УДК 004.056.5
DOI https://doi.org/10.32689/maup.it.2024.4.13

Богдан КОРНІЄНКО
доктор технічних наук, професор, професор кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського, bogdanko@gmx.net
***ORCID:** 0000-0002-2521-0878*

Леся ЛАДІЄВА
кандидат технічних наук, доцент, доцент кафедри технічних та програмних засобів автоматизації, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», lrynus@yahoo.com
***ORCID:** 0000-0002-1706-0072*

Ксенія УЛЬЯНИЦЬКА
кандидат технічних наук, доцент кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», ulianitskaya.k@gmail.com
***ORCID:** 0000-0003-0240-6250*

Лілія ГАЛАТА
доктор філософії, доцент кафедри кібербезпеки,
Національний авіаційний університет, galataliliya@gmail.com
***ORCID:** 0000-0002-7978-3954*

Андрій НЕСТЕРУК
аспірант кафедри інформаційних систем та технологій, асистент кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», aonesterukr@gmail.com
***ORCID:** 0000-0002-1563-7245*

ЗАХИСТ КРИТИЧНИХ РЕСУРСІВ ВЕБ-ЗАСТОСУНКУ З ОРЕНДИ НЕРУХОМОСТІ

***Анотація. Мета** цієї роботи полягає в розробці системи захисту веб-застосунку з використанням сучасних технологій програмування та розроблення бази даних, яка буде стійкою до змін та сторонніх втручань, буде здатна запобігати неавторизованому доступу до веб-застосунку з оренди нерухомості.*

***Методологія**, використана в роботі, полягає у розробці системи захисту веб-додатків з використанням сучасних технологій NET Framework, ASP. NET Core, EF, SSMS, Swagger. Система стійка до змін і стороннього втручання, здатна запобігти несанкціонованому доступу. Описано найпопулярніші готові сервіси для реалізації відповідного захисту.*

***Наукова новизна** роботи полягає у визначенні моделі білого списку розробки захищених веб-додатків та основні кроки реалізації моделі. Реалізовано модель білого списку для веб-додатку за допомогою системи ролей і доступу. Розроблено серверну частину веб-додатку, яка включає вбудований функціонал основних методів запобігання злому. Розроблено метод доступу до приватної інформації користувача за допомогою алгоритму шифрування Rijndael.*

***Висновки**, зроблені на основі проведених досліджень, підкреслюють важливість захисту веб-додатків від зловмисників, що залежить від технологій і компонентів, які використовуються при створенні веб-додатків, а також відможливих вразливостей цих компонентів. Існують різні класифікації вразливостей, кожна атака через вразливість має свої особливості, а причиноювразливостей є помилки в розробці, реалізації та застосуванні компонентів веб– додатків, звідси необхідність пошуку і протидії вразливостям.*

В результаті створено програмний продукт, веб-застосунок для оренди та продажу нерухомості із вбудованою системою захисту інформації. Архітектура проекту запобігає загрозі SQL-ін'єкцій. Таким чином, представлена робота робить значний внесок у сферу захисту критичних ресурсів веб-застосунку, пропонуючи інноваційні підходи до протидії загрозам та покращуючи ефективність цього процесу. Ця програма стане незамінним інструментом для інженерів та аналітиків, сприяючи підвищенню якості захисту критичних ресурсів веб-застосунку.

***Ключові слова:** веб-додаток; безпека; система захисту; загрози; білий список; модель.*

Bogdan KORNIYENKO, Lesya LADIEVA, Kseniia ULIANYTSKA, Liliia GALATA, Andrii NESTERUK. PROTECTION OF CRITICAL RESOURCES OF THE REAL ESTATE RENTAL WEB APPLICATION

***Abstract. The purpose** of this work is to develop a web application protection system using modern programming technologies and database development, which will be resistant to changes and third-party interventions, will be able to prevent unauthorized access to the real estate rental web application.*

The methodology *used in the work consists in the development of a web application protection system using modern technologies NET Framework, ASP. NET Core, EF, SSMS, Swagger. The system is resistant to changes and third-party intervention, capable of preventing unauthorized access. The most popular ready-made services for the implementation of appropriate protection are described.*

