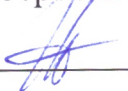


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра
на тему: «Методи та засоби захисту CMS OpenCart від
несанкціонованого доступу»

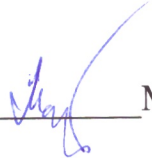
Виконав студент 2-го курсу, групи КБм-22-1
спеціальності 125 «Кібербезпека»

 Микола ЦИМБАЛЮК

Керівник доцент кафедри комп'ютерної інженерії та
кібербезпеки, кандидат технічних наук, доцент

 Юрій РОССІНСЬКИЙ

Рецензент завідувач кафедри комп'ютерних наук,
PhD

 Марина ГРАФ

Житомир – 2023

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»

ЗАТВЕРДЖУЮ

зав. кафедри комп'ютерної
інженерії та кібербезпеки

к.т.н., доцент

 Андрій ЄФІМЕНКО

«13» жовтня 2023 р.

ЗАВДАННЯ

випускню кваліфікаційну роботу магістра

Студент: Цимбалюк Микола Миколайович

Тема роботи: Методи та засоби захисту CMS OpenCart від несанкціонованого доступу

Затверджена Наказом університету від 13 жовтня 2023р. № 572/с

Термін здачі студентом закінченої роботи 22 грудня 2023 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження)

Мета роботи: розробити модуль для захисту адміністративної частини від злову шляхом включення додаткового параметру в рядок URL.

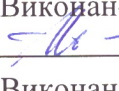
Об'єктом дослідження є безпека і захист веб-системи, побудованої на CMS OpenCart. Предмет дослідження – особливості розробки модулів для системи OpenCart з метою захисту адміністративної частини від несанкціонованого доступу (злову).

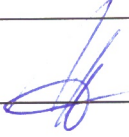
Консультанти кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Керівник	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

Керівник  Юрій РОССІНСЬКИЙ

Календарний план

№	Етап	Термін виконання етапу	Примітка
1	Визначення теми роботи, мети та завдань, об'єкта і предмета дослідження	01.10.2023 - 15.10.2023	Виконано 
2	Пошук та аналіз аналогічних модулів	15.10.2023 - 20.10.2023	Виконано 
3	Дослідження особливостей розробки модулів для OpenCart	20.10.2023 - 31.10.2023	Виконано 
4	Проектування та розробка модуля	01.11.2023 - 15.11.2023	Виконано 
5	Тестування модуля	16.11.2023 - 25.11.2023	Виконано 
6	Узагальнення і систематизація результатів роботи. Написання пояснювальної записки	25.11.2023 - 22.12.2023	Виконано 
7	Захист		

Студент  Микола ЦИМБАЛЮК

Керівник  Юрій РОССІНСЬКИЙ

РЕФЕРАТ

Темою кваліфікаційної роботи магістра є методи та засоби захисту CMS OpenCart від несанкціонованого доступу. Робота включає в себе програмний код розробленого модуля та пояснювальну записку обсягом 62 сторінки. Записка складається зі вступу, трьох розділів та висновків, а також містить список з 16 використаних джерел, 16 рисунків та 3 таблиць.

Метою роботи є розробка модуля для захисту адміністративної частини від злову шляхом включення додаткового параметру в рядок URL.

При написанні роботи було здійснено дослідження засобів та шляхів для безпеки та захисту веб-магазину від злову сторонніми особами. Серед ряду інших заходів запропоновано використати додатковий параметр (секрет) в URL для входу в панель адміністратора. Для досягнення такої цілі автором розроблено модуль, який встановлюється та активується в системі та додає необхідні модифікації в програмний код.

Ключові слова: OpenCart, безпека, захист від злову, модулі.

ABSTRACT

The theme of the master degree work is a methods and means to protect CMS OpenCart from unauthorized access. The work includes the program code of the developed module and an explanatory note of 62 pages. The note consists of an introduction, three chapters and conclusions, and also contains a list of 16 used sources, 16 figures and 3 tables.

The aim of the work is to develop a module to protect the administrative part from hacking by including an additional parameter in the URL string.

During the writing of the work, a study of means and ways for the security and protection of the web store from hacking by third parties was carried out. Among a number of other measures, it is proposed to use an additional parameter (secret) in the URL for entering the admin panel. To achieve this goal, the author has developed a module that is installed and activated in the system and adds the necessary modifications to the program code.

Keywords: OpenCart, security, anti-hacking, modules.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ МОДУЛІВ ДЛЯ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В КОНТЕКСТІ БЕЗПЕКИ ТА ЗАХИСТУ ТАКИХ СИСТЕМ.....	10
1.1. Визначення рекомендацій щодо захисту систем для здійснення електронної комерції. Постановка задачі.....	10
1.2. Огляд наявних аналогічних рішень.....	12
1.3. Обґрунтування вибору системи ОСМОД для розробки модуля	15
1.4. Вибір мови та засобів розробки	16
Висновки до розділу 1	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ МОДУЛЯ ДЛЯ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДЛЯ СИСТЕМИ OPENCART.....	18
2.1. Варіанти використання модуля.....	18
2.2. Проектування UML-діаграм діяльності, послідовностей та класів модуля.....	19
2.3. Проектування елементів візуального інтерфейсу та процес розробки модуля	23
Висновки до розділу 2	34
РОЗДІЛ 3. ПОРЯДОК ВСТАНОВЛЕННЯ, ВИДАЛЕННЯ ТА КОРИСТУВАННЯ МОДУЛЕМ. ТЕСТУВАННЯ МОДУЛЯ.....	35
3.1. Порядок встановлення системи OpenCart та модуля AdminSecure	35
3.2. Порядок використання модуля захисту адміністративної панелі від злому.....	43
3.3. Видалення, ввімкнення та вимкнення модуля	47
3.4. Тестування роботи модуля	48
Висновки до розділу 3	52
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CMS – Content Management System – Система керування контентом

SEO – Search Engine Optimization – Оптимізація пошукового рушія

B2B - Business To Business

ВСТУП

Актуальність теми дослідження. Актуальність захисту систем електронної комерції від злому залишається критичною у сучасному цифровому світі. Збільшення обсягів онлайн-торгівлі і зростання кількості транзакцій через Інтернет призводять до того, що інформація про користувачів, фінансові дані і інша конфіденційна інформація стають предметом інтересу для зловмисників.

Основні аспекти актуальності захисту систем електронної комерції включають:

Конфіденційність даних. Користувачі передають особисті дані та інформацію про кредитні картки під час онлайн-покупок. Недостатній рівень захисту може призвести до витоку конфіденційної інформації та порушити довіру споживачів.

Захист від кібератак. Зловмисники можуть використовувати різноманітні техніки, такі як SQL-ін'єкції, кросс-сайт-скриптинг (XSS), фішінг та інші для отримання доступу до систем електронної комерції. Ефективний захист від кібератак важливий для уникнення втрат даних та репутаційних проблем.

Забезпечення надійності транзакцій. Електронна комерція пов'язана з обробкою фінансових транзакцій. Забезпечення цілісності та надійності цих транзакцій є ключовим, щоб уникнути фінансових втрат та зберегти довіру клієнтів.

Дотримання вимог законодавства. Багато країн мають строгі вимоги до захисту особистих даних, особливо в галузі електронної комерції. Забезпечення відповідності цим вимогам стає важливою задачею для підприємств, щоб уникнути правових проблем і штрафів.

Розвиток технологій. Зловмисники постійно вдосконалюють свої методи, тому захист систем електронної комерції повинен здати випробування на стійкість до новітніх кіберзагроз та використовувати сучасні технології для виявлення і запобігання атакам.

Зміна природи загроз. Кіберзлочинці постійно адаптуються до нових умов та використовують соціальні інженерні методи, тому важливо не лише технічно захищати системи, але і навчати користувачів виявляти шахраїв.

Загальний підхід до захисту систем електронної комерції повинен включати в себе комбінацію технічних, організаційних та психологічних заходів для створення надійного захисного механізму.

CMS OpenCart є спеціалізованою системою для ведення комерції онлайн. Як і багато інших CMS, її можна назвати відносно захищеною платформою. Проте при її розгортанні і подальшому використанню необхідно відразу подбати про безпеку та захист свого сайту від злому сторонніми особами.

Метою роботи є розробка модуля-модифікатора для захисту адміністративної частини CMS OpenCart від злому шляхом включення додаткового параметру в рядок URL.

Завданнями роботи є:

- Постановка основної задачі щодо розробки
- Дослідження наявних рішень – модулів-аналогів
- Вивчення вимог та правил розробки модулів-модифікаторів для системи OpenCart
- Проектування модуля: розробки UML-діаграм, схеми бази даних
- Розробка модуля-модифікатора
- Розробка інструкції користувача
- Тестування системи

Об'єктом дослідження є безпека і захист веб-системи, побудованої на CMS OpenCart.

Предмет дослідження – особливості розробки модулів для системи OpenCart з метою захисту адміністративної частини від несанкціонованого доступу (злому).

РОЗДІЛ 1. АНАЛІЗ ЗАСОБІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ МОДУЛІВ ДЛЯ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В КОНТЕКСТІ БЕЗПЕКИ ТА ЗАХИСТУ ТАКИХ СИСТЕМ

1.1. Визначення рекомендацій щодо захисту систем для здійснення електронної комерції. Постановка задачі

Визначимо ряд рекомендацій щодо захисту веб-системи від злому. В першу чергу дані рекомендації стосуються OpenCart, проте можуть бути узагальнені і для інших CMS.

1) Приховання входу до адміністративної панелі.

За замовчування для входу до панелі адміністратора зазвичай використовується шлях: `http://{domain_name}/admin`. Знання шляху для хакерів до адміністративної панелі значно полегшує процес злому. Тому перша рекомендація – змінити адресу входу в адміністративну панель з `/admin` на іншу: `/manager`, `/panel` або щось складніше.

Для цього потрібно змінити назву папки “admin” в корені проекту на іншу назву, зробити таку ж заміну у файлі `config.php` всередині папки, яку ми перейменували; інколи зміни потрібно вносити у файл `config.php` у кореневій папці, якщо там зустрічається шлях `/admin`. Якщо все зроблено правильно, то адміністративна панель буде доступна за новим шляхом.

Цей варіант має певний недолік, який потрібно усувати. Якщо зайти за шляхом `domain.com/system/logs/error.log`, то в журналах можна прочитати шлях до адміністративної панелі. Тому до таких файлів необхідно забороняти доступ. Для цього в корені сайту відкрити `.htaccess` і додати блок:

```
# Prevent Direct Access to files
<FilesMatch "\.(tpl|ini|log)">
    Order deny,allow
    Deny from all
</FilesMatch>
```

Ще одним недоліком є те, що перейменування папки адміністратора може спричинити проблеми під час встановлення розширень OpenCart.

2) Зміна логіна і паролю адміністратора.

Після зміни адреси для входу в панель рекомендується змінити логін, який за замовчуванням має значення admin. Зазначимо, що це взагалі основний логін, який зазвичай використовується на багатьох CMS, тому варто його змінити на інший.

Для цього необхідно зайти в панель адміністратора, вибрати послідовно пункти «Система», «Користувачі» та знову «Користувачі». Біля рядка "admin" необхідно зайти в налаштування та поміняти логін на інший.

Аналогічно можна змінити пароль. Рекомендована мінімальна довжина паролю – десять символів. При цьому можна скористатись одним із онлайн-сервісів з генерації паролів.

Важливо не використовувати паролі, що повторюються. Кожен пароль повинен бути індивідуальним. Не можна використовувати однакові паролі для входу в панель адміністратора та, наприклад, для входу в пошту.

3) Зміна прав доступу для важливих файлів

Файли конфігурацій config.php у кореневій папці та config.php у папці admin (назва якої змінювалася вище) містять важливу інформацію, пов'язану обліковими даними для з'єднання з БД, тому рекомендується змінити права доступу для цих файлів на «лише читання».

Змінити права можна за допомогою будь-якого інструменту для роботи з файлами: наприклад, безпосередньо через “Файловий менеджер” на панелі управління хостингом. В дозволах задаємо значення 444 (тільки читання для власника і для решти). Інший варіант – скористатися FTP-клієнтом Filezilla для встановлення прав доступу.

4) Скасування виводу помилок

Як правило, при зламі хакери використовують різні способи, серед яких дозвіл на повідомлення про помилки, що виводяться в результаті неправильних дій. Тому рекомендується скасувати вивід помилок для робочої версії системи.

Для цього необхідно зайти в панель адміністратора, вибрати послідовно пункти "Система", "Налаштування", вкладка "Сервер". Біля блоку "Помилки" необхідно поставити "Ні" у параметрі "Показувати помилки". Для перегляд помилок в цьому випадку можна використовувати файл журналу помилок (його назва є в цьому ж блоці в налаштуваннях). Шлях до файлу з помилками: /system/logs.

5) Використання модуля для захисту панелі адміністратора. Сформулюємо основну задачу розробки даного модуля.

