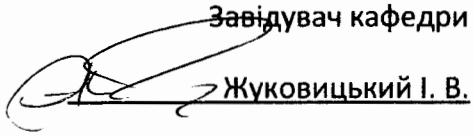
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет науки і технологій

Кафедра «Електронні обчислювальні машини»

«ДО ЗАХИСТУ»
Завідувач кафедри



Жуковицький І. В.

(підпис) (ПІБ)

«20» 12 2021 р.

ДИПЛОМНА РОБОТА

на здобуття освітнього ступеня «магістр»

Галузь знань 12 Інформаційні технології

(шифр) (назва)

Спеціальність 123 Комп'ютерна інженерія

(код) (повна назва)

Тема Дослідження методів підвищення безпеки web-додатків

Theme Research of methods to increase the security of web-applications

Керівник дипломного проекту	проф. (посада)	(підпис)	Жуковицький І. В. (ПІБ)
Консультант розділу з БЖД	доцент (посада)	(підпис)	Саблін О. І. (ПІБ)
Нормоконтролер	доцент (посада)	(підпис)	Шаповалов В. О. (ПІБ)
Студент групи	КС2021 (група)	(підпис)	Павленко І. І. (ПІБ)
Student			Pavlenko Ilya (family name)

Дніпро
2021

Довідка

про відсутність плагіату у випускній кваліфікаційній роботі

Міністерство освіти і науки України

Український державний університет науки і технологій

Кафедра Електронні обчислювальні машини

ДОВІДКА

За результатами перевірки випускної кваліфікаційної роботи здобувача вищої освіти Павленка Ілії Івановича

(прізвище, ім'я, по батькові)

на тему: Методи підвищення безпеки в веб-додатку

в роботі не виявлено порушень академічної доброчесності.

Керівник ВКР _____ Жуковицький І.В.

6 Розділи та консультанти

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв

КАЛЕНДАРНИЙ ПЛАН

Назва розділу	Термін виконан-ня	Обсяг розділу, %

Дата видачі завдання: «__»_____20____р.

Керівник дипломної роботи _____ Заєць О. П.

(підпис) (ПІБ)

Завдання прийняв до виконання _____ Павленко І. І.

(підпис) (ПІБ)

Зміст

Вступ

Поняття веб-сайту з’явилося ще на початку зародження Інтернету, коли мережа та сайти використовувалися в основному для розваг. До кінця 90-х років веб-сайти представляли собою статичні сторінки. Такі сайти були не зручними, не красивими, а також могли запропонувати лише дуже обмежений функціонал. Для написання таких сторінок достатньо знати мову розмітки HTML. Але йде час і ніщо не стоїть на місці. Розвиваються мови програмування, покращуються та вдосконалюються канали передачі інформації. Зараз інтернет представляє собою одну з найприбутковіших галузей економіки а також є місцем для проведення вільного часу, роботи та розважень для сотень мільйонів людей.

Веб-додатки грають велику роль в житті людей. Користувачі довіряють сайтам свої особисті данні, номери кредитних карток, номери телефонів, паролі та велику кількість різноманітних важливих даних, а від безпеки цих додатків та ввірених їм даних, часто, залежить навіть життя людей.

Для полегшення та прискорення процесу створення свого додатку існує велика кількість готових систем керування веб-змістом (WCMS). Веб CMS надають велику кількість різноманітних інструментів для керування та відображення графічного і текстового наповнення додатку. Доступність і легкість в освоєнні роблять системи керування веб-змістом привабливими для малих компаній та незалежних розробників. Але більшість з цих людей забуває про недоліки WCMS, такі, як наявність різноманітних вразливостей, необхідність оновлення додатку та людські прорахунки. Кожен раз, коли клієнт або випадковий відвідувач перебувають на сайті, власник повинен бути впевнений, шо їх персональні данні перебувають у безпеці.

В даній роботі були розглянуті можливі та найбільш розповсюджені вразливості веб-додатків, створених з допомогою WCMS Wordpress,

проаналізовані методи підвищення безпеки додатку та надано рекомендації для застосування цих методів.

1. ОГЛЯД ЗАГРОЗ БЕЗПЕКИ У ВЕБ-ДОДАТКУ ТА ОПИС ПРОБЛЕМИ

1.1. Головні загрози безпеці веб-додатку

Веб-додатки можуть бути атаковані як на клієнтській, так і на серверній стороні, включаючи і третю сторону, яку приймає участь в процесі передачі та отримання інформації веб-браузері клієнту. Сам по собі веб-браузер клієнту не є ні безпечною, ні довіреною стороною обміну інформацією. В ідеалі веб-додаток повинен забезпечувати захист з обох сторін, і з клієнтської, і з серверної. Але, як видно з [1], частіше всього увага приділяється саме серверній частині, а клієнтська сторона залишається відкрита для загроз. Дані з клієнтської сторони ніколи не можуть бути довіреними та завжди потребують ретельної перевірки перед використанням на сервері, додання до бази даних, тощо. Дослідження в області безпеки веб-додатків [2] показує як з часом увага зловмисників змістилася з серверної на клієнтську сторону. Серед вразливостей, показаних в дослідженні, можна виділити випадки A04, A06 та A08, для вирішення яких в більшості випадків достатньо оновлення програмного забезпечення або використання більш сучасних аналогів та компонентів. Також веб-додаток повинен перевіряти початковий код на цілісність щоб впевнитись, що частини коду, включені для безпеки, не змінені або видалені. Механізми для верифікації коду на стороні клієнту можуть в значній мірі впливати на чуйність додатку та час завантаження сторінки, що було показано в [3]. Ця проблема додає роботи розробникам і вимушує їх балансувати між необхідністю верифікувати код та забезпеченням допустимої чуйності додатку. Це також призводить до того, що велика кількість компаній та незалежних розробників з великим небажанням додають такий код до своїх проектів, т. я. вони ризикують сповільнити

його, тим самими погіршив досвіт використання додатку кінцевим клієнтом. Все це призводить до того, що веб-додаток стає менше захищеним, що в свою чергу підвергає його додатковому ризику. Оскільки код виконується на стороні клієнту, забезпечити його повний захист неможливо, зазначено в [4]. Не зважаючи на те, що недостатня кількість перевірок під час роботи додатку підвищує його вразливість до різноманітних атак, своєчасні перевірки на серверній та клієнтській частинах можуть звести ризик до мінімуму [5]. Перевірки можуть відбуватися на фоні, що забезпечує динамічний захист від підміни коду під час роботи веб-додатку.

Для полегшення та прискорення процесу створення свого додатку існує велика кількість готових систем керування веб-змістом (WCMS). Веб CMS надають велику кількість різноманітних інструментів для керування та відображення графічного і текстового наповнення додатку. Доступність і легкість в освоєнні роблять системи керування веб-змістом привабливими для малих компаній та незалежних розробників. Одним із популярних WCMS є WordPress. Як сказано на офіційному сайті [6], WordPress це програмне забезпечення для створення магазинів, блогів та новинних сторінок. Система WordPress починала як платформа для ведення блогу, після чого еволюціонувала в веб-рішення для створення інтернет магазинів, блогів та новинних сторінок. Система проста та інтуїтивно зрозуміла, через це користується популярністю серед малих фірмам та новачків при виборі способу створення веб-додатку. Також WordPress є платформою з відкритим вихідним кодом, тому кожен може впливати на його внутрішні функції. Ця відкритість дозволяє розробникам додавати нові теми або кінцевим користувачам додавати розширення для відкриття нового функціоналу в WordPress додатках.

1.2. Головні вразливості веб-додатків

Відкритість WordPress породжує можливості для великої кількості різнорідних вразливостей в системі безпеки платформи, на які ні в якому разі не можна закривати очі. Також необхідно відмітити, що кінцевий користувач не може покластися на вбудовані системи безпеки, тому необхідно приділяти особливу увагу до встановлених тем та розширень, т. я. вони можуть створити нові непередбачені вразливості в веб-додатку.

Головними вразливостями WordPress є:

1.2.1. Атака грубою силою.

Як сказано в [7], атака грубою силою – найпростіший з методів отримати доступ до сайту. Метод полягає в підборі паролю та імені користувача. Є «грубим», іноді неефективним, але показує результати коли користувачі використовують паролі по типу «12345678» та імена користувача схожі з «admin». Атака грубою силою завжди намагається скористуватися найбільш слабою частиною кожної системи – самою людиною. Характерною особливістю цієї атаки є те, що пам’ять сервера швидко вичерпується через велику кількість http-запитів. Без застосування відповідних мір до цього типу атаки вразливі всі веб-додатки, а не тільки зроблені на WordPress. Виходячи з [8], сайти, зроблені за допомогою WordPress є пріорітетною ціллю для зловмисників через ряд причин, а саме:

- WordPress – найпопулярніша WCMS у світі. Під час написання роботи, більше ніж 40% всіх сайтів використовують її [9].
- По замовчанню WordPress дозволяє безкінечну кількість спроб підключення.

- Простота пошуку сторінки для підключення. По замовчанню вона буде по наступній адресі: «yoursite.com/wp-admin». Зміну адреси необхідно проводити вручну, до чого доходять не всі «міленькі» розробники.

Сайти, зроблені з допомогою WordPress, легко виявити. Як сказано в [10], якщо користувач не увійшов до робочого кабінету та намагається потрапити на сторінку «/wordpress/wp-admin/», то він буде перенаправлений на сторінку «/wp-login.php?redirect_to=http%3A%2F%2Fexample.com%2Fwordpress%2Fwp-admin%2F&reauth=1». Ця особливість дозволяє виявити WordPress сайти через спроби входу на сторінку «/wordpress/wp-admin/».

Коли зловмисник знайшов WordPress сайт, він може провести атаку грубою силою за допомогою XSHM, навіть якщо до сайту нема прямого доступу. Згідно з [11], Cross Site History Manipulation (XSHM) це порушення безпеки, основане на том, що об’єкт на стороні клієнту не відповідає Політикам походження, що в свою чергу дає можливість через відстеження змін в об’єкті, відстежувати дії користувача клієнту. Метод з використанням XSHM застосовується для атаки через факт, що WordPress не має токенів на формах входу які могли би виключити спроби логіну через підробку запитів користувача за допомогою CSRF. Метод CSRF полягає в відправці запитів від лиця користувача. Для успішного виконання операція, яка виконується від лиця користувача, не повинна потребувати додаткових підтверджень з його сторони, наприклад таких як лист на електронній пошті або код доступу в банківському додатку. Після всіх приготувань, зловмисник може використовуючи шкідливий сайт для здійснення CSRF-атаки на основі GET параметрів які включають в себе різні комбінації ім’я користувача та паролю. Після визначення вірної комбінації та отримання прав адміністратора, зловмисник може використовувати отриманий веб-

сайти для різних цілей, таких як переадресація користувачів на шкідливі сайти тощо.

1.2.2. SQL інєкція.

Як сказано в [12], SQL (Structured Query Language – мова структурованих запитів) - це мова яка дозволяє взаємодіяти з базами даних. Сучасні веб-додатки використовують бази даних для динамічного відображення контенту та керування даними. Ін’єкція SQL - вразливість безпеки веб-додатків, яка дозволяє зловмиснику маніпулювати запитами, які додаток відсилає базі даних для отримання інформації з неї. Маніпулювання запитами дозволяє зловмиснику отримати з бази даних будь яку інформацію, додати новий або змінити існуючий запис, і навіть видалити всю базу даних, що призводить до збоїв в роботі додатку або зміни його поведінки [13]. Іноді зловмисник може використати SQL ін’єкцію для виконання атаки відказу в обслуговуванні або компрометування базового серверу. Додатки зроблені за допомогою WordPress стійкі до атаки SQL ін’єкцією тільки якщо всі файли та WordPress оновлені до останніх версій, що роблять не всі розробники. Також на вразливість додатку до SQL ін’єкції впливає наявність тем та розширень зроблених третьою стороною. Також зловмисник може отримати права адміністратора через SQL ін’єкцію.

1.2.3. Міжсайтові скрипти.

Виходячи з [14], міжсайтові скрипти (XSS) – один з найбільш розповсюджених тип атаки при якому зловмисник вводить шкідливий код JavaScript, який при завантаженні на стороні клієнту таємно збирає різноманітні данні о користувачі або перенаправляє його на шкідливі сайти. Сайти на WordPress особливо вразливі до міжсайтових скриптів якщо вони містять в собі багато різноманітних і комплексних

розширень. Чим більше и складніше розширення тим більше вірогідність того, що його автор допустив критичні помилки при розробці в відкрив шлях зловмиснику. Також до групи ризику потрапляють користувачі старих версій WordPress та застарілих розширень.

У XSS є багато різних застосувань, але згідно з [15] можна виділити п’ять основних і найрозповсюдженіших способів застосування вразливості. Основними методами використання є:

- Викрадення сеансу користувача

Більшість сайтів використовують сесії в якості унікальної мітки для визначення кожного окремого користувача. Ці сесії зберігаються в cookie файлах. За допомогою XSS зловмисник може викрасти ідентифікатор сесії користувача з cookie файлів. Після чого, використовуючи отриману інформацію про сесію користувача, заходить до особистого кабінету жерти без необхідності вводити пароль. В результаті чого шахраю стають доступні всі функції викраденого акаунту, а в деяких випадках навіть банківські карти жертви.

- Виконання неавторизованих дій

Як і в попередньому випадку, зловмисник за допомогою міжсайтових скриптів може викрасти cookie файли користувача. Після чого за допомогою отриманого ідентифікатора користувача можна викласти на сайт повідомлення (якщо функціонал сайту дозволяє). Цей тип атаки може погіршити роботу сайту або навіть заразити інших користувачів шкідливим програмним забезпеченням.

- Початок інших атак

В декількох випадках міжсайтовий скриптинг використовується для початку більших атак або направлення користувачів сайту на інші сайти, як правило, шахрайські. Ця схема дуже розповсюджена, тільки за першу половину 2020 року було виявлено більше 310 тисяч сайтів, що підпали під неї [16].

- Встановлення клавіатурного шпигуна на сайті

В цьому випадку зловмисник встановлює на сайті клавіатурний шпигун, що використовується для сбіру інформації про нажаті користувачем сайту кнопки на клавіатурі. Кожен раз коли користувач натискає кнопку, шпигун записує її, збирає додаткову інформацію, та відправляє її назад до зловмисника.

Завдяки цьому способу можна дізнатись багать конфіденційної інформації, такої як ім’я та пароль користувача для авторизації на сайті, номера та паролі банківських карт тощо.

- Викрадення важливої інформації

Цей спосіб має багато спільного з викраденням cookie файлів з першого випадку. З використання скриптів зловмисник може війти в особистий кабінет жертви і отримати всі функції та інформацію, що доступні тільки цьому користувачу.

Міжсайтові скрипти можуть бути надзвичайно небезпечними та нанести багато шкоди сайту та репутації власнику. Скоріше за все, сайти банківських компаній та крупних фірм мають добрий захист перед XSS і викрасти будь які данні звідти надзвичайно важко або взагалі неможливо. Головною ціллю XSS частіше за все стають малі та середні фірми, які використовують застарілі версії WordPress та розширень, а також не приділяють достатньої уваги забезпеченню безпеки своїх користувачів.

1.2.4. Шкідливе програмне забезпечення.

Виходячи з [17], шкідливе програмне забезпечення розміщується на сайті для зараження комп'ютерів відвідувачів сайту шкідливим програмним забезпеченням. Цілі можуть різнитися від сайту до сайту. Існує декілька основних шляхів, по яким шкідливе програмне забезпечення може потрапити на сайт:

- Проблеми з безпекою сайту та контролем доступу

  Якщо контроль доступу сайту було не правильно налаштовано, зловмисник міг отримати доступ до сайту через нього. Частіше за все використовуються методи атаки грубою силою та міжсайтовий скриптинг, які були розглянуті раніше.

- Вразливості програмного забезпечення

  Використання неякісного, застарілого та несертифікованого програмного забезпечення може привести до вразливості, що в результаті призведе до розміщення шкідливого програмного забезпечення зловмисником на сайті. Також під ризик підпадає і нове програмне забезпечення та додатки, які теж можуть викликати серйозні проблеми з безпекою в веб-додатку.

- Розширення та сценарії, зроблені третьою стороною

  Використання непровірених розширень та сценаріїв може призвести до проблем з безпекою на сайті, і я результат, до розміщення шкідливого програмного забезпечення.

- XSS вразливість сайту

Використовуючи міжсайтовий скриптинг зловмисник може отримати доступ до верифікованого кабінету користувача і розмістити посилання на шахрайські сайти (якщо функціонал сайту дозволяє). Також можливо скористатись XSS вразливістю разом з не правильно налаштованим контролем доступу для розміщення шкідливого програмного забезпечення.

- Зараження на серверному рівні

Зараження на серверному рівні особливо небезпечне через те, що може торкнутися всіх сайтів, що користуються зараженим сервером. Зараження може відбуватися через різні проблеми з безпекою серверу, такі як не правильно налаштований firewall, старі версії антивірусу та серверного програмного забезпечення, погано налаштований мережевий екран тощо. Початкове зараження може відбуватися через завантажений заражений файл з сайту або іншого джерела.

- Соціальна інженерія

Зловмисники можуть підштовхнути власника веб-додатку на встановлення шкідливого програмного забезпечення за допомогою введення в оману та інших розповсюджених практик соціальної інженерії.

Шкідливе програмне забезпечення наносить багато шкоди як і інфікованому сайту, так і користувачу, що зайшов на нього. Захисту від шкідливого програмного забезпечення потрібно приділяти особливу увагу, т. я. уражений сайт продовжує інфікувати нових і нових користувачів, уражаючи нові сайти та викликаючи цілі епідемії.

Основними причинами є використання застарілих версій розширень та програмного забезпечення, інші вразливості веб-додатку та соціальна інженерія, направлена на власника.

1.2.5. DDoS атаки.

Згідно з [18], DDoS атака це атака, яка направлена на навантаження серверу або мережі надлишковим трафіком, що може привести до зниженню продуктивності або к повному відключенню. Атаку DDoS легко замаскувати та з нею дуже важко боротися. Наявність на сайті розширень та сторонніх сценаріїв підвищують вразливість до DDoS атак. Також на вразливість до атак впливають застарілі версії WordPress та PHP.

1.2.6. Старі версії WordPress та PHP.

Застарілі версії додатків, WordPress та PHP більш схильні до загроз безпеці. З часом зловмисники знаходять нові способи здійснити атаку на сайти, а розробники постійно підвищують захист в нових версіях програмного забезпечення. Використовуючи застарілі версії власник підвергає себе додатковому ризику. Згідно з офіційною статистикою, більше 10% людей користуються застарілими версіями WordPress [9], а застарілі версії PHP - більше 9% людей [19]. Також вся небезпека використання застарілих версій була освітлена в попередніх пунктах огляду загроз безпеці веб-додатків.

1.3. Висновки

Веб-додатки можуть бути атаковані як на клієнтській, так і на серверній стороні, включаючи і третю сторону, яку приймає участь в

процесі передачі та отримання інформації веб-браузері клієнту. Сам по собі веб-браузер клієнту не є ні безпечною, ні довіреною стороною обміну інформацією.

Для полегшення та прискорення процесу створення свого додатку існує велика кількість готових систем керування веб-змістом (WCMS). Веб CMS надають велику кількість різноманітних інструментів для керування та відображення графічного і текстового наповнення додатку. Доступність і легкість в освоєнні роблять системи керування веб-змістом привабливими для малих компаній та незалежних розробників. Одним із популярних WCMS є WordPress.

Відкритість WordPress породжує можливості для великої кількості різнорідних вразливостей в системі безпеки платформи, на які ні в якому разі не можна закривати очі. Також необхідно відмітити, що кінцевий користувач не може покластися на вбудовані системи безпеки, тому необхідно приділяти особливу увагу до встановлених тем та розширень, т. я. вони можуть створити нові непередбачені вразливості в веб-додатку.

Головними вразливостями WordPress є атака грубою силою, SQL інєкція, міжсайтові скрипти, шкідливе програмне забезпечення, DDoS атаки та Старі версії WordPress і PHP.

