Kentico provides several options for storing user passwords in the database. The passwords can either be secured using a cryptographic function or saved in plain text (not recommended).

To configure the password format, open the **Settings** application and select an option using the **Security & Membership -> Passwords -> Password format** setting. Currently, the following options provide sufficient security:

- **PBKDF2** - the default and recommended option with the strongest security. Applies a cryptographic function to passwords and repeats the operation many times. To learn more, see PBKDF2.
- **SHA-2 with salt** - uses the SHA-2 hash function with an additional salt applied to the password input.

Note: Changing the password format only affects how future passwords will be stored. Existing passwords remain functional, but are stored in their original format (the *CMS_User* database table contains a column that specifies the format of each user's password). You need to reset all passwords to store them in the new format.

For this reason, we recommend setting the appropriate password format directly after installation, before you create user accounts or allow users to start registering.

Customizing the number of iterations for PBKDF2

When the **PBKDF2** option is selected for the **Password format** setting, the system uses the PBKDF2 key derivation standard to secure user passwords. A cryptographic function is applied to the original password input (with a salt), and the operation is repeated many times. By default, the number of iterations is 10000.

You can adjust the number of iterations used to create the final secured password values.

- Increasing the number of iterations generates password hashes that are more resistant to brute force attacks. However, a very high number may lead to performance problems in situations where many users authenticate or change their password at the same time.
- You can decrease the number of iterations if you experience performance problems and security is not a priority.

To customize the number of PBKDF2 iterations, you need to use the API – set the **PbkdfIterationsCount** property of the **CMS.Helpers.SecurityHelper** class.

To ensure correct behavior, set the property at the beginning of the application's life cycle (during initialization):

1. Create a custom module class.
   - Either add the class into a custom project within the Kentico solution (recommended) or directly into the Kentico web project (into a custom folder under the **CMSApp** project for *web application* installations, into the **App_Code** folder for *web site* installations).
2. Override the module's **OnInit** method and set the **PbkdfIterationsCount** property.

For basic execution of initialization code, you only need to register a "code-only" module through the API. You do NOT need to create a new module within the **Modules** application in the Kentico administration interface.

Example

```
using CMS;

using CMS.DataEngine;
using CMS.Helpers;

// Registers the custom module into the system
[assembly: RegisterModule(typeof(CustomInitializationModule))]

public class CustomInitializationModule : Module
{
        // Module class constructor, the system registers the module under the name
"CustomInit"
        public CustomInitializationModule()
                : base("CustomInit")
        {
        }

        // Contains initialization code that is executed when the application starts
        protected override void OnInit()
        {
                base.OnInit();

                // Sets the number of iterations used when generating user passwords
in the PBKDF2 format
                SecurityHelper.Pbkdf2IterationsCount = 50000;
        }
}
```

When generating new passwords in the PBKDF2 format, the system now uses the assigned number of iterations. Existing passwords created with a different number of iterations remain functional. You only need to reset existing passwords if you wish to enforce the new number of iterations.

Configuring the salt for the SHA-2 password format

If you select the **SHA2 with salt** option for the **Password format** setting, the system secures passwords using the SHA-2 hash function with the additional application of a **salt**. A salt is a string appended to passwords before they are hashed, which helps protect the passwords against dictionary or other types of brute force attacks. It also ensures that every user has a different password hash, even if multiple users have the same password.

Kentico adds two types of salt to passwords:

- **User salt** - by default, the GUID of each user (stored in the *UserGuid* column) is appended to the passwords before the hash function is applied.
- **Password salt** - to increase the length of the salt (to further improve the security of hashed passwords), you can define a custom string that the system appends to every password. Add the following key into the **<appSettings>** section of your web.config file:

  ```
  <add key="CMSPasswordSalt" value="SaltText" />
  ```

The following diagram shows how the password and salt values are composed before the hash function is applied:

Password	User salt	Password salt