Recovery and Fallback Playbook: Phishing-Resistant Account Recovery That Doesn't Become the New Attack Surface

Trust & Risk / Last updated 2026-06-11 / https://www.scrambleid.com/learn/recovery-and-fallback-playbook

In one sentence: A phishing-resistant deployment is only as strong as its recovery path; this playbook specifies how to design warm-path recovery from an enrolled device, cold-path recovery via identity proofing, and assisted recovery for users without the app, so that recovery does not silently become the new attack surface.

TL;DR (canonical)

- **Recovery is the failure mode that defeats most passwordless deployments.** When the primary authentication path fails (lost device, compromised device, no app installed), the recovery path becomes the actual security boundary. If recovery falls back to KBA, OTP, or a helpdesk script, the attacker has a clear target.
- **Three recovery paths, in order of strength:** warm-path (step-up from another enrolled device, the role XFactor (in development) is designed to fill), cold-path (identity proofing for a new device), and assisted recovery (for users without the app, requiring stronger out-of-band identity proofing and dual control).
- **Cold-path recovery should be slow on purpose.** Recovery latency that's measured in hours-to-days for high-assurance contexts is a feature, not a bug. Speed at this step is what attackers exploit.
- **Never reintroduce KBA as a "for accessibility" fallback.** If you do, the entire phishing-resistance posture collapses.
- **Audit recovery harder than primary authentication.** Recovery events are higher-risk than logins; logging, dual control, and risk correlation (Overwatch's role, in development) should be tighter, not looser.

Why recovery deserves its own playbook

Most enterprise passwordless deployments succeed at the primary path. WebAuthn works. QR(DID) works. Caller Auth works. The user with their enrolled device, on a normal day, has a strong, fast,

phishing-resistant experience.

Then something breaks. The phone gets dropped in a pool. The laptop gets stolen at the airport. The user upgrades to a new device and forgets to migrate. The contact center receives a call from a customer who insists they have always logged in this way and they cannot find their app.

What happens next is the actual security boundary. Every phishing-resistant control on the primary path is wasted if the recovery path is socially engineered. CISA's Scattered Spider advisories document this exact pattern: attackers do not break authentication; they call the helpdesk and get authentication reset.

This playbook is the answer to "what does good recovery look like?" The recovery design described here is not faster than KBA. It is harder for attackers, equally usable for legitimate users in the common cases, and explicitly slow in the rare cases where slowness is the security control.

The three recovery paths

A production deployment should support three distinct recovery paths, ordered from strongest to weakest. Each path has a different latency target, a different audit requirement, and a different threat model.

Warm-path recovery (existing enrolled device available)

The user has lost or replaced one device, but still has another enrolled device. This is the strongest recovery path because the existing device performs an XFactor step-up that authorizes the new enrollment.

Flow:

1. User opens the recovery flow on the new device (or a web UI).
2. The system prompts the user to confirm from an existing enrolled device via XFactor.
3. The user completes the XFactor chain on the existing device (typically WebAuthn UV or a signed QR(DID) confirmation).
4. The new device generates a key pair in hardware-backed storage.
5. The system enrolls the new public key against the user's `suid` with a fresh `zid`.
6. Both devices are valid during a short overlap window (recommended: 24-72 hours) before the lost device's `zid` is automatically retired.

Latency target: sub-60 seconds end-to-end.

Audit requirement: the XFactor result artifact is signed and logged. The new ZID's enrollment event references the authorizing ZID so the audit trail can be reconstructed.

Threat model: an attacker would need to compromise an existing enrolled device, which already has hardware-bound key storage and biometric/PIN unlock. This is the same threat surface as primary

authentication, not a weaker fallback.

Cold-path recovery (no enrolled device available)

The user has lost all enrolled devices, or has never enrolled a device. This path requires fresh identity proofing because the system has no cryptographic anchor to fall back on.

