

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми захисту web-сайту
розгорнутого на базі CMS Joomla»

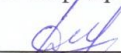
Виконав студент 4-го курсу групи КБ-21-2
спеціальності 125 «Кібербезпека»

 Ковальчук С.В.

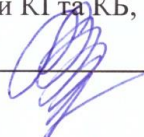
Керівник доцент кафедри КІ та КБ, к.т.н.

 Пірог О.В.

Консультант Старший викладач кафедри КІ та КБ

 Дацюк Д.В.

Рецензент професор кафедри КІ та КБ, д.т.н., доцент

 Воротніков В.В.

Житомир – 2025

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

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій Єфіменко

3 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Ковальчук Сергій Васильович

Тема роботи: «Проект підсистеми захисту web-сайту розгорнутого на базі CMS Joomla»

затверджена Наказом університету від 3 березня 2025 р. № 80/с

Термін здачі студентом закінченої роботи 27 червня 2025 р.

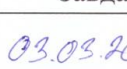
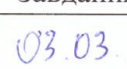
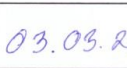
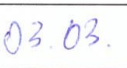
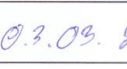
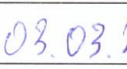
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Методи та засоби забезпечення інформаційної безпеки веб-сайтів,
розроблених на базі системи керування контентом (CMS) Joomla.

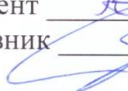
Обладнання та програмне забезпечення:

1. CMS Joomla версія 4.4.13
2. Середовище розробки : локальний сервер XAMPP
3. Інструменти безпеки: RSFirewall!

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів,
що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дацюк Д.В.	03.03.2025 	03.03.2025 
2	Дацюк Д.В.	03.03.2025 	03.03.2025 
3	Дацюк Д.В.	03.03.2025 	03.03.2025 

Студент  Сергій Ковальчук

Керівник  Олександр Пірог

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	03.03.2025	Виконано
2	Пошук теоретичної інформації	26.03.2025	Виконано
3	Вибір програмного забезпечення	01.04.2025	Виконано
4	Проведення дослідження CMS	15.04.2025	Виконано
5	Налаштування RSFierwall	01.05.2025	Виконано
6	Тестування RSFierwall	15.05.2025	Виконано
7	Написання пояснювальної записки	01.06.2025	Виконано
8	Захист	28.06.2025	Виконано

Студент  Сергій Ковальчук

Керівник  Олексій Пірог

АНОТАЦІЯ

Метою кваліфікаційної роботи було створення надійної підсистеми захисту вебзастосунку на базі CMS Joomla. Основна увага зосереджувалась на виявленні вразливостей, контролі користувацької активності, аналізі трафіку та впровадженні багаторівневого захисту. Для реалізації цих завдань було використано плагін RSFirewall!, який забезпечує комплексний захист Joomla-сайту.

У роботі проаналізовано ключові функції RSFirewall!, зокрема: антивірусне сканування, фаєрвол, захист від brute-force атак, ведення журналу подій, фільтрацію трафіку, контроль доступу та підтримку двофакторної автентифікації. Особливу увагу приділено можливостям обмеження доступу за IP-адресою, виявлення змін у файлах ядра та захисту адміністративної панелі. Описано процес інсталяції плагіна, його активації та налаштування на різних рівнях від основних до додаткових параметрів захисту. Також розглянуто переваги використання RSFirewall! у порівнянні з іншими аналогічними рішеннями для Joomla. Змодельовано повноцінний процес налаштування локального середовища на базі XAMPP, встановлення CMS Joomla та інтеграції захисного плагіна в рамках практичного завдання.

Під час тестування підтверджено ефективність основних механізмів безпеки, реалізованих за допомогою RSFirewall!. Водночас виявлено критичні налаштування, які потребують виправлення зокрема, обліковий запис "admin" і відсутність пароля до бази даних. Усунення цих проблем дозволить значно підвищити рівень безпеки сайту та забезпечити його стійкість до сучасних кіберзагроз.

Список використаних джерел містить 20 найменувань. В роботі наведено 43 зображення. Загальний обсяг роботи 76 сторінок.

Ключові слова: JOOMLA, RSFIERWALL, WAF, CMS, ЗАХИСТ, ПЛАГІНИ.

					KKIK<.KP.B – 125 –09 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект підсистеми захисту web-сайту розгорнутого на базі CMS Joomla	Літ.	Арк.	Аркушів
Розроб.		Ковальчук С.В.		27.06.25				
Керівник		Пірог О.В.		27.06.25			4	76
Реценз.		Воротніков В.В.		27.06.25				
Н. контр.		Пірог О.В.		27.06.25				
Затверд.		Єфіменко А.А.		27.06.25		ФІКТ, Гр. КБ-21-2		

ANNOTATION

The purpose of the qualification work was to create a reliable web application protection subsystem based on the Joomla CMS. The main focus was on identifying vulnerabilities, monitoring user activity, analyzing traffic, and implementing multi-level protection. To implement these tasks, the RSFirewall! plugin was used, which provides comprehensive protection for the Joomla site.

The work analyzed the key functions of RSFirewall!, including: antivirus scanning, firewall, protection against brute-force attacks, event logging, traffic filtering, access control, and support for two-factor authentication. Particular attention was paid to the capabilities of restricting access by IP address, detecting changes in core files, and protecting the administrative panel. The process of installing the plugin, activating it, and configuring it at different levels from basic to additional protection parameters is described. The advantages of using RSFirewall! in comparison with other similar solutions for Joomla are also considered. The full process of configuring a local environment based on XAMPP, installing CMS Joomla and integrating a security plugin as part of a practical task was simulated. During testing, the effectiveness of the main security mechanisms implemented using RSFirewall! was confirmed.

At the same time, critical settings were identified that require correction, in particular, the “admin” account and the absence of a password to the database. Eliminating these problems will significantly increase the level of site security and ensure its resistance to modern cyber threats.

The list of sources used contains 20 names. The work contains 43 images. The total volume of work is 76 pages. Keywords: JOOMLA, RSFIERWALL, WAF, CMS, PROTECTION, PLUGINS.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ЗАХИСТУ WEB-ДОДАТКУ	10
1.1 Важливість захисту web-додатків на базі CMS Joomla	10
1.2 Огляд основних концепції захисту веб-додатків.	12
1.3 Огляд CMS Joomla.....	19
1.4 Аналіз сучасних методів захисту веб-додатків на базі Joomla.....	21
Висновок до першого розділу	27
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПІДСИСТЕМИ ЗАХИСТУ WEBДОДАТКУ	28
2.1 Постановка задачі та вибір підходу до проєктування	28
2.2 Архітектура підсистеми захисту: рівні, компоненти, взаємодія	31
2.3 Вибір програмного забезпечення, хостингово середовища та захисних засобів	39
РОЗДІЛ 3 РЕАЛІЗАЦІЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПІДСИСТЕМИ ЗАХИСТУ.....	44
3.1 Налаштування середовища та виконання Joomla	44
3.2 Налаштування та тестування плагіна RSFirewall.....	51
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	75

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CMS – Content Management System

XSS – Cross Site Scripting

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

WAF – Web Application Firewall

CSRF – Cross-Site Request Forgery

DDoS – Distributed Denial of Service

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. У сучасному цифровому просторі, де вебсайти відіграють ключову роль у діяльності компаній, організацій та індивідуальних користувачів, питання їхнього захисту набуває особливої актуальності. Системи управління контентом, зокрема Joomla, завдяки відкритому коду, широкій функціональності та великій спільноті користувачів, нерідко стають ціллю для кіберзлочинців. Атаки на такі ресурси можуть спричинити втрату персональних даних, несанкціонований доступ до системи, порушення працездатності сайту або навіть повну його компрометацію. У зв'язку з цим забезпечення комплексної безпеки Joomla-сайту є важливою умовою його стабільного функціонування.

Метою кваліфікаційної роботи є розробка та впровадження підсистеми захисту вебсайту, розгорнутого на основі CMS Joomla, з використанням функціоналу плагіна RSFirewall!