The scientific novelty *of the work consists in defining the white list model for the development of secure web applications and the main steps of implementing the model. Implemented a whitelist model for a web application using a role and access system. The server part of the web application has been developed, which includes the built-in functionality of the main hacking prevention methods. A method of accessing private user information using the Rijndael encryption algorithm has been developed.*

The conclusions *drawn from the conducted studies emphasize the importance of protecting web applications from attackers, which depends on the technologies and components used in the creation of web applications, as well as on the possible vulnerabilities of these components. There are different classifications of vulnerabilities, each vulnerability attack has its own characteristics, and the cause of vulnerabilities is errors in the development, implementation and application of components of web applications, hence the need to find and counter vulnerabilities.*

As a result, a software product was created, a web application for renting and selling real estate with a built-in information protection system. The architecture of the project prevents the threat of SQL injections. Thus, the presented work makes a significant contribution to the protection of critical web application resources, offering innovative approaches to countering threats and improving the effectiveness of this process. This program will become an indispensable tool for engineers and analysts, helping to improve the quality of protection of critical web application resources.

Key words: *web application; security; protection system; threats; white list; model.*

Вступ. Постановка проблеми. Захист веб-ресурсів залишається одним із важливих напрямків інформаційної безпеки. Щороку кількість веб-ресурсів збільшується, зростає також кількість конфіденційної інформації, яка локалізується на серверах віддаленого доступу (особливо із використанням хмарних технологій).

У результаті цього зростають не тільки кількість атак на веб-ресурси, але й економічні наслідки таких атак. Останнім часом вразливість веб-ресурсів до атак отримала політичний вимір унаслідок як поширення гібридних війн у світі, так і зростання терористичних загроз.

Зі збільшенням залежності компаній будь-якого напрямку діяльності від ІТтехнологій, гостро постає питання забезпечення інформаційної безпеки. Одним з ключових заходів в забезпеченні інформаційної безпеки компанії є тестування на проникнення. Це дозволяє упевнитися в надійності захисту від несанкціонованого доступу та інших загроз інформаційної безпеки [1–7].

Сьогодні веб-вразливості перевершують за кількістю і можливій шкоді будь-які інші проблеми інформаційної безпеки. Більшість зовнішніх атак на корпоративні інформаційні системи націлені саме на вразливості веб-додатків.

Аналіз попередніх досліджень. Для захисту від більшості популярних видів атак достатньо належним чином перевіряти вхідні дані. Також рекомендовано використовувати шифрований протокол HTTPS та будувати програмний додаток ресурсу на одному з відомих програмних каркасів, в якому вбудовані механізми перевірки, шифрування та валідації вхідних даних [8–16].

На даний час найбільш розповсюдженими методологіями проведення тестування на проникнення є:

- The Open Source Security Testing Methodology Manual (OSSTMM);
- The National Institute of Standards and Technology (NIST) SpecialPublication 800-115;
- OWASP Testing Guide;
- Penetration Testing Execution Standard (PTES);
- Information Systems Security Assessment Framework (ISSAF);
- BSI – Study A Penetration Testing Model.

Методологія The Open Source Security Testing Methodology Manual (OSSTMM) є досить формалізованим і добре структурованим документом для тестування мережі. Документ має так звану «Карту безпеки» – візуальний показник безпеки. На карті вказуються основні галузі безпеки, які включають в себе набори елементів, які повинні бути протестовані на відповідність методиці [17-19].

Методологія NIST Special Publications 800-115 Technical Guide to Information Security Testing and Assessment. Створена і підтримується підрозділом NIST та виділяє як мінімум 3 фази проведення оцінювання інформаційної безпеки: планування, виконання, пост-експлуатація (аналізотриманих даних, виявлення причин що призвели до появи вразливостей, розробка рекомендацій до знешкодженню вразливостей і розробка звіту). У розділі «Техніки оцінки вразливостей мети», в як одна з технік описуються Тестина проникнення, а саме Фази і Логістика тестів [20].

Методологія OWASP (Open Web ApplicationSecurity Project) Testing Guide.OWASP (Open Web Application Security Project) -міжнародне відкрите співтовариство, яке орієнтоване на поліпшення безпеки програмного забезпечення. OWASP Testing Guide є більш широкою методологію в порівнянні з іншими, тому що дає вказівки не тільки по тестах на проникнення, але і з аналізу веб-додатків в цілому

(наприклад – вихідного коду), оскільки ця методика фокусує свою увагу саме на виявленнях вразливостей веб-додатків [21].