Постановка задачі. Даний модуль повинен забезпечувати захист панелі адміністратора веб-сайту від злому. Хакери, знаючи назву папки адміністратора, можуть використати метод грубої сили для злому. Модуль передбачає неможливість доступу до панелі адміністратора, якщо користувач не знає ключ аутентифікації та секретний ключ. Для отримання доступу, користувач вводить дані ключі в URL-адресі (наприклад: `https://{domain_name}/admin?your_auth_key=your_secret_key`). Невірні значення ключів унеможливають доступ до панелі адміністратора, і запит перенаправляється на домашню сторінку магазину. Даний модуль повинен бути простим в установці та використанні. Ключ повинен зберігатись в базі даних, і якщо користувач забуває ключі, їх можна легко відновити.

1.2. Огляд наявних аналогічних рішень

При пошуку наявних аналогічних рішень автором було знайдено три схожих по функціональності модулі:

- Smart Admin Security
- Secure Admin URL

- Store admin protection module for Opencart

Зробимо порівняння їх функціональних можливостей.

Smart Admin Security

https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=42842

Даний модуль забезпечує можливість захистити панель адміністратора веб-сайту. При використанні сторінка входу адміністратора не може бути доступна користувачу, якщо він не має ключ авторизації та секретний ключ. Для отримання доступу до адміністратора, користувач має ввести ключ авторизації та секретний ключ у URL-адресі (наприклад: www.yourstore.com/admin?your_auth_key=your_secret_key). При неправильному введенні ключів, запит буде перенаправлено на домашню сторінку веб-сайту. Модуль простий в установці та використанні. Ключ зберігається у базі даних з можливістю його відновлення.

Secure Admin URL (OpenCart 3.0.)*

https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=40693

Модуль для встановлення ключа авторизації та додаткового значення для захисту URL-адреси адміністратора від несанкціонованого входу. Якщо встановити лише ключ, то URL-адреса адміністратора буде змінена на формат: <http://yourdomain.com/admin/?key>. При застосуванні додаткового значення, URL-адреса адміністратора буде змінена на формат: <http://yourdomain.com/admin/?key=value>. Для налаштування необхідно перейти на вкладку «Налаштування — сервер», заповнити ключ і при бажанні додаткове значення.

Store admin protection module for Opencart

<https://ocmod.net/en/modul-zaschita-adminki>

Модуль захисту адмінпанелі магазину Opencart 2 і 3, що дозволяє захистити панель адміністратора від підбору пароля методом “грубої сили” і приховати

сторінку входу адміністратора від небажаних гостей. Модуль дозволяє змінити URL (шлях) до адмінпанелі Opencart, при переході на звичайну адресу доступу до адмінпанелі користувач перенаправляється на головну сторінку. Налаштування модуля здійснюються на вкладці “Сервер”. Необхідно заповнити поля: ключ та значення ключа (необов’язкове).

В таблиці 1.1 наведено типові функції даних модулів:

Таблиця 1.1

Порівняння функціональних можливостей аналогічних модулів для захисту адмінпанелі OpenCart

	Smart Admin Security	Secure Admin URL	Store admin protection module
Інсталяція/деінсталяція	+	+	+
Увімкнення/вимкнення	+	+	+
Аутентифікація лише через auth_key	-	+	+
Аутентифікація через auth_key та value	+	+	+
Встановлення auth_key та value через форму в адмінці	+	+	+

Здійснивши порівняння даних рішень, ми виявили типові функціональні можливості таких модулів, які пропонується реалізувати в нашому модулі.

1.3. Обґрунтування вибору системи OSMOD для розробки модуля

OSMod (OpenCart Modification) - це система, яка використовується в OpenCart для внесення змін або модифікацій до основних файлів платформи, не змінюючи їх безпосередньо. Замість безпосередньої модифікації основних файлів, OSMod використовує XML-файли для визначення змін, що полегшує управління та оновлення інсталяцій OpenCart. Наведемо декілька тез для обґрунтування використання OSMod для створення розширень для OpenCart:

Збереження вихідного коду.

OSMod дозволяє вносити зміни без безпосередньої модифікації основних файлів OpenCart. Це важливо, оскільки зміни в основних файлах можуть ускладнити майбутні оновлення та апдейти. OSMod допомагає зберегти цілісність системи OpenCart.

Зручність у встановленні та видаленні.

Розширення, розроблені за допомогою OSMod, легко встановлюються та видаляються. Зміни застосовуються динамічно без потреби ручних редагувань файлів. Це спрощує процес додавання чи видалення розширень і зменшує ризик помилок.

Сумісність з оновленнями.

OpenCart регулярно випускає оновлення та патчі для підвищення безпеки, виправлення помилок та введення нових функцій. Модифікація основних файлів може призвести до конфліктів під час процесу оновлення. OSMod гарантує, що зміни відокремлені від основних файлів. Це полегшує адаптацію до нових релізів OpenCart без втрати модифікацій.

Зручність в обслуговуванні.

OSMod спрощує обслуговування інсталяцій OpenCart. Коли розширення розробляється з використанням OSMod, його зручніше оновлювати окремо від основних файлів OpenCart, що підвищує загальну стабільність рушія.

Стандарти спільноти.

Використання OSCMod відповідає кращим практикам та стандартам спільноти. Багато розробників та постачальників розширень дотримуються цього підходу, щоб забезпечити сумісність розширень з різними інсталяціями OpenCart та зменшити ймовірність конфліктів з іншими розширеннями.

Співпраця з іншими розробниками.

При роботі в команді та співпраці використання OSCMod полегшує спільне використання та поширення модулів. Формат на основі XML надає чіткий та стандартизований спосіб опису модифікацій, що полегшує розуміння та участь інших у проекті.

Відстеження та відлагодження помилок.

OSCMod дозволяє краще відстежувати та відлагоджувати помилки. Оскільки зміни визначаються у XML-файлах, легше визначити та виправити проблеми. Це може значно заощадити час при пошуку та усуненні проблем, пов'язаних з розширеннями чи модифікаціями.

Таким чином OSCMod надає структурований та стандартизований підхід до внесення змін в OpenCart, пропонуючи переваги, такі як простота встановлення, сумісність з оновленнями, зручне обслуговування та відповідність стандартам спільноти. Ці переваги роблять OSCMod рекомендованим методом для розробки та керування розширеннями OpenCart.

1.4. Вибір мови та засобів розробки

Зазначимо передумови, необхідні для локальної розробки модуля:

- Apache Web Server – програмне забезпечення HTTP-сервера з відкритим вихідним кодом, яке широко використовується для надання веб-вмісту в Інтернеті та обробки запитів за клієнт-серверною архітектурою;
- MySQL Database Server – відкритий програмний продукт, який використовується для зберігання та управління реляційними базами даних;

- PHP (PHP 7.4) – мова програмування загального використання, яка використовується для розробки динамічних веб-сайтів і взаємодії з базами даних;
- OpenCart 3.0.3.x – відкрите програмне забезпечення для створення і управління онлайн-магазинами, що надає широкий функціонал для електронної комерції;
- IDE JetBrains PhpStorm – інтегроване середовище розробки (IDE) для мови програмування PHP, яке надає розширені можливості редагування коду, відлагодження та інші інструменти для ефективної розробки PHP-програм.
- Git – розподілена система керування версіями, призначена для відстеження змін у програмному коді під час спільної роботи розробників та керування версіями програмного продукту.

Висновки до розділу 1

В даному розділі автором розглянуто рекомендації щодо захисту систем для здійснення електронної комерції та сформульовано основну задачу роботи: необхідно розробити легкий в керуванні модуль для забезпечення додаткового захисту панелі адміністратора веб-сайту від злому методом грубої сили, шляхом додавання до URL-адреси адмін-панелі додаткового параметру як ключа доступу, з можливістю його встановлення в формі налаштувань системи.

Було виявлено три аналогічних рішення та проведено порівняння їх функціональних можливостей, в результаті чого виявлено типові функціональні можливості таких модулів, які пропонується реалізувати в нашому модулі. Обґрунтовано вибір ОСMOD для розробки модуля, мови та необхідного стеку програмного забезпечення для розробки модуля.

РОЗДІЛ 2. ПРОЕКТУВАННЯ МОДУЛЯ ДЛЯ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДЛЯ СИСТЕМИ OPENCART

2.1. Варіанти використання модуля

Модуль захисту адміністративної панелі CMS OpenCart від злому повинен відповідати загальним вимогам, які ставляться перед модулями-модифікаторами, такими як можливість встановлення, видалення, увімкнення та вимкнення, а також повинен виконувати функції, визначені в завданні:

Користувач з роллю адміністратора:

1. Встановлює та видаляє модуль;
2. Вмикає і вимикає модуль;
3. Встановлює обов'язковий секретний ключ доступу та його значення (необов'язково);
4. Здійснює вхід в панель керування з урахуванням наявності секретного ключа доступу та його значення;

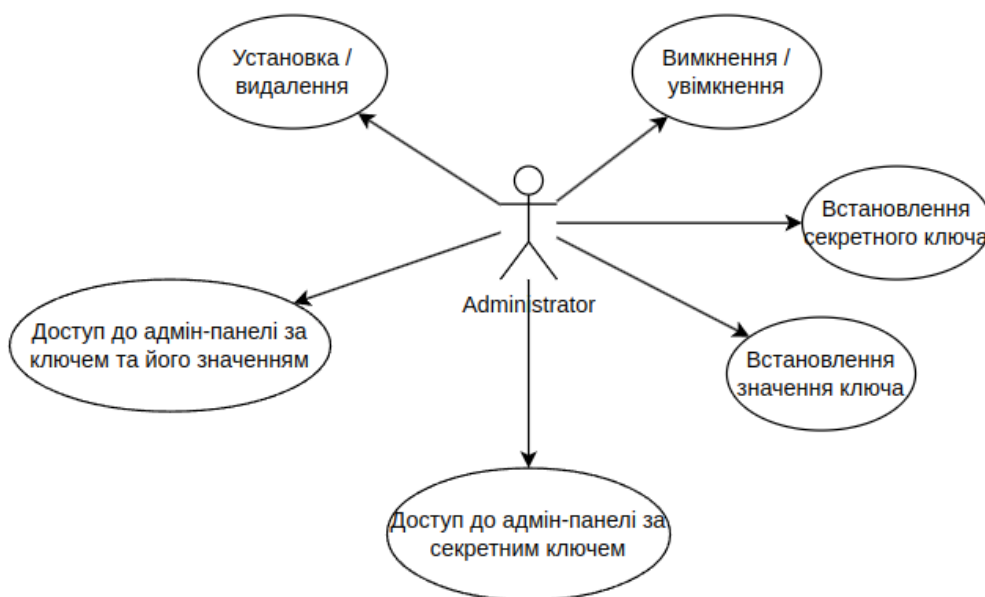


Рисунок 2.1. Use Case діаграма модуля захисту від несанкціонованого доступу для OpenCart-системи

2.2. Проектування UML-діаграм діяльності, послідовностей та класів модуля

Діаграма активності (activity diagram) - UML-діаграма, що ілюструє графічне зображення процесів та взаємодії між об'єктами в системі, яке використовується для моделювання послідовностей дій та активностей у програмному чи бізнес-проекті.

Основні компоненти діаграми активності включають:

- Активності (Activities): Представляють конкретні дії або обробки, які виконуються в системі.
- Дії (Actions): Маленькі блоки, які вказують на конкретні операції чи обчислення, що відбуваються в рамках активностей.
- Вузли рішення (Decision Nodes): Використовуються для представлення точок прийняття рішень, де в залежності від певного умови вибирається одна із кількох альтернативних гілок.
- Вузли об'єднання (Merge Nodes): Вказують на об'єднання різних гілок виконання після прийняття рішення.
- Форки (Fork Nodes) та Джойни (Join Nodes): Використовуються для паралельного виконання декількох активностей та їх об'єднання після завершення.
- Об'єкти (Objects): Представляють об'єкти, над якими виконуються активності.
- Пільги (Swimlanes): Використовуються для розділення діаграми на різні області (лінії життя), які можуть вказувати на різні ролі чи відділи у системі.
- Фрейми (Frames): Грукують діаграму активностей, надаючи структуру та вказуючи на контекст виконання дій.
- Активаційні стрілки (Activation Edges): Вказують на порядок виконання активностей та передачу управління між ними.

Ці компоненти взаємодіють між собою, утворюючи зрозумілу та структуровану діаграму для моделювання процесів та взаємодії в системі.

Спроекуємо діаграму діяльностей модуля захисту адміністративної панелі від злому (рис. 2.2).