Для досягнення поставленої мети у роботі вирішуються такі завдання:

Визначена мета обумовлює наступні завдання:

- проаналізувати основні загрози, характерні для Joomla-сайтів;
- дослідити актуальні види атак
- обґрунтувати вибір плагіна як інструменту безпеки;
- налаштувати базові та розширені функції для захисту ресурсу;
- провести тестування працездатності захисних механізмів плагіна;
- надати практичні рекомендації для користувачів щодо підвищення рівня безпеки Joomla-сайту;
- оцінити ефективність інтегрованої системи захисту у запобіганні загрозам.

Об'єктом дослідження є виступає інформаційна безпека вебресурсів, створених на базі CMS Joomla.

Предметом дослідження є методи, технології та інструменти реалізації комплексного захисту Joomla-сайтів із використанням можливостей RSFirewall!.

Під час дослідження застосовано загальнонаукові методи, зокрема: аналіз, синтез, моделювання, порівняння, логічне узагальнення та систематизація.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Практичну частину виконано на основі експериментального підходу, що передбачає розгортання середовища ХАМРР, встановлення Joomla та налаштування плагіна RSFirewall! у тестовому середовищі. Оцінювання ефективності захисних заходів проводилося через спостереження за журналами активності, поведінкою системи при моделюванні атак, функціональністю сканування на наявність шкідливого коду, перевіркою реалізації обмежень доступу, двофакторної автентифікації та фільтрації запитів.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ЗАХИСТУ WEB-ДОДАТКУ

1.1 Важливість захисту web-додатків на базі CMS Joomla

Тема є досить актуальною тому що, Joomla став однією з найпопулярніших систем керування контентом (CMS) у світі, керуючи 2.1% усіх веб-сайтів в Інтернеті. Його гнучкість, простота використання та широкий набір тем і плагінів роблять його привабливим вибором для власників веб-сайтів.

Забезпечення швидкого та легкого доступу для користувачів без технічного досвіду в Інтернеті стало однією з ключових причин популярності блогінгу по всьому світу. Ведення блогів сьогодні стало не лише популярним явищем, а й засобом комунікації для постійних користувачів мережі, а також для великих корпорацій, новинних організацій та освітніх установ. Ідея, що відсутність веб-сайту організації означає відсутність самої організації, особливо актуальна для блогів.

У перші роки свого існування Joomla була відома переважно як платформа для створення веб-сайтів і порталів. Однак з часом вона почала привертати дедалі більше уваги, оскільки користувачі помічали її гнучкість і потенціал. Спільнота Joomla активно розвивалася, об'єднуючи розробників, програмістів і дизайнерів, які спільно працювали над удосконаленням платформи та її перетворенням у багатофункціональний інструмент для створення різноманітних веб-сайтів, включаючи корпоративні портали, онлайн-магазини та блоги.

Одним із ключових аспектів успішного використання Joomla був вибір надійного хостингу, що забезпечував стабільність та ефективну роботу сайтів на цій платформі. Завдяки розширенням та шаблонам, Joomla стала універсальним рішенням для багатьох веб-проектів.

Joomla є однією з провідних CMS-платформ, що вирізняється зручним процесом інсталяції та широкими можливостями для гнучкого налаштування зовнішнього вигляду інтерфейсу завдяки використанню шаблонів та розширень.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, Joomla є безкоштовною технологією, що дозволяє використовувати її для створення як комерційних, так і приватних вебсайтів.

На сьогодні існує можливість вибору однієї з декількох програм та платформ для керування блогами та веб-сайтами загалом. Joomla виокремлюється серед провідних платформ, пропонуючи широкий функціонал для ефективного управління контентом. Відмінною особливістю Joomla є його здатність створювати веб-сайти навіть без значних технічних знань.

Однак навіть найсучасніша система управління контентом не гарантує повної безпеки Інтернет-ресурсу без належної конфігурації та заходів забезпечення безпеки.

Joomla не є винятком серед систем керування вмістом блогів, адже вона володіє найбільшою популярністю та функціональністю. Проте, разом із своєю популярністю приходить і ризик безпеки, з якими необхідно рахуватися. Ось кілька причин, чому Joomla може бути уразливим:

- **Доступність конфігураційних файлів:** Joomla використовує стандартний конфігураційний файл `configuration.php`, який за замовчуванням зберігається в кореневому каталозі сайту. У разі неправильного налаштування прав доступу або витоку даних, цей файл може стати ціллю для злоумисників.
- **Єдинна кодова база та адміністративна панель:** Адмін-панель Joomla розміщується в межах того ж домену, що й основний сайт, та використовує спільну кодову базу. У разі компрометації облікових даних адміністратора злоумисник може отримати повний доступ до системи керування вмістом і вмісту сайту.
- **Плагіни та теми:** Користувачі-адміністратори можуть інсталиувати плагіни та теми, які можуть мати вразливості або небажану функціональність. Неофіційні теми чи плагіни, а також ті, які не були оновлені або перевірені на безпеку, можуть стати джерелом потенційних загроз для веб-сайту.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Важливо бути уважними і вживати заходів безпеки, таких як регулярне оновлення платформи та плагінів, використання надійних паролів, а також перевірка безпеки та аудит коду для виявлення можливих вразливостей.

Таким чином важливість захисту вебдодатків на базі CMS Joomla обумовлена її популярністю та гнучкістю. Для надійного функціонування вебсайтів, створених на цій платформі, необхідно дотримуватись належних заходів безпеки – регулярних оновлень, контролю доступу та обережного використання плагінів і шаблонів.

1.2 Огляд основних концепції захисту веб-додатків.

Безпека вебдодатків (Web AppSec) - це невід’ємна складова роботи будь-якого веб-сайту, яка забезпечує захист від потенційних загроз. Як і в будь-якому програмному забезпеченні, існують ризики вразливостей, які можуть бути використані зловмисниками для атак [\[1\]](#). Web AppSec ставить за мету запобігання цим ризикам шляхом впровадження безпечних практик розробки та використання захисних протоколів на всіх етапах життєвого циклу розробки програмного забезпечення. Це включає не лише усунення вразливостей на етапах проектування та розробки, але і постійний моніторинг та оновлення для забезпечення безпеки веб-додатків в умовах постійно змінюючогося середовища Інтернету.

В цей же час , коли підприємства та індивідуальні користувачі широко використовують веб-додатки для різноманітних завдань, від онлайн-покупок до управління конфіденційними даними, безпека цих програм набуває надзвичайної важливості. Вразливості безпеки можуть призвести не лише до компрометації особистої інформації користувача, але й спричинити фінансові втрати, пошкодити репутацію та мати юридичні наслідки.

Вебдодатки вже давно знаходяться під постійним ризиком з боку хакерів. Їхній потенціал надати доступ до цінної інформації робить їх привабливою мішенню, а їхню відносну доступність робить їх легкими у використанні. Успішна атака може мати серйозні наслідки, включаючи фінансові втрати, пошкодження

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		12

репутації бренду та втрату довіри клієнтів. Для деяких організацій наслідки серйозного порушення безпеки можуть бути катастрофічними.

Хоча мережевий брандмауер є важливою складовою захисту в центрі обробки даних, він сам по собі не є достатнім. Багато нових атак, спрямованих на програми та кінцевих користувачів, вимагають додаткового захисту, якого не можуть забезпечити традиційні брандмауери або системи запобігання вторгненням (IPS). Методи виявлення на основі сигнатур, оцінка репутації IP-адрес та глибока аналітика пакетів можуть допомогти в зупиненні деяких з цих складних загроз, але вони мають свої обмеження. Організаціям потрібні додаткові рішення, такі як брандмауери веб-додатків, контролери доставки додатків та інтеграція ізольованих середовищ програм, щоб ефективно протистояти цим новим загрозам як для центру обробки даних, так і для кінцевих користувачів.

Кіберзлом може мати серйозні наслідки, що виходять за межі простих грошових втрат. Хоча багато веб-додатків стикаються з різними вразливостями, окремі сектори, такі як блокчейн і криптовалюти, стають особливо сприйнятливими до кіберзагроз. Забезпечення безпеки веб-додатків стає критично важливим для успіху будь-якого онлайн-бізнесу. Дані, що циркулюють в цифровій економіці, мають величезний потенціал для інновацій, але й можуть стати об'єктом різноманітних кіберзагроз.

