keepit

Key takeaways

Are you breach ready?

Learn how immutability minimizes damage

When a breach hits, every second counts, and immutability is what keeps those seconds from turning into days of downtime.

It's a tough environment out there where threat actors routinely bypass perimeter defenses. Adopting a breach-assumed mindset is no longer optional — it's essential.

This was the key focus of a a recent Keepit webinar, where Paul Robichaux, Senior Director of Product Management at Keepit and Microsoft MVP, and Bart Binder, Red Team Manager at Keepit, discussed how isolation plus immutability stops any tampering of your backup in its tracks, and which design choices (short-lived tokens, ingest-time locks, separated planes) actually matter. Read the key takeaways to turn principles into practical steps that strengthen your SaaS backups and speed up recovery.

The key takeaways

Immutability isn't a debate — it's a requirement

The question isn't whether immutability reduces breach impact; it demonstrably does. The real discussion is about which implementation you choose, because the type of immutability you choose changes your ability to quickly and completely recover.

Tamper detection ≠ immutability

Checksums and signatures are valuable to detect corruption, but preventing alteration in the first place is far stronger. Immutability should be your last line of defense when everything else (perimeter, credentials, controls) is failing.

Real-world kill chain = minutes to compromise

In many instances, attackers attempt to purge logs and bulk-delete backups. In an example discussed, the chain broke only because the storage layer enforced WORM at ingest and delete calls were refused. Breaches only take minutes to compromise, but immutability can give teams critical days to respond.

Isolation matters as much as immutability

Air-gapping has a modern form: Keep backups in an environment attackers can't pivot into. Placing Microsoft 365 data in a vendor-independent cloud with no dependencies on Azure/AWS/GCP (and no shared credentials/replication paths) removes a rich attack surface. Replication helps with disasters, but it's not true protection if an attacker can reach every replica.

“Checkbox immutability” is not enough

Storage flags that mark data “immutable” at the software layer can be undermined via misconfigurations, API abuse, or clock spoofing. Strong designs lock objects at ingest, anchor time in hardware (e.g., TPM), and prevent public APIs from ever reaching the storage plane.

Short-lived tokens reduce blast radius

Token theft and reuse are fast-rising attack vectors. Keep token lifetimes as short as practical and layer conditional access. A stolen token that expires in an hour is dramatically less useful than one valid for a day.

Trade-offs are real — choose protection over perfection

Immutable backups limit the ability to remove inadvertently stored sensitive content later. Mitigation should be earlier in the lifecycle: Data classification and data loss prevention in production will prevent unwanted data from being created or synced, rather than relying on post-hoc removal from backups.

Prioritize recovery speed

Immutability dramatically limits ransomware leverage and preserves your ability to restore, but it doesn’t replace zero trust, endpoint controls, IR processes, or business continuity planning. Recovery speed still hinges on role-based access, guided restores, tested runbooks, and organizational readiness.

Breach-ready actions for your organization:

- Demand true immutability (locked at ingest, hardware-anchored time, no-modify APIs).
- Physically/logically isolate backups from primary clouds and identities.
- Enforce short-lived tokens and conditional access.
- Separate control/storage planes; require dual control for destructive changes.
- Test restores frequently (including large-scale and identity/control-plane scenarios).
- Pair immutability with DLP/classification upstream and practiced incident response.

Keepit’s storage design: Immutability by design

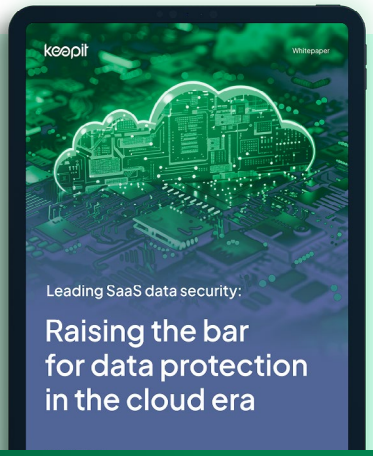
Keepit stores data in a Merkle tree object store with only two operations — read block and write block.There is no “modify block” API. Each version chains cryptographic hashes to prior blocks, delivering built-in dedupe, integrity verification, and tamper evidence.

Combined with an independent, vendor-neutral cloud, this yields practical air-gap properties in addition to provable immutability.

Leading SaaS data security

Read our whitepaper on how Keepit is raising the bar for data protection in the cloud era to learn more about our unique architecture.

Download now



Keepit provides a next-level SaaS data protection platform purpose-built for the cloud by securing data in a vendor-independent cloud to safeguard essential business applications, boost cyber resilience, and future-proof data protection.

For more information visit www.keepit.com or follow Keepit on Linkedin