Методологія PTES – Penetration Testing Execution Standard – Technical Guidelines. Стандарт, розроблений для об'єднання як бізнес вимог, так і можливостей служб безпеки, і масштабування тестів на проникнення. На першому підготовчому етапі детально розглядаються встановлюються канали комунікацій, правила взаємодії і контролю, конкретні способи реагування і моніторингу інцидентів. Далі виділені наступні етапи: збір інформації; моделювання загроз; методи аналізу вразливостей; забезпечення обходу контрзаходів і виявленнянайкращого шляху атаки; пост-експлоітація – аналіз інфраструктури, подальше проникнення вінфраструктуру, зачистка і живучість [22].

Методологія ISSAF – Information System Security Assessment Framework. Розроблено для внутрішніх контрольних перевірок. Документ охоплює величезну кількість питань, пов'язаних з інформаційною безпекою. Описана оцінка безпеки міжмережевих екранів, маршрутизаторів, антивірусних систем і багато іншого. Методологія ISSAF дозволяє змоделювати вимоги до внутрішніх заходів з безпеки, і направлена на оцінку безпеки комп'ютерних мереж, систем та додатків [23].

Методологія BSI – Study A Penetration Testing Model.Розроблено німецьким підрозділом «Federal Office for Information Security ». У документі описується проведення коректних випробувань системи на міцність. Детально описуються тільки сама методологія тестів, але і необхідні вимоги, правові аспекти застосування методології та процедури, які необхідно виконати для успішного проведення тестів. Наводиться класифікація тестів на міцність івизначені її критерії [24].

Метою статті є розробка системи захисту веб-застосунку з використанням сучасних технологій програмування та розроблення бази даних, яка буде стійкою до змін та сторонніх втручань, буде здатна запобігати неавторизованому доступу до веб-застосунку.

Виклад основного матеріалу. Проаналізувавши проблему створення веб-застосунків встановили, що ця проблема актуальна і має спільні риси з загальною концепцією створення безпечного прикладного забезпечення. Дана проблема частково залежить від механізмів захисту використовуваних веб-каркасів на яких будується веб-застосунок, використовуваній базі даних, безпеки в цілому від сервера на якому виконується веб-застосунок. Розроблено серверну частину веб-застосунку, що містить в собі функціонал запобігання методам злому.

Метод білий список для забезпечення безпеки у веб-застосунках

Концепція білого списку загально відома і використовується досить давно у багатьох сферах ще до появи інформаційних і цифрових технологій. В теорії використання моделі білого списку під час безпечної розробки веб– застосунків дозволяє запобігати деяким вразливостям які ігноруються усіма відомими веб-каркасами [25–28].

Веб-фреймворки запобігають більшості відомих вразливостям веб-додатків таких як SQL-ін'єкції, XXS, але вони не можуть запобігти деяким специфічним вразливостям які притаманні саме конкретній програмі яка розробляється деяким фреймворком, відповідальність за запобігання таких вразливостей залишається за розробниками.

Невідповідність між очікуваною і реальною поведінкою веб-застосунку може являтися індикатором атаки на веб-застосунок. Дана робота зосереджена на OWASP 4 і 7 вразливостях: небезпечні прямі посилання на об'єкти. відсутність контролю доступу функцій. Реалізовано механізм безпеки для веб-застосунку шляхом передбачення дозволених операцій. У роботі визначено, створено і впроваджено список дозволених взаємодій веб-застосунку. Ці правила керують HTTP-запитами і відповідями які обробляє веб-застосунок. Визначення елементів які входять до білого списку повинно розроблятися під час етапу проектування.