Зважаючи на глобальний характер Інтернету, веб-додатки можуть бути атаковані з будь-якого кута, а зловмисники можуть мати різний рівень кваліфікації. Безпека веб-додатків перш за все стосується захисту веб-сайтів, онлайн-інструментів та інтерфейсів програмного забезпечення (API).

Існує багато типів загроз безпеці веб-додатків, від масштабних збоїв у мережі до цілеспрямованих атак на бази даних. Список Топ-10 Open Web Application Security Project (OWASP) включає десять найбільш критичних ризиків для безпеки веб-додатків, які можуть призвести до серйозних наслідків для їхньої безпеки та надійності.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

- **Вразливості ін'єкції:** Веб-додатки можуть бути вразливими до атак, таких як міжсайтовий скриптинг (XSS) і SQL-ін'єкція. Ці атаки дозволяють зловмисникам впроваджувати шкідливі скрипти або запити до бази даних, що може призвести до отримання несанкціонованого доступу до інформації або зміни даних.
- **Атаки на відмову в обслуговуванні (DoS) і розподілені атаки на відмову в обслуговуванні (DDoS):** Ці атаки спрямовані на перевантаження цільового сервера або мережі, що призводить до відмови в обслуговуванні законних користувачів. Вони можуть спричинити затримки у відповідях сервера або повне припинення його роботи.
- **Порушення контролю доступу:** Зловмисники можуть отримати несанкціонований доступ до функціоналу або даних, обходячи механізми автентифікації та авторизації. Це може дозволити їм отримати конфіденційну інформацію або виконувати небажані дії.
- **Криптографічні збої:** Неправильне застосування криптографічних методів може призвести до розкриття конфіденційної інформації. Це може включати неналежне зберігання або передачу паролів чи інших конфіденційних даних, що порушує вимоги щодо захисту даних і може спричинити невідповідність стандартам безпеки.

Мережева безпека і безпека веб-додатків мають свої відмінності. У мережевій безпеці часто використовуються засоби захисту периметра, такі як брандмауери, для контролю та фільтрації трафіку. Наприклад, адміністратори можуть налаштувати брандмауери, щоб дозволити певним IP-адресам або користувачам доступ до певних служб, блокуючи решту.

Проте захист периметра мережі не завжди ефективний для захисту веб-додатків від зловмисних атак. Веб-сайти та програми мають бути доступні для всіх користувачів, тому адміністратори зазвичай дозволяють весь вхідний трафік на портах 80 (HTTP) і 443 (HTTPS), сподіваючись, що всі користувачі діятимуть згідно правил. Однак це відкриває можливості для атак, таких як вразливості веб-

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

додатків і атаки міжсайтового сценарію, які можуть бути використані для отримання несанкціонованого доступу до системи або витіснення користувачів

Мережеві брандмауери не здатні аналізувати веб-трафік, що надходить до або від веб-додатків, тому вони не можуть блокувати потенційно шкідливі запити, які надсилаються зловмисниками, що намагаються використати вразливості, такі як SQL-ін'єкція або міжсайтовий сценарій.

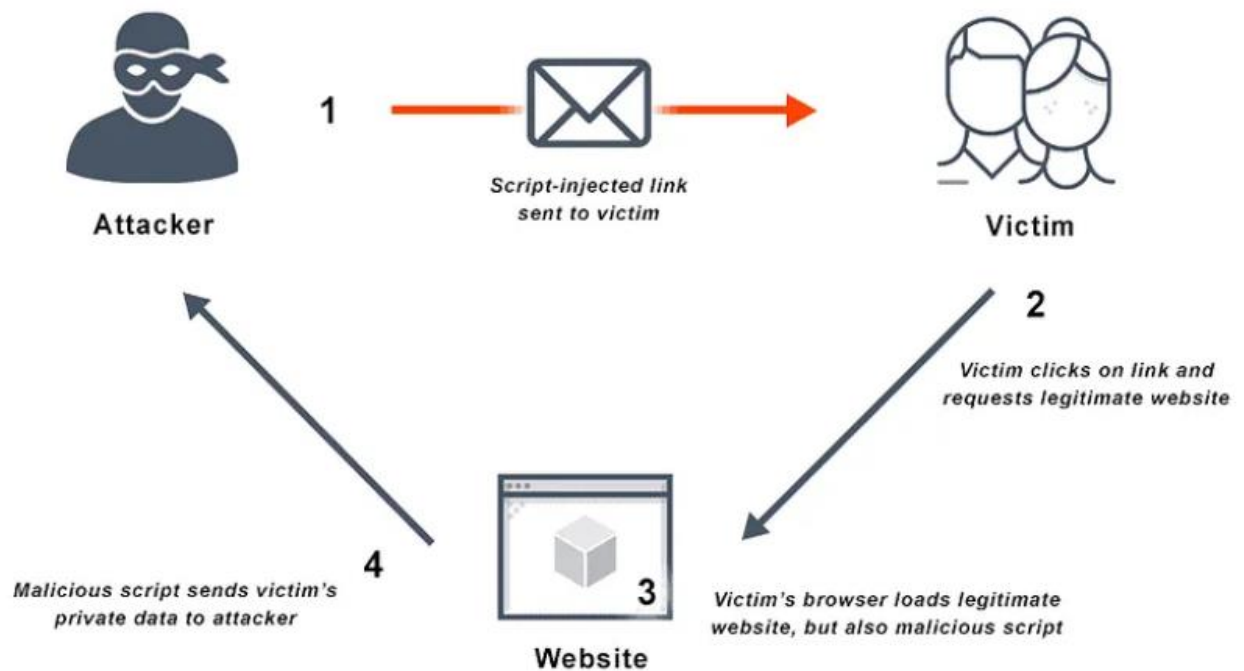


Рис 1.1 – Схема атаки XSS

Під час розробки проектів слід включати перевірку цих вразливостей у проектні документи, а також проводити тестування коду на ранніх етапах розробки. Це дозволяє переконатися, що нові вразливості не виникають, а наявні вже виправляються.

Безкомпромісна безпека веб-додатків

У 2017 році більшість успішних атак на інформаційні мережі здійснювались через уразливості у веб-додатках. У тому ж році близько 73% порушень мережевої безпеки були пов'язані саме з недоліками захисту веб-застосунків, згідно з даними галузевих досліджень. За останні роки організації активно розширюють свою веб-

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				15
Змн.	Арк.	№ докум.	Підпис	Дата		

присутність через різноманітні додатки, які охоплюють як клієнтську сферу (наприклад, онлайн-покупки або банківські послуги), так і внутрішні корпоративні системи.

Проте зростання популярності веб-додатків викликає збільшення загроз кібербезпеці. Незахищене програмне забезпечення стає серйозним ризиком для критичних інфраструктур країн, оскільки воно стає місцем атак для різних бізнес-секторів і частиною сучасної культури.

Тому надзвичайно важливо, щоб веб-програми регулярно перевірялися на безпеку, і вразливості, якщо вони виявлені, усувалися негайно. А безпека програми повинна бути пріоритетом на ранніх етапах розробки, а не тільки на завершальних етапах проекту, як це часто відбувається.

Незважаючи на нарощування популярності кіберзлочинності та її негативний вплив на потерпілих, рівень захисту, що застосовується організаціями, лишається дуже низьким. Фахівці з безпеки, такі як Nettitude, стверджують, що безпека повинна бути визнана як пріоритетний аспект на рівні управління для будь-якого бізнесу та є критичною проблемою для керівників у вищому рівні ІТ, таких як CISO та СТО.

Для перевірки та виправлення безпеки веб-додатків рекомендується залучати незалежну компанію, що має сертифікат від CREST і спеціалізується на тестуванні на проникнення у веб-додатках, наприклад, Nettitude.

Брандмауери веб-програм (WAF) виступають в ролі бар'єра між веб-додатком та Інтернетом, контролюючи весь вхідний та вихідний трафік. Їх завданням є виявлення та блокування потенційних загроз на основі попередньо встановлених правил безпеки. WAF можуть ефективно захищати від широкого спектру веб-атак, таких як SQL-ін'єкції, міжсайтові скриптові атаки (XSS) та DDoS-атаки.

