# **Cyber Security and Related Topics**

# How to Store Passwords and achieve login/authentication

Best way is hashing and salting. But lets understand why and what problems did it overcame.

## Levels of Password Authentications

## F: Using Plain Storage

This was the oldest way and totally unsecure. Remember one thing, we have seen data breach issues in past and it is always the database that is the most vulnerable part in our system. So storing password directly was not a good idea at all.

## D: Encryption

Better approach but still had some issues like where to store the decryption key? If hacker can gain access to DB, he can also read the config files / other server files.

## C: Hashing

We know why hashing is better. We are just storing the hashed password and at time of login we will just compare the hash is matching or not.

But the problem is, what if someone uses the dictionary attack. Weak passwords are present in the hacker’s dictionary already and he can easily track those passwords post hacking the DB.

Example – MD5, SHA-1

## B: Hashing + Salt

To overcome the previous problem, we could use a modern algo like SHA-256 which takes the input string and a random salt (some string) and appends it. We will now store this hashed string with the salt and compare the passwords post appending this salt.

But still a hacker can read the salt and perform dictionary attack with modern GPU based machines which can generate billions of hashes in a second.

## A: Slow Hashing (BCrypt / SCrypt)

These algos work just as previous one but better in terms of possibility of generating the guesses. Why because these are password hashing algorithms and extremely slow on purpose. We can choose the slowness.

Best Slowness is around work factor 15 ~ around 1.3s which makes the hacker to generate the passwords guesses extremely slow. That prevents the dictionary attack up to an extent and client should be choosing tough passwords so that dictionary attack can be prevented 100%

## S: No Store

Nowadays people are not storing the passwords at all and using Google / facebook login in order to prevent any data breach. The reason is that these are authentic methods and if Google is hacked, nothing is secure at all in the internet.

# HTTP, HTTPS and how it works.