2. АНАЛІЗ МЕТОДІВ ПІДВИЩЕННЯ БЕЗПЕКИ ВЕБ-ДОДАТКІВ

2.1. Загальний аналіз методів забезпечення безпеки

Веб-додатки грають велику роль в житті людей. Користувачі довіряють сайтам свої особисті данні, номери кредитних карток, номери телефонів, паролі та велику кількість різноманітних важливих даних, а від безпеки цих додатків та ввірених їм даних, часто, залежить навіть життя людей.

Безпека веб-додатків – одна з найважливіших частин розробки будь якого додатку, про що дуже часто забувають. Разом з розвитком веб технологій розвиваються і методи атак та пошуку вризливостей сайтів. Більше 95% веб-сайтів використовують HTML5 та JavaScript – мови, які легко перехопити, зламати та переглянути, що в свою чергу робить веб-додатки вризливими на стороні клієнту, особливо якщо безпеці не приділяється багато уваги. Звіт Radware, поставника послуг кібербезпеки, о стану безпеки веб-додатків показав, що більше 98% сайтів підверглись спробам атаки [20], нажаль 85% з цих випадків включають людський фактор. Інше дослідження зазначає що приблизно 50% випадків витоку даних відбуваються на рівні веб-додатку [21]. Також слід зазначити що навіть шифрування залишається вразливою частиною. Ключі шифрування в веб-додатках дуже обмежені. Більшість браузерів не мають вбудованих технологій для зберігання та передачі ключів шифрування. Безпека веб-додатку – ключовий компонент будь якого бізнесу, що покладаєтсья на свій сайт.

Атаки проти веб-додатків мають багато методів і хитрощів для обходження захисту. До них можна віднести маніпуляції з базами даних, міжсайтовий скриптинг, модифікації існуючих або створення нових прав доступу для користувача тощо.

Часто розробники веб-додатків приділяють мало уваги безпеці додатків. Команди розробки витрачають всі свою сили на візуальний дизайн та функціональність додатку, що безумовно є важливими частинами будь якого додатку, але пожертування безпекою на їх догоду завжди призводить великих проблем у майбутньому.

Можна зробити закономірний висновок що випадкової безпеки недостатньо. Для того, щоб зробити додаток стійким до атак різного типу, потрібно забезпечити систематичній захист самого веб-додатку, серверу, а також мережі. Абсолютної безпеки додатку неможливо досягти, але до неї потрібно прагнути постійно покращуючи додаток та його безпеку.

Визначавши основні загрози в безпеці веб-додатку, створеному за допомогою WordPress, можна надати рекомендації щодо забезпечення безпеки та зниження ризику підвергнутися кожному типу атаки.

2.2. Захист від атак грубою силою

Як вже було зазначено, атака грубою силою – найпростіший з методів отримати доступ до сайту. Метод полягає в підборі паролю та імені користувача. Є «грубим», іноді неефективним, але показує результати коли користувачі використовують паролі по типу «12345678» та імена користувача схожі з «admin». Офіційний сайт WordPress зазначає такі методи забезпечення захисту від атаки грубою силою як:

2.2.1. Використовувати надійні ім’я користувача та пароль.

Більшість атак в першу чергу перевіряють ім’я користувача «admin» через те, що в ранніх версіях WordPress воно використовувалось за замовченням. Рекомендується змінити стандартне ім’я на своє за допомогою створення нового облікового запису та перенесення всіх повідомлень.

Використання надійного пароля дозволить захистити веб-додаток від спроб входу за допомогою атаки грубою силою. Більшість браузерів мають вбудовану функцію генерації паролю. Її можна використовувати для створення дійсно надійного паролю. Але згенерований пароль дуже важко запам’ятати, а функцію запам’ятовування паролю не завжди можна використовувати. Тому самий кращій пароль знаходиться в балансі між надійністю та складністю запам’ятовування. Для перевірки свого пароля WordPress надає вимірювач складності пароля, що допомагає переконатися в надійності вигаданого пароля.

Для створення надійного пароля необхідно уникати будь яких перестановок справжніх імен, слів із словника будь якої мови, короткого пароля, або пароля, який містить тільки букви або тільки цифри.

2.2.2. Використання розширень для захисту

Атаки грубою силою користуються популярність через те, що за замовченням WordPress дозволяє необмежену кількість невдалих спроб логіну до сайту. Існує велика кількість розширень для WordPress для закриття цього недоліку та обмеження кількісті невдалих спроб на вхід. Повний список можна знайти на офіційному сайті WordPress [22]. Також там можна знайти розширення для повного заборони доступу людей до сторінки «wp-login.php».

2.2.3. Захист серверу

Якщо для захисту «wp-login.php» сторінки обрано спосіб з повним закриттям доступу, може виникнути ситуація, коли доступ до сторінки повністю заблокований, навіть для самого розробника. Для запобігання цього необхідно встановити стандартний документ для переходу при помилці 401.

2.2.4. Парольний захист сторінки «wp-login.php»

Для додаткового захисту можна захистити файл «wp-login.php» за допомогою паролю, створеним згідно з рекомендаціями про створення складного і надійного паролю. Також існує можливість захисту всієї папки «wp-login», хоча це і не рекомендується робити в деяких випадках, через те, що парольний захист папки призведе до збоїв в роботі розширень які використовують ajax в якості зовнішнього інтерфейсу. Тому для додаткового захисту достатньо захистити тільки саму сторінку.

2.2.5. Обмеження кількості спроб логіну до «wp-login.php» по IP

Якщо декільком людям потрібно мати доступ до сторінки «wp-login.php», то можна обмежити доступ до сторінки на підставі IP-адреси користувача. Цей метод потрібно використовувати з крайньою обережністю, так як провайдер може часто змінювати адресу користувача, тож метод доцільно використовувати тільки тоді, коли всі люди, яким потрібен доступ, використовують тільки статичну IP-адресу.

Можна сказати, що забезпечити захист від атак грубою силою не дуже складно, але якщо ігнорувати вразливості додатку, це може привести до збоїв в роботі додатку та відкриттю вразливостей до інших типів атак.

2.3. Захист від SQL ін’єкції

Як вже було зазначено, SQL (Structured Query Language – мова структурованих запитів) - це мова яка дозволяє взаємодіяти з базами даних. Сучасні веб-додатки використовують бази даних для динамічного відображення контенту та керування даними. Ін’єкція SQL - вразливість безпеки веб-додатків, яка дозволяє зловмиснику маніпулювати запитами, які додаток відсилає базі даних для отримання інформації з неї. WordPress хостинг Kinsta визначає такі методи захисту від SQL ін’єкції:

2.3.1 Екранувати данні які вводить користувач

Перевірити чи є запит, який ввів користувач, шкідливим, задача дуже складна. Тому найкращім варіантом буде екранування даних, які ввів користувач. Екранування дозволить перевести керуючі символи до їх строкового аналогу, що дозволить уникнути атаки з використанням SQL ін’єкції.

2.3.2. Використовувати заготовлені запити

В якості альтернативи екрануванню даних користувача можна використовувати заготовлені параметризовані запити. Заготовлений параметризований запит - це шаблон SQL запиту в якому на пізньому етапі вказані параметри для його виконання. Значення параметрів вказує користувач.

Основними причинами вразливості веб-сайтів, створених на платформі WordPress, можна назвати застарілі версій PHP, WordPress або MySQL, використання тем та розширень, зроблених третьою стороною а також не правильне налаштування доступу в базі даних SQL.

2.4. Захист від міжсайтових скриптів

Як вже було зазначено, міжсайтові скрипти (XSS) – один з найбільш розповсюджених тип атаки при якому зловмисник вводить шкідливий код JavaScript, який при завантаженні на стороні клієнту таємно збирає різноманітні данні о користувачі або перенаправляє його на шкідливі сайти.

Найпростіший спосіб захиститись від міжсайтового скриптингу – це відключити JavaScript в браузері. Тоді XSS, ціллю яких є JavaScript, не будуть мати ніякої сили. Але це требі робити на стороні клієнту, тобто на стороні користувача сайту.

Для адміністраторів веб-додатків існує декілька основних методів запобігання XSS. Перед тим, як сервер прийме дані користувача, вони повинні бути вивчені. Самий безпечний метод полягає в налаштування білого списку, який буде відображати які данні серверу можна приймати. Це забезпечить надійний захист від міжсайтового скриптингу.

Також в виводу даних с серверу потрібен захист. В цьому випадку необхідно замінити проблемні метасимволи на текстові посилання для того, щоб метасимволи читалися як посилання і потенційно шкідливі файли не могли буди запущені на стороні серверу.

Мережеві екрани також забезпечують захист від простих XSS-атак.

Шкоду, яку може принести вразливість сайту до XSS не можна недооцінювати. Користувач може ризикувати своїми особистими даними або стати співучасником злочину. Адміністратори веб-додатків повинні розуміти, що вони несуть відповідальність за безпеку особистих даних користувачів.

2.5. Захист від шкідливого програмного забезпечення

Як вже було зазначено, шкідливе програмне забезпечення розміщується на сайті для зараження комп'ютерів відвідувачів сайту шкідливим програмним забезпеченням. Цілі можуть різнитися від сайту до сайту.

Першим кроком до захисту веб-додатку від шкідливого програмного забезпечення є своєчасне оновлення всіх компонентів і розширень. Це відноситься до WordPress, PHP, всі встановлених розширень та тем додатку. Застарілі версій мають більше шансів містити в собі недоліки та вразливості, що призводить до вразливості додатку. Особливої уваги потребують оновлення WordPress які позначені як «Оновлення безпеки». Вони назначені для покращення захисту додатку та усунення вразливостей.

Однією з головних вразливостей WordPress є його сторінка входу на сайт. Більшість зловмисників намагаються отримати доступ до сайту саме через цю сторінку. Раніше були розглянуті методи захисту сторінки входу. Також можна піти далі та включити двофакторну аутентифікацію за допомогою додатків, що підвищить захист сторінки ще більше.

Також необхідно регулярно створювати резервні копії свого додатку для відновлення працездатності в разі непередбачених обставин.

Захист від шкідливого програмного забезпечення – це одна з найважливіших задач при створенні добре захищеного веб-додатку. Використання WordPress полегшує цю задачу, але все одно він є вразливим якщо не прийняти відповідних мір.

2.6. Захист від DDoS атак

Як вже було зазначено, DDoS атака це атака, яка направлена на навантаження серверу або мережі надлишковим трафіком, що може привести до зниженню продуктивності або к повному відключенню.

Головними методами захисту WordPress додатку від DDoS атак є встановлення розширення, націленого на захист від DDoS. Також в усуненні вразливості може допомогти WAF (Website Application Firewall) [23].

Також на вразливість до DDoS атак впливають застарілі версії WordPress та PHP, тож рекомендується оновити всі можливі компоненти та розширення.

2.7. Небезпека використання застарілих версій WordPress та PHP

Першим кроком до захисту веб-додатку є своєчасне оновлення всіх компонентів і розширень. Це відноситься до WordPress, PHP, всі встановлених розширень та тем додатку. Застарілі версій мають більше шансів містити в собі недоліки та вразливості, що призводить до вразливості додатку. Особливої уваги потребують оновлення WordPress які позначені як «Оновлення безпеки». Вони назначені для покращення захисту додатку та усунення вразливостей.

2.8. Висновки

Веб-додатки грають велику роль в житті людей. Користувачі довіряють сайтам свої особисті данні, номери кредитних карток, номери телефонів, паролі та велику кількість різноманітних важливих даних, а від безпеки цих додатків та ввірених їм даних, часто, залежить навіть життя людей.

Безпека веб-додатків – одна з найважливіших частин розробки будь якого додатку, про що дуже часто забувають.

Часто розробники веб-додатків приділяють мало уваги безпеці додатків. Команди розробки витрачають всі свою сили на візуальний дизайн та функціональність додатку, що безумовно є важливими частинами будь якого додатку, але пожертування безпекою на їх догоду завжди призводить великих проблем у майбутньому.

Дотримуючись рекомендацій та приділяючи достатню увагу захисту можна значно знизити, а в деяких випадках і повністю усунути вразливість додатку до різних типів атак, а також захистити особисті данні користувачів і свою репутацію від зловмисників.

3. ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ

3.1. Дослідження методів усунення вразливості до атаки грубою силою

Атака грубою силою – найпростіший з методів отримати доступ до сайту. Метод полягає в підборі паролю та імені користувача. Для демонстрації було обрано WordPress версії 5.8.2. В якості веб-серверу обрано локальний сервер OpenServer версії 5.3.7.

Після встановлення WordPress та вибору бази даних йде сторінка встановлення назви сайту, ім’я користувача та паролю. Сторінка встановлення назви сайту, ім’я користувача та паролю показано на рисунку 3.1.1.

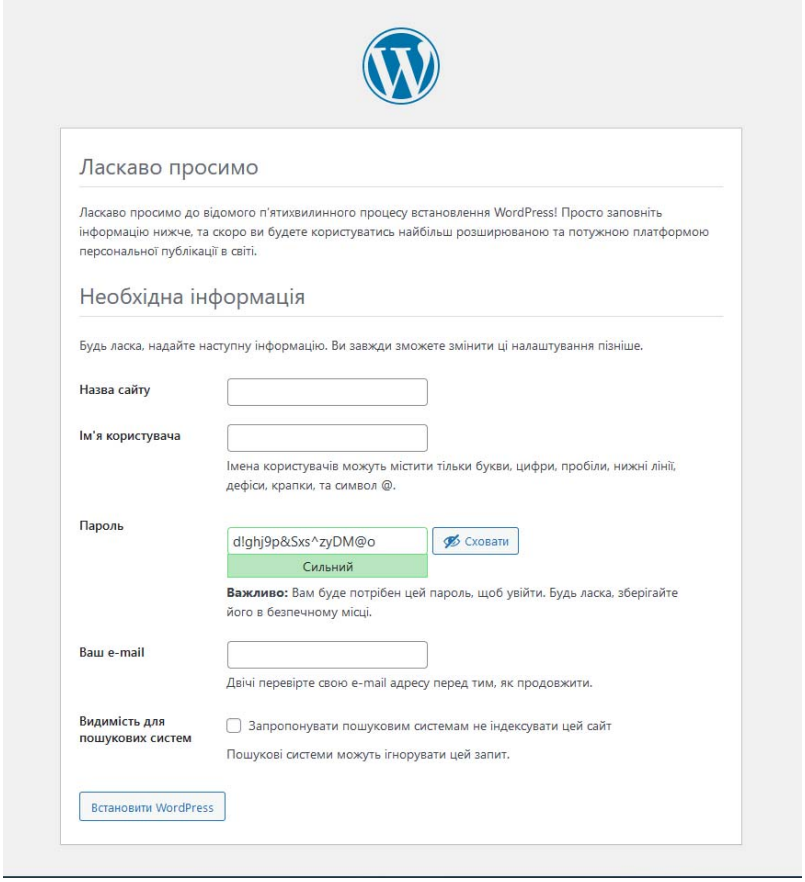


Рисунок 3.1.1 - Сторінка встановлення назви сайту, ім’я користувача та паролю

Як видно з рисунку 3.1.1, WordPress версії 5.8.2 пропонує користувачу сгенерований пароль, який містить в собі великі і малі літери, цифри а також спеціальні символи. Також проводиться перевірка складності паролю. Сгенерований пароль є дуже надійним, але запам’ятати його дуже важко, тому треба вигадати свій пароль. Якщо обрано слабкий пароль, то необхідно підтвердження. Для експерименту обрано ім’я користувача «admin» та пароль «123456789».

Після встановлення ім’я користувача та паролю сайт готовий до використання. Сайт без редагування та доповнень містить головну сторінку, а також сторінку входу, які приведено на рисунках 3.1.2 та 3.1.3 відповідно.

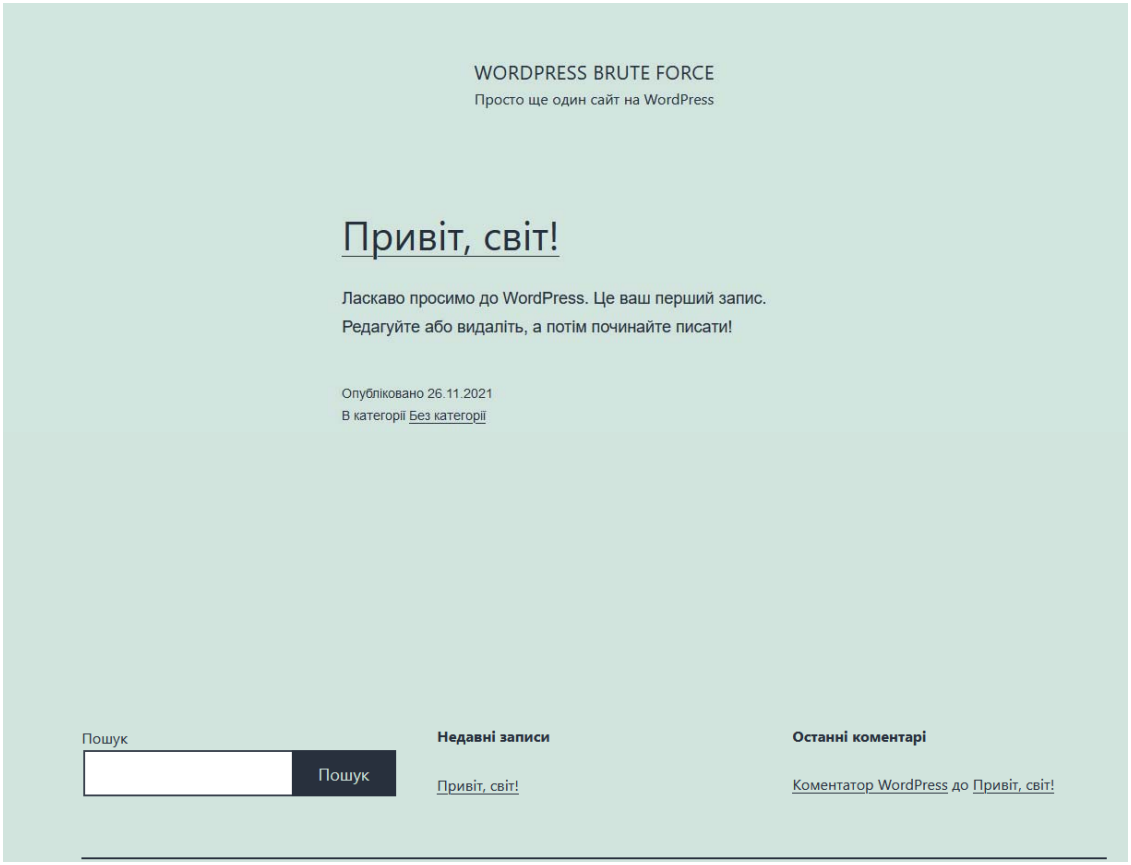


Рисунок 3.1.2 – Головна сторінка сайту

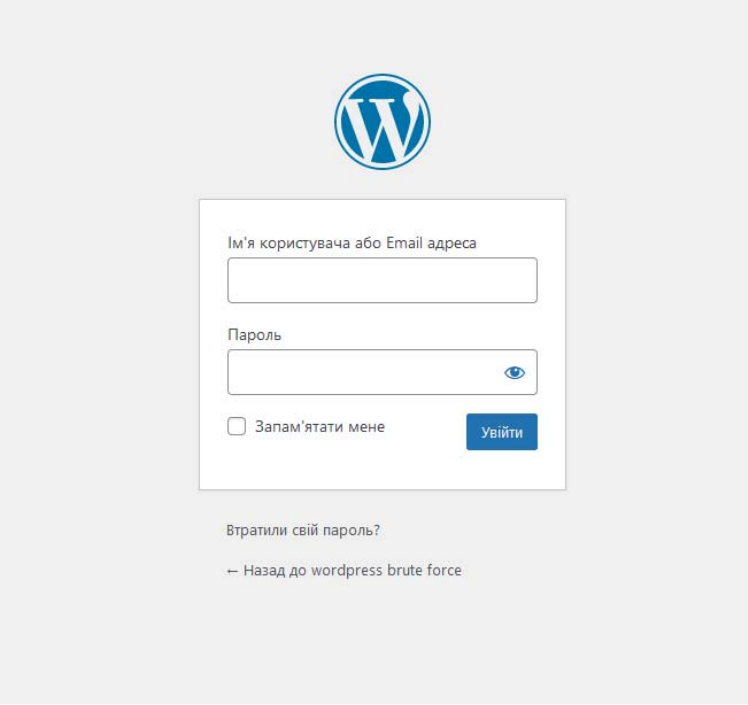


Рисунок 3.1.3 – Сторінка входу «wp-admin»

Після входу до запису адміністратора користувач потрапляє до майстерні. Загальний вид майстерні відображено на рисунку 3.1.4.