WAF функціонують на рівні додатків, аналізуючи вміст кожного пакета даних на наявність шкідливого коду або підозрілої активності. Для ідентифікації загроз вони використовують різноманітні методи, такі як виявлення за сигнатурою,

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				16
Змн.	Арк.	№ докум.	Підпис	Дата		

виявлення на основі аномалій та аналіз поведінки. За допомогою WAF організації можуть суттєво підвищити рівень безпеки своїх веб-додатків.

Управління зовнішніми атаками та поверхнею (EASM) охоплює виявлення та управління ризиками безпеки, що стосуються публічно доступних цифрових активів організації. Цей процес включає ідентифікацію, каталогізацію та постійний моніторинг всіх зовнішніх активів, таких як веб-сайти, веб-додатки, сервери та хмарні служби. Метою EASM є отримання повного розуміння цифрового сліду організації та виявлення потенційних вразливостей в ньому.

Інструменти та практики EASM допомагають організаціям виявляти незахищені активи, які можуть бути пропущені, такі як застарілі веб-додатки, незахищені бази даних або забуті цифрові сервіси. Без належного керування ці активи можуть стати легкою мішенню для зловмисників. EASM також включає постійний моніторинг поверхні атаки для виявлення змін або незвичних дій, які можуть вказувати на потенційну загрозу безпеці. Завдяки впровадженню EASM, організації можуть вчасно вирішувати ризики безпеки, зменшити площу атаки та посилити загальний рівень безпеки своїх веб-додатків.

Щодо рішення захисту веб-додатків і API (WAAP) надають комплексний захист як для веб-додатків, так і для API. Вони об'єднують у собі функціонал WAF, захист від DDoS, керування ботами та безпеку API в одному рішенні. Рішення WAAP не лише захищають від типових веб-атак, але й мають розширені можливості виявлення загроз за допомогою методів машинного навчання та аналізу поведінки.

API стали ключовим компонентом багатьох веб-додатків, що дозволяють їм взаємодіяти з іншими програмами та службами. Рішення WAAP забезпечують надійний захист для API, фільтруючи лише законні запити та запобігаючи поширеним векторам атак.

У своїй поточній формі, WAF є ефективним інструментом для захисту веб-програм від найнебезпечніших типів атак, таких як міжсайтовий скриптинг (XSS), міжсайтова підробка, включення файлів, SQL-ін'єкції або DDoS-атаки, серед

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		17

інших. Здійснювати розгортання брандмауера веб-додатків можна за допомогою апаратного, віртуального або хмарного підходу.

Однак важливо мати на увазі, що брандмауер веб-програм не повинен бути єдиним засобом захисту. WAF найбільш ефективно функціонує в тандемі з традиційним мережевим брандмауером (робота на четвертому рівні моделі OSI) або брандмауером нового покоління. Інтеграція брандмауера програми з іншими системами безпеки, такими як виявлення та запобігання вторгненням (IDS/IPS), також важлива.

Важливо розуміти, що впровадження WAF не гарантує постійного захисту активів. Оскільки веб-програма постійно змінюється, правила безпеки, які контролюють роботу брандмауера веб-програми, потребують регулярного оновлення. Проте правильно реалізований, інтегрований з іншими брандмауерами та налаштований WAF стане вашим найкращим союзником у боротьбі з потенційними атаками.

Висновуючи, захист веб-додатків від вразливостей та атак - це невід'ємна частина будь-якого інтернет-орієнтованого бізнесу чи проекту. Ефективна стратегія захисту передбачає комплексний підхід, що охоплює аутентифікацію, авторизацію, шифрування, захист від різних видів атак, валідацію даних та постійний моніторинг. Забезпечення безпеки веб-додатків є постійним процесом, і краще починати його з самого початку розробки, щоб уникнути можливих проблем у майбутньому. Тільки такий підхід може гарантувати захист інформації користувачів та стійку роботу веб-додатку у відповідь на сучасні виклики безпеки.

Захист веб-додатків є критично важливим компонентом цифрової безпеки сучасного бізнесу та організацій. У світі, де кіберзагрози постійно еволюціонують, лише багаторівневий та комплексний підхід може забезпечити ефективну протидію атакам. Впровадження безпечних практик на всіх етапах розробки, використання брандмауерів веб-додатків (WAF), систем WAAP, а також постійний моніторинг цифрових активів і поверхні атак — це не розкіш, а необхідність. Лише системне впровадження стандартів безпеки, регулярне тестування на вразливості та

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		18

оперативне реагування на інциденти дозволяє гарантувати конфіденційність, цілісність і доступність даних у веб-середовищі.

1.3 Огляд CMS Joomla

Забезпечення безпеки веб-сайту має критичне значення, оскільки будь-яке порушення може призвести до серйозних наслідків, таких як шкода репутації, фінансові втрати та юридична відповідальність. Як власник або адміністратор веб-сайту, вжиття профілактичних заходів для захисту вашого веб-сайту є надзвичайно важливим для забезпечення конфіденційності інформації та захисту як вашої організації, так і користувачів.



Рис 1.2 Логотип Joomla

Joomla, як одна з найпоширеніших систем управління вмістом (CMS), займає значну частину Інтернету. Його популярність зумовлена зручним інтерфейсом, широкою екосистемою плагінів і налаштованих тем [2]. Однак ця широка поширеність також робить Joomla привабливою мішенню для кіберзлочинців.

Оскільки Joomla має відкритий вихідний код, весь світ має доступ до його вихідного коду, включаючи зловмисників, які можуть використовувати вразливості в застарілих або неправильно розроблених плагінах і темах. Багато користувачів також не оновлюють своє ядро Joomla, теми та плагіни регулярно, залишаючи їхні веб-сайти вразливими до відомих уразливостей безпеки.

Аудит безпеки Joomla – це аналіз заходів безпеки та вразливостей веб-сайту Joomla. Під час аудиту безпеки Joomla розробники оцінюють конфігурацію веб-сайту, налаштування, теми, плагіни, ролі користувачів та інші елементи, щоб визначити потенційні ризики безпеки або слабкі місця, якими користуються зловмисники.

По суті, це повна оцінка поточного стану безпеки веб-сайту. Їхня мета — виявити слабкі місця та підвищити загальну безпеку сайту, захистити його від

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

несанкціонованого доступу, витоку даних, зараження шкідливим програмним забезпеченням та інших потенційних загроз.

Такі аудити можуть усунути вразливі місця системи безпеки, перш ніж вони призведуть до втрати даних або будь-якої іншої атаки. У цьому сенсі вони є інвестицією, яка захищає ваш бізнес від подальших, більших витрат.

Joomla можна налаштувати на безліч способів. Деякі з цих налаштувань можуть включати код та неправильні конфігурації, які можуть впливати на безпеку Joomla. Одним із популярних способів налаштування Joomla є використання плагінів та тем.

Плагіни та теми Joomla можуть значно відрізнитися за якістю та безпекою. Незважаючи на те, що команда Joomla провела багато роботи для того, щоб допомогти розробникам створювати більш безпечні плагіни та теми, не всі розробники дотримуються передових методів безпеки. Це особливо помітно під час використання плагінів, які погано підтримуються, або плагінів, отриманих із недостовірних джерел.

Видалення невикористаних тем і плагінів Joomla має велике значення для зменшення поверхні атак вашого веб-сайту. Навіть якщо теми та плагіни не використовуються, вони можуть становити загрозу безпеці, якщо вони застарілі або вразливі. Видалення їх мінімізує можливість використання та покращує загальний рівень безпеки вашого веб-сайту.

Joomla залишається однією з найпопулярніших систем управління вмістом завдяки своїй зручності використання та широкому вибору функцій, які пропонує. Однак, ця платформа також відома своєю вразливістю до кібератак через велику кількість плагінів і тем з відкритим вихідним кодом. Підтримка безпеки від Joomla відносно обмежена, тому власники сайтів повинні бути відповідальними за забезпечення безпеки своїх веб-ресурсів. Це може включати видалення застарілих плагінів та тем, регулярне оновлення ядра Joomla і додатків, а також використання надійних рішень для захисту від кіберзагроз. У цілому, з правильними заходами

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				20
Змн.	Арк.	№ докум.	Підпис	Дата		

безпеки, Joomla може бути потужним і безпечним інструментом для будь-якого веб-проекту.