Формальне визначення моделі білого списку розробки безпечних веб-застосунків

Визначимо білий список як набір чотирьох множин {C, D, W, S}, де:

C – множина елементів {u, c_1, c_2,...,c_n}, де c_1, c_2,...,c_n – складові компонент які входять в межі системи, а u компоненти які за межею системи;

D – множина елементів {d_1, d_2,...,c_n}, де d_1, d_2,...,c_n стани компонентів;

W – множина впорядкованих пар {($c_{o,}$ c_d) : $c_{o,}$ c_d є C} кожна пара представляє собою перехід з початкового компоненту c_0 до кінцевого c_d;

S – матриця розмірності |C| X |C| $S_{co,\ cd}$ = $c_{безпечний(safe)}$ визначає безпечні компоненти {c_s : c_s є C} де c_d не може слідувати з c_o.

Кожна комірка матриці розмірності |C| X |C| містить унікальну підмножину x, {x: x ⊆ D}. Якщо оцінка станів x повертає 1, то упорядкована пара переходу стану з $c_{початковий(origin)}$ до $c_{кінцевий}$-(destination) додається до множини W.

Перехід з однієї складової до іншої керується функцією переходу де перехід з $c_{початковий}$ до $c_{кінцевий}$ відбувається тоді і тільки тоді якщо ($c_{o,}$ c_d) є W, інакше функція переходу визивається через (c_o, $S_{co,\ cd}$).

$$T(c_0, c_d) = \begin{cases} c_d, якщо (c_0, c_d) \in W \text{ інакше} \\ T(c_0, S_{c_0 c_d}) \end{cases}$$

Визначено кілька операцій які виконуються за допомогою білого списку. Операції поділені на дві категорії відповідно до того коли вони можуть бути застосовані. Наступні операції будуть використовуватися під час розробки:

– створення ($c_{o,}$ c_d) у множині W: додавання впорядкованої пари до множини;
– видалення ($c_{o,}$ c_d) з множини W: видалення впорядкованої пари з множини;
– введення {d_x} до множини D: додавання станів d_x до множини D;
– видалення {d_x} з множини D: видалення станів d_x з множини D;
– додавання {d_x} до підмножини x множини $W_{co,cd}$;
– видалення {d_x} з підмножини x множини $W_{co,cd}$;
– введення {c_s} у комірку матриці $S_{co,cd}$;
– оновлення {c_s} у комірці матриці.

Операції білого списку які дозволені при виконані:

– обчислення T($c_{o,}$ c_d);
– верифікація c_o à c_d:

c_d може слідувати до c_o якщо дане твердження хибне, то усі стани які належать підмножині x у $W_{co,cd}$ будуть повертати істину. Інакше перехід до безпечного стану c_s.

Множина усіх компонентів C і відношень W можуть бути представлені у вигляді двійкової матриці, де 1 означає дозволений перехід стану а 0 означає не дозволений. Кожна комірка матриці прийматиме значення або 0 або 1 відповідно до значень підмножини станів. Матриця S міститиме безпечні переходи станів у випадку коли перехід стану з c_o до c_d не буде дозволений.

Кроки впровадження моделі

В першу чергу потрібно ідентифікувати дозволену поведінку веб– застосунку шляхом створення діаграми яка буде відображати яким чином поводить себе програма. Діаграма повинна відображати усі дозволені взаємодії міжкомпонентами застосунку. Потрібно дослідити кожну операцію і ідентифікувати підмножину переходів станів (c_o,c_d) і помістити їх у відповідні комірки матриці W. Також потрібно ідентифікувати безпечні компоненти і переходи у разі повертанні хибного значення перевірки дозволу переходу стану. Безпечні компоненти c_s слід помістити у комірки які відповідають (c_o,c_d) у матриці W до матриці S. Отже на даному етапі розробки ми маємо підмножини переходу станів які записані у матрицю W і приймають значення 1 або 0 в залежності від дозволу переходів станів. Для простоти назвемо таке представлення як M, де

|C| X |C| = M де $M_{co,cd}$ = 1 якщо (c_o,c_d) є W і $M_{co,cd}$ = 0 якщо (c_o,c_d) не належить W. Модель білого списку може змінюватися або доповнюватися в залежності від ходу розробки. Варто також зазначти що, стани в множині D повинні бути простими а не комплексними.