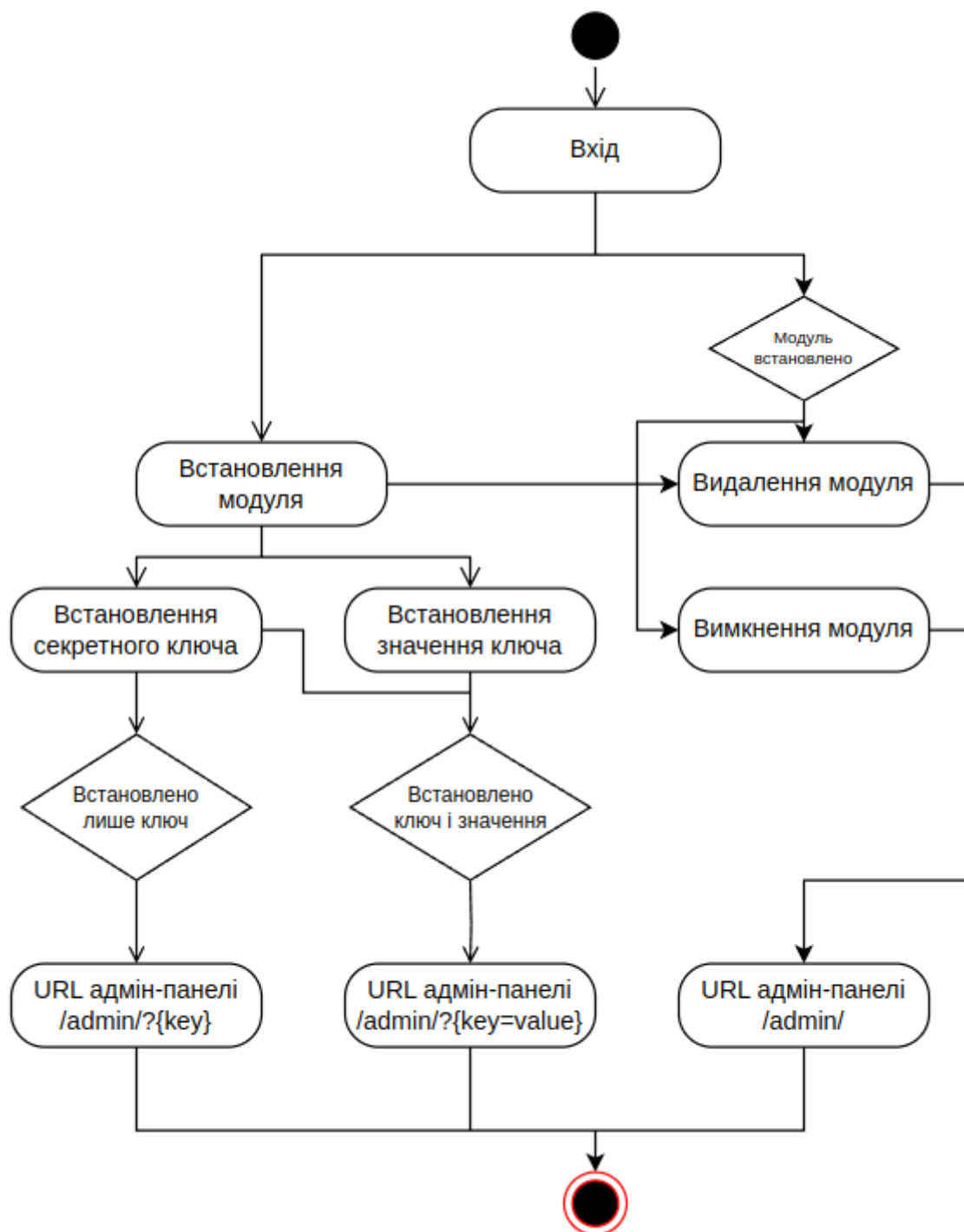


Рисунок 2.2 - Діаграма активностей модуля захисту адміністративної панелі
OpenCart від злому

Після здійснення входу в систему, адміністратор має можливість встановити модуль, а також в випадку його наявності – видалити, вимкнути та увімкнути. Якщо модуль встановлено та увімкнено, в налаштуваннях системи адміністратор встановлює секретний ключ та його значення. У випадку вимкненого модуля або його видалення налаштування по встановленню секретного ключа та його значення повинні бути відсутні. Якщо встановлено лише ключ, доступ до адмін-панелі здійснюється за посиланням `/admin/?auth_key`. Якщо встановлено ключ та його значення, URL-адреса виглядає так: `/admin/?auth_key=value`. Якщо ключ не встановлено, або модуль деактивовано чи видалено, URL-адреса має вигляд по замовчуванню: `/admin/`.

Наступною UML-діаграмою є діаграма послідовностей (Sequence Diagram) – діаграма, що використовується для моделювання взаємодій між об'єктами в системі в хронологічному порядку. Вона дозволяє нам візуалізувати та аналізувати послідовність обміну повідомленнями між об'єктами в рамках конкретного сценарію взаємодії.

Основні елементи діаграми послідовностей включають:

- Об'єкти (актори): Представлені прямокутниками з ім'ям об'єкта або ім'ям класу, якщо це клас.
- Лінії життя (Lifelines): Вертикальні лінії, які представляють існування об'єкта протягом певного періоду часу.
- Повідомлення: Представляються стрілками, що вказують напрямок обміну повідомленнями між об'єктами. Повідомлення можуть бути синхронними (виклик та очікування відповіді) або асинхронними (відправка без очікування відповіді).
- Активаційні бари (Activation Bars): Показують часовий інтервал, протягом якого об'єкт є активним та виконує певні дії.
- Опціональні та ітераційні конструкції: Можуть використовуватися для позначення умов або циклів в послідовності обміну повідомленнями.

Діаграми послідовностей особливо корисні для моделювання взаємодій між об'єктами в системі на рівні функціональності та для визначення порядку виконання дій в рамках конкретних сценаріїв взаємодії.

На рисунку 2.3 зображена спроектована діаграма послідовностей.

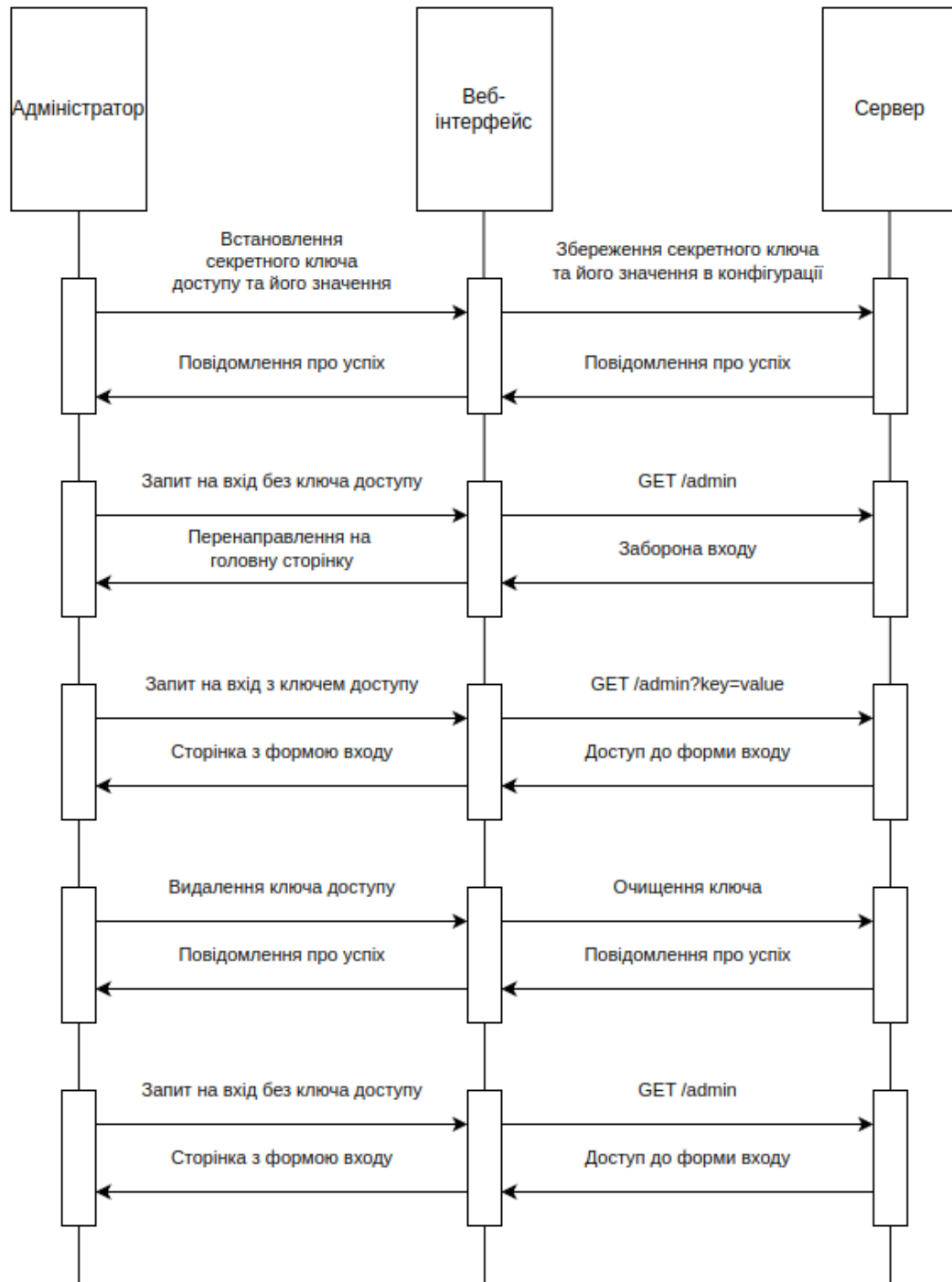


Рисунок 2.3 - Діаграма послідовностей модуля захисту адміністративної панелі OpenCart від злому

В нашій діаграмі активностей присутні такі актори: адміністратор (користувач, що володіє доступом до адмін-панелі), веб-інтерфейс (клієнтські компоненти керування на веб-сторінках) та система (серверні компоненти, що відповідають за обробку запитів та даних).

На діаграмі послідовно зображені цикли, що ілюструють обмін повідомленнями між трьома акторами. В першому циклі адміністратор встановлює налаштування ключа доступу та його значення. В другому циклі ілюструється запит на вхід через URL без параметра ключа доступу, при цьому сервер повинен заборонити вхід, перенаправивши запит на головну сторінку сайту. Третій цикл демонструє вхід з правильним URL (/admin?key або /admin?key=value). При видаленні (очищенні) ключа доступу, вхід повинен здійснюватись з URL по замовчуванню (/admin), що демонструють два останні цикли діаграми послідовностей.

2.3. Проектування елементів візуального інтерфейсу та процес розробки модуля

Одним із головних елементів керування є форма налаштувань системи OpenCart, яка доступна за шляхом System – Settings. Після встановлення модуля пропонується додати два елементи у групу (вкладку) Server даної форми шляхом правил модифікації, визначених інструментом OSMOD. Форма з додаванням полів auth_key, auth_key_value зображена на рисунку 2.4.

Процес розробки модуля враховує правила, визначені засобом OSMOD. Модифікатор OSMOD може бути як самостійним модулем, що змінює або розширює стандартний функціонал, так і входити до складу повноцінного модуля з додатковими php, tpl або twig та іншими файлами.



Рисунок 2.4 - Форма налаштувань з додаванням полів `auth_key` та `auth_key_value` шляхом модифікатора OCMoD

Назва архіву модифікатора - повинна бути задана відповідно до вимог OCMOD - "назва_файлу".osmod.zip. Таким чином, назва архіву завжди повинна закінчуватись ".osmod.zip", інакше система проігнорує модифікатор.

Починаючи з версії OpenCart 3x передбачається, що модифікатор буде використовуватися у складі повного модуля у вигляді архіву. Відповідно, при використанні файлу модифікатора у складі повного модуля та при його завантаженні через адміністративну частину в розділі Розширення > Установка розширень назва файлу-модифікатора в архіві має бути `install.xml`.

В OpenCart 3x файл-модифікатор також можна завантажити в папку system на постійне розміщення, використовуючи клієнт FTP, за умови, що назва файлу модифікатора буде відповідати вимогам OSMOD у вигляді "назва_файлу".osmod.xml.

Приклад структури файлів для архіву OSMOD:

- Папка upload. Усі файли в цьому каталозі буде завантажено в каталог OpenCart.
- install.xml – файл модифікації XML. Даний файл, що створює віртуальну копію будь-яких файлів, які потребують змін. Дана система використовується для перезапису будь-яких інсталяційних файлів за замовчуванням. До одного файлу можна застосувати декілька змін.

Приклад XML-файлу OSMOD:

```
<?xml version="1.0" encoding="utf-8"?>
<modification>
  <name>Modification Default</name>
  <version>1.0</version>
  <author>OpenCart Ltd</author>
  <link>http://www.opencart.com</link>
  <file path="catalog/controller/common/home.php">
    <operation>
      <search><![CDATA[
        $data['column_left'] = $this->load-
      >controller('common/column_left');
      ]]></search>
      <add position="replace"><![CDATA[
        test123
      ]]></add>
    </operation>
  </file>
</modification>
```

Лістинг 2.1 - Приклад файлу install.xml файлу OSMOD

Теги, що використовується в install.xml:

`<file>...</file>` – вказує шлях до файлу, що модифікується. Наприклад,
`<file path="admin/controller/common/column_left.php">`

Використання фігурних дужок дозволяє вибрати кілька файлів і не повторювати операцію з кодом кілька разів:

```
<file  
path="system/engine/action.php|system/engine/loader.php|system/library/co  
nfig.php|system/library/language.php">
```

Звернемо увагу, що всі шляхи до файлів мають починатися з admin, catalog або system. Також можна використовувати символ підстановки для пошуку кількох каталогів і файлів.