Безпека веб-сайту на основі Joomla потребує постійної уваги та відповідального підходу з боку власника або адміністратора [4]. Попри зручність та функціональність CMS Joomla, її відкритий вихідний код і широка екосистема плагінів створюють додаткові ризики, які можуть бути використані зловмисниками. Регулярне оновлення компонентів, аудит безпеки, видалення застарілих або невикористаних плагінів і тем, а також використання лише перевірених розширень — це ключові дії, що забезпечують захист ресурсу. Впровадження цих заходів значно знижує ризик атак і робить Joomla надійною платформою для побудови захищених веб-сайтів.

1.4 Аналіз сучасних методів захисту веб-додатків на базі Joomla

Безпека в Joomla не лише про захист контенту сайту. Вона має критичне значення для збереження майбутніх доходів, довіри клієнтів і загальної стабільності вашого веб-проекту. Від зломів до SEO-спаму, загрози, що чекають на ваш веб-сайт, можуть бути численними та різноманітними. Розробка та підтримка веб-платформи — це значна інвестиція, і вона повинна бути захищена від можливих ризиків, які можуть шкодити майбутнім прибуткам і довірі користувачів.

Однією з основних причин, чому безпека в Joomla настільки важлива, є захист конфіденційної інформації. Багато сайтів Joomla обробляють особисті дані користувачів, такі як інформація про платіжні дані, дані для входу та інші особисті дані. Порушення безпеки може призвести до витоку або втрати цих даних, що може стати причиною фінансових втрат та юридичних проблем. Забезпечення надійного захисту цих даних — це не лише моральний, а й юридичний обов'язок у багатьох сферах діяльності.

З кіберзагрозами, які постійно еволюціонують, зловмисники постійно розробляють нові методи використання вразливостей [3]. Несанкціонований доступ до вашого веб-сайту загрожує його функціональності та знижує продуктивність.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Крім того, зловмисники можуть спричинити перебої в роботі вашого сайту шляхом DDoS-атак, навіть не отримуючи прямого доступу. Joomla регулярно оновлює свої функції безпеки, щоб усунути ці вразливості. Тому підтримка актуальних версій ядра, тем і плагінів Joomla є життєво важливою для захисту вашого сайту від цих нових загроз.

Уразливості у Joomla дійсно досить поширені, і, на жаль, вони можуть з'являтися одна за одною. Нижче перераховано декілька найпоширеніших і найбільш тривожних типів атак, а також тактики запобігання та пом'якшення, які користувачі Joomla повинні застосовувати:

Атаки грубою силою: Спроби входу методом проб і помилок утворюють основу атаки грубої сили, де зловмисники перебирають безліч можливих комбінацій паролів, намагаючись знайти правильний. Це може бути досить просто, оскільки багато користувачів використовують надзвичайно слабкі паролі або такі, які легко вгадати. На жаль, як тільки хакерам вдасться проникнути, вони можуть отримати доступ до вашого сайту і здійснити різноманітні шкідливі дії.

Рішення : Вимоги до пароля стали строгішими. Користувачам тепер необхідно створювати складні комбінації літер, цифр і спеціальних символів. Ще краще — використання двофакторної автентифікації, яка включає в себе не лише стандартний пароль, а й додаткові коди, надіслані на мобільні пристрої або авторизовані облікові записи електронної пошти.

Застаріле програмне забезпечення, плагіни та теми: Без вчасного оновлення тем, плагінів та програмного забезпечення Joomla ризик вразливостей значно зростає. Оновлення містять важливі патчі, які забезпечують надійний захист від багатьох типових проблем. Особливу увагу слід звертати на уразливості плагінів, оскільки вони часто використовуються хакерами для компрометації сторінок Joomla.

Рішення: Рішення цієї проблеми просте: зобов'язуйтеся регулярно оновлювати всі версії Joomla, теми та особливо плагіни. Не має значення, чи

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		22

здається, що ваш сайт працює ідеально; ваша нездатність ввести оновлення зараз може спричинити хаос на вашому сайті в майбутньому. Ми також рекомендуємо регулярно сканувати Joomla на наявність уразливостей .

Неавторизовані адміни: Недоліки плагіна можуть дозволити неавторизованим користувачам мати доступ до адміністративних функцій, що дає їм можливість виконувати дії, які зазвичай доступні лише користувачам з високим рівнем привілеїв. Це одна з проблем, яка часто залишається непоміченою, але має великий потенціал завдати серйозної шкоди вашому сайту.

Рішення: Запобігаючи атакам грубої сили, ми також убезпечуємо себе від потенційних несанкціонованих адміністраторів, які можуть намагатися отримати доступ до привілеїв, що не належать їм. Дотримання загального принципу найменших привілеїв також відіграє ключову роль у цьому процесі, обмежуючи доступ лише до необхідного мінімуму функцій для кожного учасника.

Шкідливе програмне забезпечення: Шкідливе програмне забезпечення, спеціально створене для завдання максимальної шкоди, може стати причиною серії проблем для комп'ютерів, мереж або серверів. Цей вид загрози, також відомий як «зловмисне програмне забезпечення», здатне інфікувати системи та викрадати вразливі дані. Це лише одна з численних небезпек, що виникають через застарілі теми та плагіни.

У зв'язку з тим, що шкідливе програмне забезпечення має багато різних виявів, його складно виявити або запобігти, а наслідки його дій можуть бути надзвичайно серйозними. Такі програми можуть включати в себе шпигунське програмне забезпечення, яке збирає конфіденційну інформацію, або розшифровувальне програмне забезпечення, яке блокує доступ до системи або даних, вимагаючи викуп. Боротьба з цими загрозами може виявитися витратною і складною задачею.

Рішення: Не існує універсального рішення, яке б миттєво вирішило всі проблеми зі зловмисним програмним забезпеченням, проте найбільш ефективним

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		23

є комплексний та багаторівневий підхід до безпеки. Оновлені плагіни та теми становлять надійне вихідне положення, проте настільки ж важливі безпечні методи входу, регулярні резервні копії та обмежений доступ адміністратора.

Міжсайтовий сценарій (XSS): Міжсайтові сценарії (XSS) — це атаки, які використовуються для впровадження шкідливих сценаріїв у веб-сайти або програми, навіть тих, які видавалися надійними. Це може призвести до вразливості для відвідувачів сайту, а також витоку конфіденційної інформації, такої як файли cookie та інші дані, що зберігаються у браузері.

Рішення: Щоб зменшити ризик вразливостей міжсайтових сценаріїв, можна вжити різних заходів для поліпшення очищення даних. Фільтрування вхідних даних та кодування вихідних може значно допомогти у цьому. Також використання спеціальних заголовків відповідей дозволяє браузерам правильно інтерпретувати дані, що відправляються.

Файл `configuration.php` містить важливу інформацію про конфігурацію вашого сайту на Joomla і є одним з ключових елементів забезпечення безпеки вашого блогу. Захищаючи цей файл, ви збільшуєте безпеку ядра вашого сайту.

Переміщення файлу `configuration.php` на рівень вище кореневого каталогу є ефективною тактикою, яка ускладнює завдання хакерам. Це робить файл недосяжним для зовнішніх атак і додає додатковий шар захисту до вашого сайту.

Процес переміщення файлу `configuration.php` досить простий і не вимагає складних навичок програмування. Ви просто переносите файл на рівень вище, ніж кореневий каталог, і оновлюєте посилання в конфігураційних файлах Joomla. Це мінімізує ризик доступу до важливої інформації і забезпечує додатковий захист вашого сайту.

Якщо користувач має права адміністратора в панелі керування Joomla сайту, він може вносити зміни у будь-які файли, що входять до складу системи. Це означає, що він має можливість редагувати файли плагінів та тем.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Проте, якщо заборонити можливість редагування файлів з панелі адміністрування, ніхто, навіть якщо хакер отримає доступ до вашого адміністративного облікового запису, не зможе внести зміни у файли вашої інсталяції Joomla. Це може стати ефективним заходом безпеки, що додасть додатковий шар захисту до вашого сайту.