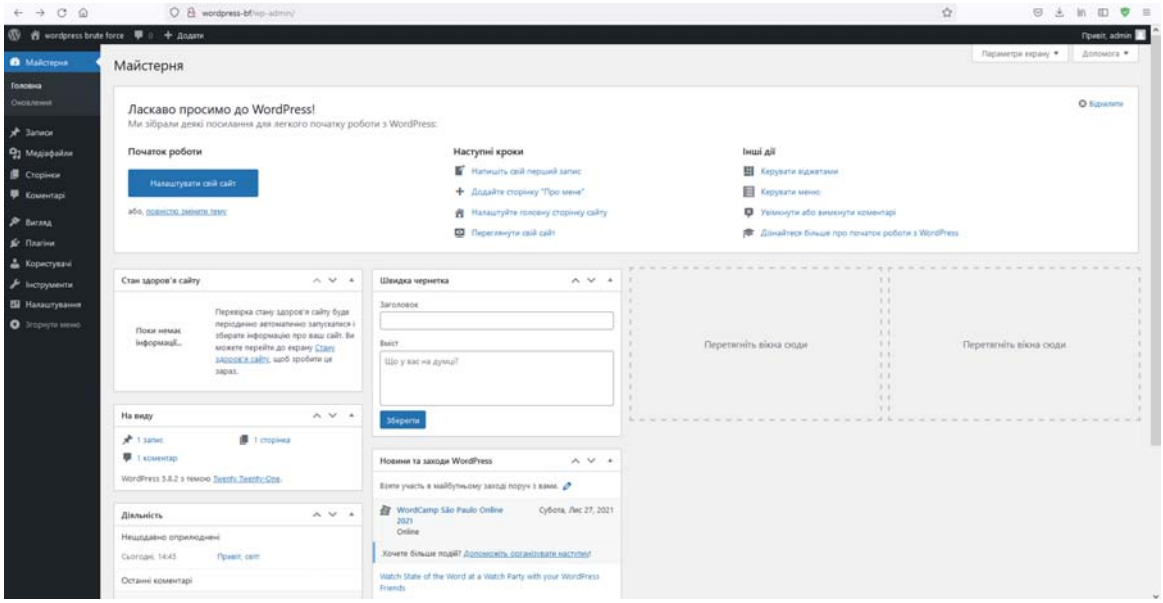


Рисунок 3.1.4 – Загальний вид майстерні

В майстерні можна переглядати, видаляти, додавати та редагувати записи, коментарі та сторінки. Також доступні функції налаштування головної сторінки, включення або виключення коментарів, керування меню тощо. Можна сказати що той, хто має доступ до запису адміністратора, може модифікувати та змінювати сайт як завгодно по своєму бажанню. Слід зазначити можливість зміни паролю та пошти, що означає, що зловмисник, отримавши доступ до запису адміністратору, може змінити ці данні на свої, а законний власник вже ніколи не отримає доступ до свого сайту.

Для проведення атаки грубою силою було використано утилиту WPForce. Як вказано в [24], WPForce – це набір інструментів проведення атак на WordPress. Зараз він містить 2 сценарії – WPForce для проведення атак грубою силою та Yertle, який завантажує шкідливі

сторінки. Також Yertle містить ряд модулів для полегшення експлуатації.

Також для проведення атаки грубою силою необхідні два словники – словник імен та словник паролів. Обидва словники можна легко знайти в інтернеті. До них включені найбільш звичайні паролі та імена користувачів.

Коли всі вхідні дані визначені можна починати атаку грубою силою. Вікно виконання WPForce показано на рисунку 3.1.5.

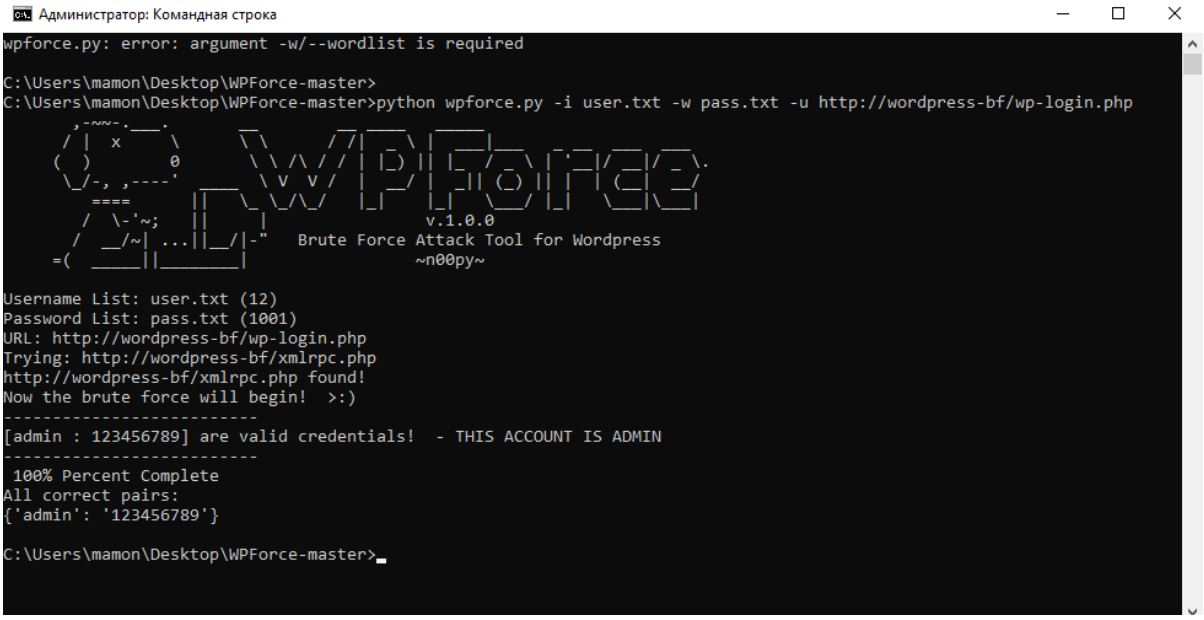


Рисунок 3.1.5 – Виконання програми WPForce

Як видно з рисунку 3.1.5, було знайдено підходящу пару логіну та паролю. Цією парою є admin : 1234556789.

Для підвищення захисту від атаки грубою силою рекомендується змінити пароль на складний а також встановити розширення для обмеження кількості невдалих спроб входу до запису адміністратора.

Для зміни паролю необхідно зійти до запису адміністратора, після чого увійти до розділу редагування профілю, де все можна змінити пароль на більш надійний, що і відображено на рисунку 3.1.6.

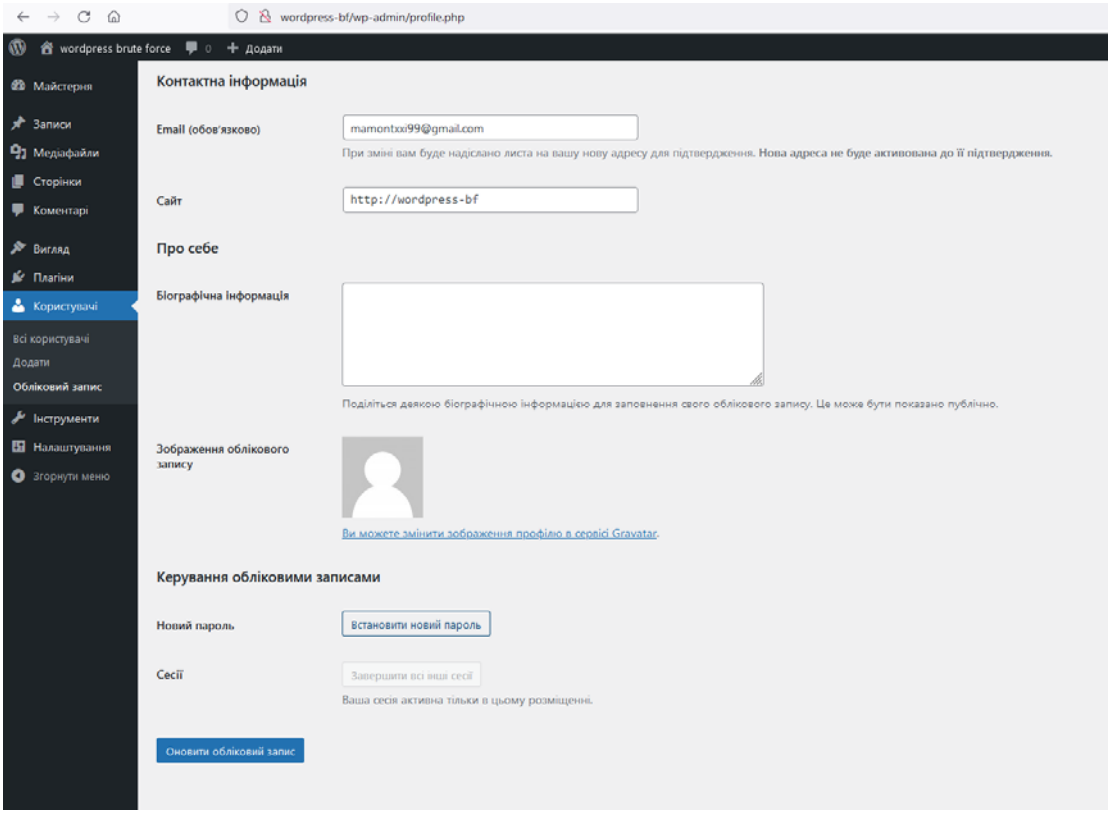


Рисунок 3.1.6 – Спосіб зміни паролю в особистому кабінеті

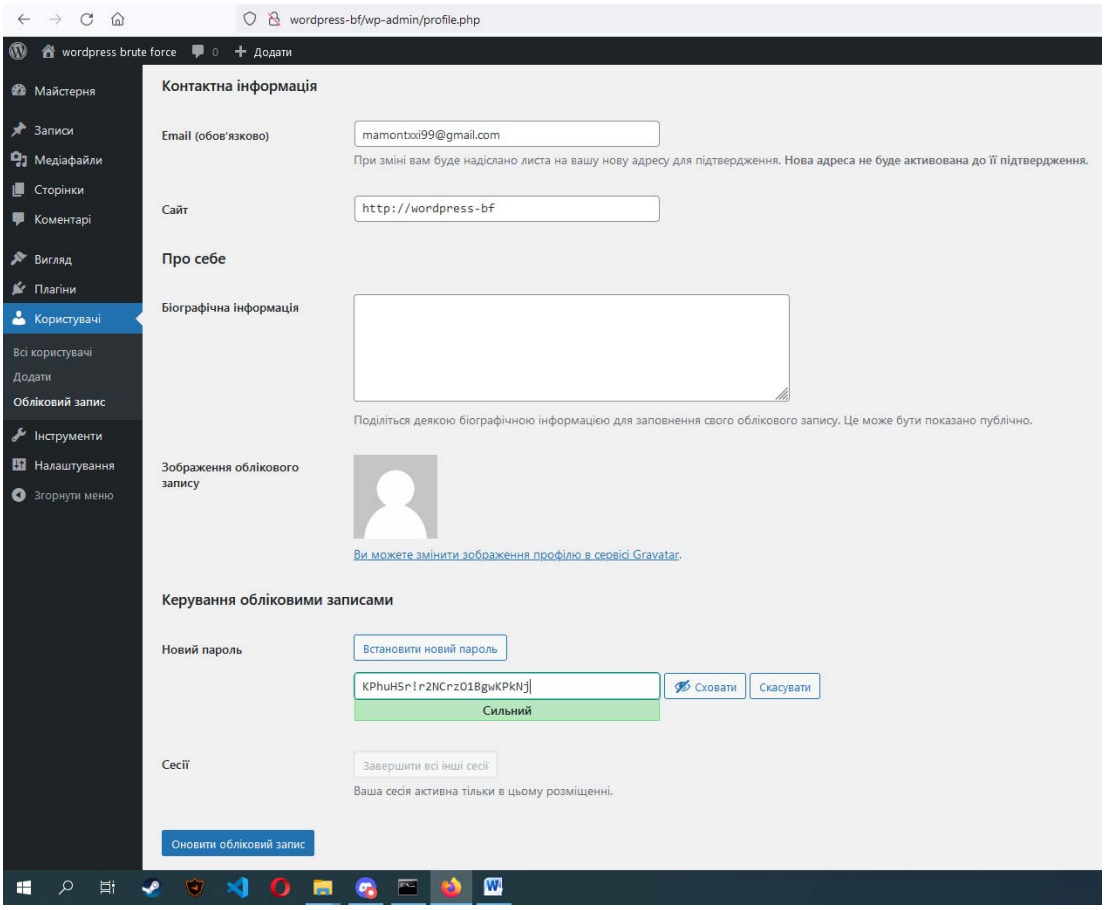


Рисунок 3.1.7 – Зміна паролю

Як видно на рисунку 3.1.7, після натискання кнопки зміну паролю WordPress знову пропонує сгенерований надійний пароль.

Після встановлення нового паролю використання WPForce не дасть ніякого результату, що відображено на рисунку 3.1.8. Для зламу такого запису методом грубої сили необхідно витратити набагато більше часу та сил.

Рисунок 3.1.8 – Атака грубою силою не вдалась

Для зламу паролю, який містить маленькі та великі літери латинського алфавіту, цифри а також складається з 8 символів, потрібно витратити більше 90 тисяч років безперервного перебору паролів.

Також для додаткового захиситу від атак грубою силою можна встановити розширення для обмеження кількості невдалих спроб входу до запису адміністратора. Для цих цілей було обрано розширення Limit Login Attempts від команди розробників WordPress. Як сказано в [25], Limit Login Attempts Reloaded зупиняє атаки грубою та оптимізує продуктивність сайту, обмежуючи кількість спроб входу в систему, які можливі при звичайному вході, а також через XMLRPC, Woocommerce та сторінки користувача входу. Це розширення буде блокувати Інтернет-адресу (IP) та/або ім'я користувача від подальших спроб після досягнення зазначеного ліміту повторних спроб, що зробить атаку методом грубої сили важкою чи неможливою. Після встановлення розширення кількість спроб для невірного вводу паролю дорівнює 4, а після їх вичерпання доступ до сторінки входу блокується на 20 хвилин, що робить будь які атаки грубою силою з використанням великих

словників неможливими. На рисунку 3.1.9 відображено вікно входу після встановлення розширення.

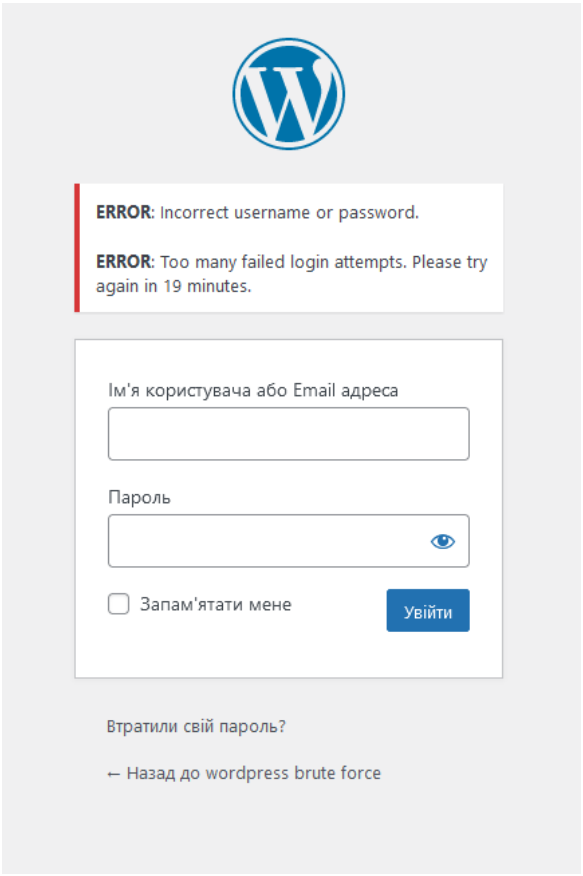


Рисунок 3.1.9 – Вікно входу після встановлення розширення Limit Login Attempts Reloaded

Тепер для проведення атаки грубою силою необхідно ще більше часу. В результаті після декількох нескладних кроків захист веб-додатку від атак грубою силою значно підвищився, а для складних паролів навіть зовсім стало неможливо виконати цей вид атак.

3.2. Дослідження методів усунення вразливості до SQL ін’єкції

Ін’єкція SQL - вразливість безпеки веб-додатків, яка дозволяє зловмиснику маніпулювати запитами, які додаток відсилає базі даних

для отримання інформації з неї. Для демонстрації було обрано WordPress версії 5.8.2. В якості веб-серверу обрано локальний сервер OpenServer версії 5.3.7. Для демонстрації було створено сторінку, яка виводить данні з бази даних на підставі номеру користувача. Загальний вид сторінки і таблиці бази даних наведено показано на рисунках 3.2.1 та 3.2.2.

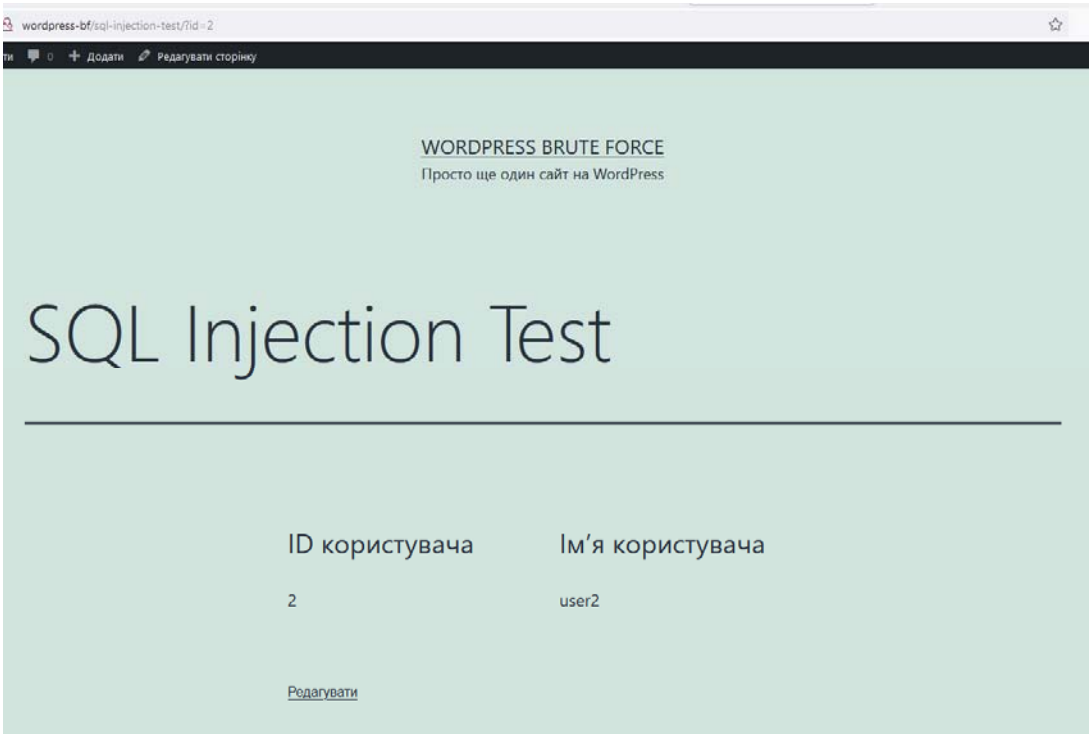


Рисунок 3.2.1 – Загальний вид сторінки тестування SQL ін'єкції

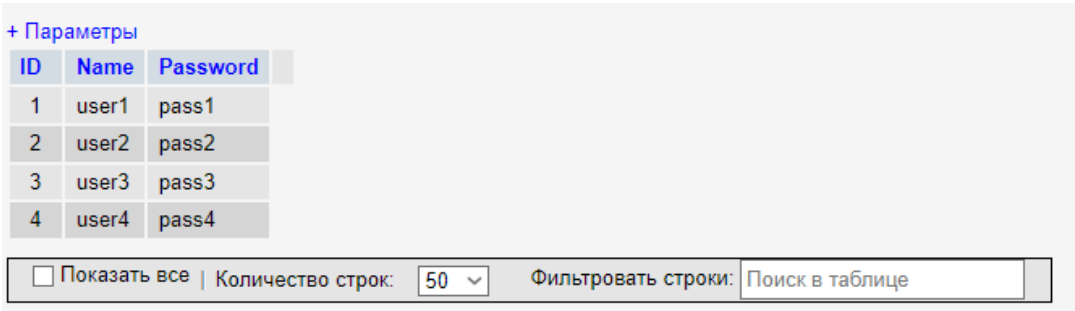


Рисунок 3.2.2 – Загальній вид таблиці бази даних

Для пошуку та експлуатації вразливостей використовується утиліта SQLMap. З офіційного сайту [26], SQLMap – це інструмент тестування на проникнення з відкритим кодом, який автоматизує

процес виявлення та використання недоліків SQL та захоплення серверів баз даних. Він поставляється з потужним механізмом виявлення, багатьма нішевими функціями для найкращого проникнення та широким набором перемикачів, що тривають від відбитків пальців бази даних, отримання даних з бази даних, до доступу до базової файлової системи та виконання команд в операційній системі через вихідні дані.