Основні кроки побудови моделі білого списку:

– створити діаграму яка відображає назначену допустиму поведінку веб –застосунку;
– задати підмножини станів за для змоги аналізу і визначення дозволених переходів і взаємозв'язків застосунку;
– розмістити кожну підмножину переходу станів c_o,c_d відповідну коміркуматриці |C|x|C|;
– ідентифікувати безпечні компоненти c_s і розмістити їх у комірки які відповідають (c_o,c_d) у матриці W до матриці S;
– присвоїти значення 1 або 0 для кожної підмножини переходу станів взалежності відповідності білого списку;
– $M_{co,cd}$ = 1 якщо (c_o,c_d) є W і $M_{co,cd}$ = 0 якщо (c_o,c_d) не належить;
– належним чином налаштувати процес розробки для впровадження методики білого списку в уже існуючий процес розробки на будь-якому етапі.

Реалізація методу білий список для веб-застосунків

Розглядається веб-застосунок, який вимагає обов'язкову аутентифікацію користувача для можливості його використання. Застосунок дозволяє 3 спроби аутентифікації. Якщо користувач успішно пройде аутентифікацію застосунок пере направить користувача на його персональну сторінку. Користувач матиме змогу редагувати свій профіль або зв'язатися з іншими користувачами. Користувач також має змогу вийти зі свого профілю в будь-який час.

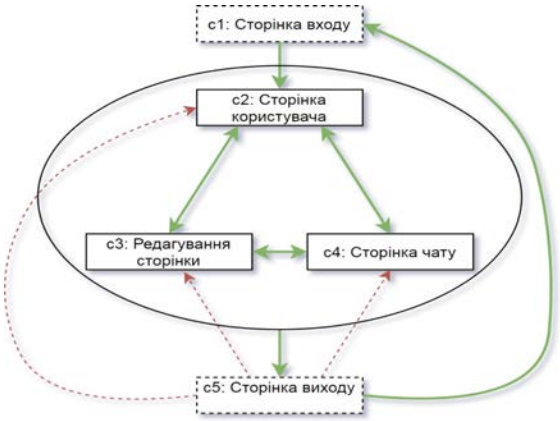

Рис. 1. Приклад діаграми, яка відображає поведінку програми

Модель за стосунку даного прикладу складається з 5 компонент. C = {u, c_1,c_2, c_3, c_4, c_5}. U представляє собою компоненти за границею системи і включаєтьсядо C за для повноти. Щодо глобальної множини станів D, припустимо що вони означають наступні стани:

- d1: користувач анонімний.
- d2: коистувач авторизований.
- d3: час існування сесії валіний.
- d4: попереднє представлення.
- d5: представлення підпослідовності
- d6: спроби аутентифікації < 3.

Білий список містить підмножину D у кожній комірці матриці. Для прикладу, білий список нижче відображає підмножину дозволених переходів станів з c_1, c_2 і іншу підмножину яка приводить до недозволеного переходу з c_5 до c_4. Для дозволеного переходу з c_1 до c_2, підмножина станів: x = {d_2, d_3, d_4 = login view, d_5 = user portal view, d_6}. Усі стани у x мають повертати істину. Для недозволених переходів з c_5 до c_4, підмножина станів x = {d_2, d_3, d_4 = edit profile view або user portal view, d_6}.

Очевидно, перехід з стану c_5 до c_4 не дозволений, бо перший стан у підмножині d_2 не може бути досягненим якщо користувач вийшов з аканту. Заповнимо наші переходи у матрицю переходів станів.

Нехай множина впорядкованих пар переходу станів наступна

W = {(u; u) ; (u; c1) ; (c1; c1) ; (c1; c2) ; (c2; c2) ; (c2; c3) ; (c2; c4) ; (c2; c5) ;(c3; c2) ; (c3; c3) ; (c3; c4) ; (c3; c5) ; (c4; c2) ; (c4; c3) ; (c4; c4) ; (c4; c5) ; (c5; c1)}

Наступник крок це заповнити матрицю S безпечними компонентами для перенаправлення переходу на безпечний стан якщо він не дозволений.

Таким чином, проаналізована концепція типового використання білого списку у IT а також розроблена модель білого списку і методика використання даної моделі для розробки безпечних веб-застосунків. А також приклад використання. Можна зробити висновок, що будь-яку концепцію захисту чи розмежування доступу можна адаптувати під будь-який процес, а саме під процес розробки захищених веб-застосунків. Згідно аналізу створеного прикладу модель являється працюючою і успішно забезпечує безпеку спроектованого застосунку.

