**Лабораторна робота № 4**

1. **Механізми парольного захисту**
   * + - 1. **інформаційних систем.**

**Мета роботи:** ознайомити студента з перевагами та недоліками парольного захисту, а також механізмами реалізації парольного захисту інформаційних систем. Набути знань із реалізації механізмів парольного захисту у вигляді програмних модулів.

**Завдання:** Освоїти основні механізми парольного захисту інформаційних систем, сформувати структуру та реалізувати модуль парольного захисту власного програмного продукту.

**1.** **Опис методу реалізації завдання**

Пароль – це секретна інформація, якою має володіти тільки авторизований суб’єкт.

Паролем може бути слово, комбінація для замка чи персональний ідентифікаційний номер (PIN).

Парольний механізм може бути досить легко реалізований і мати низьку вартість. Але є суттєві мінуси: зберегти пароль в секреті часто буває проблематично, зловмисники постійно придумують нові методи крадіжки, зламу й підбору пароля. Це обумовлює слабку захищеність парольного механізму.

Для виконання завдання використаємо такі способи підвищення захищеності пароля - Перевірка мінімальної довжини пароля (10 символів) та Перевірка допустимої кількості разів введення пароля (7 разів).

**2.** **Алгоритм реалізації завдання**

1. Перед створенням користувача в базі перевіримо чи довжина його пароля входить в заданий діапазон.
2. Зашифруємо пароль використавши bcrypt алгоритм
3. Перед спробою входу користувача перевіримо чи вірно введена комбінація емейлу і пароля
4. Якщо користувач використав неправильні дані то відхиляємо його запит і збільшуємо лічильник неправильних спроб
5. Якщо користувач використав правильні дані то перевіримо чи він вичерпав ліміт введень. Якщо так то відхиляємо його, якщо ні то здійснєюмо його вхід до системи.

##### **3. Лістинг програми**

Лістинг 1. Модель користувача

**class User < ApplicationRecord**

**MAX\_ATTEMPTS = 7**

**has\_secure\_password**

**validates :name, :email, presence: true**

**validates :email, uniqueness: true, format: { with: URI::MailTo::EMAIL\_REGEXP }**

**validates :password, length: { minimum: 10, maximum: 32 }, if: :password**

**scope :locked, -> { where('attempts >= ?', MAX\_ATTEMPTS) }**

**before\_save { self.email = email.downcase }**

**def locked?**

**attempts >= MAX\_ATTEMPTS**

**end**

**end**

Лістинг 2. Контролер регістрацій

**class UsersController < ApplicationController**

**before\_action :set\_labs**

**def new**

**@user = User.new**

**end**

**def create**

**@user = User.new(user\_params)**

**if @user.save**

**flash[:notice] = I18n.t('users.registrations.success', name: @user.name)**

**redirect\_to new\_session\_path**

**else**

**flash.now.alert = I18n.t('users.registrations.error')**

**render :new**

**end**

**end**

**private**

**def user\_params**

**params.require(:user).permit(:name, :email, :password, :password\_confirmation)**

**end**

**end**

Лістинг 3. Контролер сесій

**class SessionsController < ApplicationController**

**before\_action :set\_labs, except: :destroy**

**before\_action :set\_user, only: :create**

**def new**

**redirect\_to root\_url if logged\_in?**

**end**

**def create**

**if @user&.authenticate(params[:session][:password])**

**reject\_locked!**

**else**

**@user&.increment!(:attempts) unless @user&.locked?**

**flash.now[:error] = I18n.t('users.sessions.error')**

**render :new**

**end**

**end**

**def destroy**

**log\_out**

**redirect\_to root\_url, notice: I18n.t('users.sessions.logout')**

**end**

**private**

**def set\_user**

**@user = User.find\_by email: params[:session][:email].downcase**

**end**

**def reject\_locked!**

**if @user.locked?**

**redirect\_to new\_session\_path, alert: I18n.t('users.sessions.locked')**

**else**

**log\_in @user**

**redirect\_to root\_url, success: I18n.t('users.sessions.success', name: @user.name)**

**end**

**end**

**end**

Лістинг 4. Допоміжні методи

**module SessionsHelper**

**def log\_in(*user*)**

**session[:user\_id] = user.id**

**end**

**def current\_user**

**@current\_user ||= User.find\_by(id: session[:user\_id]) if session[:user\_id]**

**end**

**def logged\_in?**

**current\_user.present?**

**end**

**def log\_out**

**session.delete(:user\_id)**

**@current\_user = nil**

**end**

**end**

##### **4. Результати виконання програми**

##### Рис 1. Реєстрація

##### Рис 2. Вхід

##### 

##### Рис 3. Акаунт заблокований

##### 

##### Рис 4. Успішний вхід

##### 

##### 

##### **5. Висновок**

В даній лабораторній роботі я освоїв основні механізми парольного захисту інформаційних систем, сформувати структуру та реалізував модуль парольного захисту власного програмного продукту.

Захист здійсює шифрування паролю, перевірку довжини на перевірку кількості неправильних введень.