Recovery

CoreBond Research Series — CBRS-010
August 2026
Todd Kirton
DOI: 10.5281/zenodo.22242558

1. Restore the Last Good State

A system fails in a way that matters to trust. A directory is corrupted. Firmware is altered. Credentials may have been exposed. An administrator reaches for the obvious recovery instruction: restore the last known-good state.

If yesterday's state was trusted and today's state is not, returning to yesterday appears to return the system to trust. Recovery plans preserve backups, recovery images, configuration, keys, and other material precisely because starting from nothing is expensive and often unnecessary.

Real recovery procedures complicate that story almost immediately. Microsoft's Active Directory forest-recovery guidance does not merely instruct administrators to restore a backup. It requires selection of a trusted backup, isolation of the recovered environment, and recovery actions intended to prevent restored domain controllers from simply resuming potentially compromised relationships (Microsoft, 2025a, 2025b, 2025d). NIST recovery guidance likewise treats restoration as a controlled process that includes validation of recovery assets and recovered data (Bartock et al., 2016; Nelson et al., 2025).

The problem is not restoration. It is what restoration is being asked to prove. A previous state can supply data, configuration, identity records, or executable material. It does not automatically establish that every trust-bearing condition represented by that state remains legitimate now.

Not every recovery event requires this full trust analysis. A storage failure or accidental deletion may leave the relevant trust assumptions intact. The preserve, break, and re-establish questions become necessary to the extent that the failure invalidated—or plausibly invalidated—the basis on which the system was trusted.

2. The Backup Is Part of the Question

A backup is often described as known-good as though the label were stored with the media. It is not. The claim depends on why the backup is believed to precede corruption or compromise, whether its integrity remains intact, and whether the authority and environment that produced it can still be relied upon.

NIST SP 800-61 Rev. 3 calls for verifying the integrity of backups and other restoration assets before using them in recovery (Nelson et al., 2025). NIST SP 800-184 similarly treats recovery as more than copying old data back into place; restoration and integrity validation belong to the same recovery problem (Bartock et al., 2016).

Time does not settle the issue. Microsoft's forest-recovery guidance addresses situations in which the time of failure is unknown and directs administrators to investigate further to identify a backup containing the last safe state of the forest (Microsoft, 2025a). A backup can predate the visible incident and still be too late.

The recovery material is therefore an input to a current decision. Age, provenance, integrity, exposure, isolation, recovery authority, and policy can all affect whether prior state is admissible for a particular recovery purpose. Evidence supports that judgment; evidence does not make the policy decision by itself.

That judgment may never reach certainty. Incident response can support a last safe point without proving an exact compromise boundary. Recovery still has to proceed. What it cannot legitimately do is turn an unknown boundary into a known one merely because a convenient backup exists.

3. Restored Is Not Trusted Again

The distinction becomes clearer when recovery procedures restore operational state while deliberately changing the mechanisms that previously carried trust.

In Active Directory forest recovery, historical directory state can be restored from backup while krbtgt, computer-account, and, after a security breach, trust passwords are reset so restored controllers do not automatically resume prior trust relationships (Microsoft, 2025b, 2025c, 2025d). The directory has been restored, but some of its prior trust continuity is intentionally interrupted.

Microsoft Entra recovery guidance shows the same separation at the identity layer. Authentication-method data can be recoverable, yet methods potentially involved in malicious activity should not be reused merely because Backup and Recovery can restore them. Suspect methods are removed and trusted authentication is established again (Microsoft, 2026a).

These procedures are not new. Mature recovery practice already distinguishes restoration, validation, credential reset, re-registration, and rekeying. The narrower point here is that those mechanisms expose a common reasoning boundary: restoration reconstructs operational or historical state; trust re-establishment determines whether the recovered actor, component, credential, or relationship may participate in present trust decisions.

> **A system can therefore be restored accurately and still lack sufficient grounds to trust what it does next.**

4. What May Be Preserved?

That separation does not imply that recovery should distrust everything produced before the failure. The evidence rejects that opposite absolute as well.

Active Directory recovery depends on historical directory state. NIST platform-firmware resiliency supports recovery to authorized firmware or data through protected recovery mechanisms (Regenscheid, 2018). Remote attestation architectures compare current Evidence against Reference Values that can predate the current appraisal (Birkholz et al., 2023).

Old state can remain useful for particular purposes. A validated backup may preserve historical objects. An authorized recovery image may remain an acceptable source of firmware. A reference value may remain an appropriate input for evaluating current Evidence. Identity records and history may survive even when the credentials used to authenticate that identity do not.