Захист критичних ресурсів веб-застосунку

Для реалізації проекту використовували ASP.NET – технологію створення веб-застосунків і веб-сервісів, яка була створена компанією Microsoft. Ця технологія є основною частиною платформи Microsoft.NET і розвитком старішої технології Microsoft ASP. На цей час останньою версією цієї технології є ASP.NET Core 2.0 [29-30].

Засіб Web API засноване на додаванні в додаток ASP.NET MVC Framework контролера спеціального виду. Цей різновид контролерів, яка називається контролером API, володіє двома характеристиками:

- методи дій повертають об'єкти моделей, а не об'єкти типу ActionResult;
- методи дій вибираються на основі HTTP-методу, використовуваного в запиті.

На рис. 2 зображено Структура додатку, написаного за допомогою технології ASP.NET Web API.

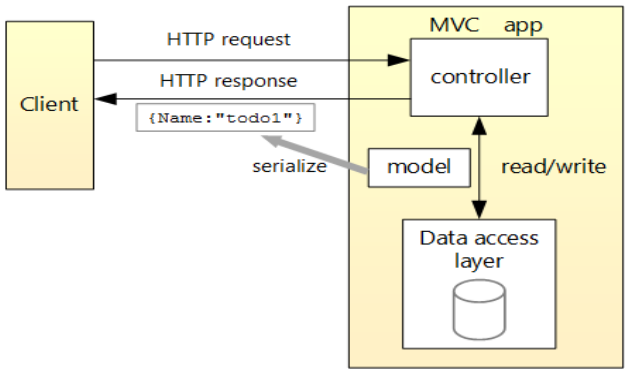

Рис. 2. Структура додатку, написаного за допомогою технології ASP.NET Web API

Кожен великий проект використовує паттерни для кращої структуризації коду та його підтримки в майбутньому.

Основні можливості програми показано на діаграмі прецедентів (рис. 3):

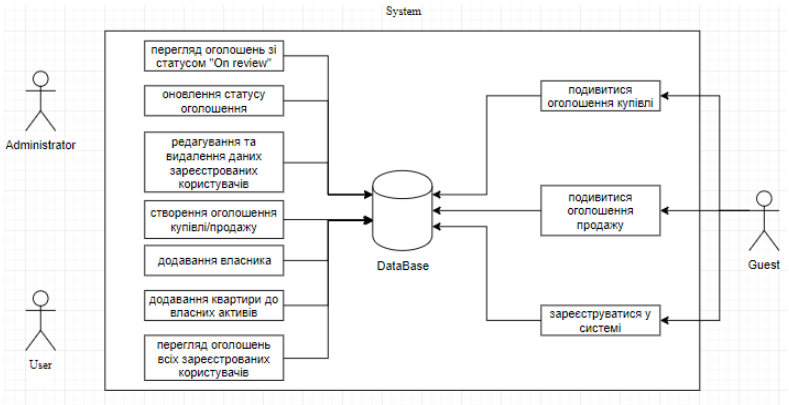

Рис. 3. Діаграма прецедентів

Для розробки системи використовувалась об’єктно-орієнтована мова програмування C# та наступні технології: .NET Framework, ASP.NET Core, EF, SSMS, Swagger. Для побудови запитів до БД обрано мову SQL. На рис. 4 наведено вигляд сайту для користувача.