Спочатку необхідно провести попередньою розвідку, яка допоможе визначити вразливості сайту до SQL-ін'єкції, якщо така взагалі є. З цим допоможе команда, представлена на рисунку 3.2.3.

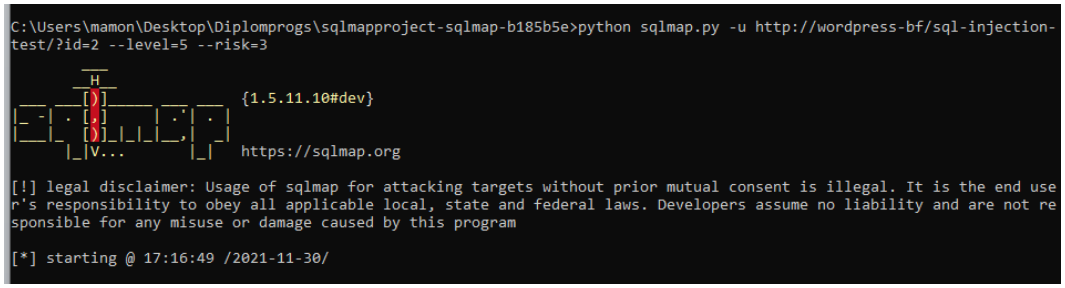


Рисунок 3.2.3 – Команда для попередньої розвідки

Після вдалого проведення розвідки необхідно визначити назви доступних баз даних. Команда для виявлення та результат її роботи приведено на рисунках 3.2.4 і 3.2.5 відповідно.

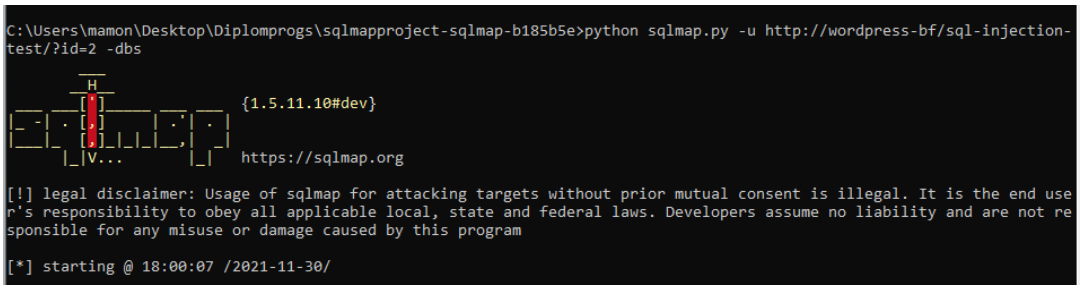


Рисунок 3.2.4 – Команда для визначення назв баз даних

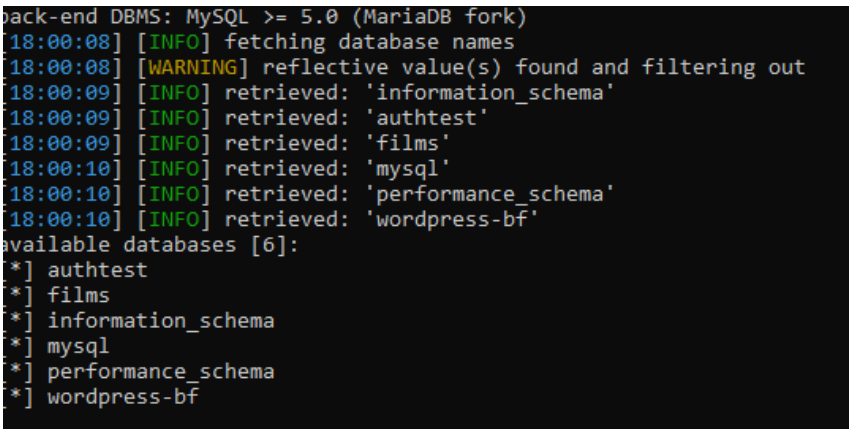


Рисунок 3.2.5 – Результат виконання команди

Після визначення назв всіх доступних баз даних необхідно визначити назви всіх таблиць цікавої для зловмисника бази даних. Для цього підходить команда відображена на рисунку 3.2.6. Результат її виконання показано на рисунку 3.2.7.

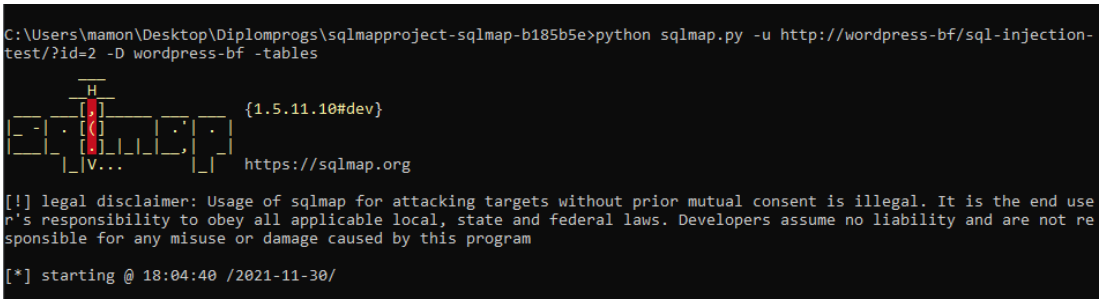


Рисунок 3.2.6 – Команда для визначення всіх таблиць конкретної бази даних

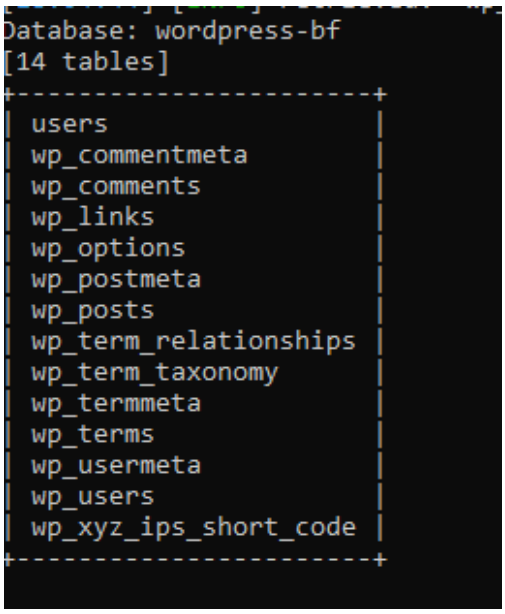


Рисунок 3.2.7 – Результат роботи команди для визначення назв всіх таблиць бази даних

Надалі існує можливість отримати всі данні з конкретної таблиці конкретної бази даних. Команда для цього відображена на рисунку 3.2.8. Кінцевий результат відображено на рисунку 3.2.9.

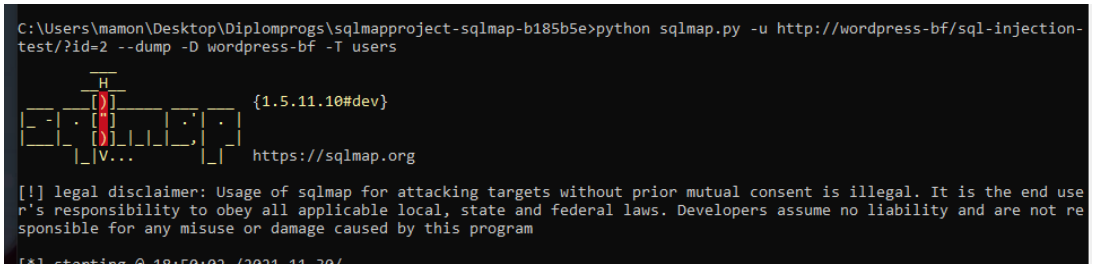


Рисунок 3.2.8 – Команда для отримання віх даних із таблиці

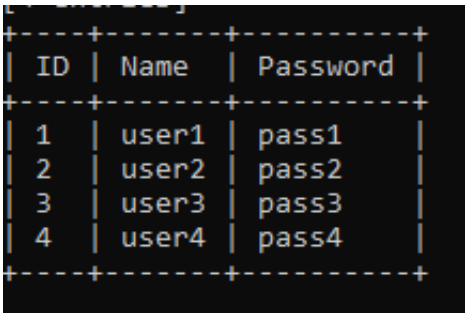


Рисунок 3.2.9 – Результат виконання команди

Як видно з рисунку 3.2.9, отримано повну таблицю імен користувачів та їх паролів. В реальних таблицях для підвищення безпеки зберігаються хеші паролів користувачів. Після отримання таблиці хешів користувачів зловмисник може спробувати розшифрувати хеші та встановити реальні паролі за допомогою різних програм для встановлення паролів, наприклад HashCat.

Через можливість SQL-ін’єкції в веб-додатку стався виток даних і конфіденційні данні користувачів потрапили до зловмисників. Для запобігання цього рекомендується виконати дії для закриття вразливості до SQL-ін’єкції.

Перш за все необхідно встановити брандмауер. Було обрано «Shieldfy». Від відрізняється від аналогів вбудованим антивірусом, а також широкими безкоштовними функціями.

Після встановлення брандмауеру також рекомендується замінити всі запити до бази даних на параметризовані. Нижче на рисунках 3.2.10 та 3.2.11 наведено неправильний і правильний способи формувати запити до бази даних.

```php
<?php
include_once("wp-config.php");
include_once("wp-includes/wp-db.php");
global $wpdb;

$id = $_GET['id'];
$result = $wpdb->get_row("SELECT Name FROM users WHERE ID = ", $id);
echo $result->Name;
?>
```

Рисунок 3.2.10 – Неправильний варіант формування запитів

```php
<?php
include_once("wp-config.php");
include_once("wp-includes/wp-db.php");
global $wpdb;

$id = $_GET['id'];
$wpdb->prepare("SELECT Name FROM users WHERE ID = %1$s", $id);

$result = $wpdb->get_row($wpdb->prepare("SELECT Name FROM users WHERE ID = %s", $id)
);
echo $result->Name;
?>
```

Рисунок 3.2.11 – Правильний варіант формування запитів

Як видно з рисунків 3.2.10 та 3.2.11 головна різниця полягає в використанні методу «prepare» в правильному варіанті. Метод «prepare» дозволяє виключити можливість доступу до запиту та його модифікації чи пошкодження зловмисником, що дозволить уникнути SQL-ін’єкції.

Після проведення всіх робот по закриттю вразливостей до SQL-ін’єкції необхідно провести повторне тестування утилітою SQLMap. Результат тестування наведено на рисунку 3.2.12.

Як видно з рисунку 3.2.12, утиліта SQLMap не знайшла жодних вразливостей до SQL-ін’єкції.

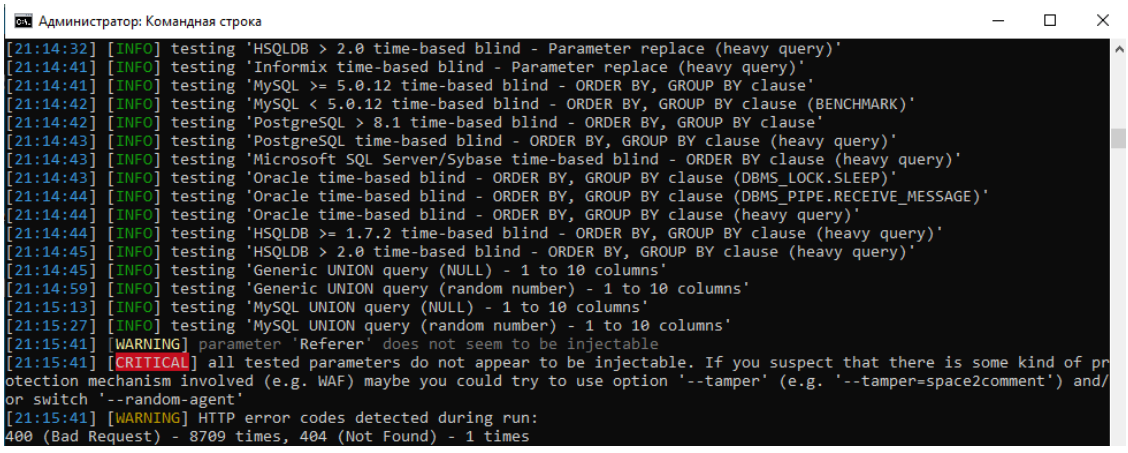


Рисунок 3.2.12 – Результати повторної перевірки утилітою SQLMap

В результаті виконання декількох нескладних кроків захист додатку значно виріс, а популярна утиліта для пошуку вразливостей до SQL-ін’єкції не знайшла жодного вразливого параметру.

3.3. Дослідження методів усунення вразливості до міжсайтового скриптингу

Міжсайтові скрипти (XSS) – один з найбільш розповсюджених тип атаки при якому зловмисник вводить шкідливий код JavaScript, який при завантаженні на стороні клієнту таємно збирає різноманітні данні о користувачі або перенаправляє його на шкідливі сайти. Для демонстрації було обрано WordPress версії 5.8.2. В якості веб-серверу обрано локальний сервер OpenServer версії 5.3.7. Також необхідно зазначити, що для демонстрації роботи методів експлуатації XSS вразливості веб-додатку було відключено антивірус та брандмауер, які було встановлено раніше для захисту від SQL-ін’єкції. Для вивчення отриманих даних та автоматичної генерації корисних даних було використано сервіс XSS Hunter. Сервіс XSS Hunter дозволяє досліджувати вразливості сайту до XSS, збирати інформацію про вразливості та відправляти з листи на електронну пошту з рекомендаціями адміністраторам сайту.

Після створення особистого запису на XSS Hunter користувачу стає доступна велика кількість різноманітних корисних даних для тестування сайту на XSS вразливість та збір даних в разі цієї вразливості. Серед даних які можна отримати таким способом є IP жертви, адреса вразливої сторінки, Cookie файли користувача, назву та версію браузеру тощо. Загальний вигляд звіту відображено на рисунку 3.3.1.

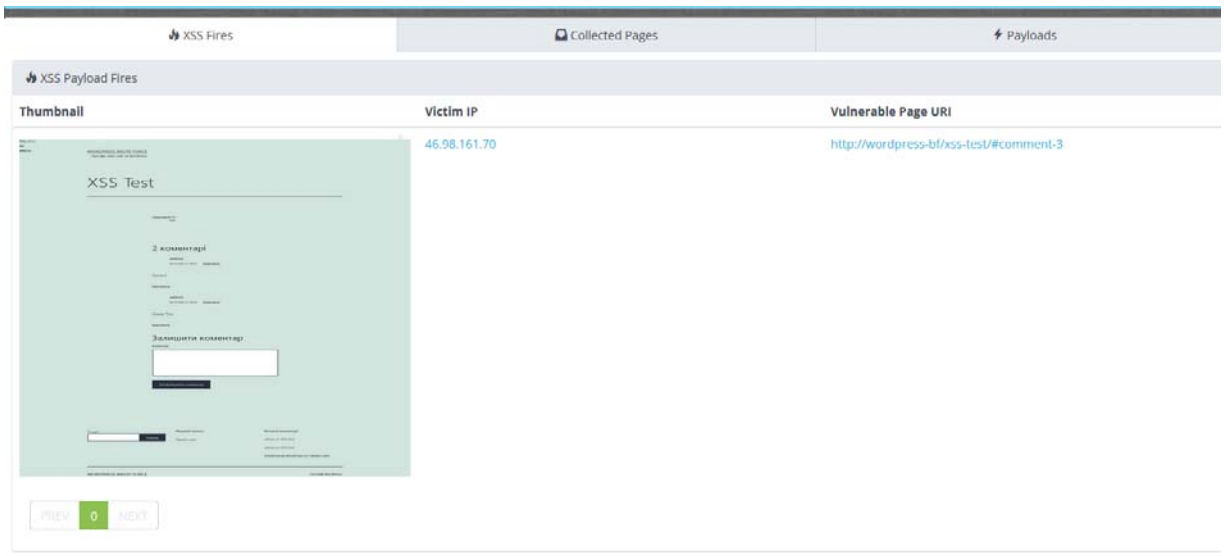


Рисунок 3.3.1 – Загальний вигляд звіту

Також XSS Hunter надає велику кількість різноманітних даних для тестування, в разі, якщо деякі вхідні дані жертви фільтруються. Сторінка можливих корисних даних наведена на рисунку 3.3.2.

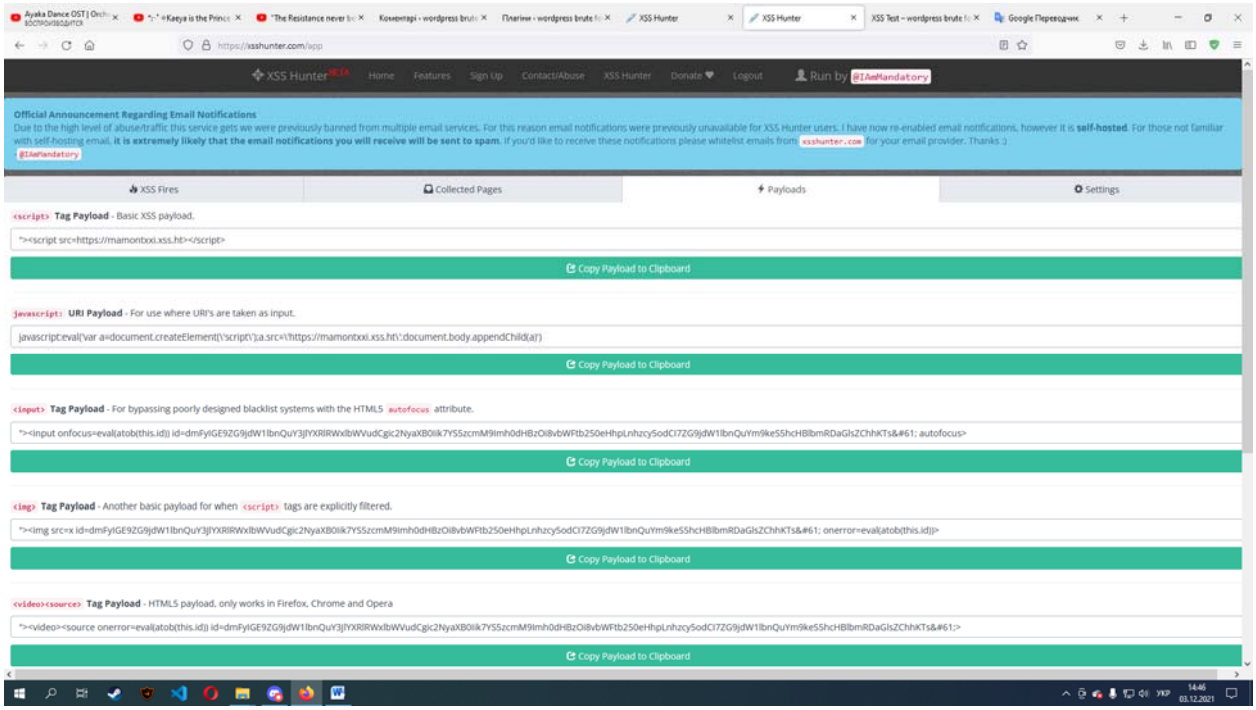


Рисунок 3.3.2 – Можливі варіації корисних даних для тестування

З рисунку 3.3.2 видно, що для експлуатації вразливості веб-додатку до міжсайтового скриптингу можна використовувати не тільки тег «script», але і «img», «video» тощо. Також можливо скористатися вразливість і тоді, коли від користувача потребується ввід посилання. Все це показує, що будь яке місце, де можливий ввід користувачем даних є потенційно небезпечним та вразливим до міжсайтового скриптингу. Також потенційно вразливими та небезпечними є поля пошуку на сайті.

