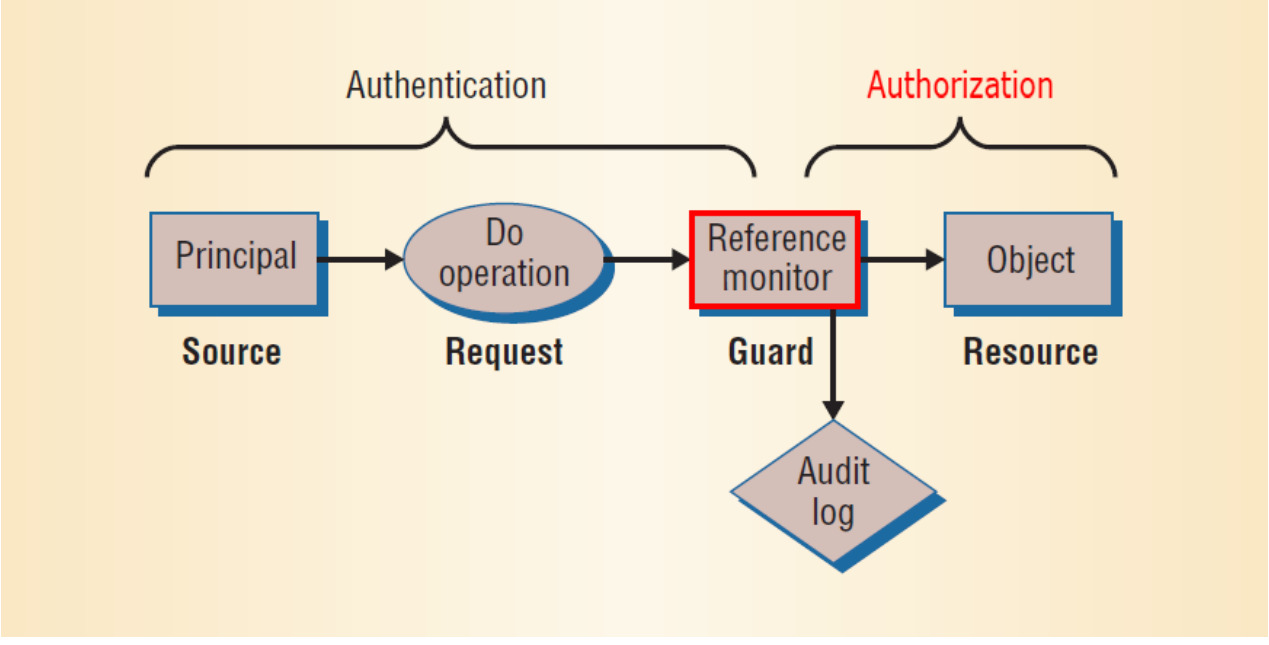
Andreas Ecke

Summary

This article presents an alternative, simplified approach for enforcing authorization policies and evaluates its practicability. It is based on transactional memory, a concurrency control mechanism where access to shared memory is performed in atomic transactions.

Summary

This article presents an alternative, simplified approach for enforcing authorization policies and evaluates its practicability. It is based on transactional memory, a concurrency control mechanism where access to shared memory is performed in atomic transactions.



from B.W. Lampson, "Computer Security in the Real World," doi:10.1109/MC.2004.17

Appreciative comment

Authorization enforcement is done inside transactions

- everything is rolled back if one authorization check fails
- writes to the security-relevant resources are reverted
- deferred authorization checks preserve integrity

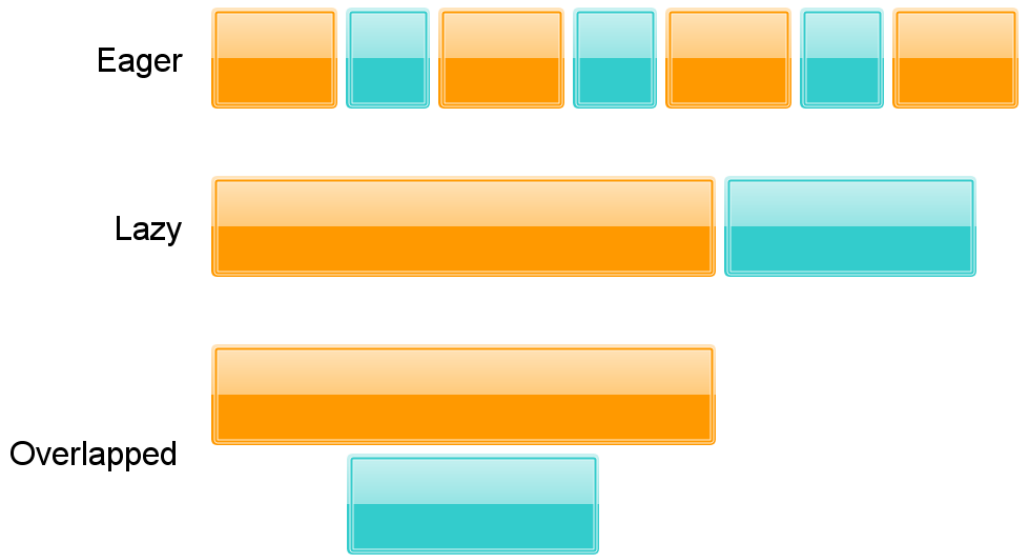
$\Rightarrow$ lazy and overlapped enforcement

Appreciative comment

Authorization enforcement is done inside transactions

- everything is rolled back if one authorization check fails
- writes to the security-relevant resources are reverted
- deferred authorization checks preserve integrity

$\Rightarrow$ lazy and overlapped enforcement



Appreciative comment

Lazy and overlapped authorization enforcement

- speculative execution of transactions
- yields better performance
- not suitable to ensure confidentiality

Critical comment

Measurement with a single STM system only

- DSTM2 is a unoptimized research STM library
- overhead of TMI compared to pure DSTM2-version is less then 11%
- real-world STM implementations might yield a bigger overhead for TMI
- tests with other optimized STM systems needed!
- measurements based on single client settings only

Question for you

The authors propose transactional memory introspection for implementing the guard which enforces authorization policies. Can TMI also be used in the information flow control model, i.e. for implementing a guard that controls outgoing information?

Question for you

The authors propose transactional memory introspection for implementing the guard which enforces authorization policies. Can TMI also be used in the information flow control model, i.e. for implementing a guard that controls outgoing information?