Flow:

1. User initiates recovery through a designated channel (web form, phone call to a recovery line, or in-person request).
2. The system requires identity proofing appropriate to the assurance level the deployment demands. For an AAL3-aligned design, proofing requires real-time video verification with a government-issued ID document and a liveness check; for AAL2, document verification plus a few additional signals may suffice. The exact proofing requirements live in the deployment's security plan, not in the product.
3. After successful proofing, the new device enrollment proceeds as a fresh enrollment, not as a delta from a prior identity. The user's prior `suid` is preserved (so historical audit trails remain coherent), but every enrolled `zid` against that `suid` is treated as untrusted until re-attested.
4. For high-risk users (admins, finance approvers, executives), the recovery itself requires dual approval before the new ZID becomes active (Lockstep, in development, is designed to enforce this). This prevents an attacker who passes proofing from immediately self-approving high-risk actions.
5. Audit logging records the proofing artifact, the operator who reviewed it (if applicable), the dual approver, and the new ZID enrollment event.

Latency target: hours to a business day. This is intentional. A 30-second cold-path recovery is an attack surface; a 30-minute supervised cold-path recovery is a feature.

Audit requirement: every cold-path recovery is a structured event that includes proofing evidence references, approver identity, and the resulting ZID. Recovery events are sampled by audit and reviewed monthly.

Threat model: the proofing step is the security boundary. Identity proofing must be appropriate to the assurance level, must include liveness detection, and must reject document-only or knowledge-based recovery. CISA's guidance and NIST SP 800-63A define what proofing looks like at each IAL.

Assisted recovery (user without the app)

A subset of users may not have the ScrambleID app or any enrolled device. This is more common in consumer contexts (B2C) than in workforce deployments, and is the path most likely to be socially engineered. Design assisted recovery on the assumption that an attacker is on the phone.

Flow:

1. The customer contacts the support channel.

2. The agent does not authenticate the customer in the call. KBA is explicitly off the table.
3. Instead, the agent initiates a structured recovery flow that puts the customer in front of the proofing requirement on a different channel (typically a one-time secure link sent to a previously verified email or postal address, with mailbox-control proof and ID document upload).
4. The customer completes the proofing flow asynchronously (recommended TTL: 24-72 hours).
5. After proofing succeeds, the standard cold-path recovery flow above applies, including dual approval for high-risk users.
6. The agent never makes the trust decision; the agent's role is to route the customer into the proofing flow.

Latency target: initial response in minutes (the agent confirms the issue and dispatches the recovery link); completion in 24-72 hours.

Audit requirement: every assisted recovery is logged with the agent identity, the recovery vector (email, mail, in-person), and the proofing outcome. Agents who handle assisted recovery should be specifically trained, audited, and rotated; unfamiliar customers should not see the same agent twice if practical.

Threat model: the agent is the most-attacked surface. Train agents that they are not gatekeepers, they are routers. Provide them with scripts for handling pressure ("I understand this is urgent; the recovery flow takes 24 hours and I cannot accelerate it on this call, here is the link"). Monitor wrong-recovery-rate, repeat-recovery-rate per customer, and per-agent recovery volumes for anomaly patterns.

Decision tree

A user reports loss of access. The agent or self-service flow walks the following decision tree:

```
1. Does the user have any other enrolled device they can access?
   YES → Warm-path recovery (XFactor step-up from existing device).
         Latency: sub-60s. Audit: standard.
   NO  → Continue to step 2.

2. Does the user have the ScrambleID app installed somewhere
   (even if it has no enrolled device for this account)?
   YES → Cold-path recovery: install-then-enroll.
         The user opens the app, enters the recovery flow, and
         completes identity proofing in-app via video and ID scan.
         Latency: minutes for proofing review (or hours if queued
         for human reviewer in high-assurance contexts).
   NO  → Continue to step 3.

3. Is this a high-risk user (admin, finance approver, executive,
   or any account flagged for elevated treatment)?
   YES → Assisted recovery + Lockstep dual approval.
         Initial routing in minutes; completion in 24-72 hours.
         Two distinct approvers required before ZID activation.
   NO  → Assisted recovery (single-approver flow if policy allows).
         Initial routing in minutes; completion in 24-72 hours.

4. Has the user failed identity proofing within the last 7 days?
   YES → Pause the request, escalate to fraud review, do not retry
         on the same channel for 24 hours, log the pattern.
   NO  → Proceed.

5. Does the proofing evidence pass review?
   YES → Enroll new ZID, mark old ZIDs as retired (not just
         revoked, retired with audit trail), notify the user
         on a known-good channel that recovery completed.
   NO  → Deny, log, escalate to fraud team for follow-up.
         Do not allow the user to retry on the same channel
         immediately; require a 24-hour cooldown.
```