`<search>...</search>` – знаходить код у файлі. Дозволені атрибути тегу
`<search>`

```
trim="(true|false) "  
regex="(true|false) "  
index="(number) "
```

`<add>...</add>` – додає код в даний файл. Дозволені атрибути:

```
trim="(true|false) "  
position="(replace|before|after) "  
offset="(number) "
```

Перейдемо безпосередньо до розробки. Визначимо файли та операції по змінах в них:

1. admin/language/en-gb/setting/setting.php – словниковий файл для відображення текстових елементів інтерфейсу на відповідній мові:
 - Додавання текстових написів для даних auth_key, auth_key_value, help_auth_key, help_auth_key_value
2. admin/view/template/setting/setting.twig – шаблон форми налаштувань:
 - Додавання в форму налаштувань полів auth_key та auth_key_value
3. admin/controller/setting/setting.php – контролер для роботи з налаштуваннями:
 - POST: підготовка полів до збереження як налаштування

- GET: отримання значень полів з наявних налаштувань
4. `admin/controller/common/login.php` – контролер форми входу:
- Формування правильного рядка запиту залежно від встановлення ключа та його значення
 - Формування змінної `$data_action` залежно від встановлення ключа та його значення

Етапи розробки даних блоків наведено на рис. 2.5:

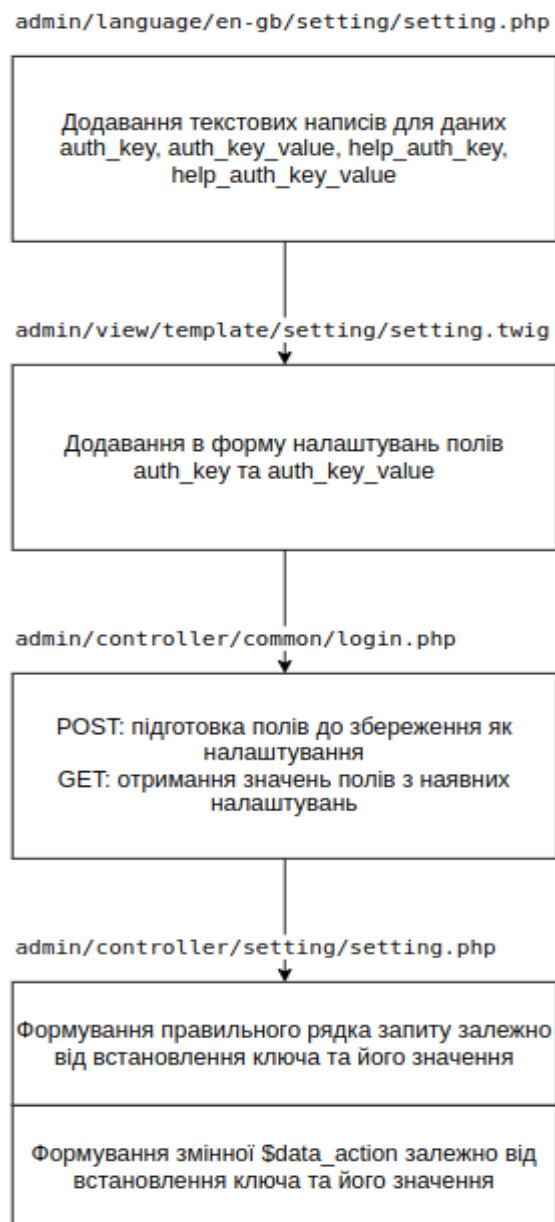


Рисунок 2.5 - Послідовність операцій для внесення змін в файли адміністративної частини

Розробка програмного коду модуля.

Крок 1. Створимо новий проект та в ньому створимо папку `adminsecure.ocmod`. В папці модуля створимо папку *upload* та файл *install.xml*.

Крок 2. Створимо базову структуру файлу *install.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<modification>
    <name>AdminSecureFromHacking</name>
    <code>admin-secure-from-hacking</code>
    <version>1.0.0</version>
    <author>@authorName</author>
    <link>https://opencart.com</link>
    <file path="admin/language/en-gb/setting/setting.php">

    </file>
    <file path="admin/view/template/setting/setting.twig">

    </file>
    <file path="admin/controller/setting/setting.php">

    </file>
    <file path="admin/controller/common/login.php">

    </file>
</modification>
```

Лістинг 2.2 - Структура файлу *install.xml* файлу OCMOD

Крок 3. Першою модифікацією є словниковий файл `admin/language/en-gb/setting/setting.php`. До даного файлу додано 4 нові текстові змінні:

- `entry_auth_key` – напис для поля встановлення секретного ключа доступу;
- `entry_auth_key_value` – напис для поля встановлення додаткового значення секретного ключа доступу;
- `help_auth_key` – довідковий напис для поля секретного ключа доступу;

- `help_auth_key_value` – довідковий напис для поля додаткового значення ключа доступу.

В наступному лістингу наведено код модифікації словникового файлу:

```
<file path="admin/language/en-gb/setting/setting.php">
    <operation>
        <search>
            <![CDATA[// Entry]]>
        </search>
        <add position="after">
            <![CDATA[
$_['entry_auth_key'] = 'Admin Secure Auth Key';
$_['entry_auth_key_value'] = 'Admin Secure Auth Key Value';
$_['help_auth_key'] = 'Set Secure Auth Key to protect your Admin
URL preventing unauthorized entry.<br /><br /><b><i> CAUTION !!
</b></i><br /> <br />Your Admin URL will be changed to this format:
http://yourdomain.com/admin/?key';
$_['help_auth_key_value'] = 'Set the Additional Value to protect
your Admin URL preventing unauthorized entry.<br /><br /><b><i>
CAUTION !! </b></i><br /> <br />Your Admin URL will be changed to
this format: http://yourdomain.com/admin/?key=value';
]]>
        </add>
    </operation>
</file>
```

Лістинг 2.3 - Модифікація словникового файлу admin/language/en-gb/setting/setting.php

Крок 4. Модифікація файлу шаблону admin/view/template/setting/setting.twig.

В даному файлі ми створили два поля:

- `config_auth_key[auth_key]` – параметр для введення та редагування секретного ключа доступу адміністратора;
- `config_auth_key[auth_key_value]` – параметр для введення та редагування додаткового значення ключа доступу адміністратора.

В наступному лістингу наведено код модифікації файлу шаблону форми:

```
<file path="admin/view/template/setting/setting.twig">
    <operation>
        <search>
            <![CDATA[<label class="col-sm-2 control-label"><span
data-toggle="tooltip"      title="{{      help_maintenance      }}">{{
entry_maintenance }}</span></label>]]>
        </search>
        <add position="before" offset="1">
            <![CDATA[
<div class="form-group">
                <label          class="col-sm-2          control-label"
for="admin_auth_key">
                    <span          data-toggle="tooltip"          title="{{
help_auth_key }}">{{ entry_auth_key }}</span>
                </label>
                <div class="col-sm-10">
                    <input          type="text"
name="config_auth_key[auth_key]"          value="{%          if
config_auth_key.auth_key %} {{ config_auth_key.auth_key }} {% endif
%}" id="admin_auth_key" class="form-control" />
                </div>
            </div>
            <div class="form-group">
                <label          class="col-sm-2          control-label"
for="admin_auth_key_value">
                    <span          data-toggle="tooltip"          title="{{
help_auth_key_value }}">{{ entry_auth_key_value }}</span>
                </label>
                <div class="col-sm-10">
                    <input          type="text"
name="config_auth_key[auth_key_value]"          value="{%          if
```

```

config_auth_key.auth_key_value %) {{ config_auth_key.auth_key_value
}} {% endif %}" id="admin_auth_key_value" class="form-control" />
        </div>
    </div>

]]>

</add>

</operation>

</file>

```

Лістинг 2.4 - Модифікація файлу шаблону налаштувань

admin/view/template/setting/setting.twig

Крок 5. Модифікація файлу admin/controller/setting/setting.php. Даний файл є контролером, що здійснює обробку запиту на додавання або редагування налаштувань. При цьому потрібно врахувати, що запит може бути POST і GET. Для POST-запиту необхідно перевірити, якщо визначений POST-параметр config_auth_key, передаємо його для подальшого внесення в масив налаштувань. Для GET-запиту отримуємо значення config_auth_key з масиву налаштувань. В наступному лістингу наведено код модифікації контролера setting.php:

```

<file path="admin/controller/setting/setting.php">
    <operation>
        <search><![CDATA[
            if (isset($this->request->post['config_auth_keyword'])) {
        ]]></search>
        <add position="before"><![CDATA[
            if (isset($this->request->post['config_auth_key']))
        {
                $data['config_auth_key']      =      $this->request-
>post['config_auth_key'];
            } else {
                $data['config_auth_key']      =      $this->config-
>get('config_auth_key');
            }
        ]]></add>
    </operation>

```

</file>

Лістинг 2.5 - Модифікація файлу шаблону налаштувань admin/controller/setting/setting.php

Крок 6. Модифікація файлу admin/controller/common/login.php. Даний файл є контролером, який безпосередньо обробляє запит на вхід. Визначимо дві модифікації, які потрібно внести в даний файл:

- Формування правильного рядка запиту залежно від встановлення ключа та його значення
- Формування змінної \$data_action залежно від встановлення ключа та його значення

Код формування рядка запиту в залежності від налаштувань config_auth_key наведено в наступному лістингу 2.6:

```
<operation error="log">
<search>
    <![CDATA[$this->load->language('common/login');]]>
</search>
    <add position="after"><![CDATA[
        $configAuthKey = $this->config->get('config_auth_key');
        $snpass = null;
        if (!empty($configAuthKey['auth_key'])          &&
empty($configAuthKey['auth_key_value'])) {
            if (!isset($_GET[$configAuthKey['auth_key']])) {
                $this->response-
>redirect('http://'.$_SERVER['HTTP_HOST']."/");
            } else {
                $snpass = '&' . $configAuthKey['auth_key'];
            }
        } else if (!empty($configAuthKey['auth_key'])  &&
!empty($configAuthKey['auth_key_value'])) {
            if (!isset($_GET[$configAuthKey['auth_key']]) ||
$_GET[$configAuthKey['auth_key']] !=
$configAuthKey['auth_key_value']) {
```

```

        $this->response->
>redirect('http://'.$_SERVER['HTTP_HOST']."/");
    } else {
        $snpass = '&' . $configAuthKey['auth_key'] . '='
. $configAuthKey['auth_key_value'];
    }
}
]]></add>
</operation>

```

Лістинг 2.6 - Модифікація файлу контролеру admin/controller/common/login.php.
Формування правильного рядка запиту залежно від встановлення ключа та його
значення

Другою операцією в цьому файлі є формування змінної \$data_action шляхом додавання до неї визначеного на попередньому кроці рядку запиту, що включає в себе параметри налаштувань config_auth_key. Код даної модифікації наведений в лістингу 2.7:

```

<operation>
    <search><![CDATA[
        $data['action'] = $this->url->link('common/login', '',
true);
    ]]></search>
    <add position="replace"><![CDATA[
        if (!empty($configAuthKey['auth_key'])) {
            $data['action'] = $this->url->link('common/login',
'', true) . $snpass;
        } else {
            $data['action'] = $this->url->link('common/login',
'', true);
        }
    ]]></add>
</operation>

```

Лістинг 2.7 - Модифікація файлу контролеру admin/controller/common/login.php.

Формування змінної \$data_action залежно від встановлення ключа та його значення

Крок 7. Здійснивши розробку модуля-модифікатора, його необхідно архівувати в файл з іменем adminsecure.osmod.zip.

Таким чином, нами було здійснено розробку модуля-модифікатора для захисту адміністративної панелі від злому. Даний модуль дає можливість додавати в URL-рядок параметру auth_key та при необхідності його значення auth_key_value. Дані параметри мають можливість встановлюватись через адміністративну панель в налаштуваннях системи у вкладці “Server”.

Висновки до розділу 2

В даному розділі нами було визначено варіанти використання модуля для користувача, що має доступ до адміністративної панелі. Адміністратор зокрема повинен мати можливість встановлювати та видаляти модуль, вмикати та вимикати модуль, встановлювати ім'я секретного ключа доступу та його додаткового значення, здійснювати вхід в панель керування з урахуванням наявності секретного ключа доступу та його значення. Було спроектовано UML-діаграми варіантів використання, активностей та послідовностей, здійснено проектування елементів графічного інтерфейсу та опис процесу розробки модуля-модифікатора.

РОЗДІЛ 3. ПОРЯДОК ВСТАНОВЛЕННЯ, ВИДАЛЕННЯ ТА КОРИСТУВАННЯ МОДУЛЕМ. ТЕСТУВАННЯ МОДУЛЯ

3.1. Порядок встановлення системи OpenCart та модуля AdminSecure

Наведемо порядок встановлення системи OpenCart та модуля AdminSecure.

Крок 1. Реєстрація доменного імені. Для цього використаємо реєстратор NIC.UA та зареєструємо безкоштовне доменне ім'я web-marketplace.pp.ua. Після реєстрації доменного імені, його необхідно активувати, використовуючи чат-бот pp.ua. Зареєстроване та активоване доменне ім'я має вигляд:

☐ Domain	NS	Expiry date	Order	AR	
☐ web-marketplace.pp.ua	Custom NS (3)	20 Dec 2024	2401671	☐	<button>Renew</button>

Рисунок 3.1 - Зареєстроване доменне ім'я, готове до використання

Крок 2. Створення веб-сайту. Для цього використаємо хостинг-провайдер Хостинг Україна (Ukraine.com.ua). Нам достатньо вибрати мінімальний тарифний план та в панелі керування у вкладці “Хостинг” створити новий сайт та у вкладці “Домени” додати зареєстроване доменне ім'я:

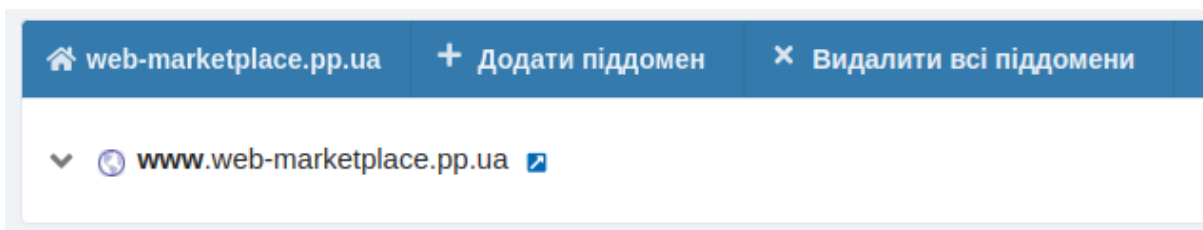


Рисунок 3.2 - Створений веб-сайт у вкладці “Хостинг”

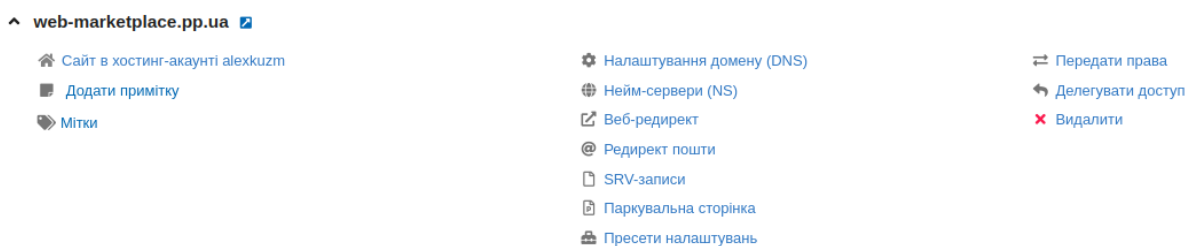


Рисунок 3.3 - Додане доменне ім'я у вкладці “Домени”

Крок 3. Налаштування NS-серверів доменного імені. В налаштуваннях даного доменного імені в панелі адміністратора NIC.UA задамо налаштування NS-серверів – сервери, на яких зберігається вся необхідна інформація про домен, потрібна для коректного функціонування сайту. На ns-серверах вказується інформація про доменне ім'я, поштові записи та субдомени. Таким чином, доменне ім'я зареєстроване на NIC.UA, а інформація про нього, в тому числі і веб-сайт, знаходиться на Ukraine.com.ua:

NS-servers

Custom name servers ▼

List of your own name servers [Change](#)

1. ns310.inhostedns.org
2. ns210.inhostedns.net
3. ns110.inhostedns.com

Рисунок 3.4 - Налаштування NS-серверів для доменного імені.

Крок 4. Створення бази даних. На панелі керування хостингом знайдемо інструмент для створення реляційної бази даних. Створимо нову базу даних webmarketplace, з однойменним користувачем та паролем. Ім'я БД, логін, пароль та адресу сервера БД необхідно скопіювати в безпечне місце для подальшого використання при встановленні OpenCart.

Крок 5. Підготовка OpenCart для встановлення. З офіційного сайту OpenCart.com завантажимо дистрибутив OpenCart та використовуючи файловий менеджер панелі керування хостингом завантажимо вміст папки upload в папку web-marketplace.pp.ua/www на сервері. Перейменуємо файли config-dist.php в корені та в папці /admin в файл config.php відповідно:

	Ім'я	Тип	Розмір	Дата	Права
☐	..				
☐	admin	Folder	—	20.12.2023	rwxf-xf-x
☐	catalog	Folder	—	09.04.2019	rwxf-xf-x
☐	image	Folder	—	09.04.2019	rwxf-xf-x
☐	install	Folder	—	09.04.2019	rwxf-xf-x
☐	system	Folder	—	09.04.2019	rwxf-xf-x
☐	.htaccess.txt	File	2.44 KB	09.04.2019	rw-r--r--
☐	config.php	File	1.17 KB	20.12.2023	rw-r--r--
☐	index.php	File	0.3 KB	11.04.2019	rw-r--r--
☐	opencart-3.0.3.2.zip	File	16.24 MB	20.12.2023	rw-r-----
☐	php.ini	File	0.4 KB	09.04.2019	rw-r--r--

Рисунок 3.5 - Файли OpenCart для наступного встановлення.

Крок 6. Встановлення OpenCart відбувається в браузері. Виконаємо запит за адресою нашого веб-сайту: <https://web-marketplace.pp.ua>:

1/4

opencart

License agreement
Please read the OpenCart licence agreement

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble
The GNU General Public License is a free, copyleft license for software and other kinds of works.

License

Pre-Installation

Configuration

LANGUAGE ▼

CONTINUE

Рисунок 3.6 - Встановлення OpenCart. Ліцензійні умови та вибір мови встановлення.

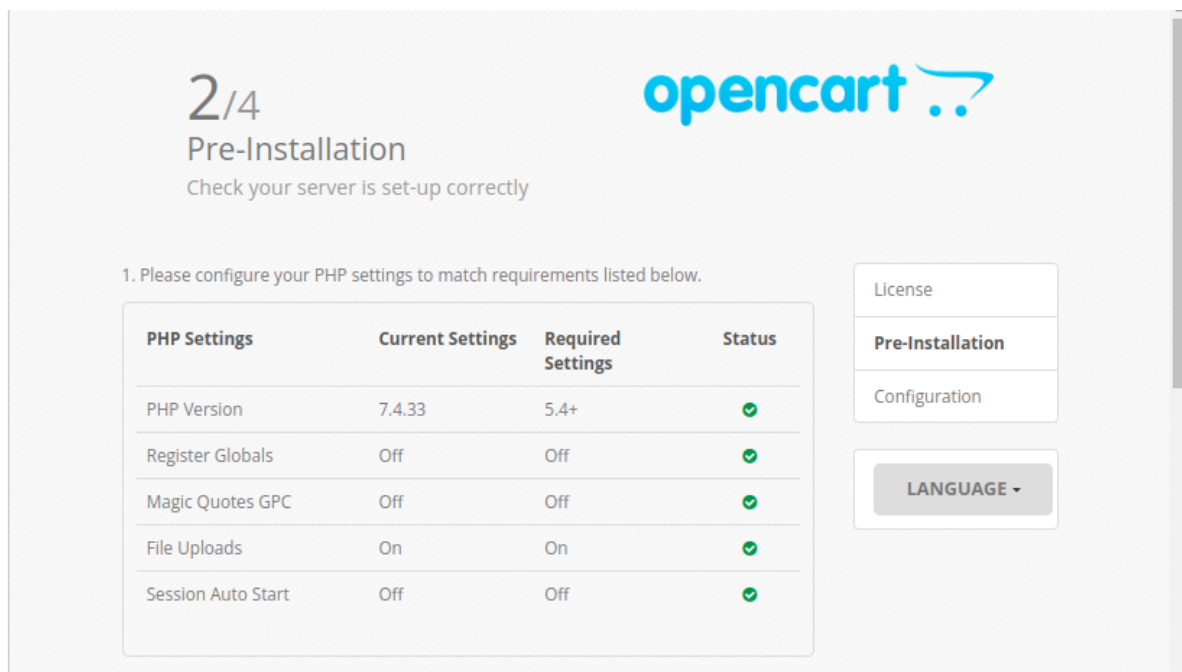


Рисунок 3.7 - Встановлення OpenCart. Перевірка системних вимог.

Configuration - Chromium

Configuration

web-marketplace.pp.ua/install/index.php?route=ins...

1. Please enter your database connection details.

DB Driver: MySQLi

* Hostname: localhost

* Username: root

Password:

* Database:

* Port: 3306

Prefix: oc_

License

Pre-Installation

Configuration

LANGUAGE

2. Please enter a username and password for the administration.

* Username: admin

* Password:

* E-Mail:

BACK CONTINUE

[Project Homepage](#) | [Documentation](#) | [Support Forums](#)

Рисунок 3.8 - Встановлення OpenCart. Введення облікових даних для з'єднання з базою даних а також облікових даних (Username, Password, E-Mail) адміністратора системи.

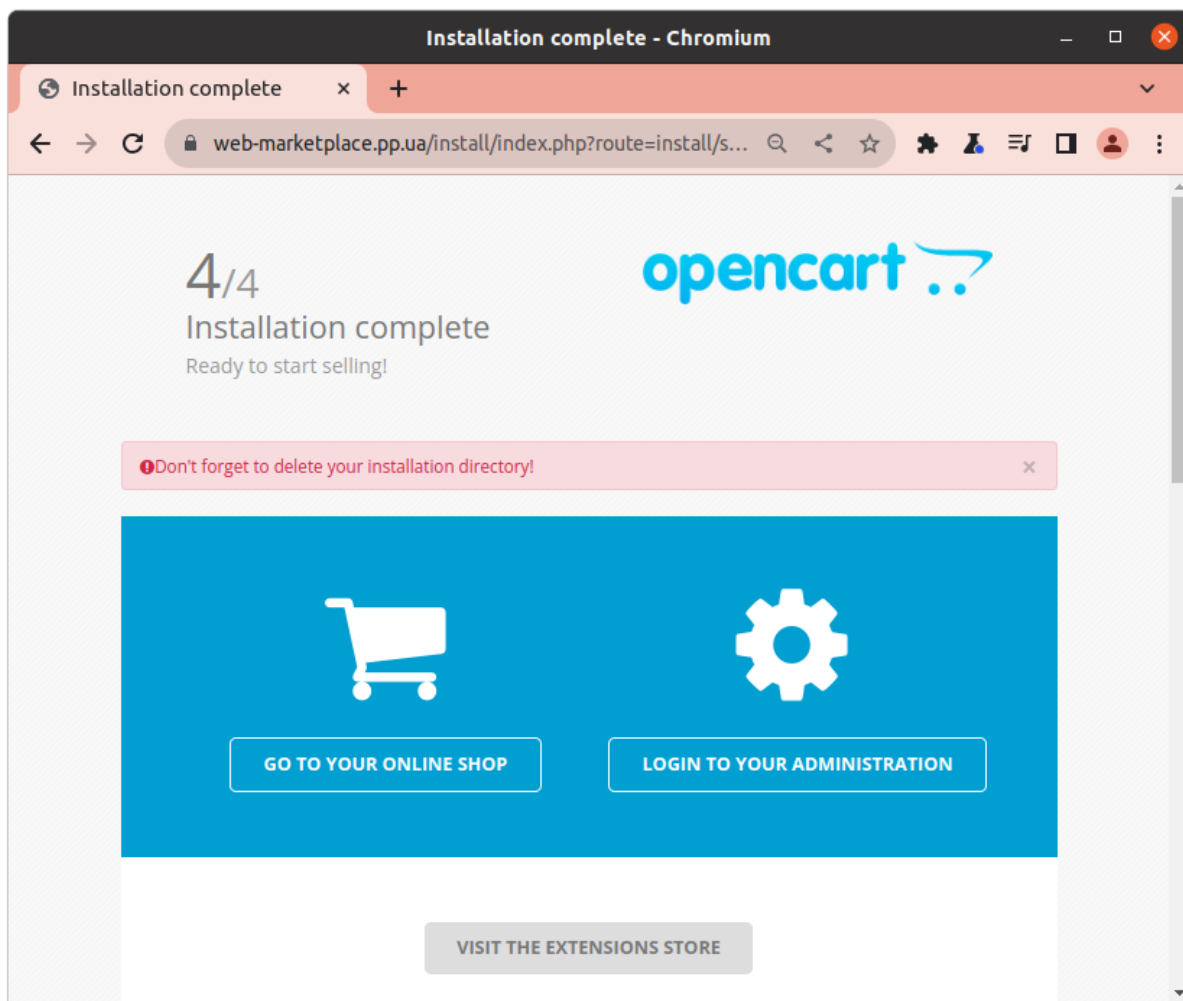


Рисунок 3.9 - Встановлення OpenCart. Успішне встановлення

Після успішного встановлення можна видалити директорію `install` в корені проекту. Вона необхідна лише для встановлення системи.

Для доступу в адміністративну панель використаємо шлях `/admin`:

Рисунок 3.10 - Форма для входу в адміністративну панель

При успішному вході, адміністратор має доступ до функцій керування веб-системою на CMS OpenCart:

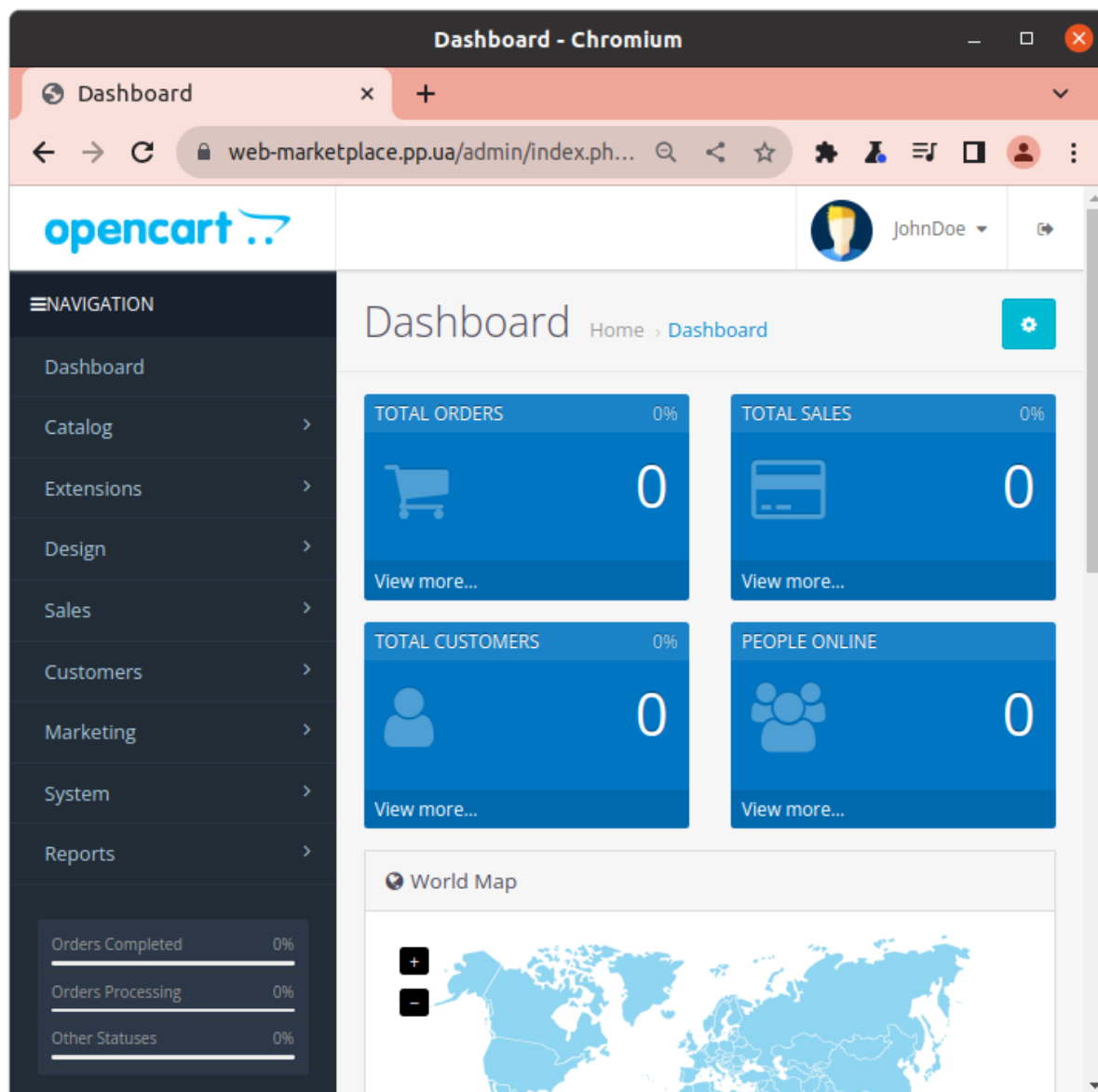


Рисунок 3.11 - Панель адміністратора OpenCart.

Крок 7. Встановлення модуля AdminSecure. Перейдемо по навігації в Extensions – Installer, натиснемо кнопку Upload, виберемо підготовлений архівний файл модуля adminsecure.ocmod.zip та здійнимо установку:

opencart

JohnDoe

Extension Installer

Home > Extension Installer

Upload your extensions

Upload your extensions

Upload File **Upload**

Install Progress

Progress

Success: You have modified extensions!

Install History

Filename	Date Added	Action
adminsecure.ocmod.zip	22/12/2023	

Showing 1 to 1 of 1 (1 Pages)

Orders Completed 0%

OpenCart © 2009-2023 All Rights Reserved.

Рисунок 3.12 - Установка модуля AdminSecure.ocmod

Крок 8. Після установки модуля необхідно оновити модифікатори. Для цього перейдемо в “Extensions – Modifications” та натиснемо кнопку “Оновити”. Оновлення модифікацій фактично приміняє модифікації щойно встановленого модуля:

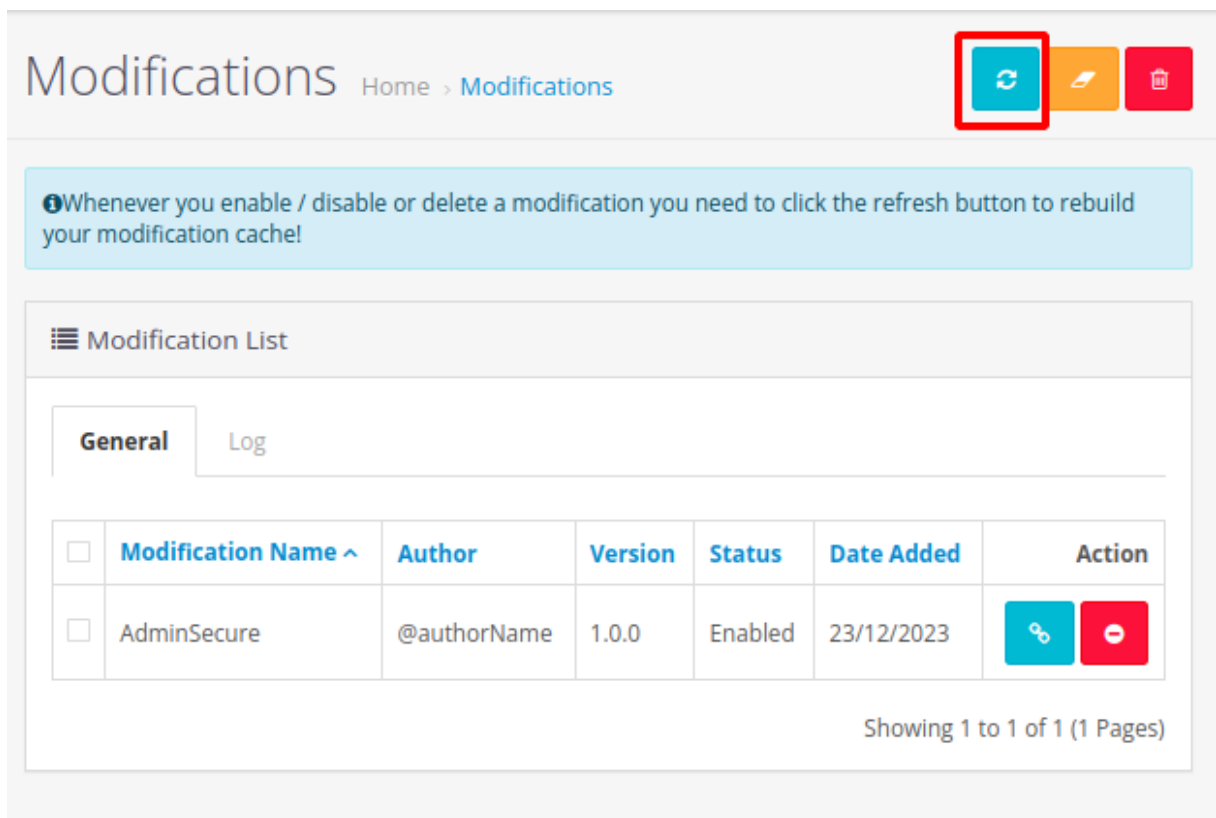


Рисунок 3.13 - Оновлення модифікацій

Таким чином, ми розглянули порядок встановлення системи OpenCart в робочому режимі та послідовність дій для встановлення модуля AdminSecure. В наступному підрозділі розглянемо порядок користування модулем.

3.2. Порядок використання модуля захисту адміністративної панелі від злому

Після встановлення модуля, його можна використовувати. Для цього необхідно:

- 1) зайти в налаштування системи: “System – Settings – Редагувати”:

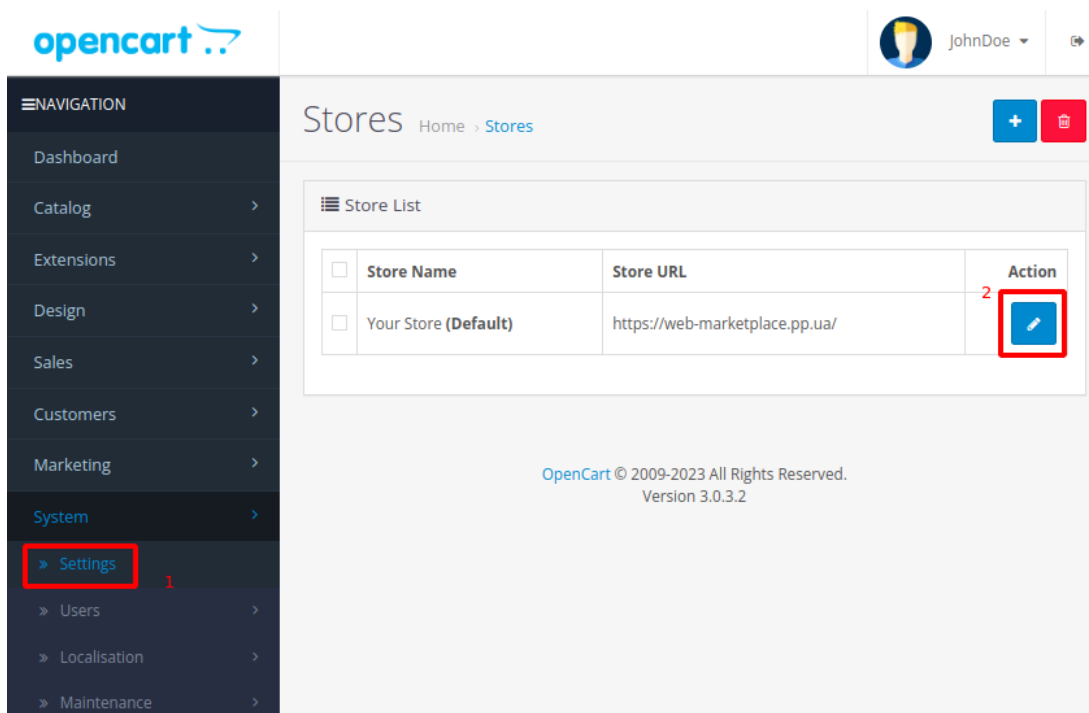


Рисунок 3.14 - Редагування налаштувань

- 2) перейти у вкладку “Server” та переконатись у наявності параметрів Admin Secure Auth Key та Admin Secure Auth Key Value:

Рисунок 3.15 - Налаштування секретного ключа та його необов’язкового значення

- 3) ввести ім'я секретного ключа та при необхідності його значення, наприклад:

Admin
Secure
Auth Key ?

auth_key

Admin
Secure
Auth Key
Value ?

qwerty1234

Рисунок 3.16 - Встановлення налаштувань параметрів захисту

- 4) зберегти налаштування, натиснувши відповідну кнопку зверху справа сторінки. Результат збереження повинен супроводжуватись повідомленням про успіх:

Stores Home > Stores

+ 🗑️

✔ Success: You have modified settings!

☰ Store List

☐	Store Name	Store URL	Action
☐	Your Store (Default)	https://web-marketplace.pp.ua/	

Рисунок 3.17 - Повідомлення про успішне збереження налаштувань

- 5) якщо користувач встановив лише ім'я секретного ключа доступу, то доступ до форми логування повинен відбуватись з таким URL-рядком: /admin?{auth_key}, де auth_key – встановлене ім'я:

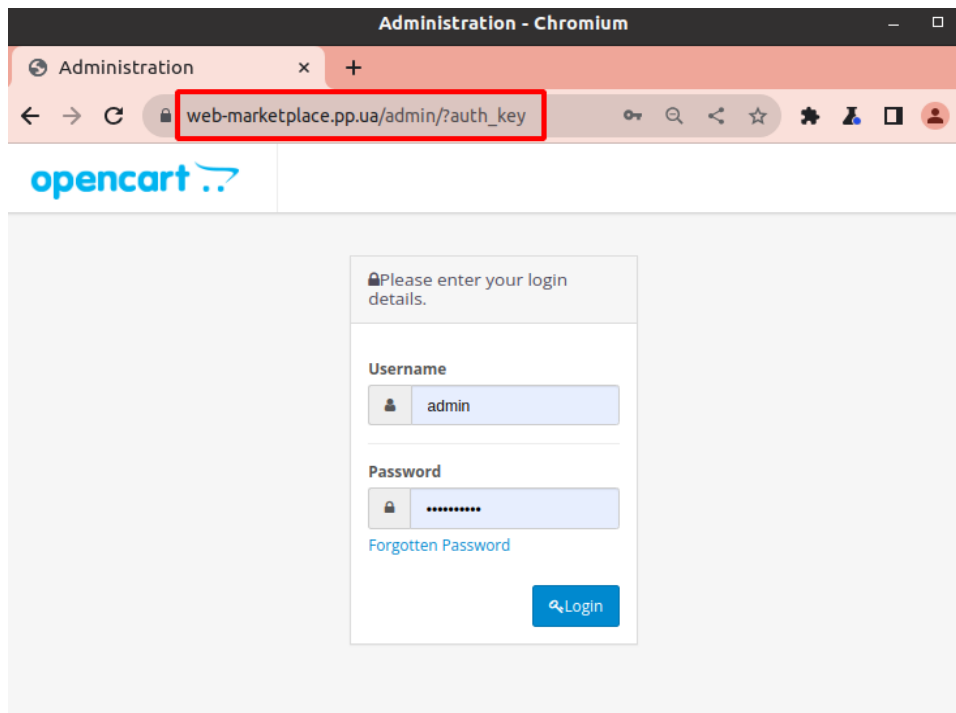


Рисунок 3.18 - Доступ до сторінки логування з лише встановленим іменем секретного ключа доступу

- 6) Якщо користувач встановив ім'я та значення секретного ключа, доступ до форми логування повинен здійснюватись за рядком запиту у форматі: `/admin/?{auth_key}={auth_key_value}`, де `auth_key` – ім'я ключа доступу, `auth_key_value` – його значення:

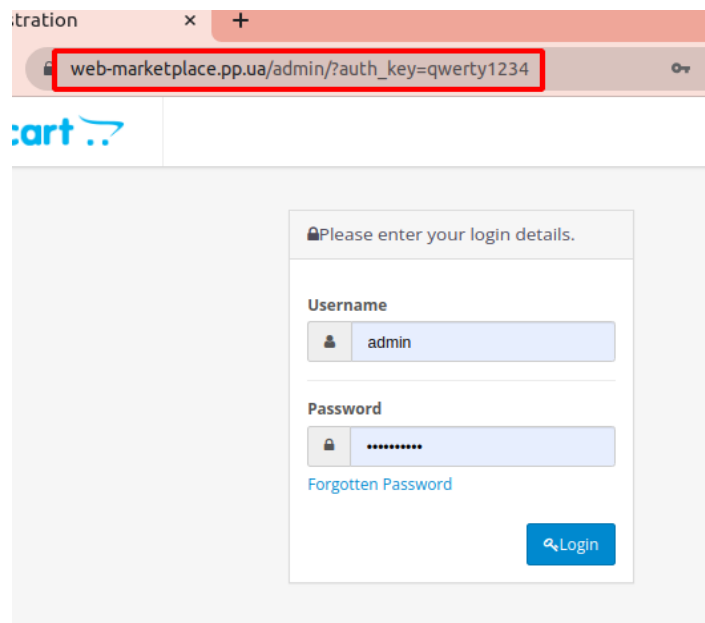


Рисунок 3.19 - Доступ до сторінки логування зі встановленим іменем та значенням секретного ключа доступу

- 7) Для скасування роботи секретного ключа доступу достатньо очистити налаштування Admin Secure Auth Key. Тоді URL-адреса до сторінки з формою входу матиме вигляд по замовчуванню: /admin.

3.3. Видалення, увімкнення та вимкнення модуля

Для повної деінсталяції модуля необхідно перейти в розділ Extensions, далі Installer, та у таблиці Install History натиснути кнопку Uninstall:

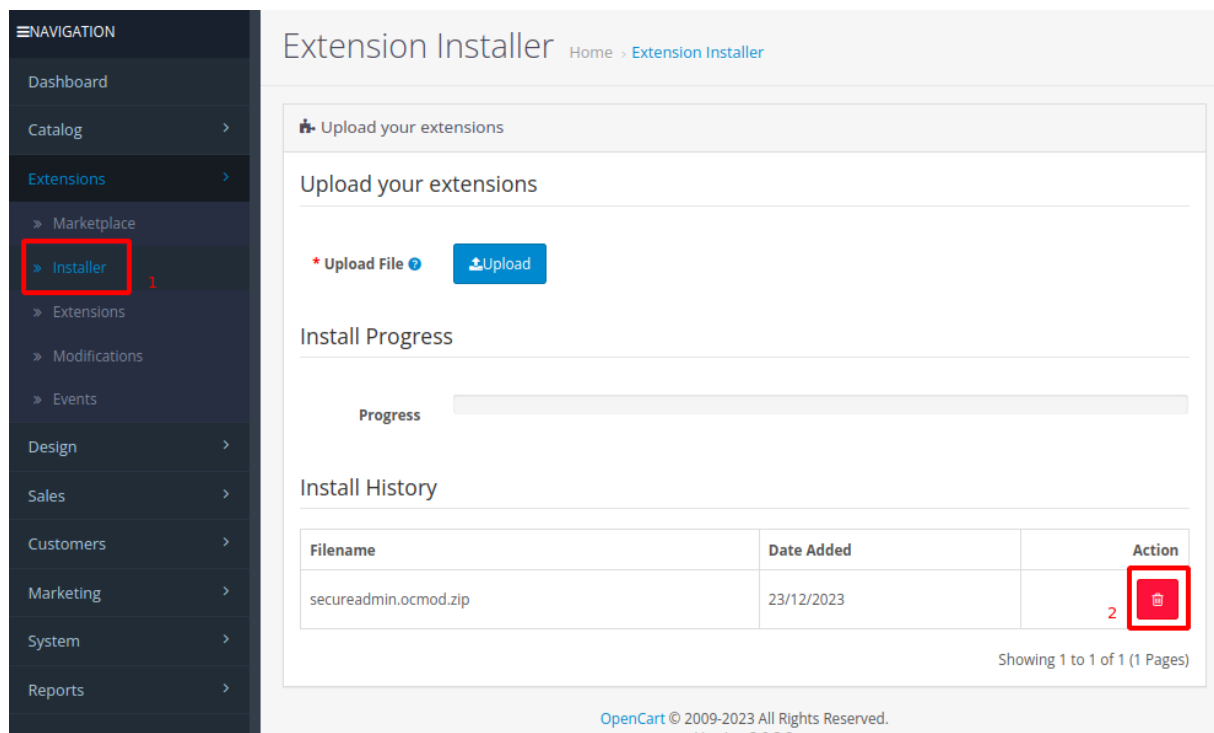


Рисунок 3.20 - Деінсталяція модуля AdminSecure

Після деінсталяції необхідно обов'язково оновити кеш модифікаторів аналогічно як і після інсталяції.

Для увімкнення чи вимкнення модуля необхідно перейти в розділ Extensions, далі Modifications. У таблиці Modification List біля даного модуля в стовпчику Action натиснути кнопку Увімкнути (Вимкнути):

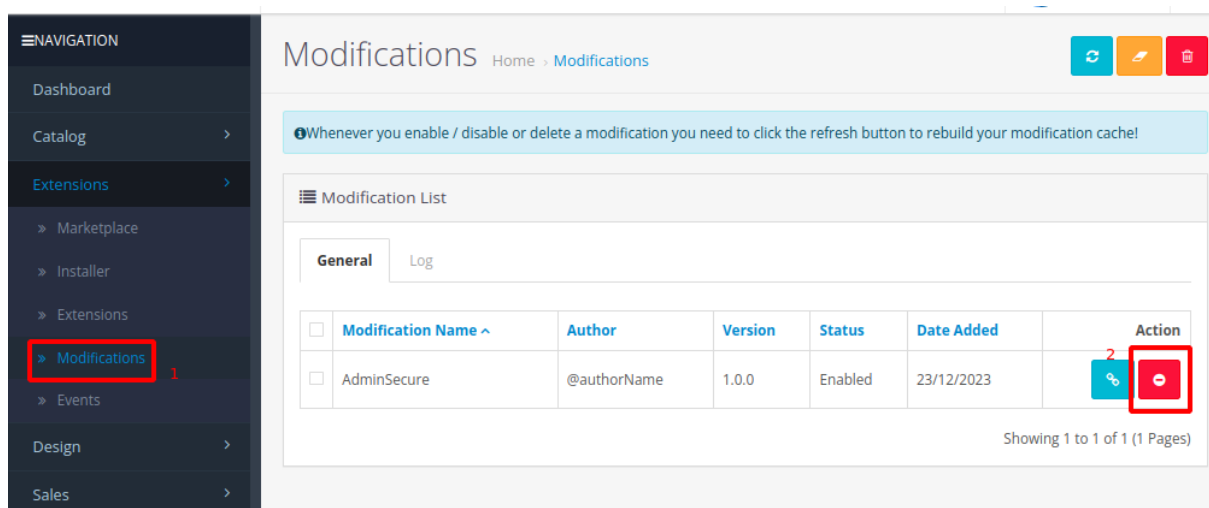


Рисунок 3.21 - Увімкнення та вимкнення модуля AdminSecure.

Після виконання даних дій аналогічно до встановлення чи видалення необхідно також оновлювати кеш модифікаторів.

3.4. Тестування роботи модуля

Тестування модуля для системи OpenCart є важливою частиною розробки та включає ряд загальних кроків для забезпечення якості роботи модуля належним чином перед випуском його робочої версії. Наведемо нижче такі кроки:

Налаштування тестового середовища. Передбачає встановлення локального середовища для OpenCart або наявність тестового сервера, встановлення модуля на тестовий сервер.

Функціональне тестування. Перевірка правильності роботи всіх функцій модуля. Перевірка правильності обробки даних і взаємодії з базою даних OpenCart. Пересвідчення, що модуль не порушує основні функції OpenCart.