The operative question is not whether the state is old. It is whether a current recovery judgment—supported by evidence and bounded by recovery authority and policy—still permits reliance on it for the purpose at hand.

Strong isolation can make that judgment easier. NIST SP 800-193 describes Roots of Trust for Update and Recovery intended to protect the mechanisms that authenticate and restore firmware (Regenscheid, 2018). When those roots remain adequately protected, continuity can be legitimate.

But preservation is conditional. Calling something a recovery root, trusted backup, or reference value does not place it outside compromise by definition. Recovery preserves what remains justified, not whatever happened to survive.

5. What Must Be Broken?

Some of the most important recovery actions prevent prior trust continuity from carrying forward automatically. Depending on the domain, that may mean revocation, reset, suspension, isolation, replacement, rotation, or additional validation.

A cryptographic key is the clearest example. NIST key-management guidance treats compromise of high-level or root keying material as an event capable of invalidating the trust that depends on it, even though the key bytes themselves may still exist (Barker, 2020). Physical survival and authoritative survival are different properties.

NIST firmware-resiliency guidance recognizes the same problem for firmware-signing keys. A recovery architecture may strongly protect its recovery path, but it still needs a way to respond when the keying material used to authorize firmware is compromised (Regenscheid, 2018). The existence of a root does not prove the continuing authority of the root.

Identity recovery gives the distinction a less cryptographic form. An account can remain the same account while passwords, authentication methods, sessions, or other proof mechanisms are replaced or revalidated. Microsoft recovery guidance preserves the object or service relationship while changing material used to establish control over it (Microsoft, 2026a).

Recovery therefore does not have to choose between preserving identity and destroying continuity. It can preserve history, names, objects, permissions, and other legitimate state while preventing suspect trust-bearing relationships from silently resuming their former authority.

Sometimes that interruption is temporary; sometimes it is permanent. The important point is that survival alone does not entitle prior trust continuity to resume.

6. What Must Be Established Again?

Once suspect continuity has been interrupted, restoration still leaves a present-tense question: what is true of the recovered system now?

A backup can describe what configuration existed at an earlier point. It cannot, by itself, establish that the current recovered platform is executing that configuration without unauthorized modification. A restored account can preserve identity history without proving who currently controls it. A recovered credential record can describe prior enrollment without proving that the corresponding authenticator remains acceptable.

When the failure could have changed a property being asserted about the recovered present, that assertion needs an appropriate current basis. NIST recovery guidance pairs restoration with integrity validation (Bartock et al., 2016; Nelson et al., 2025). Microsoft identity recovery uses replacement or re-registration when prior authentication methods cannot be relied upon (Microsoft, 2026a).

Remote attestation makes the composition visible. In the RATS architecture, a Verifier appraises Evidence using Reference Values and appraisal policy that may predate the current appraisal (Birkholz et al., 2023). Recovery therefore does not require every trustworthy input to be newly generated.

Nor does current evidence become trustworthy merely because it is current. Measurement paths, reference values, appraisal policy, verifiers, and other dependencies can themselves fail or be compromised. The narrower requirement is that present-state claims have an appropriate present basis rather than being inferred solely from pre-failure state when the failure could have changed them.

7. Recovery Is Allowed to Go Backward

Paper 9 treated rollback as a reason memory can be security-relevant. A system that has already accepted a newer state may need to remember that fact so an older state cannot silently return. Recovery appears to violate that principle on purpose.

It sometimes does. Recovery after corruption or destructive attack may intentionally restore an older validated image, configuration, or dataset (Bartock et al., 2016; Regenscheid, 2018). The older state has not become newer. The architecture has entered an authorized recovery procedure that is permitted to reconstruct from it.

This makes rollback context-dependent. During ordinary operation, accepting an older state can indicate attack, replay, or loss of security progress. During authorized recovery, returning to an earlier validated state may be the safest available action.

The exception cannot be merely 'because we are recovering.' Recovery authority, provenance, validation, and policy distinguish deliberate reconstruction from attacker-induced rollback. Because the recovery path is allowed to cross rules that ordinary operation cannot, the recovery mechanism and the authority controlling it become part of the security-critical attack surface.

Trust-bearing material created or exposed after the selected restore point may also need to be invalidated rather than resurrected with the restored snapshot. Recovery does not erase monotonic security rules; it creates a controlled exception whose legitimacy must itself be justified.