Наприклад, якщо додати коментар, зображений на рисунку 3.3.3, то сторінка кожний раз після завантаження буде показувати повідомлення.

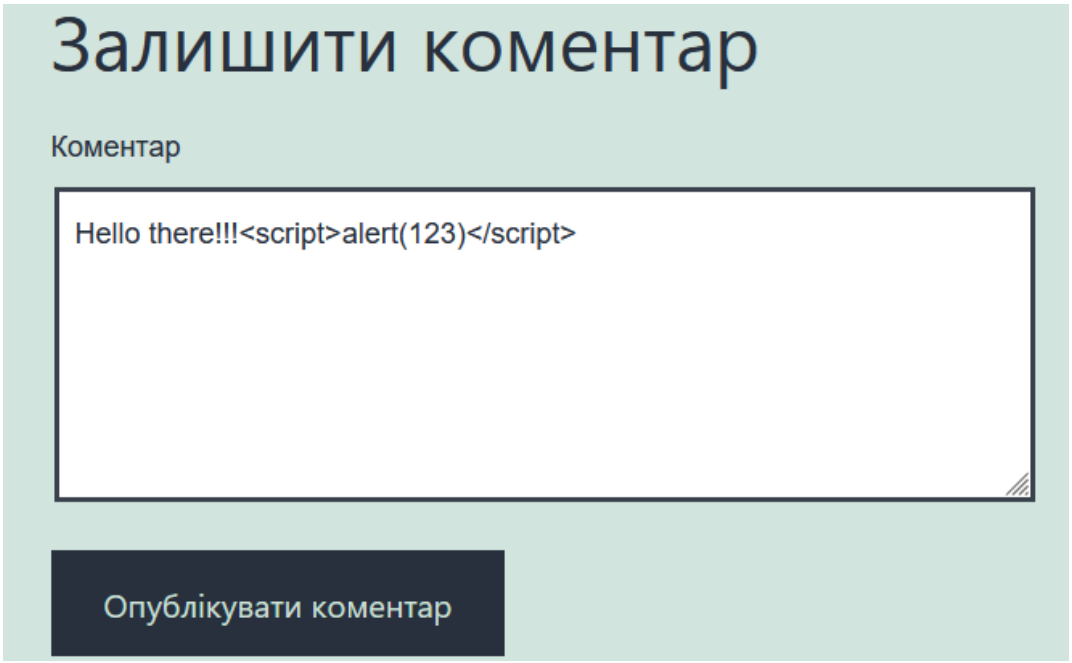


Рисунок 3.3.3 – Коментар який містить XSS

На рисунку 3.3.4 зображено результат виконання цього сценарію.

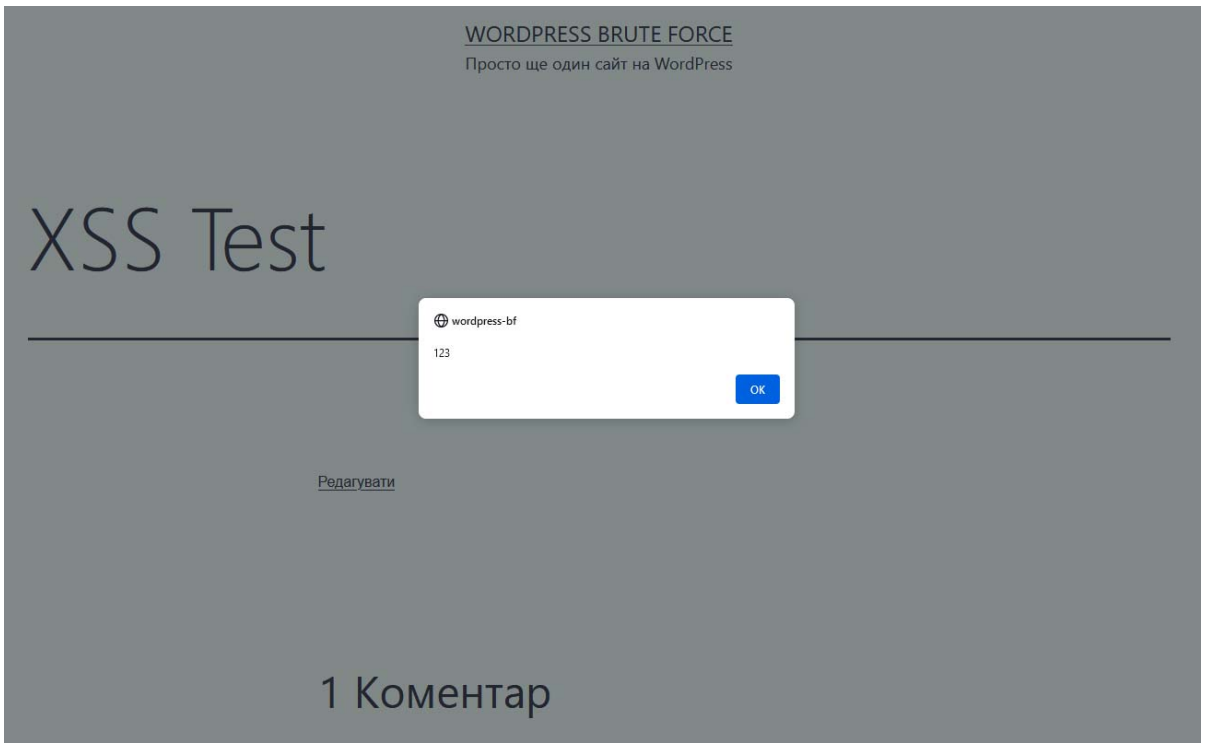


Рисунок 3.3.4 – Результат виконання сценарію

Також, якщо переглянути коментар, який призвів до показу повідомлення, можна помітити, що частина, яка містить шкідливий код, зникла. Це сталося тому, що браузер відніс частину зі шкідливим кодом до коду самої сторінки, а не до тексту коментаря, і виконав його. Найпростіший спосіб відключити показ цього повідомлення це видалення всього коментаря з сайту в вкладці керування коментарями в майстерні адміністратора сайту. Але цей спосіб добре працює тільки тоді, коли можна виявити шкідливий коментар. На великих сайтах з великою кількістю коментарів немає можливості постійно переглядати нові коментарі для запобігання додавання шкідливого кодую також потрібно розуміти, що цей спосіб працює тільки після виконання шкідливого коду, що може привести до того, що к часу, коли коментар буде видалений, деяка кількість користувачів додатку все постраждають. Тому необхідно в першу приймати міри саме проти потрапляння шкідливого коду на сайт.

Більшість браузерів використовують сесії для ідентифікації свого користувача. Після успішного входу на сайт номер сесії зберігається в «cookie» файлах. Тепер будь які дія, яка потребує авторизації, буде містити номер сеансу. Таким чином сервер визначає який саме користувач відправив запит. Використовуючи вразливість до міжсайтового скриптингу можна викрасти номер сеансу з «cookie» файлів користувача, що буде означати викрадення всього сеансу. Після викрадення сеансу зловмисник може відправляти запити на сервер на виконувати інші дії на сайті від імені користувача. Використовуючи коментар, приведений на рисунку 3.3.5, можна дізнатися номер поточного сеансу.

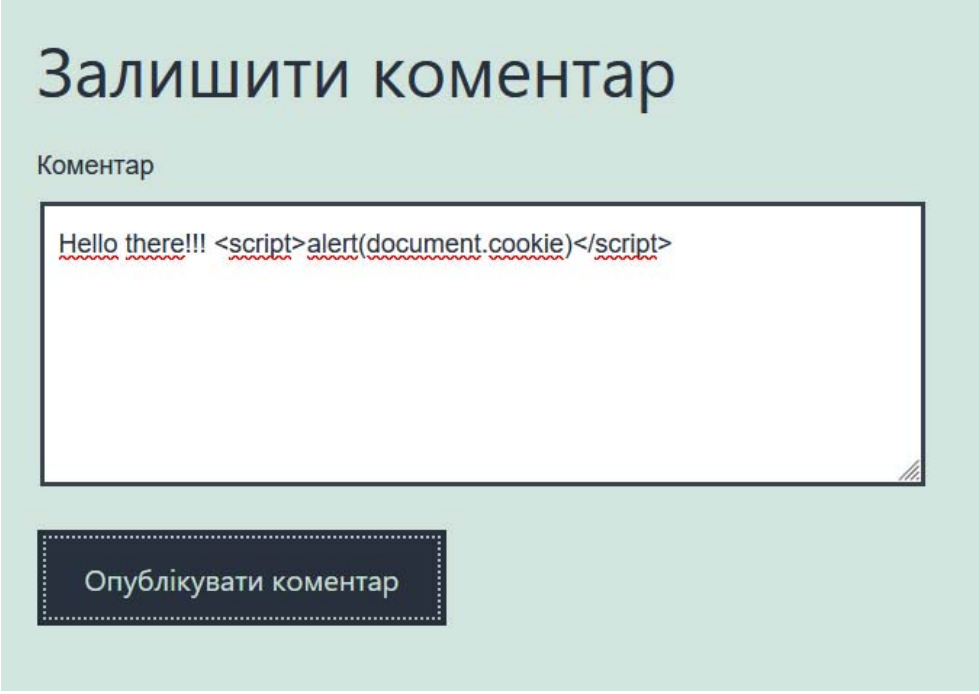


Рисунок 3.3.5 – Коментар для визначення номеру поточного сеансу

Після додавання цього коментаря користувача при воді на сайт отримає повідомлення, яке відображено на рисунку 3.3.6.

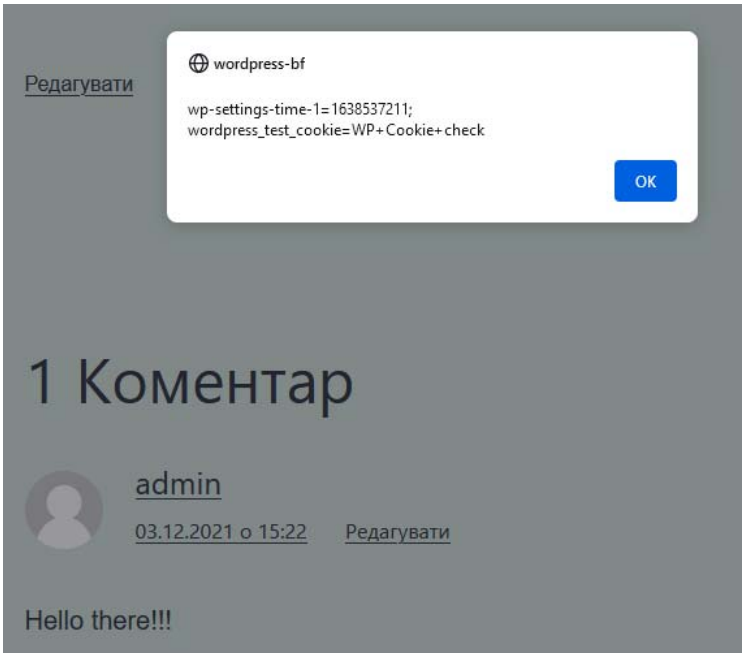


Рисунок 3.3.5 – Результат виконання сценарію

Як видно з рисунку 3.3.5, існує можливість отримати сеанс користувача, що дозволить зловмиснику відправляти запити від імені користувача, а іноді навіть зписувати гроші з картки.

Використовуючи сценарій, який показано на рисунку 3.3.6 можна отримати «cookie» файли користувача, після чого використовувати номер сеансу в своїх цілях.

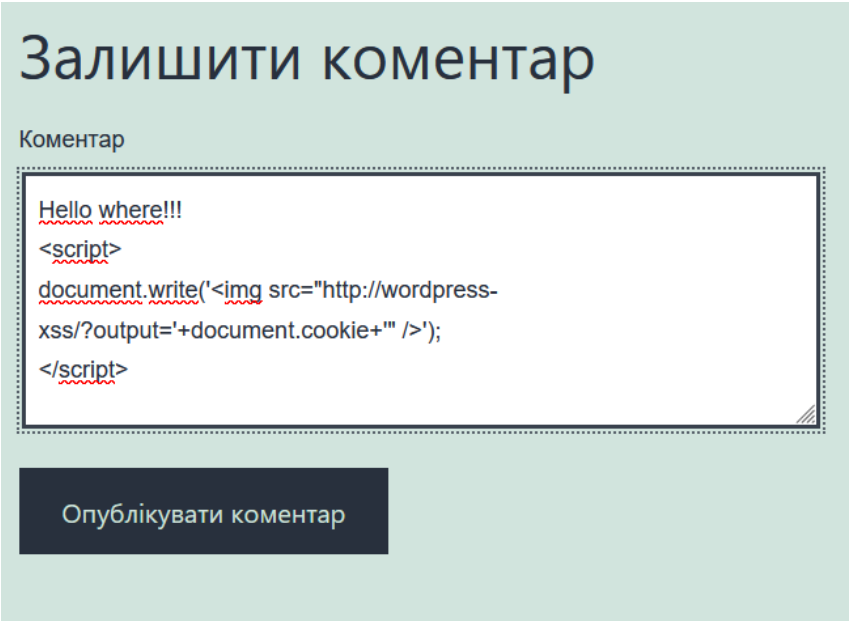


Рисунок 3.3.6 – Сценарій для викрадення сеансу користувача

Після того, як користувач заходить на сторінку, шкідливий сценарій відправляє на сайт зловмисника запит який містить в собі «cookie» файли. Використовуючи програму для перехоплення мережевих пакетів зловмисник може перехопить запит до свого сайту і дізнатися інформацію з «cookie» файлів жертви, що і показано на рисунку 3.3.7.

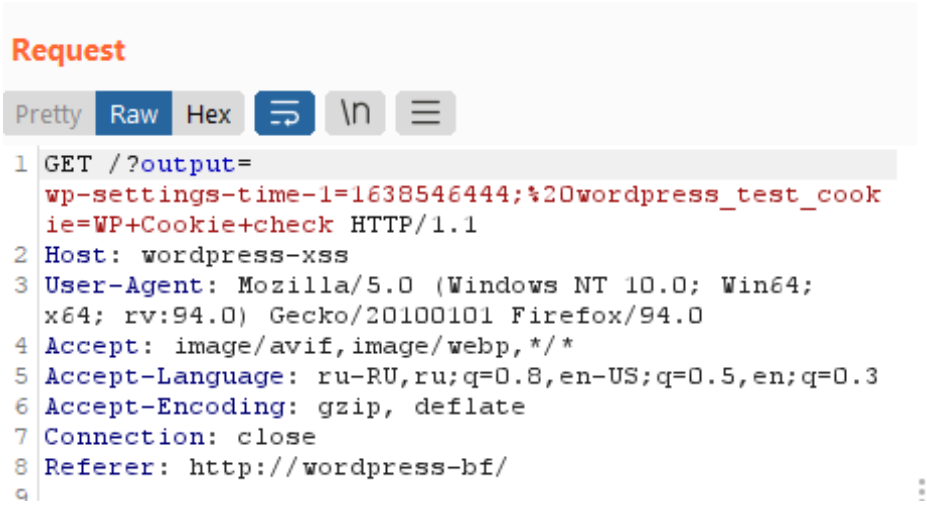


Рисунок 3.3.7 – Запит до сайту зловмисника

Як видно з рисунку 3.3.7, запит було відправлено з сайту, на який зайшов користувач (wordpress-bf), до сайту зловмисника (wordpress-xss), і заголовок запиту містить всі налаштування сесії поточної користувача, такі як номер. Далі все зловмисник може використовувати сеанс користувача для своїх цілей.

Найпростіший спосіб захиститись від міжсайтового скриптингу – це відключити JavaScript в браузері. Тоді XSS, ціллю яких є JavaScript, не будуть мати ніякої сили. Але це требі робити на стороні клієнту, тобто на стороні користувача сайту.

Головний метод забезпечення захисту від XSS для адміністраторів - замінити проблемні метасимволи на текстові

посилання для того, щоб метасимволи читалися як посилання і потенційно шкідливі файли не могли буди запущені на стороні серверу.

Для відключення всіх метасимволів можна використовувати розширення «Prevent XSS Vulnerability». Як говориться на офіційному сайті [27], розширення передбачене для виключення вразливості до XSS. Воно перевіряє URL-адресу та перенаправляє її, якщо ви ввімкнули параметр «Увімкнути блокування» і URL-адреса містить будь-який вразливий код. Він блокує лише деякі параметри, які не дозволені в URL-адресі та в розділі Параметри блоку. Ви можете пропустити деякі параметри з нього, якщо вам все ще подобається їх використання. Сторінка налаштування відображена на рисунку 3.3.8.

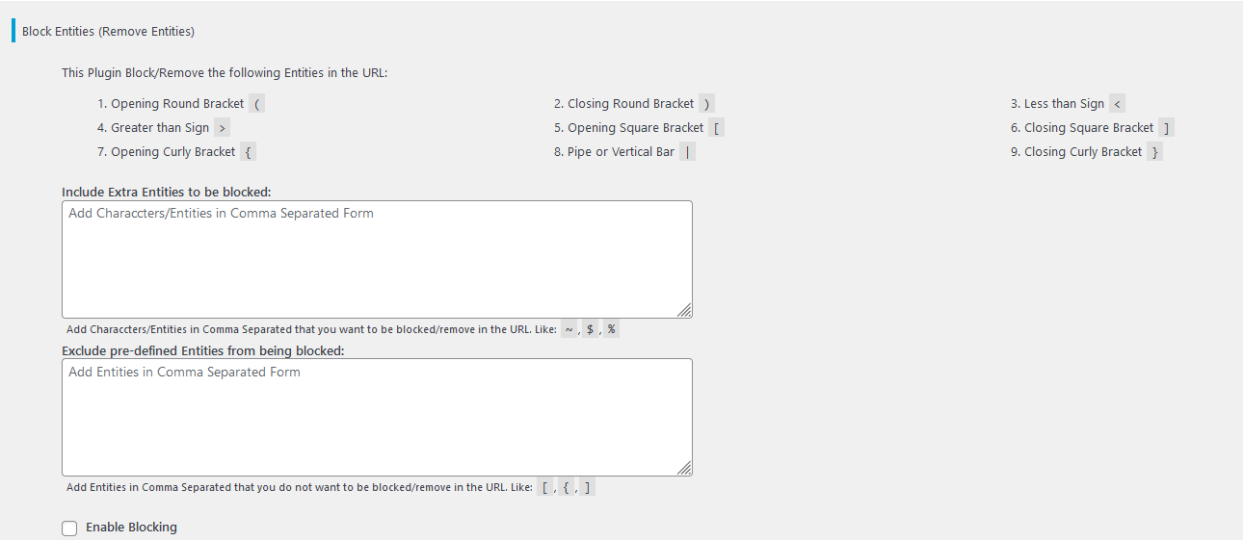


Рисунок 3.3.8 – Вигляд сторінки налаштування розширення

Як видно з рисунку 3.3.8, розширення має можливість тонкого налаштування, можна вказати символи, які будуть заблоковано, або символи, які не будуть заблоковані.

Після включення всіх фільтрів розширення необхідно оновити всі коментарі. Після оновлення коментарів всі шкідливі сценарії перестануть виконуватись і відображатись, а запити не сайт

зловмисника з «cookie» файлами користувача не будуть відправлятися, що і відображено на рисунках 3.3.9 та 3.3.10.

