**Bcrypt**

**What is bcrypt?**

* Bcrypt is a password hashing algorithm designed by Niels Provos and David Mazières based on the **Blowfish** cipher. The name “bcrypt” is made of two parts: b and crypt, where b stands for Blowfish and crypt is the name of the hashing function used by the Unix password system.
* Bcrypt was created as a result of the failure of Crypt to adapt to technology and hardware advancement. Bcrypt is designed to be a slow algorithm, which is a good thing when it comes to password hashing. Therefore, bcrypt is perfect for password hashing because it reduces brute-force attacks.

**How does bcrypt work?**

* bcrypt takes a user-submitted plain password and converts it into a **hash**. The hash is what is stored in the database. This prevents attackers from accessing users’ plain passwords in the event of a data breach. Unlike some other password-hashing algorithms that just hash the plain password, bcrypt uses the concept of salt.
* This unique randomly generated string provides an additional level of security for a generated hash. Before the plain password is hashed, a salt is generated. Then, it is appended to the plain password, and everything is hashed (the plain password and salt). This help protects against rainbow table attacks because attackers can randomly guess users’ passwords, but they can’t guess the salt.
* Bcrypt also uses a cost factor (or work factor) to determine how long it takes to generate a hash. This cost factor can be increased to make it slower as hardware power increases. The higher the cost factor, the more secure the hash and the slower the process. Therefore, you need to find the right balance between security and speed.
* The generated hash will include the salt and other things, like the hash algorithm identifier prefix, the cost factor, and the hash. The hashing process is irreversible. The hash cannot be converted back to the original plain password. Therefore, to determine whether a user provides the correct password, the provided password is hashed (using the original salt) and compared against the hash stored in the database.

**Benefits of password hashing by bcrypt**

- Bcrypt has significant advantages over other hashing methods like MD5, SHA1, SHA2, and SHA3. They can all perform hashing of a large number of data in less time. Suppose an attacker has a robust system capable of trying 700-900 million passwords in seconds. Your password containing alphanumeric and special character values will be cracked in a few seconds.

-So, all of these hashing methods cannot be used to encrypt the password.

Now, the main question is, how does bcrypt provide a significant advantage here?

Bcrypt was built upon Blowfish keying schedule and used a work factor, which decides how expensive the hash function will be. After knowing it, bcrypt will get slower if an attacker makes multiple requests in a single time frame. So generally, cracking one password will take 12 damn years. Also, bcrypt uses salt, which helps prevent attacks like rainbow table attacks and is suitable for securing passwords.