УДК:004.4
Ребриков М. М.[1], Неласа Г.В.[2]
[1] студ. гр. РТ-818 НУ «Запорізька політехніка»
[2] проф. НУ «Запорізька політехніка»

РОЗРОБКА TELEGRAM-БОТА ДЛЯ ЗБЕРЕЖЕННЯ ДАНИХ КОРИСТУВАЧА ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ NODE.JS ТА СУБД MONGODB

На даний час є велика кількість веб-сайтів і додатків, кожен з яких вимагає від користувача реєстрації в системі. Щоб виключити ймовірність злому свого логіна, користувачеві необхідно використовувати складні паролі, які відрізняються один від одного. Внаслідок цього виникає потреба запам'ятовувати велику кількісь паролів, що доволі складно.

Надійним рішенням зберігати паролі є використання менеджера паролів. Деякі з них виконані у вигляді десктопних, мобільних або веб додатків, таких як, 1Password, KeePass, LastPass, RoboForm та багато інших. Більшість подібних сервісів є пропрієтарними, тобто користувач не може побачити початковий код і зрозуміти, чи є методи шифрування і зберігання даних, який використовує той чи інший сервіс не застарілим або безпечним.

Так як останнім часом дуже багато людей користуються месенджером telegram, виникла ідея знайти аналоги десктопних і мобільних додатків для нього. Під час досліджень було знайдено вже готового telegram бота (@PasswordWizardBot), але відкритим залишилось питання безпеки обробки та збереження даних користувачів. Так як не було знайдено відкритий репозиторій, в якому могли б зберігатися файли з вихідним кодом даного сервісу, переконатися в його безпеці не є можливим. Тому було прийняте рішення розробити власний telegram-бота для зберігання паролів.

Основні поставлені завдання такі:

зберігання паролів в зашифрованому вигляді в базі даних;

можливість додавати, редагувати і видаляти паролі;

захист бази даних і доступу до неї;

перевірка паролів на унікальність і вразливість;

генерація паролів з заданими користувачем параметрами;

На даний момент реалізовано зберігання паролів в зашифрованому вигляді в базі даних. Також в базі даних зберігається унікальний ідентифікатор користувача, за яким і здійснюється доступ до полів з паролями в базі даних, при цьому для отримання відповіді від сервера і подальшим виведенням всіх паролів даного користувача, йому необхідно ввести генеральний пароль, який він вказує при реєстрації в чат боті. Також створено функціонал для генерації нового пароля.

Важливим моментом проекту є те, що на стадії розгортання бота на віддаленому сервері, його вихідний код буде перебувати у відкритому доступі, тому будь-хто може побачити початковий код розробленого бота і переконатися в його безпеці, запропонувати правки і поліпшення щодо коду програми. При цьому всі дані конфігурації проекту, включаючи і ключі шифрування, винесені в окремі файли, які не потраплять у відкритий репозиторій.

Для покращення роботи та швидкодії бота були проведені тести за результатами яких знайдений кращий в даному випадку метод шифрування даних.

Незабаром планується доробка користувальницького інтерфейсу, додавання функціоналу для перевірки електронної пошти користувача на факт злому її на будь-яких сторонніх сайтах, розміщення сайту на віддаленому сервері і, можливо, додавання функціоналу для шифрування і зберігання в зашифрованому вигляді текстових повідомлень і різних файлів. Також планується розробити захист від «ін'єкцій» в базу даних.

Так як обмежені можливості месенджера telegram, а саме взаємодія користувача і інтерфейсу чат-бота, не дозволяють реалізувати деякий функціонал, в подальшому планується створення десктопного, мобільного та веб-додатків з допомогою таких фреймворків для Node, як React і React Native.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Node.js [Електронний ресурс] – https://nodejs.org
2. Express.js [Електронний ресурс] – https://expressjs.com
3. MongoDB [Електронний ресурс] – https://www.mongodb.com
4. Telegram Bot Api [Електронний ресурс] – https://core.telegram.org/bots/api
5. node-telegram-bot-api [Електронний ресурс] – https://github.com/yagop/node-telegram-bot-api
6. crypto-js [Електронний ресурс]– https://www.npmjs.com/package/crypto-js