POLICY

**IDENTITY PROOFING -** establishes that a person is who they say they are, and the strength of this proof is based on one or more pieces of identity evidence. Identity proofing is defined in the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-63-3, Digital Identity Guidelines, and by Identity Assurance Levels (IALs), which ranges from 1 to 3.

This guideline focuses on the enrollment and verification of an identity for use in digital authentication. Central to this is a process known as identity proofing in which an applicant provides evidence to a credential service provider (CSP) reliably identifying themselves, thereby allowing the CSP to assert that identification at a useful identity assurance level. This document defines technical requirements for each of three identity assurance levels. This publication supersedes corresponding sections of NIST Special Publication (SP) 800-63-2.

According to the NIST Special Publication 800-63B, password length has been found to be a primary factor in characterizing password strength. NIST requires that **all user-created passwords be at least 8 characters in length and all machine-generated passwords are at least 6 characters in length**.

Password policy:

**1. Enable Show Password**

It’s unlikely that the person behind you is going to record your password data, so there’s little reason to hide your password as you type. You’re more likely to make mistakes in typing if you can’t see the characters, and mistakenly think you’ve forgotten your password. This error leads to potential data exposure every time you need to reset your password.

**2. Use a Password Manager**

NIST suggests that companies use a password manager to help their employees and stakeholders encrypt and generate strong passwords. Even if you’re securing your own servers, you will want to help reduce human error by giving your users access to a password manager, which will automatically generate long, strong passwords and passphrases for them.

**3. Store Securely**

NIST requires that organizations remove the user-generated password from their server as soon as it is created, either using a [zero-knowledge password protocol](https://csrc.nist.gov/glossary/term/zero_knowledge_password_protocol) or [zeroization](https://csrc.nist.gov/glossary/term/zeroization). They also suggest “hashing” and “salting” stored passwords. NIST defines a hash as “a function that maps a bit string of arbitrary length to a fixed-length bit string.” Using hashes to store password data will ensure that you never expose a database of passwords to a hacker; instead, they’d get a list of hashes that would take much longer to crack and give you more time to recuperate. Salting adds unique markers to each password, so that two people with the same password (say, a default password) are assigned two distinct hashes.

**4. Lock After Multiple Attempts**

NIST suggests locking a user out of password-protected programs if they use an incorrect password multiple times; per [Section 5.22 of Special Publication NIST 800-63b](https://pages.nist.gov/800-63-3/sp800-63b.html#throttle), which provides guidelines for “rate-limiting” on authentication attempts, the verifier (that’s you) should allow no more than 100 attempts to input a password. However, most good programs limit far before that threshold, and also use strategies like making a user wait a period of time before attempting to sign on again.

**5. Employ Two-Factor Authentication**

Two-factor or multi-factor authentication requires that someone entering their password authenticates their login from another device or through a code sent to an alternate location (email or text, for example), or with another form of data (fingerprint, face scan, etc.).

Password rotation should be implemented across every account, system, networked hardware, IoT device, application, service, etc. **Passwords should be unique, never reused or repeated, and randomized on a scheduled basis, upon check-in, or in response to specific threat or vulnerability**.

C - federated identities and assertions to convey the result of authentication at a given Federation Assurance Level (FAL)

federated identities and assertions

Federated identity **allows authorized users to access multiple applications and domains using a single set of credentials**. It links a user's identity across multiple identity management systems so they can access different applications securely and efficiently.

Federated identity management gives you control over your password standards and login procedures for your centralized SaaS application database, which improves your security immensely. You can set character limits, reset schedules, enforce multifactor authentication, and even monitor when and how your employees are accessing your FIM system.

Most federated identity management systems also include [multifactor authentication](https://www.fool.com/the-ascent/small-business/identity-management/articles/multifactor-authentication/) to prevent malicious intruders from accessing assets through exposed passwords. Microsoft Authenticator App