Один із найефективніших способів забезпечити безпеку сайту - це регулярне оновлення програмного забезпечення. Відвідавши панель адміністратора Joomla, ви зможете перевірити наявність оновлень та встановити їх. Joomla пропонує зручний інтерфейс для масового оновлення плагінів. Проте не забудьте зробити резервну копію свого сайту та перевірити його після оновлення, щоб упевнитися у його правильному функціонуванні.

Оновлення грають важливу роль у забезпеченні безпеки вашого сайту, допомагаючи уникнути вразливостей у вашому коді, які можуть бути використані хакерами для різних видів атак, таких як ін'єкція коду, XSS або CSRF. Регулярна перевірка наявності оновлень для вашого сервера, CMS (наприклад, Joomla), фреймворків (наприклад, Laravel) та плагінів (наприклад, VirtueMart) є обов'язковою. Після виявлення оновлень важливо встановлювати їх якомога швидше, щоб забезпечити найвищий рівень безпеки для вашого сайту.

Файрволи та антивіруси є важливими засобами для запобігання атакам, таким як атаки на відмову у обслуговуванні та прослуховування, а також іншим шкідливим діям. Встановлення та налаштування файрволу на сервері допомагає блокувати небажані запити та забезпечує захист вашого сайту від несанкціонованого доступу. Крім того, важливо встановлювати та регулярно оновлювати антивірусне програмне забезпечення на комп'ютері для сканування та видалення будь-яких шкідливих файлів або програм. Такий комплексний підхід допомагає забезпечити безпеку вашого веб-сайту та комп'ютера.

Запровадження надійних методів автентифікації є критично важливим для захисту веб-сайту від несанкціонованого доступу до його серверної частини. Розглянувши можливість використання багатофакторної автентифікації (MFA),

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		25

можна додати додатковий шар безпеки. MFA передбачає, що користувачі повинні підтвердити свою ідентичність двома або більше способами, наприклад, введенням пароля та одноразового коду, який надсилається на їхній мобільний пристрій. Такий підхід значно ускладнює завдання потенційним зловмисникам і забезпечує більший рівень захисту для вашого веб-сайту.

Організаційно важливим етапом у створенні системи захисту веб-додатків є тестування на проникнення, або етичне вторгнення. Цей метод є оптимальним для перевірки захищеності інформаційної системи шляхом імітації спрямованих атак. Тест на проникнення надає можливість оцінити захищеність веб-додатків від несанкціонованого доступу, застосовуючи різні моделі вторгнень.

У випадку тестування на проникнення веб-додатків, увага зосереджена виключно на оцінці рівня захисту самого додатка. Процес полягає у активному аналізі додатків та пошуку в них вразливостей, технічних помилок та інших проблем. Всю знайдену інформацію про слабкі місця відображається в підсумковому звіті, що допомагає покращити безпеку системи.

Вибір відповідного плагіна безпеки може суттєво вплинути на керування та ефективність нашого веб-сайту. Потрібно урахувати внутрішні та зовнішні фактори, пов'язані з нашим сайтом та нашими потребами.

У репозиторії Joomla доступно тисячі плагінів безпеки з різними рівнями надійності, функціональності та зручності використання. Один із найпопулярніших розширень безпеки для Joomla — **RSFierwall!**, який надає вбудований **брандмауер** для захисту від відомих вразливостей, автоматичне оновлення файлів ядра, а також можливість налаштування прав доступу. Крім того, деякі хостинг-провайдери пропонують інтеграцію з **CDN-сервісами** для покращення швидкодії та доступності сайту. Вибір захисного розширення має базуватись на конкретних потребах сайту, щоб забезпечити оптимальну безпеку та продуктивність.

Безпека Joomla-сайту — це не одноразова дія, а постійний процес, що включає оновлення компонентів, впровадження надійної автентифікації, захист критичних файлів, а також використання спеціалізованих розширень, як-от Admin

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Tools чи RSFirewall!. Ігнорування цих аспектів може призвести до втрати конфіденційних даних, фінансових втрат і втрати довіри користувачів. Забезпечуючи всебічний захист сайту, ви не лише зберігаєте його стабільність і продуктивність, але й гарантуєте довготривалу ефективність веб-проекту в умовах постійно зростаючих кіберзагроз.

Висновок до першого розділу

Отже, важливість захисту вебдодатків на основі CMS Joomla обумовлена її широким розповсюдженням, відкритим вихідним кодом та гнучкістю в налаштуванні. Ці переваги водночас створюють потенційні вектори атак, які можуть бути використані зловмисниками для компрометації системи. Тому забезпечення безпеки Joomla-сайту є першочерговим завданням для розробників і адміністраторів.

Надійний захист ресурсу вимагає впровадження комплексного підходу: регулярних оновлень ядра системи, шаблонів і плагінів, налаштування прав доступу, а також обережного використання сторонніх розширень. Не менш важливою є систематична перевірка на вразливості, проведення аудитів безпеки та очищення системи від застарілих або непотрібних компонентів.

У сучасних умовах, коли кіберзагрози постійно ускладнюються, ефективним є багаторівневий захист: поєднання брандмауерів (WAF), систем WAAP, механізмів автентифікації, моніторингу активності та автоматизованих засобів реагування на інциденти. Важливу роль відіграють спеціалізовані інструменти, як-от RSFirewall, що забезпечують додатковий рівень безпеки.

Загалом, безпека Joomla-сайту — це не одноразовий захід, а безперервний процес, який вимагає технічної грамотності, системного підходу та постійного вдосконалення. Лише дотримання сучасних практик інформаційної безпеки дозволяє зберегти стабільність, продуктивність і репутацію вебпроекту в умовах зростаючих кіберзагроз.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПІДСИСТЕМИ ЗАХИСТУ WEBДОДАТКУ

2.1 Постанова задачі та вибір підходу до проєктування

Метою є створення надійної, масштабованої підсистеми захисту web-додатку, реалізованого на базі CMS Joomla, з урахуванням сучасних загроз. Ця підсистема має забезпечувати багаторівневий захист, включаючи виявлення та запобігання атак на рівні додатку, так і на рівні мережі. Важливо передбачити автоматичне оновлення компонентів і регулярний аудит безпеки для мінімалізації ризиків.

Необхідно інтегрувати механізми захисту від ботів, брутфорс-атак і соціальної інженерії, а також забезпечити надійну систему автентифікації користувачів. Крім того, ця підсистема має бути гнучкою для адапації до нових вразливостей і відповідати вимогам масштабування при зростанні навантаження.

Для підвищення надійності захисту важливо впровадити моніторинг активності користувачів і системних логів, що дозволить оперативно виявляти підозрілі дії. Система має підтримувати централізоване управління політиками безпеки, щоб спростити адміністрування у великих проєктах. Особлива увага приділяється захисту від найпоширеніших видів атак, таких як SQL-ін'єкції, XSS, CSRF, а також атакам на сесії. Важливо реалізувати шифрування критичних даних, як на рівні зберігання, так і під час передачі між клієнтом і сервером.

Резервне копіювання та швидке відновлення після інцидентів мають стати невід'ємною частиною підсистеми безпеки підприємства, такими як SIEM і WAF, для більш комплексного захисту. Необхідно передбачити механізми автоматичного виявлення і блокування підозрілої активності, з можливістю оповіщення адміністратора.

Важливим аспектом є підтримка багатофакторної автентифікації для всіх рівнів доступу. Підсистема має бути побудована з урахуванням принципів безпеки за замовчуванням, мінімалізації прав і захисту від помилок конфігурації. В кінцевому результаті, рішення повинно забезпечити баланс між безпекою, зручністю користування та продуктивністю веб-додатку.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Хоча ядро Joomla активно підтримується розробниками, час від часу у ньому все ж таки з'являються певні уразливості. Однією з найбільш поширених загроз є SQL-інєкції, які дозволяють зловмисникам вставляти шкідливі SQL-запити через форми або URL-параметри.

Також існує ризик атак типу XSS (Cross-Site Scripting), що дають можливість виконання шкідливих скриптів у браузері користувача. Особливо небезпечним є вразливості типу RCE (Remote Code Execution), які дозволяють зловмиснику віддалено запускати довільний код на сервері. Крім того, можуть виникати уразливості, що ведуть до підвищення привілеїв, завдяки яким зловмисник може отримати адміністративний доступ до системи.