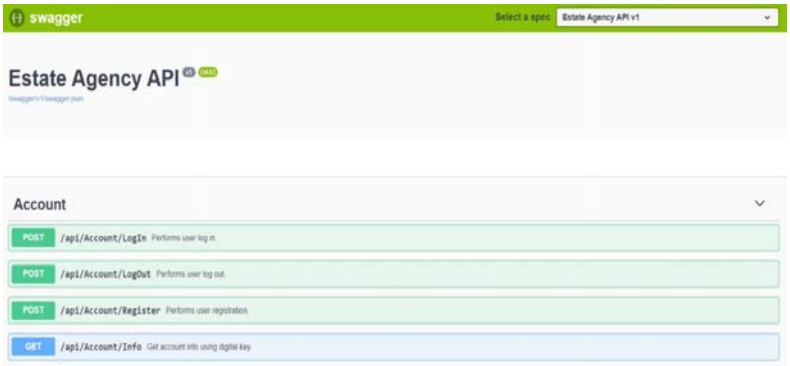

Рис. 4. Вигляд сайту

Сервер автентифікації Microsoft Identity Web – це набір бібліотеки ASP.NET Core, розширення підтримки авторизації веб-додатків та веб-API, інтегрованих з платформою Microsoft Identity. Він надає зручний високопродуктивний API рівня, зв'язуючий ASP.NET Core, за допомогою проміжного слова для перевірки справжності та бібліотеки перевірки підлинності Майкрософт (MSAL) для .NET.

Реалізоване запобігання SQL-ін’єкціям. SQL-ін’єкція є виконанням довільного запиту до бази даних додатка за допомогою поля форми або параметра URL. У разі використання стандартної мови Transact

SQL можливо вставити шкідливий код. Внаслідок чого будуть отримані, змінені або видалені дані таблиць. Щоб запобігти цьому, використайте запити, які параметризуються, які підтримуються більшістю мов веб-програмування.

Висновки. В результаті створено програмний продукт, веб-застосунок для оренди та продажу нерухомості із вбудованою системою захисту інформації. Архітектура проекту запобігає загрозі SQL-ін'єкцій. За допомогою системи ролей та доступів реалізовано модель білого списку. За допомого алгоритму шифрування Rijndael розроблено метод доступу до приватної інформації користувача.

Захист веб-додатків від зловмисників залежить від технологій і компонентів, що використовуються при створенні веб-додатків, а також відможливих вразливостей цих компонентів. Існують різні класифікації вразливостей, кожна атака через вразливість має свої особливості, а причиноювразливостей є помилки в розробці, реалізації та застосуванні компонентів веб– додатків, звідси необхідність пошуку і протидії вразливостям.

Список використаних джерел:

1. Галата Л. П., Корнієнко Б. Я., Заболотний В. В. Математична модель протидії загрозам у системі захисту критичних інформаційних ресурсів. *Наукоємні технології*. 2019. Том 43. № 3. С. 300–306.
2. Корнієнко Б. Я., Галата Л. П. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK. *Моделювання та інформаційні технології*. 2018. Вип. 83. С. 34–42.
3. Корнієнко Б. Я. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів. *Наукоємні технології*. 2017. № 4 (36). С. 316–322.
4. Корнієнко Б. Я., Юдін О. К., Снігур О. С. Безпека аутентифікації у web-ресурсах. *Захист інформації*. 2012. № 1 (54). С. 20–25. DOI: 10.18372/2410– 7840.14.2056 (ukr).
5. Корнієнко Б. Я., Максімов Ю. О., Марутовська Н. М. Прикладні програми управління інформаційними ризиками. *Захист інформації*. 2012. № 4 (57). С. 60–64. DOI: 10.18372/2410-7840.14.3493 (ukr).
6. Корнієнко Б. Я. Безпека інформаційно-комунікаційних систем та мереж. Навчальний посібник для студентів спеціальності 125 «Кібербезпека». К.: НАУ, 2018. 226 с.
7. Корнієнко Б. Я., Галата Л. П. Оптимізація системи захисту інформації корпоративної мережі. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2019. Випуск 19. С. 56–62.
8. Корнієнко Б. Я. Дослідження моделі взаємодії відкритих систем з погляду інформаційної безпеки. Наукоємні технології. 2012. № 3 (15). С. 83–89, doi.org/10.18372/2310– 5461.15.5120 (ukr).
9. Корнієнко Б. Я., Галата Л. П. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів. *Наукоємні технології*. 2017. № 4 (36). С. 316–322. doi.org/10.18372/2310-5461.36.12229.
10. Корнієнко Б. Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив :монографія. К.: Вид-во Аграр Медіа Груп. 2014. 288 с.
11. Корниенко Б. Я. Кибернетическая безопасность – операционные системы и протоколы. ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland. 2017. 122 P.
12. Корниенко Б. Я. Информационная безопасность и технологии компьютерных сетей: монография. ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland. 2016. 102 с.
13. Galata L., Korniyenko B., Yudin A. Research of the simulation polygon for the protection of critical information resources. CEUR Workshop Proceedings, Information Technologies and Security, Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017), 30 Nov 2017, Kyiv, Ukraine. vol. 2067. P. 23–31. urn:nbn:de:0074-2067-8.
14. Kornienko Y. M., Liubeka A. M., Sachok R. V., Korniyenko B. Y. Modeling of heat exchangement in fluidized bed with mechanical liquid distribution. *ARPN Journal of Engineering and Applied Sciences*. 2019. No14 (12). P. 2203–2210.
15. Korniyenko B. Modeling of information security system in computer network. *Безпека інформаційних систем і технологій*. 2019. Том №1 (1). С.36–41.
16. Korniyenko B., Galata L. Implementation of the information resources protection based on the CentOS operating system. Conference Proceedings of 2019 IEEE 98 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON -2019) July 2-6, 2019, Lviv, Ukraine. P. 1007–1011.
17. Korniyenko B., Yudin A., Galata L. Risk estimation of information system. // *Wschodnioeuropejskie Czasopismo Naukowe*. 2016. № 5. P. 35–40.
18. Korniyenko B., Galata L., Ladieva L. Research of Information Protection System of Corporate Network Based on GNS3. Conference Proceedings of 2019 IEEE Inernational Conference on Advanced Trends in Information Theory (IEEE ATIT-2019) Dezember 18-20, 2019, Kyiv, Ukraine. P. 244–248.
19. Korniyenko B., Galata L., Ladieva L. Mathematical model of threats resistance in the critical information resources protection system. CEUR Workshop Proceedings, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019) Kyiv, Ukraine, November 28, 2019. Vol-2577. P. 281–291.
20. Korniyenko B. Y., Galata L. P. Design and research of mathematical model for information security system in computer network. *Наукоємні технології*. 2017. № 2 (34). С. 114–118.
21. Korniyenko B., Galata L., Kozuberda O. Modeling of security and risk assessment in information and communication system // *Sciences of Europe*. 2016. V. 2. No 2 (2). P. 61–63.
22. Korniyenko B. The classification of information technologies and control systems // *International scientific journal*. 2016. № 2. P. 78–81.