Тестування на сумісність. Перевірка взаємодії модуля з іншими встановленими модулями, перевірка сумісності модуля з різними версіями OpenCart.

Тестування безпеки. Перевірка модуля на наявність вразливостей безпеки, таких як SQL-ін'єкції, XSS, CSRF тощо. Пересвідчення, що модуль не допускає несанкціонований доступ до даних або функцій.

Тестування продуктивності. Пересвідчення, що модуль не впливає негативно на швидкість сайту, визначення часу виконання ключових функцій модуля.

Тестування візуального вигляду. Оцінка візуального вигляду елементів інтерфейсу модуля. Пересвідчення в інтеграції з дизайном сайту та без впливу на зовнішній вигляд.

Тестування мобільної сумісності. Перевірка роботи модуля на мобільних пристроях. Забезпечення зручного інтерфейсу для мобільних пристроїв.

Тестування оновлення. Перевірка обробки процесу оновлення. Пересвідчення, що оновлення модуля не порушує існуючі дані або налаштування.

Формування документації. Пересвідчення, що є достатня документація для модуля, включно з інструкціями зі встановлення та налаштування.

Автоматизоване тестування. Розгляд можливості створення автоматизованих тестів для модуля для полегшення процесів тестування під час майбутніх змін.

Тестування взаємодії з користувачем. Пересвідчення, що модуль забезпечує зручну та інтуїтивно зрозумілу роботу.

Тестування локалізації та інтернаціоналізації. Тестування модуля різними мовами та пересвідчення у підтримці функцій інтернаціоналізації.

Кросбраузерне тестування. Перевірка сумісності модуля з різними веб-браузерами для забезпечення стабільної роботи користувача.

Тестування обробки помилок і журналювання. Перевірка обробки помилок модулів та реєстрації відповідної інформації. Пересвідчення про чіткість та корисність повідомлень про помилки для усунення несправностей.

Таким чином, виконавши дані кроки, можна систематично протестувати модуль OpenCart для забезпечення його надійності, функціональності та безпеки перед випуском його робочої версії.

В таблиці 3.1 наведено хід та результати функціонального тестування:

Таблиця 3.1

Хід та результати функціонального тестування модуля для захисту адміністративної панелі від злому

Функція	Послідовність дій	Результат тестування
Встановлення модуля	● Перейти в Extensions – Installer; ● Натиснути Upload; ● Вибрати файл модуля AdminSecure.ocmod.zip; ● Оновлення кешу модифікаторів.	Модуль встановлено
Наявність полів auth_key та auth_key_value в формі налаштувань системи	● Перейти в System – Settings; ● Відкрити вкладку Server; ● Перевірити наявність полів Secure Auth Key та Secure Auth Key Value; ● Перевірити роботу tooltips (підказок) до полів;	Виконано
Збереження налаштувань auth_key та auth_key_value	● Перейти в System – Settings; ● Відкрити вкладку Server; ● Ввести значення в поля auth_key та auth_key_value ● Натиснути “Зберегти” ● Перевірити наявність даних значень в формі після збереження	Виконано
Перевірка запиту на сторінку з формою входу з заданим	● Встановити значення auth_key а auth_key_value; ● Здійснити вихід з адмін-панелі; ● Ввести адресу /admin для	Виконано

значенням auth_key та auth_key_value	входу; ● Переконатись, що відбулась переадресація на головну сторінку; ● Ввести адресу у форматі /admin?{auth_key}={auth_key_value}, де auth_key та auth_key_value – встановлені значення параметрів; ● Переконатись, що запит отримав сторінку з формою для входу в систему.	
Перевірка запити на сторінку з формою входу з заданим значенням лише auth_key	● Встановити значення лише auth_key; ● Здійснити вихід з адмін-панелі; ● Ввести адресу /admin для входу; ● Переконатись, що відбулась переадресація на головну сторінку; ● Ввести адресу у форматі /admin?{auth_key}, де auth_key встановлене значення параметру; ● Переконатись, що запит отримав сторінку з формою для входу в систему.	Виконано
Перевірка запити на сторінку з формою входу без задання значення auth_key	● Очистити значення auth_key; ● Здійснити вихід з адмін-панелі; ● Ввести адресу /admin для входу; ● Переконатись, що запит отримав сторінку з формою для входу в систему.	Виконано
Перевірка запити на сторінку з формою при	● Встановити значення auth_key; ● Вимкнути модуль; ● Оновити кеш модифікаторів;	Виконано

вимкненому модулі	● Ввести адресу /admin для входу; ● Переконались, що запит отримав сторінку з формою для входу в систему;	
Видалення модуля	● Перейти в Extensions – Installer; ● В таблиці Install History в рядку модуля натиснути кнопку Uninstall; ● Оновити кеш модифікаторів; ● Переконались, що параметри auth_key та auth_key_value відсутні в налаштуваннях системи;	Виконано
Перевірка запиту на сторінку з формою при видаленому модулі	● Здійснити вихід; ● Ввести адресу /admin для входу; ● Переконались, що запит отримав сторінку з формою для входу в систему;	Виконано

Отже, нами було здійснено тестування модуля захисту адмін-панелі від злому для системи OpenCart, зокрема було проведено функціональне тестування, продуктивності, зручності використання, сумісності, безпеки, адаптивності для мобільних пристроїв.

Висновки до розділу 3

В третьому розділі роботи автором розроблено порядок встановлення системи OpenCart та модуля AdminSecure, використання модулю захисту адміністративної панелі від злому, інструкції щодо видалення, ввімкнення та вимкнення модуля.

Запропоновано та описано ряд загальних кроків для забезпечення якості роботи модуля належним чином перед випуском його робочої версії. Наведено

деталізовану послідовність дій при проведенні функціонального тестування. Всі виявлені помилки модуля були виправлені та зроблено висновок про відповідність розробленого модуля всім поставленим завданням.

ВИСНОВКИ

В кваліфікаційній роботі було здійснено розробку модуля для захисту адміністративної частини від злому шляхом включення додаткового параметру в рядок URL.

В ході роботи було розглянуто рекомендації щодо захисту систем для здійснення електронної комерції та сформульовано основну задачу роботи. Було виявлено три аналогічних рішення та проведено порівняння їх функціональних можливостей, в результаті чого виявлено типові функціональні можливості таких модулів, які пропонується реалізувати в нашому модулі. Обґрунтовано вибір ОСМОД для розробки модуля, мови та необхідного стеку програмного забезпечення для розробки модуля.

Визначено варіанти використання модуля для користувача, що має доступ до адміністративної панелі. Адміністратор зокрема повинен мати можливість встановлювати та видаляти модуль, вмикати та вимикати модуль, встановлювати ім'я секретного ключа доступу та його додаткового значення, здійснювати вхід в панель керування з урахуванням наявності секретного ключа доступу та його значення. Було спроектовано UML-діаграми варіантів використання, активностей та послідовностей, здійснено проектування елементів графічного інтерфейсу та опис процесу розробки модуля-модифікатора.

Розроблено порядок встановлення системи OpenCart та модуля AdminSecure, використання модуля захисту адміністративної панелі від злому, інструкції щодо видалення, ввімкнення та вимкнення модуля.

Запропоновано та описано ряд загальних кроків для забезпечення якості роботи модуля належним чином перед випуском його робочої версії. Наведено деталізовану послідовність дій при проведенні функціонального тестування. Всі виявлені помилки модуля були виправлені та зроблено висновок про відповідність розробленого модуля всім поставленим завданням.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenCart documentation. Developing Modules [Електронний ресурс]. – Режим доступу: <https://docs.opencart.com/en-gb/developer/module/>, вільний. – (дата звернення 05.10.2023).
2. OSMOD Modification System [Електронний ресурс]. – Режим доступу: <https://github.com/opencart/opencart/wiki/Modification-System>, вільний. – (дата звернення 15.10.2023).
3. Smart Admin Security [Електронний ресурс]. – Режим доступу: https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=42842, вільний. – (дата звернення 01.11.2023).
4. Secure Admin URL (OpenCart 3.0.*) [Електронний ресурс]. – Режим доступу: https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=40693, вільний. – (дата звернення 01.11.2023).
5. Store admin protection module for Opencart [Електронний ресурс]. – Режим доступу: <https://osmod.net/en/modul-zaschita-adminki>, вільний. – (дата звернення 01.11.2023).
6. OpenCart Basic Security Practices [Електронний ресурс]. – Режим доступу: <https://docs.opencart.com/en-gb/administration/security/>, вільний. – (дата звернення 15.11.2023).
7. 25 website security measures for eCommerce developer – Opencart [Електронний ресурс]. – Режим доступу: <https://webocreation.com/25-opencart-security-tactics-a-cheat-sheet-for-opencart-ecommerce-users>, вільний. – (дата звернення 15.11.2023).
8. Opencart : Security Vulnerabilities [Електронний ресурс]. – Режим доступу: https://www.cvedetails.com/vulnerability-list/vendor_id-9599/Opencart.html, вільний. – (дата звернення 15.11.2023).
9. PHP Documentation [Електронний ресурс]. – Режим доступу: <https://www.php.net/docs.php>, вільний. – (дата звернення 20.11.2023).

- 10.OpenCart Basics [Електронний ресурс]. – Режим доступу: <https://github.com/opencart/opencart/wiki/OpenCart-Basics>, вільний. – (дата звернення 20.11.2023).
- 11.Using OpenCart as a lightweight MVC framework [Електронний ресурс]. – Режим доступу: <https://forum.opencart.com/viewtopic.php?t=168126>, вільний. – (дата звернення 20.11.2023).
- 12.Route Flow Of Opencart [Електронний ресурс]. – Режим доступу: <https://webkul.com/blog/route-flow-opencart/>, вільний. – (дата звернення 01.12.2023).
- 13.OpenCart Theme Development: Basic Tutorial for Beginners [Електронний ресурс]. – Режим доступу: <https://isenselabs.com/posts/opencart-theme-development>, вільний. – (дата звернення 01.12.2023).
- 14.OpenCart Local testing environment [Електронний ресурс]. – Режим доступу: <https://forum.opencart.com/viewtopic.php?t=220476>, вільний. – (дата звернення 05.12.2023).

ДОДАТКИ

ОСМОД-файл install.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<modification>
<name>AdminSecure</name>
<code>admin-secure</code>
<version>1.0.0</version>
<author>@authorName</author>
<link>https://opencart.com</link>
<file path="admin/language/en-gb/setting/setting.php">
<operation>
<search>
<![CDATA[// Entry]]>
</search>
<add position="after">
<![CDATA[
$_['entry_auth_key'] = 'Admin Secure Auth Key';
$_['entry_auth_key_value'] = 'Admin Secure Auth Key Value';
$_['help_auth_key'] = 'Set Secure Auth Key to protect your Admin URL preventing
unauthorized entry.<br /><br /><b><i> CAUTION !! </b></i><br /> <br />Your Admin
URL will be changed to this format: http://yourdomain.com/admin/?key';
$_['help_auth_key_value'] = 'Set the Additional Value to protect your Admin URL
preventing unauthorized entry.<br /><br /><b><i> CAUTION !! </b></i><br /> <br
/>Your Admin URL will be changed to this format:
http://yourdomain.com/admin/?key=value';
]]>
</add>
</operation>
</file>
<file path="admin/view/template/setting/setting.twig">
<operation>
<search>
<![CDATA[<label class="col-sm-2 control-label"><span data-toggle="tooltip"
title="{{ help_maintenance }}">{{ entry_maintenance }}</span></label>]]>
</search>
<add position="before" offset="1">
<![CDATA[
<h2>Admin secure parameters</h2>
<div class="form-group">
<label class="col-sm-2 control-label" for="admin_auth_key">