8. When Nothing Deserves to Survive

Selective reconstruction assumes that at least some recovery basis remains defensible. Sometimes that assumption fails.

If defenders cannot justify a backup, recovery root, credential, signing authority, or other continuity after a serious compromise, the preserve set can shrink dramatically. A clean rebuild, new trust anchor, re-enrollment, key replacement, or complete reprovisioning may be safer than trying to salvage continuity whose basis is unknown.

A clean rebuild is not state-free. It still inherits installation media, source repositories, provisioning authority, trust anchors, identity records, policy, operators, and recovery procedures. Starting over can reduce suspect continuity without eliminating inherited trust.

Nor is clean rebuilding the universal ideal. It is expensive, disruptive, and can destroy useful continuity unnecessarily when strong recovery anchors and validated state remain available. Platform-resiliency architectures exist precisely to avoid rebuilding everything after every recoverable fault.

The point is conditional. Recovery should not preserve continuity merely because preserving it is convenient. If no relevant basis can be justified, there may be nothing trust-bearing worth carrying forward.

9. Controlled Reconstruction

Recovery standards and mature operational guidance already validate backups, isolate restored systems, reset credentials, replace keys, restore authorized firmware, re-register authentication methods, and validate recovered operation. The mechanisms are not the contribution.

The synthesis is the separation of the decisions underneath them. What prior state may be preserved because continued reliance remains justified? What trust-bearing continuity must be interrupted because its authority no longer survives? What properties of the recovered present need an appropriate current basis because the failure could have changed them?

That is controlled reconstruction: not a new protocol or formal taxonomy, but a cross-domain way to describe the reasoning already visible in recovery architectures.

It avoids two opposite mistakes. Blind restoration treats historical validity as present authority. Total reinvention discards state and continuity that may remain legitimate. A trustworthy present may contain a great deal of the past; the recovery problem is deciding which parts still deserve to participate in present trust decisions.

10. What If We Cannot Know?

The hardest recovery cases do not end with a clean answer.

Defenders may not know when compromise began. They may not be able to prove that a recovery key was never exposed, that a backup predates every malicious change, or that every reference value and recovery authority remained outside the failure. Microsoft's forest-recovery guidance explicitly contemplates uncertainty about the last safe state rather than assuming it away (Microsoft, 2025a).

Recovery procedures can validate surviving assets, isolate restored systems, replace suspect roots, and obtain evidence about the recovered environment. Those actions can reduce risk without recursively proving every root, operator, policy, backup, and assumption forever.

A recovered system may contain yesterday's data, yesterday's identity, yesterday's configuration, and yesterday's reference values. Trust in the recovered present still depends on a current judgment about which of those things deserve to carry forward and what must be established again.

Sometimes that judgment is strongly supported. Sometimes material uncertainty remains. Paper 11 asks what trust should mean when the system cannot eliminate it.

References

Barker, E. (2020). Recommendation for key management: Part 1 - General (NIST Special Publication 800-57 Part 1 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-57pt1r5

Bartock, M., Cichonski, J., Souppaya, M., Smith, M., Witte, G., & Scarfone, K. (2016). Guide for cybersecurity event recovery (NIST Special Publication 800-184). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-184

Birkholz, H., Thaler, D., Richardson, M., Smith, N., & Pan, W. (2023). Remote attestation procedures architecture (RFC 9334). Internet Engineering Task Force. https://doi.org/10.17487/RFC9334

Microsoft. (2025a). Active Directory Forest Recovery - Determine how to recover the forest. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-determine-how-to-recover

Microsoft. (2025b). Active Directory Forest Recovery - Perform the initial recovery. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-perform-initial-recovery

Microsoft. (2025c). AD Forest Recovery - Reset the krbtgt password. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-reset-the-krbtgt-password

Microsoft. (2025d). Active Directory Forest Recovery - Reset a trust password on one side of the trust. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-reset-trust

Microsoft. (2026a). Recover user authentication methods using Microsoft Entra Backup and Recovery. Microsoft Learn. https://learn.microsoft.com/en-us/entra/backup/recover-user-secrets

Nelson, A., Rekhi, S., Souppaya, M., & Scarfone, K. (2025). Incident response recommendations and considerations for cybersecurity risk management: A CSF 2.0 community profile (NIST Special Publication 800-61 Rev. 3). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-61r3

Regenscheid, A. (2018). Platform firmware resiliency guidelines (NIST Special Publication 800-193). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-193