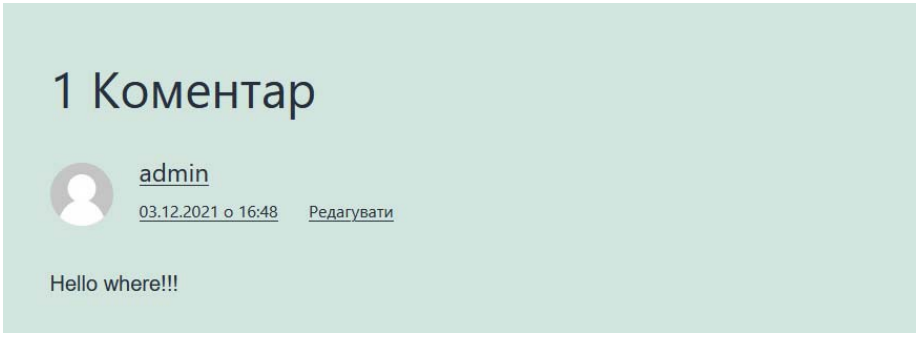


Рисунок 3.3.9 – Шкідливий сценарій в коментарях більше не виконується

	Host	Method	URL	Params	Edited	Status	Length	MIME t
	http://wordpress-bf	POST	/wp-admin/comment.php	✓		302	455	HTML
	http://wordpress-bf	GET	/wp-admin/comment.php?action=edit...	✓		200	52309	HTML
	http://wordpress-bf	GET	/xss-test/?preview_id=27&preview_no...	✓		200	24620	HTML
	http://wordpress-bf	POST	/wp-admin/comment.php	✓		302	455	HTML
	http://wordpress-bf	GET	/wp-admin/comment.php?action=edit...	✓		200	52162	HTML
	http://wordpress-bf	POST	/wp-admin/admin-ajax.php	✓		200	523	JSON
	http://wordpress-bf	GET	/wp-admin/plugins.php?deactivate=tr...	✓		200	84330	HTML
	http://wordpress-bf	GET	/wp-admin/plugins.php?action=deacti...	✓		302	448	HTML
	http://wordpress-bf	POST	/wp-admin/admin-ajax.php	✓		200	523	JSON

Рисунок 3.3.10 – Запит на сайт зловмисника не відправився

Виходячи з результатів дослідження можна зробити висновок, що вразливість додатку до міжсайтового скриптингу була усунута, а користувачам нема необхідності відключати JavaScript в своїх браузерах.

3.4. Дослідження методів усунення вразливості до шкідливого програмного забезпечення

Шкідливе програмне забезпечення розміщується на сайті для зараження комп'ютерів відвідувачів сайту шкідливим програмним забезпеченням. Цілі можуть різнитися від сайту до сайту. Для демонстрації було обрано WordPress версії 5.8.2. В якості веб-серверу обрано локальний сервер OpenServer версії 5.3.7.

Головною причиною проникнення шкідливого програмного забезпечення на сайт є бекдори в програмному забезпеченні. Бекдори виникають через допущені помилки та недоліки під час розробки програмного забезпечення. Але потрібно розуміти, що бекдори постійно виявляються та закриваються розробниками в нових версіях програмного забезпечення. Тому найважливішою міру про забезпеченні захисту від наявності бекдорів, і як наслідок проникнення шкідливого програмного забезпечення на сайт, є повне оновлення всіх розширень і самого WordPress до останніх версії. Також необхідно перевіряти сумісність версій WordPress і розширень, що використовуються. Для спрощення цієї задачі WordPress містить спеціальну мітку яка показує сумісність розширення і поточної версії. Загальний вигляд мітки наведено на рисунку 3.4.1.

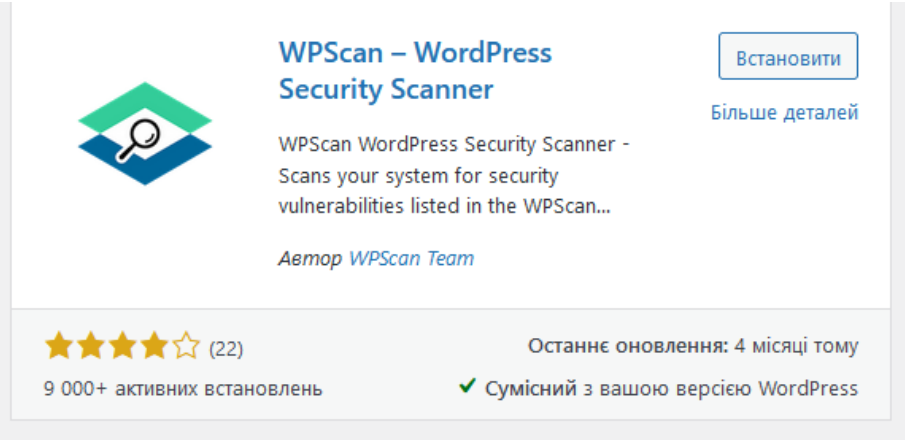


Рисунок 3.4.1 – Розширення сумісно з поточною версією WordPress

Також необхідно приділяти увагу оновленням самого WordPress, які з’являються в майстерні в вкладці «Оновлення». Загальній вигляд наведено на рисунку 3.4.2. Особливої уваги заслуговують оновлення з поміткою «Оновлення безпеки».

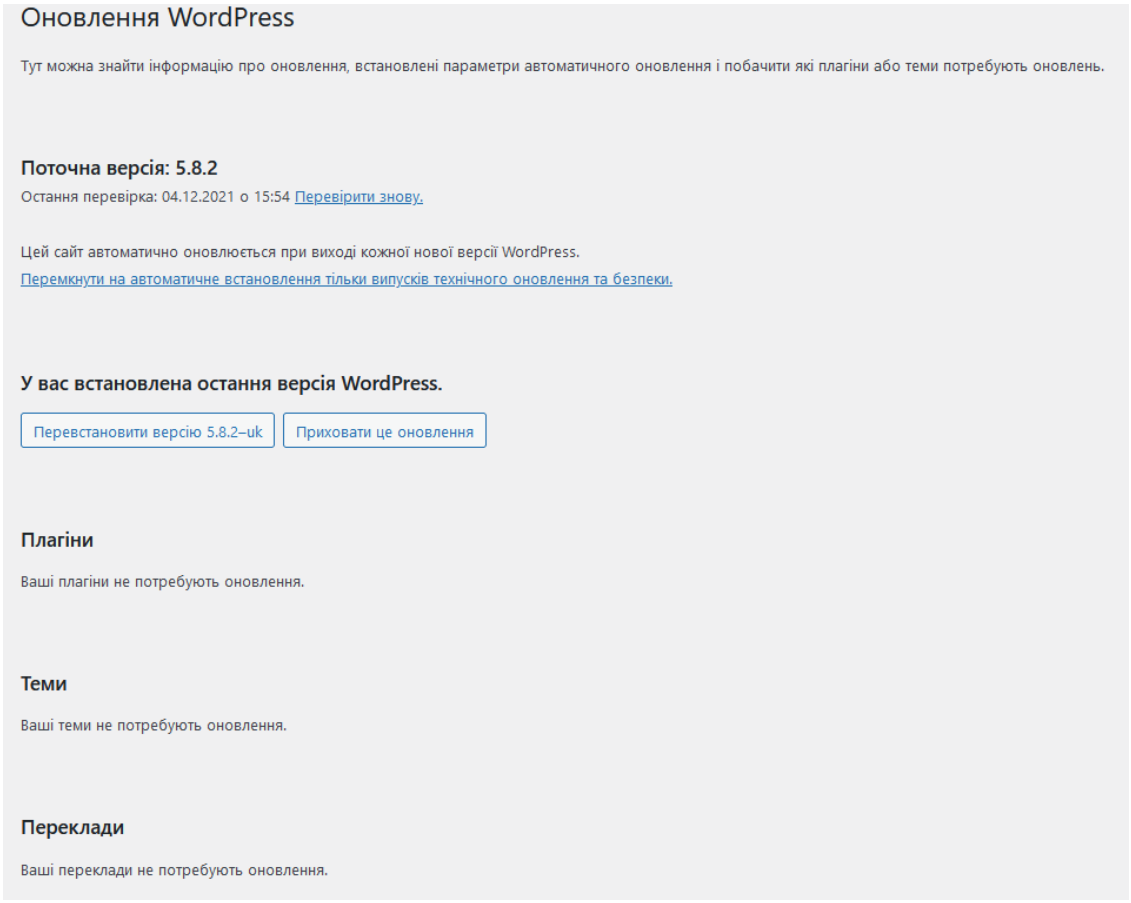


Рисунок 3.4.2 – Сторінка оновлення WordPress

Для виявлення та закриття всіх бекдорів необхідно використовувати розширення «WPScan». Виходячи з [28], WPScan – це безкоштовне розширення для некомерційного використання яке представляє з себе сканер безпеки WordPress, доданих розширень і бази даних. Сканер включає в себе базу даних які містить 24 тисячі відомих вразливостей WordPress. Сторінка налаштування сканеру представлена на рисунку 3.4.3.

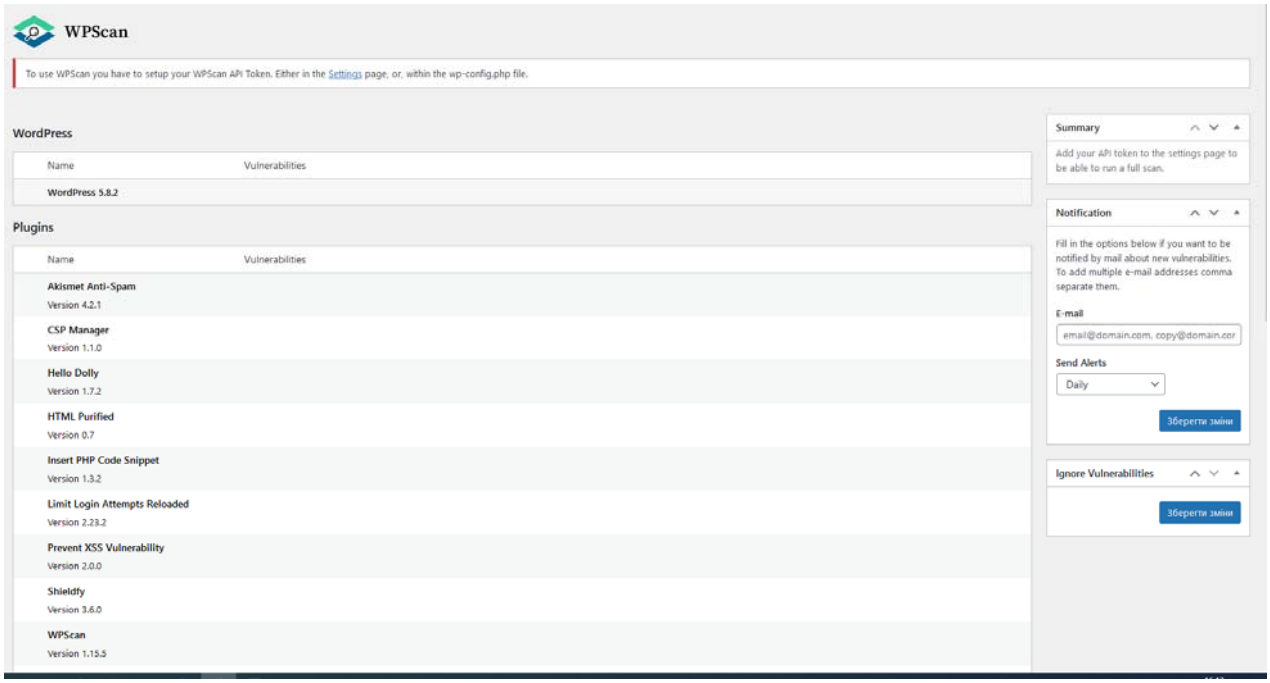


Рисунок 3.4.3 – Сторінка налаштування сканеру

Як видно з рисунку 3.4.3, сканер виявляє список всіх встановлених розширень, теми та версії WordPress, а також їх вразливості. Сканер тільки було встановлено і сканування не проводилося, тому список виявлених вразливостей пустий.

Для використання сканеру необхідно зареєструватися на офіційному сайті WPScan, після чого користувачу стане доступне сканування. При використанні безкоштовного тарифу на один день доступно 25 сканувань. За одне сканування рахується сканування розширення чи самого WordPress. Тобто при наявності 10 розширень буде за один раз витрачено 11 спроб, що необхідно ураховувати.

Після сканування сканер виявив декілька незакритих вразливостей. результат сканування представлено на рисунках 3.4.4, 3.4.5 та 3.4.6.

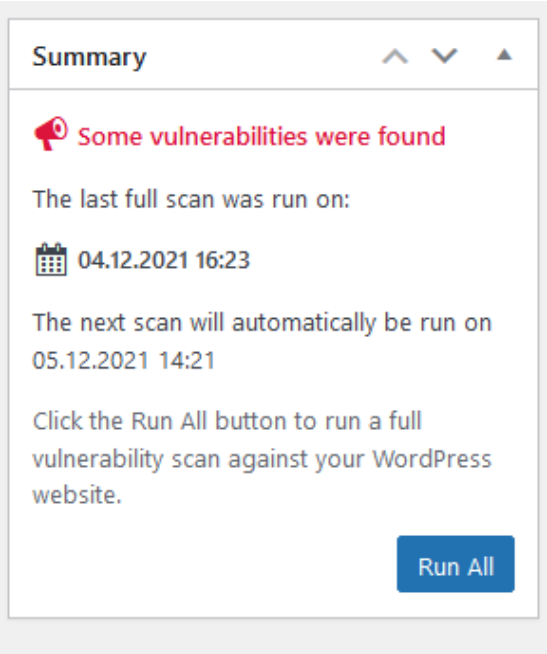


Рисунок 3.4.4 – Загальні результат сканування.

Name	Vulnerabilities
Akismet Anti-Spam Version 4.2.1	No known vulnerabilities found to affect this version
CSP Manager Version 1.1.0	No known vulnerabilities found to affect this version
Hello Dolly Version 1.7.2	No known vulnerabilities found to affect this version
HTML Purified Version 0.7	No known vulnerabilities found to affect this version
Insert PHP Code Snippet Version 1.3.2	No known vulnerabilities found to affect this version
Limit Login Attempts Reloaded Version 2.23.2	No known vulnerabilities found to affect this version
Prevent XSS Vulnerability Version 2.0.0	No known vulnerabilities found to affect this version
Shieldfy Version 3.6.0	No known vulnerabilities found to affect this version
WPScan Version 1.15.5	No known vulnerabilities found to affect this version
WP Shieldon Version 1.6.3	WP Shieldon 1.6.3 - Unauthenticated Cross-Site Scripting (XSS) We are not aware of a fix for this vulnerability. Click here for further details

Рисунок 3.4.5 – Результати сканування розширень

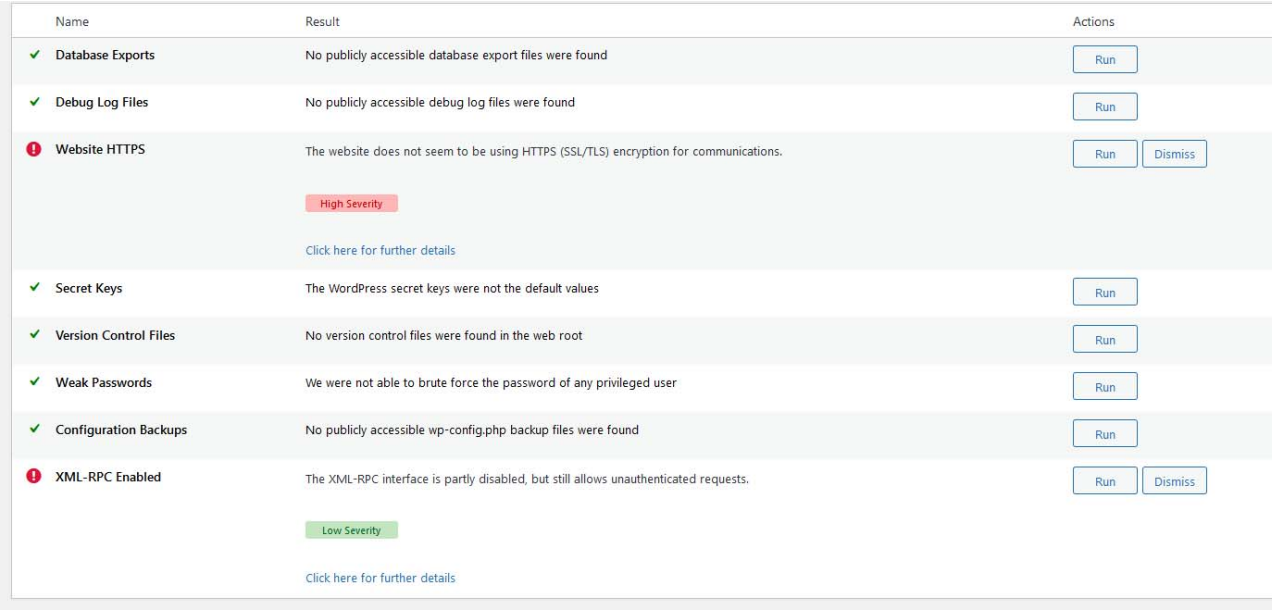


Рисунок 3.4.6 – Результат загального сканування

З рисунку 3.4.5 видно, що виявлено вразливість розширення «WP Shieldon» до міжсайтового скриптингу. Рішення цієї проблеми може бути повне видалення вразливого розширення та заміна його на більш нове та захищене. Після видалення розширення можна помітити зміни в списку плагінів, який представлено на рисунку 3.4.7.

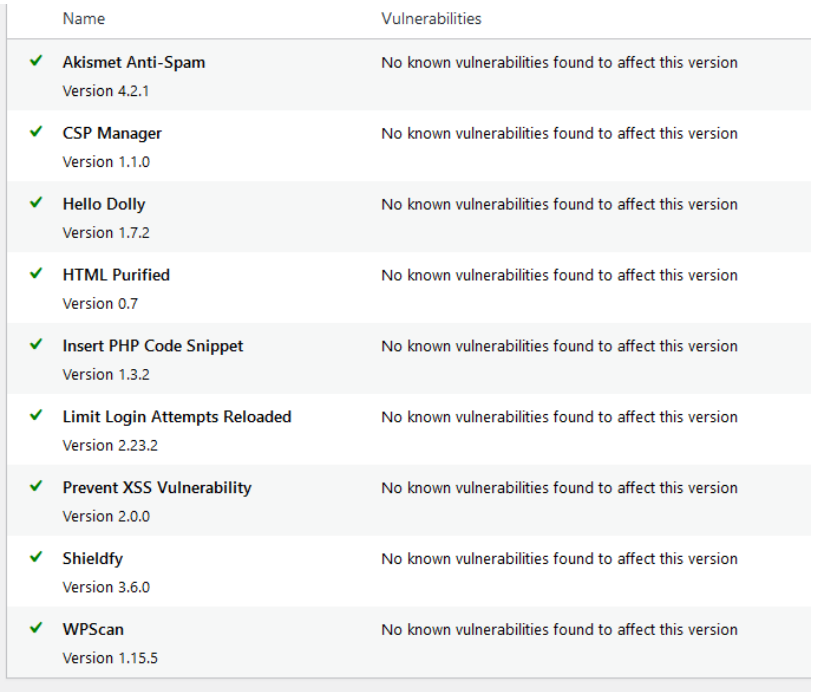


Рисунок 3.4.7 – Зміни в списку розширень

З рисунку 3.4.6 видно, що виявлено дві вразливості. Першою вразливістю є використання HTTP замість HTTPS. Для закриття цієї вразливості необхідно отримати SSL сертифікат. Отримати його можна тільки після того, як сайт буде викладено на хостинг, тому цю вразливість поки прийдеться ігнорувати.

Другою вразливістю є не відключений протокол XML-RPC, через що існує можливість відправки не аутентифіковані запитів на сервер. Для закриття цієї вразливості можна повністю відключити XML-RPC. Основним способом відключення є встановлення розширення, яке буде блокувати XML-RPC API. Для цього було обрано розширення «Disable XML-RPC API».

Після виконання всіх кроків для закриття всіх вразливостей необхідно провести сканування для того, щоб переконатися що всі бекдори закриті. Результати сканування наведено на рисунку 3.4.8 та 3.4.9.