```

```

<span data-toggle="tooltip" title="{{ help_auth_key }}">{{ entry_auth_key
}}</span>
</label>
<div class="col-sm-10">
<input type="text" name="config_auth_key[auth_key]" value="{% if
config_auth_key.auth_key %} {{ config_auth_key.auth_key }} {% endif %}"
id="admin_auth_key" class="form-control"/>
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label" for="admin_auth_key_value">
<span data-toggle="tooltip" title="{{ help_auth_key_value }}">{{
entry_auth_key_value }}</span>
</label>
<div class="col-sm-10">
<input type="text" name="config_auth_key[auth_key_value]" value="{% if
config_auth_key.auth_key_value %} {{ config_auth_key.auth_key_value }} {% endif
%}" id="admin_auth_key_value" class="form-control"/>
</div>
</div>
]]>
</add>
</operation>
</file>
<file path="admin/controller/setting/setting.php">
<operation>
<search><![CDATA[
if (isset($this->request->post['config_auth_keyword'])) {
}}></search>
<add position="before"><![CDATA[
if (isset($this->request->post['config_auth_key'])) {
$data['config_auth_key'] = $this->request->post['config_auth_key'];
} else {
$data['config_auth_key'] = $this->config->get('config_auth_key');
}
}}></add>
</operation>
</file>
<file path="admin/controller/common/login.php">
<operation error="log">
<search>