Такі уразливості можуть стати серйозною загрозою для конфіденційності, цілісності та доступності даних веб-додатку. Навіть тимчасова експлуатація однієї з них може призвести до витоку персональної інформації користувачів або до повного контролю над сайтом.

Особливу небезпеку становить поєднання кількох вразливостей, що дозволяє зловмиснику здійснити комплексну атаку. Значна частина загроз виникає через не своєчасне оновлення системи або використання застарілих компонентів. Важливо зрозуміти, що Joomla, як і будь-яка інша CMS є привабливою цілью для автоматизованих ботів, які постійно сканують сайти на предмет відомих слабких місць.

Не менш критичною є наявність сторонніх розширень, що часто стають джерелом нових векторних атак. Деякі розширення містять погано написаний код або залишають "бекдори", через які може відбутися несанкціонований доступ. Крім технічних вразливостей, велику роль відіграє людський фактор, зокрема використання слабких паролів або нехтування двофакторною автентифікацією.

Адміністратори повинні бути обізнаними щодо найпоширеніших методів атак і регулярно проходити навчання з кібербезпеки. Також варто налаштувати моніторинг логів, щоб виявляти спроби підбору паролів або сканування системи. Важливим елементом захисту є контроль прав доступу, користувачі повинні мати лише ті повноваження, які дійсно необхідні.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Потрібно заборонити виконання PHP-скриптів у папках де зберігаються завантаження, наприклад у каталозі /images. Бажано реалізувати механізми обмеження кількості спроб входу до адміністративної панелі, щоб запобігти брутфорс-атакам. Крім цього, варто використовувати веб-додатки фаєрволи (WAF), які блокують підозрілу активність ще до обробки запиту Joomla.

Система керування вмістом Joomla залишається популярною серед розробників завдяки своїй гнучкості та широкими можливостями розширення. Однак саме ці переваги нерідко стають джерелом серйозних вразливостей. Однією з основних причин атак на Joomla сайти є використання сторонніх розширень.

Велика кількість плагінів і модулів, створених різними розробниками часто містять небезпечні вразливості, особливо якщо вони більше не підтримуються або створені без належного фокусу на безпеку. Особливо небезпечно встановлювати плагіни з неперевірених джерел, адже вони можуть містити шкідливі скрипти або приховані бекорди, які дають зловмиснику повний контроль над сайтом.

Ще однією критичною проблемою є неправильна конфігурація сайту. Адміністратори нерідко залишають стандартний логін “**admin**” який автоматизовані боти перевіряють у першу чергу. Крім того, відсутність обмеження доступу до директорії /**administrator**/ дозволяє будь-кому намагатися увійти в панель керування. Належним чином незахищений файл **configuration.php** також може стати джерелом витоку конфіденційної інформації, якщо сервер дозволяє доступ до нього через браузер. Публічний доступ до резервних копій або логів відкриває ще один шлях до потенційного зламу.

Серверне середовище, на якому розміщено Joomla-сайт теж має критичне значення. Відсутність HTTPS, увімкненні небезпечні функції PHP, такі як **allow_url_fopen**, або дозвіл на виконання PHP коду в медіа папках – усе це створює серйозні ризики. Крім того якщо серверне програмне забезпечення (PHP, MySQL, Apache або Nginx) не оновлюється, сайт може залишатися відкритим для атак.

Не менш небезпечнішим є людський фактор. Соціальна інженерія та фішингові атаки часто спрямовані саме на адміністраторів сайтів. Зловмисники

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		30

створюють підроблені форми входу, які виглядають як справжня адмінка Joomla, щоб вкрасти логін та пароль. Якщо власник сайту не використовує двофакторну автентифікацію, злоумисник отримає повний доступ до керування сайтом. Більше того, використання одного і того самого пароля на різних платформах значно підвищує ризик компрометації.

Сучасні кібератаки як правило здійснюються не вручну, а за допомогою автоматизованих скриптів і ботів. Вони сканують тисячі сайтів у пошуках відомих вразливостей, перевіряють встановленні плагіни та їхні версії і навіть намагаються зламати пароль до адмін-панелі методом брутфорсу. Якщо на сайті не встановлено обмеження кількості спроб входу або захист у вигляді CAPTCHA, ботам нічого не остається як спробувати знову і знову.

Усі ці загрози дозволяють що безпека Joomla сайту залежить не лише від самої CMS, а на сам перед від адміністратора. Комплексний підхід до захисту, регулярне оновлення розширень, захист серверного середовища, навчання персоналу та впровадження сучасних засобів автентифікації є ключем до стабільної та безпечної роботи ресурсу. Лише поєднання технічних, організаційних і поведінкових заходів може ефективно протистояти сучасним кіберзагрозам.

2.2 Архітектура підсистеми захисту: рівні, компоненти, взаємодія

Система керування контентом Joomla – це одна з найпопулярніших платформ для створення веб-сайтів. Однак її популярність робить її привабливою ціллю для злоумисників. Захист CMS Joomla реалізується як сукупність взаємопов'язаних рівнів, що взаємодіють між собою через окремі компоненти. Така багаторівнева архітектура дозволяє забезпечити комплексну безпеку системи та гнучкість у налаштуванні.

Захист на рівні сервера

Використання Web Application Firewall (WAF), такого як mod_security, дозволяє ефективно блокувати небезпечні HTTP-запити ще до того, як вони досягнуть CMS. WAF аналізує запити на наявність шкідливих шаблонів, наприклад SQL-ін'єкцій чи XSS, і зупиняє їх на рівні сервера. Часто використовуються бази

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		31

правил OWASP для виявлення найпоширеніших загроз. Додатково можна налаштовувати власні фільтри відповідно до структури сайту. Журнали, які веде WAF, дозволяють оперативно реагувати на інциденти.

Серверні правила також використовуються для обмеження доступу до конфігураційних файлів і директорій. Наприклад, файл `configuration.php` можна захистити за допомогою правил у `.htaccess`, які повністю забороняють доступ до нього ззовні. Права доступу до файлів обмежуються до мінімально необхідних, часто на рівні 600 або 640. Відключення індексації вмісту директорій (Options - Indexes) запобігає їхньому перегляду сторонніми користувачами. Для адміністративної частини сайту доцільно встановити додаткову авторизацію через `.htpasswd` або обмеження за IP.

Файл `.htaccess` відіграє ключову роль у локальному захисті Joomla. Він допомагає запобігти поширеним атакам, наприклад SQL-ін'єкціям та XSS, блокуючи запити з підозрілими параметрами. Також у ньому можна заборонити виконання PHP-скриптів у директоріях, де вони не повинні знаходитися (наприклад, у медіа-каталогах). Через цей файл додаються захисні HTTP-заголовки (X-XSS-Protection, Content-Security-Policy), які ускладнюють ін'єкції зловмисного коду.

Крім цього, `.htaccess` дозволяє задавати правила перепису URL (Rewrite Rules), які використовуються для блокування небезпечних запитів. Наприклад, можна обмежити доступ до URL із певними ключовими словами або параметрами, які часто використовуються під час атак. Можна перенаправляти користувачів зі застарілих шляхів, фільтрувати запити ботів і створювати фільтри для захисту API.

Безпека паролів

Забезпечення складних паролів та двофакторної автентифікації є критично важливим. Joomla дозволяє використовувати спеціальні плагіни для впровадження 2FA (наприклад, через Google Authenticator). Це суттєво ускладнює несанкціонований доступ, навіть якщо пароль було викрадено. Також важливо впровадити політику зміни паролів, заборонити повторне використання попередніх комбінацій та мінімізувати кількість користувачів із правами адміністратора.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Обмеження доступу до адміністративної панелі за IP-адресами — додатковий рівень захисту. Можна дозволити доступ лише з певної IP-адреси або блоку IP, а також використовувати VPN. Ведення журналу спроб входу дозволяє виявити підозрілу активність і швидко зреагувати на потенційну загрозу.

Оновлення CMS та розширень

Регулярне оновлення Joomla і встановлених розширень — це ключ до уникнення відомих вразливостей. Розробники часто випускають оновлення з виправленнями безпеки, тому затримка з оновленням може призвести до серйозних проблем. Joomla підтримує систему сповіщення про нові версії, а також є розширення, які дозволяють автоматизувати процес оновлення. Перед оновленням важливо робити резервну копію, щоб уникнути втрати даних.