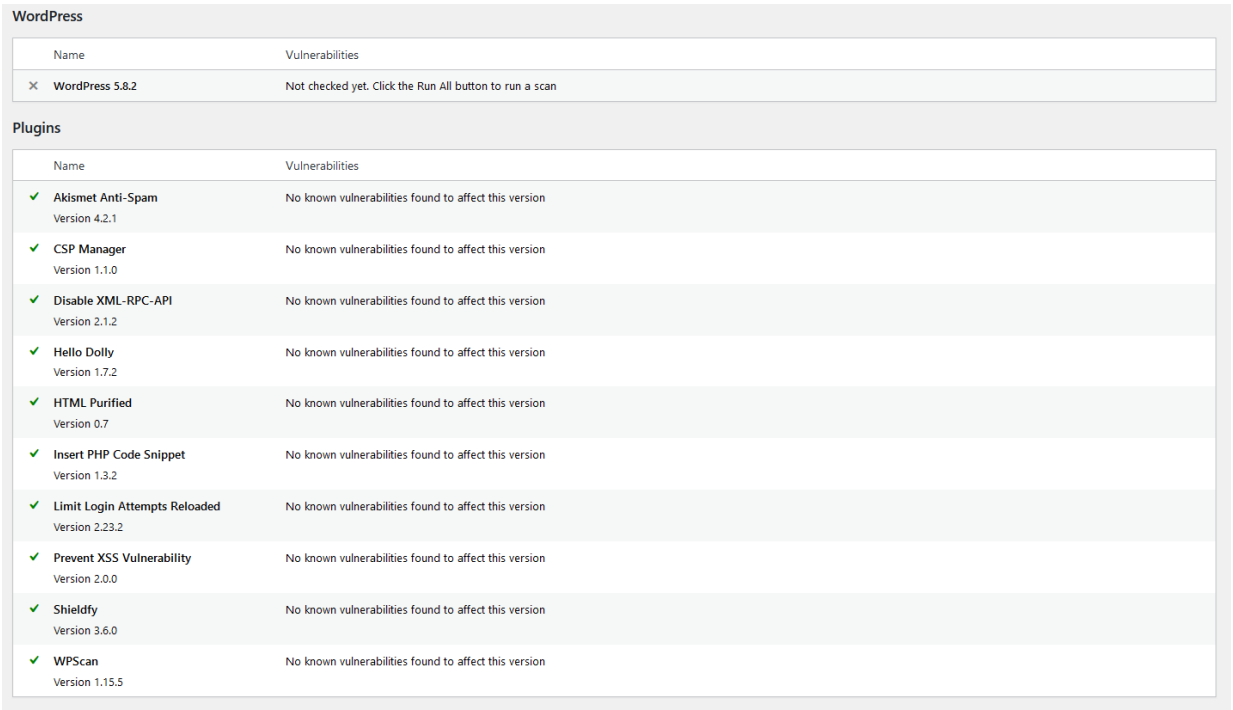


Рисунок 3.4.8 – Результати сканування розширень

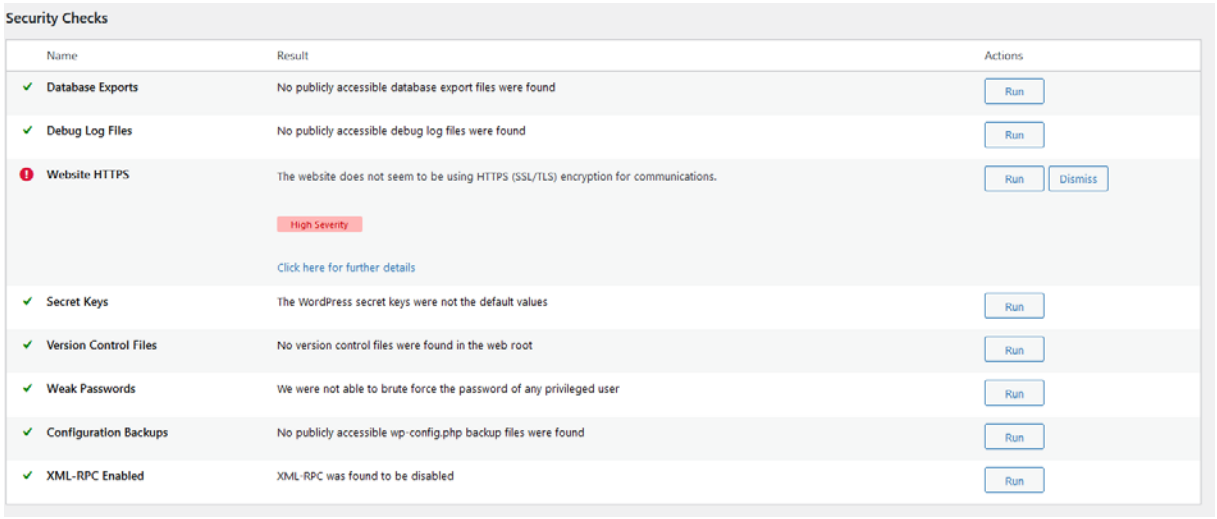


Рисунок 3.4.9 – Результати сканування безпеки додатку

Як видно з рисунків 3.4.8 та 3.4.9, всі вразливості додатку було усунуто (окрім зміна протоколу на HTTPS). За декілька нескладних кроків безпеку додатку було сильно підвищено, а вірогідність зараження шкідливим програмним забезпеченням значно знижено.

3.5. Дослідження методів усунення вразливості до DDoS атак

DDoS - це атака, яка направлена на навантаження серверу або мережі надлишковим трафіком, що може привести до зниженню продуктивності або к повному відключенню.

Головними причинами, чому сайти, зроблені за допомогою WordPress, стають жертвами DDoS атак є недоліки самого WordPress. Наприклад, як показав Брак Тавілі, для обробки запитів користувачів використовується файл «load-scripts.php», який об’єднує декілька JavaScript файлів для зручності адміністраторів сайтів. Всі файли, які необхідно поєднати, передаються в якості аргументу через кому в

рядку пошуку. Але через необачність ця функціональність ніяк не захищена, що дозволяє зловмиснику заставити сайт завантажити всі можливі сценарії, що дуже сильно підвищить навантаження на сервер. В свою чергу підвищення навантаження від кожного окремого запиту дозволяє провести DDoS атаку при невеликих витратах на ресурси. Наприклад, в експерименті, приведеному в [29], сайт витримав всього 500 запитів з повним навантаженням «load-scripts.php». На даний момент вразливість виправлена в пізніх версіях WordPress, але версії до 4.9.2. вразливість все ще присутня.

Для захисту від DDoS атак дуже добре допомагає відключення XML-RPC. Його відключення було продемонстровано в попередньому пункті.

Також буде корисно встановити розширення брандмауер для фільтрації трафіку та захисту від DDoS атак. Необхідно зазначити, що існує можливість заборонити доступ до сайту при великому навантажені на сервер, але цей спосіб не дуже вигідний, так як він може відлякати потенційних клієнтів. Найкращім вибором в даному випадку буде вибір кращих серверів, які могли би витримати велике навантаження, а також перед входом користувача на сайт ввести перевірку, яка допоможе відрізнити реальну людину від боту, які широко використовуються при DDoS атаках.

В якості розширення було обрано «WP Cerber Security». Я говориться на офіційному сайті [30], розширення містить в собі декілька важливих функцій, таких як автоматичне резервне збереження файлів сайту, сканування всіх встановлених тем та розширень, сканування на наявність нових або модифікованих файлів в разі зараження шкідливим програмним забезпеченням. Також із важливого треба відмітити функцію брандмауера які включають в себе і

фільтрацію трафіка яка виявляє всі підозрілі запити на сервер, що найчастіше означає спроби DDoS атаки на сайт.

Після встановлення розширення стає доступна велика кількість різноманітних корисних функцій. Всі функції представлено на рисунку 3.5.1.

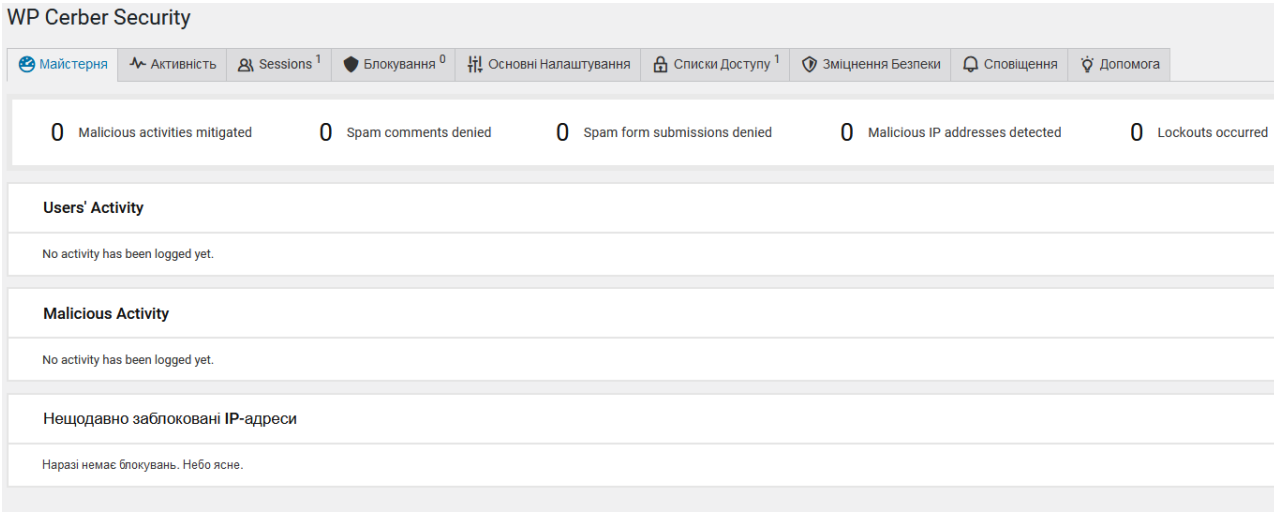


Рисунок 3.5.1 – Основні функції розширення WP Cerber Security

Як видно з рисунку 3.5.1, доступні функції моніторингу трафіку, кількість та опис активних сесій, список заблокованих IP-адрес, списки доступу тощо.

Після вдалого встановлення розширення його необхідно правильно налаштувати. Для допомоги користувачам написана велика кількість документації по всім пунктам налаштування розширення. Вона доступна на офіційному сайті.

Перш за все необхідно провести повне сканування сайту. Результати сканування наведено на рисунках 3.5.2. та 3.5.3.

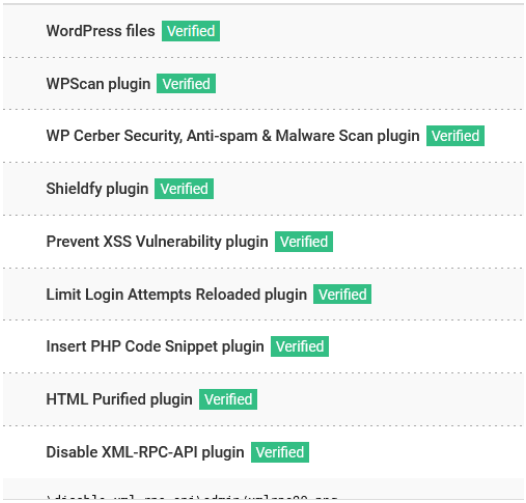


Рисунок 3.5.2 – Результати сканування

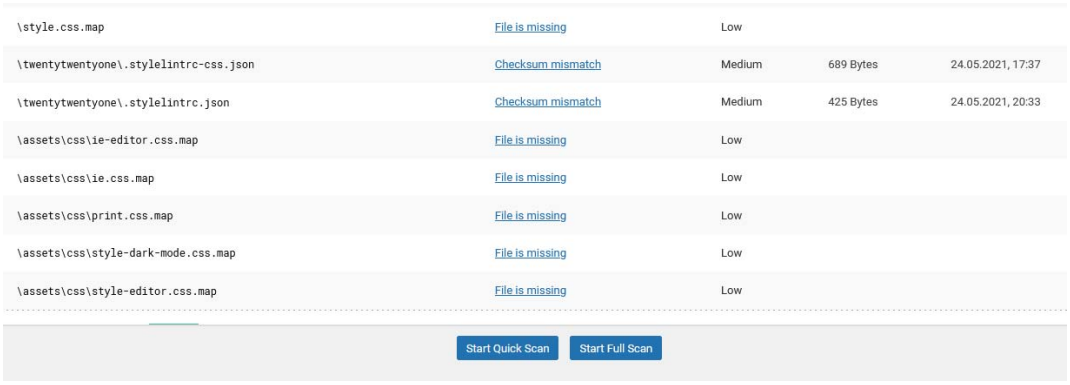


Рисунок 3.5.3 – Результати сканування

Як видно з рисунку 3.5.3, виявлено декілька неіснуючих тем, що пояснюється використанням нової версії WordPress. Ці помилки не критичні та їх можна ігнорувати перший час. Ніяких серйозних вразливостей виявлено не було.

Далі необхідно включити антиспам систему. Робиться це за допомогою перемикання всіх необхідних прапорів на необхідне положення. Приблизний вигляд сторінки налаштування антиспам системи наведено на рисунку 3.5.4.

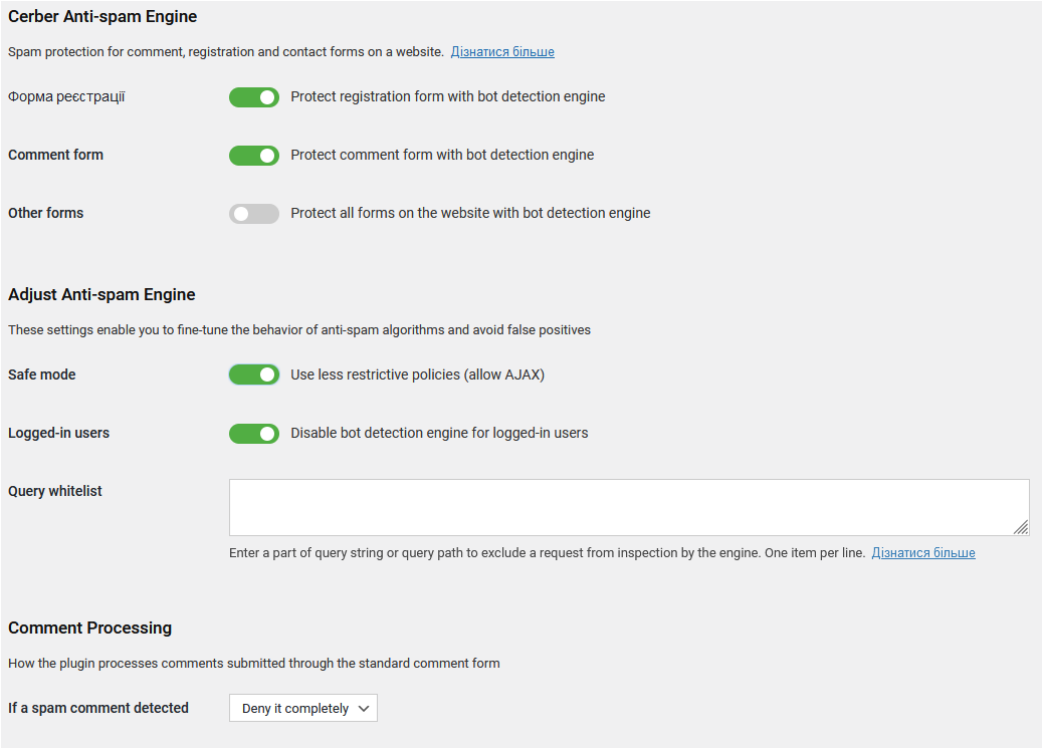


Рисунок 3.5.4 – Вигляд сторінки налаштування антиспам системи

Як видно з рисунку 3.5.4, доступні функції захисту різних форм, такі як форма реєстрації, додавання коментарів та інші.

Також необхідно підключити резервне копіювання файлів сайту. Для налаштування резервного копіювання необхідно встановити розширення «UPdraftPlus». Це розширення автоматично зберігає файли до Dropbox, Google Drive та FTP. Основною причиною обрання саме цього розширення є надійність і простота в використанні. Сторінку налаштування розширення наведено на рисунку 3.5.6.

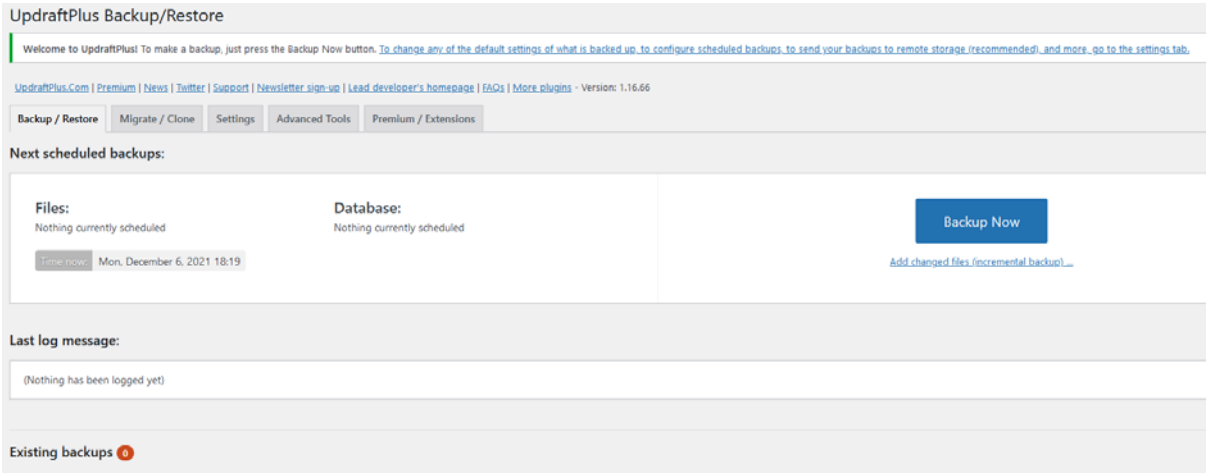


Рисунок 3.5.6. – Сторінка налаштування UPdraftPlus

Після початку налаштування виникає вікно тонкого налаштування резервного копіювання. Загальний вигляд вікна наведено на рисунку 3.5.7.

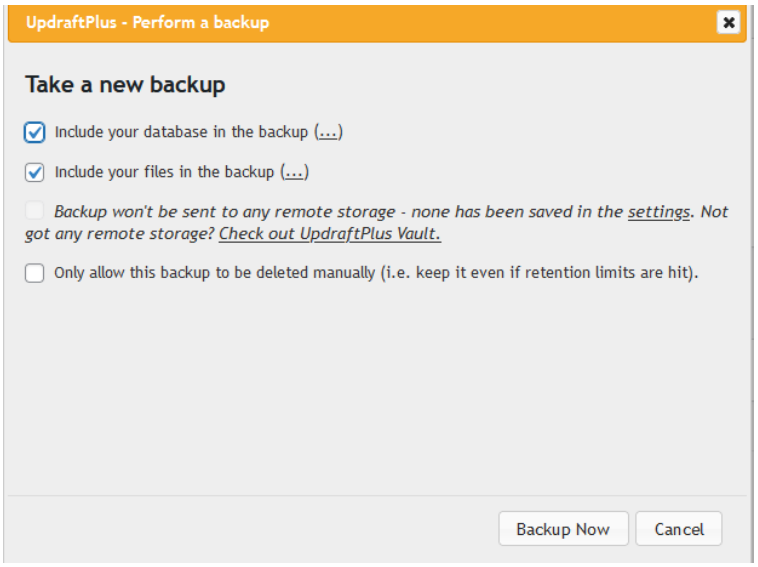


Рисунок 3.5.7 – Вигляд вікна налаштування

Як видно з рисунку 3.5.7, існує можливість включити або виключити базу даних і файлів з резервного копіювання. Рекомендується зберігати всі файли і бази даних. Також доступна можливість виключити автоматичне видалення збережень після вичерпання їх терміну зберігання. Після обрання всіх налаштування для зберігання починається процес копіювання файлів до сховища.

Після створення резервної копії вона з’являється в списку резервних копій, який наведено на рисунку 3.5.8.

