Object Lock prevents deletion.

It does not prevent compromise.

True immutability isn't a storage feature. It's a system property.

Swipe to see the 5 layers →

Most teams equate immutability with:

S3 Object Lock

Enable it. Set retention.
Check the compliance box.

WORM Storage

Write-once media.
Air quotes around the air gap.

Both are legitimate controls. Neither is sufficient on its own.

Attackers don't have to touch the storage layer to make recovery impossible.

01

Layer 01 — Storage Immutability

Object Lock & WORM Enforcement

PROTECTS AGAINST

Prevents deletion and modification at the object level. Retention policies enforce a time-locked floor.

FAILURE MODE

Does NOT protect against control plane compromise or backup software manipulation.

02

Layer 02 — Backup Software Enforcement

Retention Logic & Immutability Flags

PROTECTS AGAINST

Backup software enforces its own retention policies independently of the storage layer.

FAILURE MODE

Attacker disables immutability flags in the backup software before the attack begins.

03

Layer 03 — Access Control Plane

IAM, RBAC & Credential Isolation

PROTECTS AGAINST

Controls who can modify retention settings, disable Object Lock, or delete backup jobs.

FAILURE MODE

Admin credentials compromised — immutability bypassed at the policy level before storage is touched.

04

Layer 04 — Air Gap

True vs. Connected Separation

PROTECTS AGAINST

Logical separation, delayed replication, and offline copies keep a copy genuinely unreachable.

FAILURE MODE

Most air gaps are connected. Shared credentials across environments = no real gap under compromise.

05

Layer 05 — Recovery Validation

Proven Restorability Under Adversarial Conditions

PROTECTS AGAINST

Restore tests, corruption detection, and ransomware-safe recovery paths confirm real recoverability.

FAILURE MODE

If you haven't tested recovery, you haven't proven immutability. Immutable data you can't restore is an archive.

Attackers don't delete your backups.

They make them unusable.

01 Object Lock ON → attacker deletes the backup catalog. Objects immutable. Recovery: impossible.

02 Snapshots intact → recovery toolchain compromised. You can't trust the restore.

03 Air gap exists → credentials span environments. The gap is logical, not physical.

04 Immutable storage → backup jobs silently failing 30 days. No clean restore point.

01 The Catalog Attack

Object Lock is configured correctly. Retention is enforced. The data is immutable.

The attacker doesn't touch storage. They delete the backup catalog — the index that tells the recovery system where the clean copies live.

The objects exist. Recovery is undefined. RTO is now measured in days, not hours.

02 The Credential Bridge

The backup target lives in a separate AWS account. Object Lock is enabled. It looks textbook.

But the IAM role that writes backups is accessible from the production environment. Under credential compromise, the attacker crosses the account boundary through the same credentials that own production.

The gap is logical. Logical gaps are traversable.

03 The Silent Failure

Immutable storage is configured. The compliance dashboard shows green.

Backup jobs have been silently failing for 23 days due to a misconfigured agent. The ransomware was present before the oldest retained copy.

Immutability enforced. No clean restore point. The protection worked exactly as designed — and it didn't matter.

Object Lock is the floor.

Immutability is enforced at the system level — or not at all.

L1	L2	L3	L4	L5
Storage Immutability	Backup Software	Access Control	Air Gap (True)	Recovery Validation

Full breakdown: 5-Layer Model, real failure modes, Architect's Verdict

rack2cloud.com/immutable-backup-object-lock/