This tree is opinionated. The opinions are intentional. Each branch encodes a security decision that matters in production.

Anti-patterns to avoid

These are the recovery design choices that look reasonable in a planning meeting and turn out to be the actual attack surface in production. Reviewers and auditors should treat each one as a red flag.

"We left KBA on for accessibility"

The most common failure mode. A team retires KBA from the primary authentication path but leaves it as the recovery option for users who can't complete the new flow. Attackers find this within weeks. KBA's failure modes (OSINT, breach data, social engineering pressure, agent judgment under pressure) are exactly as bad in recovery as they are in primary authentication. Possibly worse, because recovery often gates higher-risk actions.

The right response: KBA is gone. Period. Accessibility is solved with assisted recovery (real human, structured proofing), not with security questions.

"Helpdesk can override after verifying employee details"

A variant of KBA at the helpdesk level. Attackers research employee details, call the helpdesk, and pressure the agent to reset. The agent has been trained to be helpful. The override happens.

The right response: helpdesk agents do not authenticate; they route the user into the structured recovery flow. The trust decision lives in proofing artifacts, not in agent judgment. Agents who can override authentication should not exist in this design; if they must exist for true emergencies, every override is a Lockstep operation requiring two distinct supervisors and a recorded justification.

"We mail a recovery code to the address on file"

A reasonable-sounding fallback that fails when the address itself was compromised in the original breach (which is most large breaches). It also fails when the user has moved and the mail goes to the previous occupant of their old address.

The right response: mailed codes can be one factor in a multi-factor cold-path recovery, but they are never sufficient alone. The mailed code combined with a video-verified ID document and a liveness check is reasonable; the mailed code alone is not.

"We text a recovery link"

SMS to a phone number on file. Defeated by SIM swap, defeated by an attacker who has already taken over the phone, defeated by carrier social engineering. SMS recovery is functionally KBA with extra steps.

The right response: never use SMS as a sole recovery factor. SMS can be one signal among many; it cannot be the trust anchor.

"We let users self-recover with their security questions"

KBA renamed. Same failure modes.

"We have a 24-hour cooldown on recovery, but it can be waived for VIPs"

The cooldown is the security control. Waiving it for VIPs makes VIPs the attack target. VIPs are typically the most valuable accounts.

The right response: cooldowns apply uniformly. If the deployment cannot tolerate cooldowns for some users, those users should have multiple enrolled devices so warm-path recovery is always available.

"We allow recovery from any device, including a brand-new one with no prior context"

Without device-binding, the recovery flow has no context to evaluate. New IP address, new device fingerprint, new geography. None of those alone are damning, but combined with a recovery request they should trigger heightened scrutiny.

The right response: the risk layer evaluates recovery requests against device-context signals (the job Overwatch is designed to do once it ships); high-risk-context recovery requires elevated proofing or dual approval. The signals are inputs to a layered decision, not single-decision controls.

What recovery should log

Recovery events are higher-stakes than authentication events. Logging should be more detailed, retention should be longer, and review should be more frequent.

