# CWE Detail – CWE-521

## Description

The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.

## Extended Description

Authentication mechanisms often rely on a memorized secret (also known as a password) to provide an assertion of identity for a user of a system. It is therefore important that this password be of sufficient complexity and impractical for an adversary to guess. The specific requirements around how complex a password needs to be depends on the type of system being protected. Selecting the correct password requirements and enforcing them through implementation are critical to the overall success of the authentication mechanism.

## Threat-Mapped Scoring

Score: 3.0

Priority: P2 - Serious (High)

## Observed Examples (CVEs)

**•** CVE-2020-4574: key server application does not require strong passwords

## Related Attack Patterns (CAPEC)

* CAPEC-112
* CAPEC-16
* CAPEC-49
* CAPEC-509
* CAPEC-55
* CAPEC-555
* CAPEC-561
* CAPEC-565
* CAPEC-70

## Attack TTPs

**•** T1110.001: Password Guessing (Tactics: credential-access)

**•** T1133: External Remote Services (Tactics: persistence, initial-access)

**•** T1110.002: Password Cracking (Tactics: credential-access)

**•** T1021.002: SMB/Windows Admin Shares (Tactics: lateral-movement)

**•** T1021: Remote Services (Tactics: lateral-movement)

**•** T1078.001: Default Accounts (Tactics: defense-evasion, persistence, privilege-escalation, initial-access)

**•** T1110.003: Password Spraying (Tactics: credential-access)

**•** T1110: Brute Force (Tactics: credential-access)

**•** T1114.002: Remote Email Collection (Tactics: collection)

**•** T1558.003: Kerberoasting (Tactics: credential-access)

## Modes of Introduction

**•** Architecture and Design: COMMISSION: This weakness refers to an incorrect design related to an architectural security tactic.

**•** Implementation: Not enforcing the password policy stated in a products design can allow users to create passwords that do not provide the necessary level of protection.

## Common Consequences

**•** Impact: Gain Privileges or Assume Identity — Notes: An attacker could easily guess user passwords and gain access user accounts.

## Potential Mitigations

**•** Architecture and Design: A product's design should require adherance to an appropriate password policy. Specific password requirements depend strongly on contextual factors, but it is recommended to contain the following attributes: Enforcement of a minimum and maximum length Restrictions against password reuse Restrictions against using common passwords Restrictions against using contextual string in the password (e.g., user id, app name) Depending on the threat model, the password policy may include several additional attributes. Complex passwords requiring mixed character sets (alpha, numeric, special, mixed case) Increasing the range of characters makes the password harder to crack and may be appropriate for systems relying on single factor authentication. Unfortunately, a complex password may be difficult to memorize, encouraging a user to select a short password or to incorrectly manage the password (write it down). Another disadvantage of this approach is that it often does not result in a significant increases in overal password complexity due to people's predictable usage of various symbols. Large Minimum Length (encouraging passphrases instead of passwords) Increasing the number of characters makes the password harder to crack and may be appropriate for systems relying on single factor authentication. A disadvantage of this approach is that selecting a good passphrase is not easy and poor passwords can still be generated. Some prompting may be needed to encourage long un-predictable passwords. Randomly Chosen Secrets Generating a password for the user can help make sure that length and complexity requirements are met, and can result in secure passwords being used. A disadvantage of this approach is that the resulting password or passpharse may be too difficult to memorize, encouraging them to be written down. Password Expiration Requiring a periodic password change can reduce the time window that an adversary has to crack a password, while also limiting the damage caused by password exposures at other locations. Password expiration may be a good mitigating technique when long complex passwords are not desired. See NIST 800-63B [REF-1053] for further information on password requirements. (Effectiveness: N/A)

**•** Architecture and Design: Consider a second
 authentication factor beyond the password, which prevents the
 password from being a single point of failure. See CWE-308 for
 further information. (Effectiveness: N/A)

**•** Implementation: Consider implementing a password complexity meter to inform users when a chosen password meets the required attributes. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)