

CLOUD SECURITY

AWS Transaction Integrity Lab

Applying Clark-Wilson concepts with Lambda, S3, and IAM

Implemented academic lab

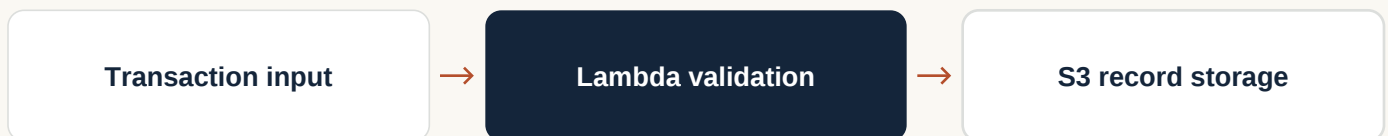
Bellevue University · CIS 312 · May 2026

Built an AWS lab that channels transaction-record writes through a validating Lambda function. Combined a scoped execution role, an explicit S3 bucket-policy deny, and audit logging to explore Clark-Wilson transaction-integrity concepts.

Project objective

Protect transaction records from direct modification while allowing validated writes through a designated procedure. The lab translates a formal integrity model into a small, testable cloud architecture.

Implemented lab architecture



The execution role authorizes the intended write path. A bucket policy denies direct writes by other principals. CloudWatch and CloudTrail support logging and audit review.

Controls implemented

01 Validate before writing

A Python Lambda function checks required fields, positive numeric amounts, and distinct submitter/approver values before writing a transaction record.

02 Limit the write path

The execution role grants S3 write access to the designated objects and CloudWatch logging, without S3 read or delete permissions. An explicit bucket-policy deny restricts direct writes by other principals.

03 Preserve history and audit evidence

Configured S3 public-access blocking and versioning, plus a multi-region CloudTrail trail with management events and relevant S3 data events. Enabled log-file validation to generate digest files for later integrity checks.

SKILLS & TOOLS

AWS Lambda / Amazon S3 / IAM / Python / CloudTrail / CloudWatch



CLOUD SECURITY

Evidence, scope & next steps

Four documented test cases

The lab report records the following outcomes. The tested write path remained usable while direct uploads were denied.

INPUT / ACTION	RECORDED OUTCOME
Valid transaction; distinct submitter and approver values	Accepted; record written to S3
Matching submitter and approver values	Rejected; 403 response value
Direct S3 upload using an administrator identity	AccessDenied
Valid Lambda write after bucket-policy deny	Accepted; intended write path still worked

SCOPE & LIMITATIONS

The approval rule compares supplied values; it does not authenticate a second approver. This is a Clark-Wilson-inspired lab, not a complete implementation or a compliance assessment. Versioning and CloudTrail support recovery and audit review, but do not by themselves provide immutable retention or verify the validity of stored records.

What I learned

The key lesson was how application validation, identity permissions, and resource policies reinforce one another. Testing the denied path was as useful as testing successful writes: it showed that the applied bucket policy blocked a direct upload while preserving the intended Lambda workflow.

Next steps

1. Bind approval to independently authenticated identities.
2. Add a procedure that checks stored-record integrity.
3. Evaluate protected record/audit retention and alerts for denied writes.

FURTHER READING

AWS: IAM policy evaluation
AWS: CloudTrail log-file integrity validation
NIST: Integrity models workshop report