Per recovery event, log:

- **Recovery type:** warm-path / cold-path / assisted.
- **User identifiers:** `suid`, prior `zid` (if applicable), new `zid`.
- **Recovery channel:** web, phone, in-person, support form.
- **Operator:** if assisted, the agent's identity. If self-service, none.
- **Proofing artifacts:** references to proofing evidence (not the evidence itself; that lives in a separate, more tightly access-controlled store with PII handling).
- **Approver identity:** if dual approval (Lockstep) is required, both approvers are recorded with the LSID.
- **Decision:** approved / denied / pending.
- **Decision reason:** structured field, not free text. Examples: `proofing_video_failed`, `lockstep_quorum_not_reached`, `cooldown_active`, `fraud_team_review_pending`.
- **Outcome:** new ZID active, prior ZIDs retired, notification sent.
- **Correlation IDs:** session id, Overwatch alert id (if applicable), case id (if escalated).

Recovery logs should be retained at least as long as authentication logs, typically a minimum of 7 years for financial services, 6 years for HIPAA-aligned health, and 3 years for general enterprise context. Retention should be set by the regulatory regime, not by infrastructure cost optimization.

SLAs and SLOs

Recommended targets. Tune per use case.

Path	Action	Target
Warm-path recovery	End-to-end completion	< 60 seconds p95
Cold-path recovery (self-service in-app)	Proofing review	< 5 minutes p95 (automated review with human escalation queue)
Cold-path recovery (high-risk users with Lockstep)	Approver round-trip	< 30 minutes p95, < 4 hours p99
Assisted recovery	Initial agent response	< 5 minutes p95
Assisted recovery	End-to-end completion	< 24 hours p95, < 72 hours p99
Recovery event logging	Decision logged to durable store	< 5 seconds from decision
Cooldown enforcement	Prior failed recovery blocks retry	24 hours minimum, 72 hours for high-risk users

These targets are starting points. Adjust based on user research, fraud metrics, and audit feedback. The right test for an SLA is whether legitimate users complete recovery without abandonment **and** attackers cannot complete recovery faster than your fraud team can detect them.

Recovery in regulated industries

Different regulatory regimes set different recovery expectations. The playbook above is a baseline; layer on industry-specific requirements:

- **Financial services (PCI DSS, SOX):** dual approval for any recovery affecting accounts that can move money. Audit retention 7 years. Identity proofing must include document verification.
- **Healthcare (HIPAA):** recovery cannot expose PHI in agent-visible context (no patient data on the agent's screen during routing). Audit retention 6 years.
- **Federal-aligned (NIST 800-63):** identity proofing must align with the IAL the deployment claims. AAL3 deployments require IAL2 or IAL3 proofing for cold-path recovery; AAL2 deployments can use IAL1 with additional signals.
- **EU (GDPR):** recovery operations are processing of personal data; document the lawful basis (contract performance or legitimate interest, typically), include retention windows in the privacy

notice, and support right-to-erasure flows that account for prior recovery records.

How recovery integrates with the rest of the corpus

Recovery is not a standalone product feature. It is a system-level design that touches several ScrambleID primitives:

- **Device key lifecycle** specifies how new ZIDs are enrolled and how prior ZIDs are retired during recovery.
- **Session binding cryptography** specifies how the recovery session is bound to prevent cross-session replay.
- **XFactor step-up** (in development) is the primitive designed to drive warm-path recovery (step-up from existing enrolled device).
- **Lockstep dual control** (in development) is the primitive designed to drive high-risk cold-path recovery (two-distinct-approver requirement).
- **Overwatch risk engine** (in development) is designed to evaluate recovery requests against cross-channel signals and elevate or block based on policy.
- **Compliance mapping (NIST/CISA)** maps the recovery design to NIST 800-63-4 identity proofing requirements and NIST 800-53 controls (IA-5 Authenticator Management, AC-2 Account Management).

Recovery is the test of whether all of these primitives are wired together coherently. A deployment that gets primary authentication right but recovery wrong is a deployment that has not finished.