```

```

<![CDATA[$this->load->language('common/login');]]>
</search>
<add position="after"><![CDATA[
$configAuthKey = $this->config->get('config_auth_key');
$snpass = null;
if (!empty($configAuthKey['auth_key']) &&
empty($configAuthKey['auth_key_value'])) {
if (!isset($_GET[$configAuthKey['auth_key']])) {
$this->response->redirect('http://'.$_SERVER['HTTP_HOST']."/");
} else {
$snpass = '&' . $configAuthKey['auth_key'];
}
} else if (!empty($configAuthKey['auth_key']) &&
!empty($configAuthKey['auth_key_value'])) {
if (!isset($_GET[$configAuthKey['auth_key']]) ||
$_GET[$configAuthKey['auth_key']] != $configAuthKey['auth_key_value']) {
$this->response->redirect('http://'.$_SERVER['HTTP_HOST']."/");
} else {
$snpass = '&' . $configAuthKey['auth_key'] . '=' .
$configAuthKey['auth_key_value'];
}
}
}]]></add>
</operation>
<operation>
<search><![CDATA[
$data['action'] = $this->url->link('common/login', '', true);
]]></search>
<add position="replace"><![CDATA[
if (!empty($configAuthKey['auth_key'])) {
$data['action'] = $this->url->link('common/login', '', true) . $snpass;
} else {
$data['action'] = $this->url->link('common/login', '', true);
}
}]]></add>
</operation>
</file>
</modification>

```

Послідовність файлів та операцій для внесення змін в файли адміністративної частини

admin/language/en-gb/setting/setting.php

Додавання текстових написів для даних
auth_key, auth_key_value, help_auth_key,
help_auth_key_value

admin/view/template/setting/setting.twig

Додавання в форму налаштувань полів
auth_key та auth_key_value

admin/controller/common/login.php

POST: підготовка полів до збереження як
налаштування
GET: отримання значень полів з наявних
налаштувань

admin/controller/setting/setting.php

Формування правильного рядка запиту залежно
від встановлення ключа та його значення

Формування змінної \$data_action залежно від
встановлення ключа та його значення

Форма налаштувань з додаванням полів auth_key та auth_key_value шляхом модифікатора OSMOD

The screenshot displays the OpenCart admin interface. At the top, there is a browser window with the address bar showing the URL: `https://{domainName}/admin/index.php?route=setting/setting`. Below the browser window, the OpenCart logo is visible. The main content area is divided into a left sidebar and a main panel. The sidebar, titled 'Navigation', contains buttons for 'Dashboard', 'Catalog', 'Extensions', 'System' (highlighted in blue), '>> Settings', and '>> ...'. The main panel has a tabbed interface with tabs labeled 'Tab 1', 'Tab 4', 'Tab 3', and 'Server' (highlighted in blue). A 'Зберегти' (Save) button is located in the top right corner of the main panel. The 'Server' tab is active, showing the 'General' section. This section contains two input fields: 'Admin Key' and 'Admin Key Value'. The 'Admin Key Value' field has a dropdown menu with an ellipsis ('...') indicating more options.