23. Korniyenko B., Galata L., Ladieva L. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources. CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018) Kyiv, Ukraine, November 27, 2018. Vol-2318. P. 176–187. urn:nbn:de:0074-2318-4

24. Korniyenko B., Yudin O., Novizkij E. Open systems interconnection model investigation from the viewpoint of information security. *The Advanced Science Journal*. 2013. Issue 8. P. 53–56.

25. Korniyenko B., Ladieva L., Galata L. Control system for the production of mineral fertilizers in a granulator with a fluidized bed. 2020 2nd IEEE International Conference on Advanced Trends in Information Theory. 2020. No 9349344. P. 307–310.

26. Kornienko Y. M., Haidai S. S., Sachok R. V., Liubeka A. M., Korniyenko B. Y. Increasing of the heat and mass transfer processes efficiency with the application of non-uniform fluidization. *ARPN Journal of Engineering and Applied Sciences*. 2020. No 15(7). P. 890–900.

27. Korniyenko B., Kornienko Y., Haidai S., Liubeka A., Huliienko S. Conditions of Non-uniform Fluidization in an Autooscillating Mode. Advances in Computer Science for Engineering and Manufacturing. ISEM 2021 Lecture Notes in Networks and Systems. 2022. No 463. P. 14–27.

28. Korniyenko B., Kornienko Y., Haidai S., Liubeka A. The Heat Exchange in the Process of Granulation with Non-uniform Fluidization. Advances in Computer Science for Engineering and Manufacturing. ISEM 2021 Lecture Notes in Networks and Systems. 2022. No 463. P. 28–37.

29. Korniyenko B. Y. The two phase model of formation of mineral fertilizers in the fluidized–bed granulator. *The Advanced Science Journal*. 2013. № 4. P. 41–44.

30. Zhulynskyi A. A., Ladieva L. R., Korniyenko B. Y. Parametric identification of the process of contact membrane distillation. *ARPN Journal of Engineering and Applied Sciences*. Volume 14. Issue 17. September 2019. P. 3108–3112.