Key Takeaway

Recovery and fallback design is the failure mode that defeats most passwordless deployments. The right design uses three paths: warm-path recovery via XFactor step-up from an existing enrolled device (designed to complete in moments, not hours), cold-path recovery via identity proofing for users without an enrolled device (hours-to-days target with intentional friction), and assisted recovery for users without the app (agent routes into a structured proofing flow rather than authenticating in the call). KBA, SMS-only recovery, helpdesk override authority, and mailed-code-only recovery are explicit anti-patterns. High-risk users (admins, finance approvers, executives) require dual approval (Lockstep) on cold-path recovery. Recovery events are higher-stakes than authentication events; logging, retention, and review must be more rigorous, not less.

FAQ

How long should cold-path recovery take?

Hours to a business day for most use cases. Speed at this step is the attack surface. A 30-minute supervised cold-path recovery is the design intent. For genuinely time-sensitive cases (an executive locked out before a board meeting), warm-path recovery from a backup device should be available; if it is not, the right answer is to enroll the executive on multiple devices upfront, not to weaken cold-path recovery.

Can helpdesk agents authenticate users in a recovery call?

No. The agent's role is to route the user into the structured recovery flow, not to make the trust decision. Agents who can override authentication on the call become the attack target. If exceptions for true emergencies exist, every exception should be a Lockstep operation with two distinct supervisor approvals and a recorded justification.

Is mailed recovery code acceptable?

As one factor in a multi-factor cold-path recovery, yes. As the sole recovery factor, no. Mailed codes fail when the address was compromised in the original breach or when the user has moved. Pair with video-verified ID and liveness check.

How do you handle users who refuse identity proofing?

Deny the recovery request and offer a path to escalation through a known-good channel (a corporate ID at an in-person verification desk, for example). The user has the right to refuse proofing; the deployment has the right to refuse recovery without it. Refusal patterns are also a fraud signal; track them.

What's the right retention window for recovery audit logs?

At minimum, 3 years for general enterprise; 6 years for HIPAA-aligned health; 7 years for financial services. Set retention by regulatory requirement, not by infrastructure cost. Recovery logs are higher-evidentiary-value than authentication logs because they document trust-decision moments.

Can cooldowns be waived for VIPs?

No. The cooldown is the security control. Waiving it for VIPs makes VIPs the attack target, and VIPs are typically the most valuable accounts. If the deployment cannot tolerate cooldowns for some users, those users should be enrolled on multiple devices upfront so warm-path recovery is always available.

How do you measure whether recovery is working?

Track: legitimate-user completion rate by path, abandonment rate, fraud-attempt rate, time-to-detect on fraud attempts, agent override rate (should be zero in steady state), and Overwatch escalation rate on recovery events. The right combination is high completion for legitimate users, low abandonment, near-zero successful fraud attempts, and a healthy detection rate on attempted fraud (high detection means the system is seeing what it should).

References (public)

- NIST SP 800-63A-4 Identity Proofing: https://csrc.nist.gov/pubs/sp/800/63a/4/final
- NIST SP 800-63B-4 Authenticator and Lifecycle Management: https://csrc.nist.gov/pubs/sp/800/63b/4/final
- NIST SP 800-53 Rev. 5 (IA-5 Authenticator Management, AC-2 Account Management): https://csrc.nist.gov/pubs/sp/800/53/r5/upd1
- CISA Cybersecurity Advisory AA23-320A (Scattered Spider, helpdesk social engineering patterns): https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a
- CISA Phishing-Resistant MFA Fact Sheet: https://www.cisa.gov/sites/default/files/publications/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf

---

Related reading

- Architecture: Device Key Lifecycle
- Architecture: Session Binding Cryptography
- XFactor: Step-Up Authentication
- Lockstep: Dual Control
- Overwatch: Risk Engine
- Compliance Mapping: NIST and CISA
- KBA Is Dead: Contact Center Playbook
- ScrambleID Glossary