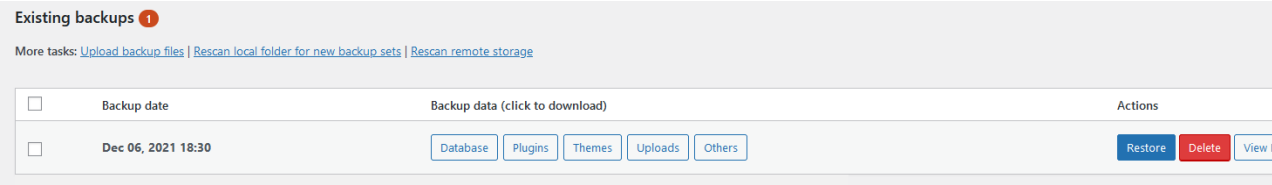


Рисунок 3.5.8 – Резервні копії сайту

Як видно з рисунку 3.5.8, адміністратору стають доступні функції встановлення файлів з резервної копії а також її видалення.

В підсумку можна зазначити, що за допомогою декількох розширень було закрито вразливість сайту до DDoS атак а також налаштовано резервне копіювання для усунення можливих проблем з втратою файлів після DDoS атак.

3.6. Висновки

В результаті виконання декількох кроків захист веб-додатку від всіх найрозповсюдженіших типів атак значно підвищився. Також було встановлено систему автоматичного створення резервних копій веб додатку, а антивірус за розкладом проводить сканування з ціллю виявлення випадків потрапляння шкідливого програмного забезпечення.

4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Правила безпеки про роботі за персональним комп’ютером

В даному розділі наведено основні фактори небезпеки під час роботи з персональним комп’ютером. Також було надано рекомендації щодо безпечного проводження робіт за ПК.

Згідно з [31], закон України «Про охорону праці» визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні. Виходячи з закону роботодавець несе повну відповідальність за створення необхідних умов для безпечної роботи працівників, в тому числі і підтримання робочого місця в придатному до роботи стані, впровадження нових технологій, усунення причин нещасних випадків а також проведення інструктажів тощо. Також необхідно зазначити, що згідно з доповненням №1213-XI від 02.04.2021 при укладенні трудового договору про дистанційну роботу, про надомну роботу на роботодавця покладається обов’язок систематичного проведення інструктажу (навчання) працівника з питань охорони праці і протипожежної безпеки в межах використання таким працівником обладнання та засобів, рекомендованих або наданих роботодавцем.

Основні положення щодо створення належних умов під час роботи з екранними пристроями наведено в нормативно-правових актах «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [32] та «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [33].

Нормативно-правовий акт «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» затверджує такі положення щодо безпеки робочих місць працівників з екранними пристроями:

- Робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.
- Для забезпечення безпеки та захисту здоров’я працівників усе випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня з погляду безпеки та охорони здоров’я працівників.
- Організація робочого місця працівника з екранними пристроями має забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.
- Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями має підтримуватись на постійному рівні та відповідати вимогам санітарних норм мікроклімату виробничих приміщень [34].
- Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.
- Робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення.

Необхідно зазначити, що сидіння крісла має регулюватися по висоті, а спинка – як по висоті, так і по нахилі. Також слід передбачити підніжку для зручності, якщо це необхідно працівнику.

Також необхідно організувати перерви на відпочинок для працівників. Тривалість перерв при 8 годинах роботи та їх частота залежать від виконуваних робіт:

- Перерва тривалістю 15 хвилин через кожну робочу годину за комп'ютером.
- Перерва тривалістю 10 хвилин через кожну годину роботи для операторів комп'ютерного набору.
- Перерва тривалістю 15 хвилин через кожні дві години роботи для операторів ЕОМ.

Слід зазначити, якщо умови не дозволяють організувати перерви під час виробничого процесу, то тривалість безперервної роботи не повинна перевищувати 4 годин.

При 12 годинах роботи перерви в перші 8 годин рахуються як при 8 годинах роботи, а після 15 хвилин на кожну годину не залежно від діяльності.

Також нормативно-правовий акт «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» затверджує вимоги до пристроїв, на яких проводяться виробничі роботи:

- Екранні пристрої не мають бути джерелом ризику для працівників.
- Усе випромінювання, за винятком видимої частини електромагнітного спектра, має бути зведене до незначного рівня з погляду безпеки і охорони здоров'я працівників.

- Символи на екранних пристроях мають бути чіткими, відповідного розміру. Між символами і рядками символів має бути належна відстань.
- Зображення на екрані має бути стабільним, без миготінь або інших видів нестабільності.
- Яскравість та/або контрастність символів має легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов.

Слід зазначити, що працівнику має бути доступна можливість регулювати нахил екрану та відстань до нього. Також клавіатура має бути автономною, для того, щоб працівник міг встановити зручне положення клавіатури незалежно від положення екрану.

Нормативно-правовий акт «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» зазначає такі основні вимоги до виробничих приміщень для експлуатації ЕОМ та ПЕОМ, як:

- Розміщення робочих місць у підвальних приміщеннях та на цокольних поверхах заборонено.
- Площа на одне робоче місце має становити не менше ніж 6,0 м2, а об'єм не менше ніж 20,0 м3.
- Приміщення для роботи повинне мати природне та штучне освітлення згідно з ДБН В.2.5-28-2006.
- Природне освітлення має здійснюватись через світлові прорізи, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природної освітленості (КПО) не нижче ніж 1,5%. Розраховується КПО за методикою, викладеною в ДБН В.2.5-28-2006.

- Виробничі приміщення для роботи з ЕОМ не повинні межувати з приміщеннями, в яких рівні шуму і вібрації перевищують допустимі значення.
- Приміщення для роботи з ВДТ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією відповідно до СНиП 2.04.05-91.

Також нормативно-правовий акт «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» затверджує гігієнічні вимоги до параметрів виробничого середовища, таких як мікроклімат:

- У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату, а саме температури, відносної вологості і рухливості повітря.
- Рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати санітарно-гігієнічним нормам.

Вимоги до інших параметрів виробничого середовища, таких як освітлення, шум та вібрація, було наведено раніше в основних вимогах нормативно правового акту.

Оптимальні параметри мікроклімату, згідно з ДСН 33.6.042-99 [34], наведено в таблиці 4.1.1.

Таблиця 4.1.1 – Оптимальні параметри мікроклімату для легких категорій робот

Період року	Категорія робіт	Температура повітря	Відносна вологість	Швидкість руху повітря, м/сек
Холодний період	Легка Іа	22-24	60-40	0.1
	Легка Іб	21-23	60-40	0.1
Теплий період	Легка Іа	23-25	60-40	0.1
	Легка Іб	22-24	60-40	0.2

Допустимі рівні звуку при роботі з екранними пристроями, згідно з ДСанПіН 3.3.2-007-98 [33], наведено в таблиці 4.1.2.

Таблиця 4.1.2 - Допустимі рівні звуку при роботі з екранними пристроями

Вид трудової діяльності	Рівні звукового тиску в дБ в октавних смугах із середньогеометричними частотами, Гц									
	31.5	63	125	250	500	1000	2000	4000	8000	Рівні звуку, дБА/дБАекв
Програмісти ЕОМ	86	71	61	54	49	45	42	40	38	50
Оператори в залах обробки інформації, набору тексту	96	83	74	68	63	60	57	55	54	65
В приміщеннях з розташуванням шумних агрегатів ЕОМ	103	91	83	77	73	70	68	66	64	75

Допустимі рівні іонізації при роботі з екранними пристроями, згідно з ДСанПіН 3.3.2-007-98 [33], наведено в таблиці 4.1.3.

Таблиця 4.1.3 - Допустимі рівні іонізації при роботі з екранними пристроями

Рівні	Число іонів в 1 см3 повітря	
	n+ (позитивні)	n- (негативні)
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

4.2. Дії працівників і надзвичайних ситуаціях

Типова інструкція щодо дій персоналу підприємств при загрозі або виникненні надзвичайних ситуацій надано відповідно до статті 130 Кодексу цивільного захисту України:

4.2.1. Залежно від існуючої або прогнозованої обстановки з питань цивільного захисту та надзвичайних ситуацій на підприємстві, в установі, організації, закладі (*далі* — підприємство) може бути встановлено один з трьох режимів функціонування об'єктової ланки функціональної або територіальної підсистеми єдиної державної системи цивільного захисту:

– режим повсякденного функціонування;

– режим підвищеної готовності;

– режим надзвичайної ситуації.

Режими встановлюються органами виконавчої влади, а у окремих випадках на території підприємства — його керівником.

4.2.2. Усі працівники підприємства, незалежно від займаних посад, повинні знати та суворо виконувати вимоги Типової інструкції щодо дій персоналу підприємства при загрозі або виникненні надзвичайних ситуацій.

За невиконання вимог інструкції персонал підприємства може бути притягнутий до адміністративної відповідальності.

Порядок оповіщення адміністрації та персоналу про загрозу виникнення надзвичайної ситуації наступний: оповіщення адміністрації та працівників проводиться за заздалегідь розробленим планом; адміністрація в неробочій час сповіщуються телефоном, працівники сповіщуються за необхідністю. Також кожен працівник повинен знати сигнал сповіщення при загрозі або виникненні надзвичайної ситуації.

В залежності від типу надзвичайної ситуації відрізняються і дії персоналу. Наприклад:

- На випадок виникнення НС, пов’язаної з загрозою або початком забруднення повітря небезпечною хімічною чи радіоактивною речовиною всі працівники підлягають укриттю в захисній споруді цивільного захисту. Також видаються засоби індивідуального захисту при отриманні відповідного розпорядження або за рішенням керівника підприємства.

- На випадок виникнення небезпеки розповсюдження особливо небезпечного інфекційного захворювання усі працівники повинні суворо виконувати вимоги санітарно-епідеміологічної служби щодо проведення термінової профілактики та імунізації, ізоляції та лікування виявлених хворих.

- При виникненні пожежі всі працівники зобов’язані суворо виконувати вимоги інструкції з пожежної безпеки приміщення.

- При загрозі або виникнення стихійних лих працівники повинні припинити виконання виробничих робот, провести за розпорядженням адміністрації необхідні протипожежні дії, вимкнути всі електричні прилади з мережі та приготуватися до евакуації.

- При находженні анонімної інформації про загрозу на території підприємства або поблизу нього, працівники. який прийняв її повинен негайно доповісти керівнику підприємства та до правоохоронних органів, після чого діяти згідно з наданими розпорядженнями та інструкціями.

Якщо виникли постраждалі, надати їм першу медичну допомогу та вжити заходи для їх госпіталізації.

Керівництво повинно постійно проводити інструктажі для забезпечення обізнаності працівників щодо дій під час всіх типів надзвичайних ситуацій.

Згідно з наказом «Про затвердження Правил безпечної експлуатації електроустановок споживачів» [35], дії щодо забезпечення електробезпеки персоналу складається з:

- призначити відповідального за справний стан і безпечну експлуатацію електрогосподарства з числа інженерно-технічних працівників, які мають електротехнічну підготовку і пройшли перевірку знань у встановленому порядку
- забезпечити достатню кількість електротехнічних працівників
- встановити такий порядок, щоб працівники, на яких покладено обов'язки з обслуговування електроустановок, вели ретельні спостереження за дорученим їм обладнанням і мережами - оглядом, перевіркою дії, випробуванням і вимірюванням
- забезпечити проведення протиаварійних, приймально-здавальних і профілактичних випробувань та вимірювань електроустановок згідно з правилами і нормами

- У випадку, якщо працівник самостійно не спроможний вжити дійових заходів з усунення виявлених ним порушень, він зобов'язаний негайно повідомити про це безпосереднього керівника, а у випадку його відсутності - керівника вищого рівня.
- В разі нещасних випадків з людьми зняття напруги для звільнення потерпілого від дії електричного струму має бути виконано негайно, без попереднього дозволу.
- Працівники, що припустилися порушення вимог Правил, без позачергової перевірки знань до робіт в електроустановках не допускаються.

Міністерством енергетики та вугільної промисловості України затверджено збір Правил улаштування електроустановок [36], а саме:

- Електроустановки та пов’язані з ними конструкції мають бути стійкими до впливу навколишнього середовища або захищеними від цього впливу.
- Електроустановки мають задовольняти вимогам відповідних нормативних документів з охорони навколишнього середовища за допустимими рівнями шуму, вібрації, напруженості електричного і магнітного полів, електромагнітної сумісності, відходів хімічних речовин, масла, сміття тощо.
- У разі небезпеки виникнення електрокорозії або ґрунтової корозії треба передбачати відповідні заходи щодо захисту споруд, устаткування, трубопроводів та інших підземних комунікацій.
- Усі огороджувальні та закриваючі пристрої відповідно до місцевих умов повинні мати достатню механічну міцність.

- При роботі необхідно дотримувати правила техніки безпеки при роботі з високою напругою, не підключати і не відключати кабелі при включеній напрузі мережі, технічне обслуговування і ремонт проводити тільки при вимкнутому живленні

Висновки

Веб-додатки можуть бути атаковані як на клієнтській, так і на серверній стороні, включаючи і третю сторону, яку приймає участь в процесі передачі та отримання інформації веб-браузері клієнту. Сам по собі веб-браузер клієнту не є ні безпечною, ні довіреною стороною обміну інформацією.

Для полегшення та прискорення процесу створення свого додатку існує велика кількість готових систем керування веб-змістом (WCMS). Веб CMS надають велику кількість різноманітних інструментів для керування та відображення графічного і текстового наповнення додатку. Доступність і легкість в освоєнні роблять системи керування веб-змістом привабливими для малих компаній та незалежних розробників. Одним із популярних WCMS є WordPress.

Відкритість WordPress породжує можливості для великої кількості різнорідних вразливостей в системі безпеки платформи, на які ні в якому разі не можна закривати очі. Також необхідно відмітити, що кінцевий користувач не може покластися на вбудовані системи безпеки, тому необхідно приділяти особливу увагу до встановлених тем та розширень, т. я. вони можуть створити нові непередбачені вразливості в веб-додатку.

Головними вразливостями WordPress є атака грубою силою, SQL інєкція, міжсайтові скрипти, шкідливе програмне забезпечення, DDoS атаки та Старі версії WordPress і PHP.

Веб-додатки грають велику роль в житті людей. Користувачі довіряють сайтам свої особисті данні, номери кредитних карток, номери телефонів, паролі та велику кількість різноманітних важливих даних, а від безпеки цих додатків та ввірених їм даних, часто, залежить навіть життя людей.

Безпека веб-додатків – одна з найважливіших частин розробки будь якого додатку, про що дуже часто забувають.

Часто розробники веб-додатків приділяють мало уваги безпеці додатків. Команди розробки витрачають всі свою сили на візуальний дизайн та функціональність додатку, що безумовно є важливими частинами будь якого додатку, але пожертування безпекою на їх догоду завжди призводить великих проблем у майбутньому.

Дотримуючись рекомендацій та приділяючи достатню увагу захисту можна значно знизити, а в деяких випадках і повністю усунути вразливість додатку до різних типів атак, а також захистити особисті данні користувачів і свою репутацію від зловмисників.

В результаті виконання декількох кроків захист веб-додатку від всіх найрозповсюдженіших типів атак значно підвищився. Також було встановлено систему автоматичного створення резервних копій веб додатку, а антивірус за розкладом проводить сканування з ціллю виявлення випадків потрапляння шкідливого програмного забезпечення.

Джерела

1. P. De Ryck, L. Desmet, F. Piessens, and M. Johns, Primer on Client-side Web Security. Springer, 2014.
2. Документація з сайту «OWASP Top Ten», доступ за посиланням: https://owasp.org/www-project-top-ten/
3. R. C. Marchany and J. G. Tront, "E-commerce Security Issues," in Conference on System Sciences. IEEE, Jan. 2002, pp. 2500–2508
4. P. De Ryck, L. Desmet, W. Joosen, and F. Piessens, "Automatic and Precise Client-side Protection Against CSRF Attacks," in Conference on Research in Computer Security. Springer-Verlag, 2011, pp. 100–116
5. T.-Y. Li and Y. Wu, "Trust on Web Browser: Attack vs. Defense," in Conference on Applied Cryptography and Network Security. Springer, Oct. 2003, pp. 241–253.
6. Документація з сайту «WordPress.org», доступ за посиланням: https://ru.wordpress.org/about/
7. Документація з сайту «WordPress.org – Brute Force Attacks», доступ за посиланням: https://wordpress.org/support/article/brute-force-attacks/
8. Документація з сайту «WpMarmite - WordPress Blog – WordPress Tips», доступ за посиланням: https://wpmarmite.com/en/brute-force-attack-wordpress/
9. Документація з сайту «W3Techs – Usage statistics and market share of WordPress», доступ за посиланням: https://w3techs.com/technologies/details/cm-wordpress
10. Стаття з сайту «Netsparker - Identifying WordPress Websites On Local Networks (behind Firewalls) and Bruteforcing the Login Pages», доступ за посиланням: https://www.netsparker.com/blog/web-security/bruteforce-wordpress-local-networks-xshm-attack/
11. Документація з сайту «OWASP – Cross Site History Manipulation», доступ за посиланням: https://owasp.org/www-community/attacks/Cross_Site_History_Manipulation_(XSHM)

12. Стаття з сайту «Kinsta - SQL Injection: A Beginner's Guide for WordPress Users», доступ за посиланням: https://kinsta.com/blog/sql-injection/
13. Стаття з сайту «PortSwigger – SQL Injection», доступ за посиланням: https://portswigger.net/web-security/sql-injection
14. Стаття з сайту «Websiterating – Top 6 Most Common WordPress Vulnerabilities», доступ за посиланням: https://www.websiterating.com/wordpress/most-common-wordpress-vulnerabilities/
15. Стаття з сайту «Malcare – WordPress XSS Attacks», доступ за посиланням: https://www.malcare.com/blog/wordpress-xss/
16. Стаття з сайту «Maclare – How to Safely Remove Phishing from Your WordPress WebSite?», доступ за посиланням: https://www.malcare.com/blog/how-to-remove-phishing/
17. Стаття з сайту «Sucuri – Website Malware», доступ за посиланням: https://sucuri.net/guides/website-malware/
18. Стаття з сайту «Cisco – What Is a DDoS Attack?», доступ за посиланням: https://www.cisco.com/c/en/us/products/security/what-is-a-ddos-attack.html
19. Статистика з сайту «PHP Versions Stats», доступ за посиланням: https://blog.packagist.com/php-versions-stats-2021-1-edition/
20. Статистика з сайту «TechRepublic - Report», доступ за посиланням: https://www.techrepublic.com/article/report-5-ways-web-apps-suffered-in-2020-and-will-continue-to-suffer-in-2021/
21. Стаття з сайту «Imperva – Lessons Learned from Analyzing 100 Data Breaches», доступ за посиланням: https://www.imperva.com/resources/resource-library/white-papers/lessons-learned-from-analyzing-100-data-breaches/
22. Документація з сайту «WordPress.org – Plugin Tag: Brute Force», доступ за посиланням: https://wordpress.org/plugins/tags/brute-force/

23. Стаття з сайту «WpBeginner – How to Stop and Prevent a DDoS Attack on WordPress», доступ за посиланням: https://www.wpbeginner.com/wp-tutorials/how-to-stop-and-prevent-a-ddos-attack-on-wordpress/
24. Документація з сайту «GitHub – n00py/WPForce», доступ за посиланням: https://github.com/n00py/WPForce
25. Документація з сайту «WordPress – Limit Login Attempts», доступ за посиланням: https://ru.wordpress.org/plugins/limit-login-attempts-reloaded/
26. Документація з сайту «SQLMap», доступ за посиланням: https://sqlmap.org/
27. Документація з сайту «WordPress - Plugins», доступ за посиланням: https://uk.wordpress.org/plugins/prevent-xss-vulnerability/
28. Документація з сайту «WPScan», доступ за посиланням: https://wpscan.com/wordpress-security-scanner
29. Документація з сайту «Information Security», доступ за посиланням: https://baraktawily.blogspot.com/2018/02/how-to-dos-29-of-world-wide-websites.html
30. Документація з сайту «WP Cerber», доступ за посиланням: https://wpcerber.com/
31. Закон України «Про охорону праці» згідно з постановою № 1667-XI від 15.07.2021
32. Нормативно-правовий акт «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» від 25.04.2018 за № 508/31960
33. Нормативно-правовий акт «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДНС 3.3.6.042-99 від 10.12.1998 № 7
34. Постанова «Санітарні норми мікроклімату виробничих приміщень» ДНС 3.3.6.042-99 №42 від 01.12.1999

35. Наказ «Про затвердження Правил безпечної експлуатації електроустановок споживачів» № 93/2533 від 09.01.1998
36. Звід правил «Правила улаштування електроустановок» Київ 2017