HIPAA AUDITORS

Person or Entity Authentication Implementation Checklist

Implementation and evidence checklist for 45 CFR 164.312(d) — Person or Entity Authentication

Issued	9 August 2026
Prepared by	HIPAA Auditors — certified HIPAA lead auditors
Contact	info@hipaaauditors.com · +91-95851-60363
Address	100 Congress Ave, Suite 2000, Austin, TX 78701

How to use this document. This is a working template, not a finished compliance artefact. Replace every bracketed placeholder with your organisation’s own detail, delete the guidance notes, and have the result reviewed and formally adopted by your Privacy or Security Officer before you rely on it.

Regulatory basis. References are to the HIPAA Administrative Simplification regulations at 45 CFR Parts 160, 162 and 164, and to NIST SP 800-66 Rev. 2, *Implementing the HIPAA Security Rule*.

The requirement

Implement procedures to verify that a person or entity seeking access to ePHI is the one claimed.

45 CFR 164.312(d) · Technical Safeguards

Risk level	Difficulty	Typical cost	Timeframe
Critical	Moderate	Medium	2-4 months

Implementation steps

	Implement procedures to verify that a person or entity seeking access to ePHI is the one claimed. The standard at 164.312(d) has no implementation specifications, which means it is required in full: you must authenticate, and the method must be reasonable and appropriate to the risk you identified
	Authentication factors:
	Something known — password, passphrase or PIN
	Something held — hardware token, smart card, authenticator app or certificate
	Something inherent — fingerprint, facial or other biometric
	Implementation approach:
	- Set password requirements from current NIST SP 800-63B thinking: favour length over composition, screen against known-breached password lists, and stop forcing arbitrary periodic rotation, which drives predictable, weaker choices
	- Deploy multi-factor authentication for remote access, administrative accounts, and any internet-facing system holding ePHI. The rule does not name MFA, but a risk analysis will almost always conclude it is warranted, and its absence features repeatedly in enforcement
	- Authenticate systems and services, not only people: API keys, certificates and service accounts all need managed identity and rotation
	- Lock accounts after repeated failures, and alert on password spraying and impossible-travel patterns
	- Re-authenticate before high-risk actions such as bulk export or privilege change
	- Never transmit or store credentials in reversible form; hash with a modern algorithm and per-credential salt
	- Verify identity before issuing or resetting credentials — help-desk reset is a well-worn attack path, and social engineering of it is common
	- Manage the full credential lifecycle: issue, rotate, revoke on termination, and review dormant accounts

Evidence to produce and retain

Tick only where the artefact exists and you can say where it lives.

- [ ] **Authentication policy covering people, systems and services**
  Location: _____ Owner: _____
- [ ] **Password standard, including length, screening against breached lists and rotation position with rationale**
  Location: _____ Owner: _____
- [ ] **Multi-factor authentication scope, and the risk rationale for any system excluded**
  Location: _____ Owner: _____
- [ ] **Identity verification procedure for credential issue and reset, including help-desk steps**
  Location: _____ Owner: _____
- [ ] **Account lockout and failed-attempt alerting configuration**
  Location: _____ Owner: _____
- [ ] **Credential storage standard, naming the hashing algorithm in use**
  Location: _____ Owner: _____
- [ ] **Service account and API credential inventory, with owners and rotation schedule**
  Location: _____ Owner: _____
- [ ] **Credential lifecycle records: issue, rotation, revocation and dormant account review**
  Location: _____ Owner: _____
- [ ] **Biometric enrolment and handling procedures, where biometrics are used**
  Location: _____ Owner: _____
- [ ] **Authentication-related incident records**
  Location: _____ Owner: _____

How this gets tested

These are the procedures an auditor is likely to run. Self-test first.

- [ ] **Confirm authentication is enforced on every path to ePHI, including APIs, database tools and mobile apps**
  Result: _____ Tested by / date: _____
- [ ] **Attempt authentication with a known-breached password and confirm it is rejected**
  Result: _____ Tested by / date: _____
- [ ] **Verify multi-factor authentication is enforced for remote and administrative access, and attempt to bypass it**
  Result: _____ Tested by / date: _____
- [ ]

	Result: _____ Tested by / date: _____
	Inspect the lockout threshold and confirm it is within the required limit and per-credential salt Result: _____ Tested by / date: _____
	Attempt a credential reset without satisfying identity verification and confirm refusal Result: _____ Tested by / date: _____
	Review the service account inventory and confirm rotation is happening as scheduled Result: _____ Tested by / date: _____
	Confirm credentials are revoked promptly on termination, checking actual timestamps Result: _____ Tested by / date: _____
	Identify dormant accounts and confirm they are disabled Result: _____ Tested by / date: _____
	Confirm no default or vendor-supplied credentials remain active Result: _____ Tested by / date: _____

Common failures to check yourself against

- No multi-factor authentication on remote or administrative access to systems holding ePHI
- Shared credentials, which defeat both authentication and attribution
- Default or vendor-supplied passwords left active on devices and applications
- Credentials stored reversibly, or hashed with an obsolete algorithm
- Help-desk resets performed without verifying identity, enabling straightforward social engineering
- Static API keys and service account passwords never rotated, sometimes committed to code repositories
- Dormant accounts left enabled long after the user stopped needing access
- Password policies that force frequent rotation and drive predictable variants
- Authentication enforced in the application but bypassable through direct database access

Beyond the minimum

- Require multi-factor authentication for all remote access, administrative accounts and internet-facing systems
- Follow NIST SP 800-63B: long passphrases, breached-password screening, no arbitrary forced rotation
- Prefer phishing-resistant factors such as FIDO2 security keys or platform authenticators over SMS codes
- Use single sign-on so credential policy is enforced centrally and revocation is immediate everywhere
- Give service accounts managed identities or short-lived certificates instead of static shared secrets
- Verify identity out of band before any credential reset, and script the help-desk procedure
- Alert on password spraying, impossible travel and MFA fatigue patterns
- Re-authenticate before bulk export or privilege escalation
- Review dormant accounts monthly and disable automatically

NIST mapping: NIST SP 800-66 Rev. 2: Section 3.3.4

Sign-off

Completed by	Title	Date
Reviewed by	**Title**	**Next review due**