Bcrypt Generation

Contents

Overview

This document is intended to provide Bcrypt concept and Encryption Type.

What is Bcrypt

Bcrypt is a **cryptographic hash function** specifically designed for **secure password hashing**. It is used to protect passwords by converting them into fixed-length, irreversible strings (hashes), making it extremely difficult for attackers to retrieve the original password even if they gain access to the stored data.

How Bcrypt works(Hash Generation Process)

- **Input**: A plain-text password.

- **Salt Generation**: Bcrypt automatically generates a **random salt** — a unique string added to the password before hashing. This ensures that even identical passwords produce different hashes.

- **Work Factor (Cost)**: Bcrypt uses a configurable **cost parameter** (e.g., 10, 12, 14), which controls how computationally intensive the hash calculation is. A higher cost means more security, but slower hashing.

- **Hashing**: The password and salt are combined and passed through the bcrypt algorithm multiple times (based on the cost), producing the final hash.

- **Output**: A string in the format:

```
$2a$12$salt22characters......hash31characters......
```

Why Bcrypt is Not "Encryption"

- **Encryption** is a two-way process (you can decrypt).

- **Hashing** (like bcrypt) is one-way — designed to be **non-reversible**.

- Bcrypt hashes cannot be decrypted. They can only be **compared** by re-hashing the input and checking if it matches the stored hash.

Why Bcrypt is Secure

- Prevents **rainbow table attacks** via salting.

- Defends against **brute-force attacks** through adjustable computation cost.

- Resistant to **hash collisions** and **timing attacks**

Use Cases

- Prevents **rainbow table attacks** via salting.
- Defends against **brute-force attacks** through adjustable computation cost.
- Resistant to **hash collisions** and **timing attacks**