IA.L2-3.5.10

Identification & Authentication

Cryptographically-Protected Passwords

"Store and transmit only cryptographically-protected passwords."

Objectives

[a]

Passwords are cryptographically protected in storage.

[b]

Passwords are cryptographically protected in transit.

Key Discussion Points

Hashing Required:
Passwords must be salted one-way hashes — bcrypt, PBKDF2, or Argon2. AES encryption of the plaintext does not satisfy this requirement.

In Storage AND Transit:
Both objectives must be met — hashed at rest AND encrypted in transit (TLS). A system that hashes but transmits in cleartext fails [b].

Weak Hashes Fail Too:
MD5 and SHA-1 are considered cryptographically weak — use strong modern hashing algorithms with salting.

Legacy Apps Are a Risk:
Many older applications store passwords in plaintext or reversible encryption — a legacy app in the CUI environment is a direct finding.

Assessment Methods

EXAMINE

Identification and authentication policy; system security plan; procedures addressing authenticator management; procedures addressing user identification and authentication; system design documentation; list of system authenticator types; system configuration settings; system audit logs.

INTERVIEW

Personnel with authenticator management responsibilities; personnel with information security responsibilities; system or network administrators.

TEST

Mechanisms supporting or implementing authenticator management capability.

IA.L2-3.5.10

IDENTIFICATION & AUTHENTICATION — Cryptographically-Protected Passwords

Plain English

What this control is really saying:

If passwords are stored in plaintext or in reversible form, a database breach instantly exposes every user's credential. If they're transmitted without encryption, anyone on the network can capture them. This control requires one-way hashing for storage and encrypted transmission — so even if the data is stolen, passwords cannot be recovered.

How it is used:

- Active Directory stores all domain account passwords using Kerberos with NTLM hash disabled — passwords are never stored in plaintext.
- All web application passwords are stored using bcrypt with a unique salt per password — database exports cannot be used to recover plaintext passwords.
- All authentication traffic occurs over TLS 1.2 or higher — plaintext authentication protocols are blocked at the firewall and host level.
- The SSP documents the hashing algorithm, salting approach, and transmission encryption protocol for all systems handling passwords.

Real World Example

The Scenario

Acme Defense uses a project management web application that was built in-house five years ago. The developer stored passwords using MD5 hashing with no salt. The application runs on HTTP without TLS. Nobody has reviewed its authentication architecture since it was deployed.

What the assessor finds

The application's password hashes are MD5 without salt — a freely available rainbow table cracks 80% of the passwords within seconds of database access. Credentials are transmitted in cleartext over HTTP — captured immediately by the assessor using Wireshark on the local network.

SPRS Score Impact

3.5.10 carries a point value of 5. Plaintext or weakly hashed password storage combined with unencrypted transmission represents complete credential exposure — a single breach gives an attacker every password in the environment.

What Good Looks Like

Passwords stored using strong one-way salted hashing (bcrypt, PBKDF2, or Argon2), all authentication traffic over TLS 1.2+, no plaintext or reversible storage anywhere in the environment, hashing and transit protection documented in SSP.

Common Gaps

What assessors actually find in the field:

✗ Plaintext password storage

An application stores passwords in plaintext in its database — a single SQL injection exposes every user's password.

✗ Reversible encryption used

Passwords are encrypted with AES — technically 'protected' but decryptable with the key. This does not satisfy one-way hashing.

✗ Weak hashing algorithm

Passwords are hashed with MD5 without salting — rainbow table attacks can crack the entire database in minutes.

✗ Authentication over HTTP

A web application login form transmits credentials over plain HTTP — credentials are captured in cleartext on the network.

✗ Legacy app not evaluated

An older CNC configuration app handles login but nobody has verified how it stores or transmits passwords.