Document on

Login Password Encryption

Login Security Module

LSM-1.0

Document Introduction:

What is encryption?

In computing, encryption is the method by which plaintext or any other type of data is converted from a readable form to an encoded version that can only be decoded by another entity if they have access to a decryption key. Encryption is one of the most important methods for providing data security, especially for end-to-end protection of data transmitted across networks.

Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private. Organizations and individuals also commonly use encryption to protect sensitive data stored on computers, servers and mobile devices like phones or tablets.

About Crypt()

The crypt() function returns a hashed string using DES, Blowfish, or MD5 algorithms. Programing technology checks what algorithms are available and what algorithms to use when it is installed. The salt parameter is optional. However, crypt() creates a weak password without the **salt**. We use salt for better security.

Passwords are encrypted by the UNIX crypt encrypting algorithm before they are stored in the directory. When crypt is used, only the 1st 8 characters of a password are used. Passwords longer than 8 characters are truncated.

Objectives

As per view of organization the application should not be accessible, crack able by unauthorised user, as concern with security issues the organizational application should be robust and secure. To protect our systems from unauthorised users we have implemented login security with strong encryption key of password.

Requirements:

1. **Secure** The security system should not store passwords in plain text in the database.
2. Once password is encrypted in cipher text, then password **should be decrypt in plain text**.
3. By default password also be **generated in encryption format**.
4. If someone got encrypted password then, he/she will **not be use that encrypted password as a login credentials**.
5. Password pattern **should be contain digit, special character, capital, small case format**.
6. **1st to max characters** of a password can used.

Standard Operating Procedure:

Login Procedure:

1. The operating procedure starts with process of user registration, when we register user at that event we setup by default **encrypted password against that employee number in database row**.
2. When employee try to login on our portal, that time employee have to **submit credentials as plain text.**
3. We verify that credentials and try to **match with existing database record**.
4. If credentials is correct then we **allowing user to enter in portal**, if credentials incorrect then we **terminate the process**.

Encryption Procedure:

1. We are encrypting password with **crypt()** function, using **strong encryption key**.
2. Once **password encrypted then we cannot make it as a decrypt**.
3. We can encrypt plain text including **1 to max characters**
4. When user try to reset password at that time we also updating password as encrypted format.

Bulk Encryption Procedure:

1. Instead of updating single encrypt password in database, we have develop **bulk encrypted password update page**.
2. This page fetch employee details like **employee code, name, plain text password and show that details on table format** of page.
3. As per the fetch count of table records we are **looping the method of password encryption**.
4. After got encrypted password we are sending that list to **update password method**.
5. Update password method **updating the bulk password according to employee code**.
6. After password encryption employee can use **credentials as plain text** on login form and they can enter to portal.
7. The bulk encryption procedure not accessible to other users **it will only enable for developers**.

Conclusion:

As we developing enterprise application tool for our company, we need more secure methods and infrastructure to protect our sensitive data from un-authorized user, for this purpose IT Development team created their own encryption methods with hidden key to protect user's sensitive password data

Once password data is encrypted then developer of algorithm cannot be able to break the password

Document & Development By

IT Development Team (R&D)

1. Rahul Gokhle
2. Pradnya P. Shroff
3. Kaustubh A. Khadke