Встановлення тільки перевірених розширень — ще одна важлива умова. Не варто завантажувати компоненти, модулі чи плагіни з невідомих джерел, адже вони можуть містити шкідливий код. Краще користуватися офіційним каталогом JED (Joomla Extensions Directory), де публікуються лише перевірені розробки. Розширення, які не оновлювалися тривалий час або мають мало відгуків, варто уникати.

Контроль доступу

Joomla має потужну систему ACL (Access Control List), яка дозволяє чітко розмежувати повноваження користувачів. Наприклад, можна налаштувати права доступу окремо для створення, редагування, публікації або видалення матеріалів. Це дозволяє реалізувати принцип найменших привілеїв — кожен користувач має лише ті права, які необхідні для виконання його функцій.

Ролі користувачів мають бути чітко визначені. Адміністратор, редактор, автор — кожен має різні повноваження, і не повинен мати доступу до функцій, які йому не потрібні. Також важливо регулярно перевіряти список користувачів та їхні права, видаляти або блокувати неактивні акаунти та стежити за журналом змін прав доступу.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Захист від атак

Joomla оснащена механізмами захисту від поширених типів атак, таких як SQL-ін'єкції, XSS, Brute Force. У роботі з базами даних використовуються безпечні методи, зокрема підготовлені запити та спеціальні функції кодування. Форми захищаються токенами CSRF, що запобігає виконанню шкідливих дій із підроблених запитів. Доступ до форми входу можна обмежити кількістю спроб або часом затримки між ними.

Спеціальні розширення для моніторингу безпеки, як-от Admin Tools, RSFirewall!, дозволяють виявляти зміни у файлах, підозрілі дії в адміністративній зоні, спроби несанкціонованого входу. Вони можуть автоматично блокувати підозрілі IP-адреси, перевіряти систему на наявність вразливостей і повідомляти адміністратора про будь-які загрози. Такі інструменти є незамінними для забезпечення цілісності та стабільності Joomla-сайту.

Компоненти. У Joomla компоненти є головними структурними елементами, які забезпечують основну функціональність сайту. Вони виступають самостійними програмними одиницями, що реалізують складну логіку обробки даних. Компоненти дозволяють додавати функціональність, яка виходить за межі базових можливостей CMS. Вони керують взаємодією з користувачами, обробляють дані з бази та відповідають за відображення складного контенту. З їхньою допомогою можна будувати масштабні функціональні модулі, як-от блоги, каталоги або інтернет-магазини.

Компоненти зазвичай мають дві частини: адміністративну (бекенд) та сайтову (фронтенд). Сайтовий інтерфейс відповідає за відображення вмісту для відвідувачів, тоді як адміністративна частина дозволяє керувати налаштуваннями компонента через панель керування Joomla. Вони діють у тісному зв'язку між собою, обробляючи запити, зберігаючи налаштування та формуючи відповіді. Адміністраторська частина дає змогу створювати, редагувати або видаляти дані, а сайдова — виводить ці дані в зручному вигляді. Це дозволяє адаптувати сайт як до потреб адміністраторів, так і до очікувань кінцевих користувачів. Обидві частини

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

компонента працюють у злагодженій взаємодії, що забезпечує цілісність і стабільність системи.

Joomla постачається з базовим набором компонентів, які охоплюють основні потреби. Наприклад, `com_content` дозволяє керувати статтями та категоріями, `com_contact` відповідає за створення контактних форм, а `com_users` дає змогу керувати обліковими записами користувачів. Компонент `com_menus` слугує для створення навігації на сайті, а `com_banners` дозволяє запускати банерну рекламу. Крім того, `com_tags` дозволяє гнучко фільтрувати контент за тегами. Усі ці компоненти працюють разом, формуючи повноцінний функціональний каркас сайту.

Joomla є надзвичайно розширюваною системою, що дозволяє створювати власні компоненти. Кожен компонент можна адаптувати до специфіки конкретного проєкту — від шкільного сайту до великого корпоративного порталу. Компоненти можуть об'єднуватися з модулями, плагінами, шаблонами та мовними файлами для досягнення складної логіки. Вони підтримують налаштування прав доступу, що дозволяє обмежити або розширити функціональність для різних користувачів. Система дозволяє створювати мультимовні компоненти, що зручно для інтернаціональних сайтів. Також можливе масштабування компонентів без втрати швидкодії.

Багато компонентів можуть викликатись напряду через HTTP-запити, що робить їх універсальними для створення кастомних сторінок. Наприклад, URL-адреса типу `index.php?option=com_content` дозволяє одразу звернутися до компонента статей. За допомогою параметрів у запиті можна вказувати конкретну дію — перегляд, редагування або фільтрацію. Це відкриває можливості створення інтерактивних веб-сервісів без потреби у створенні пунктів меню. Компоненти можуть приймати дані через GET чи POST, що дозволяє працювати з формами або з API-запитами. Вони також можуть повертати відповіді у форматі JSON або XML, що зручно при розробці AJAX-функціоналу.

Компоненти в Joomla дотримуються архітектурного шаблону Model-View-Controller (MVC). Модель (Model) керує доступом до даних і логікою їх обробки.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Вигляд (View) відповідає за відображення інформації користувачеві у зрозумілому форматі. Контролер (Controller) обробляє запити, координує дії моделі та обирає потрібне представлення. Така структура дозволяє відокремити логіку від дизайну, що спрощує обслуговування та оновлення коду. MVC підвищує масштабованість і модульність, а також полегшує командну розробку, оскільки розробники можуть працювати з різними частинами незалежно. Це також сприяє тестуванню й автоматизації процесів розробки.

Для спрощення створення компонентів існують спеціальні інструменти, наприклад, Component Architect. Вони дозволяють розробникам автоматично згенерувати структуру MVC і швидко налаштувати поля, форми та перегляди. Такі засоби знижують поріг входу у розробку, оскільки частину рутинної роботи вони виконують автоматично. Інструменти часто мають візуальні редактори, які дають змогу створювати компоненти без глибоких знань PHP. Деякі підтримують експорт ZIP-файлів для подальшого встановлення через інтерфейс Joomla. Крім того, ці утиліти можуть бути інтегровані з системами керування версіями (наприклад, Git), що зручно для командної роботи. Вони також мають шаблони, що запобігають типовим помилкам і сприяють стандартизації розробки.

Взаємодія між ними

Joomla використовує багаторівневу систему безпеки, де кожен рівень виконує свою захисну функцію і водночас взаємодіє з іншими компонентами, формуючи комплексну архітектуру. Підсистема захисту базується на принципі оборони в глибину (defense-in-depth), що дозволяє виявляти, блокувати й ізолювати загрози на різних етапах обробки запиту.

Зовнішній рівень системи складається з налаштувань веб-сервера, файлу .htaccess і використання засобів файрволу вебзастосунків, таких як mod_security. На цьому етапі відбувається первинна фільтрація запитів, що надходять до сайту, і блокується доступ до небезпечних файлів або директорій. Запити, які не відповідають заданим правилам безпеки, не досягають ядра Joomla.

Серверна частина CMS, реалізована на PHP, є наступним рівнем захисту. Саме тут знаходиться ядро Joomla, яке обробляє всі запити, перевіряє маркери

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		36

безпеки (зокрема CSRF-токени), фільтрує введення користувача, виконує маршрутизацію та перевірку прав доступу. Ядро тісно взаємодіє з системою ACL, яка регулює доступ до функцій, модулів і контенту на основі ролі користувача. Усі дії, що стосуються обробки даних, контролюються засобами аутентифікації, авторизації та журналювання.

Важливу роль у захисті відіграють розширення Joomla — плагіни, модулі та компоненти. Вони працюють через стандартизоване API та викликаються в межах визначених подій ядра. Це дозволяє забезпечити контрольований і безпечний доступ до внутрішніх ресурсів системи. Багато розширень мають вбудовані власні механізми захисту, зокрема CAPTCHA, блокування ботів, двофакторну автентифікацію або захист від brute force-атак.

Наступний рівень — це взаємодія з базою даних і файловою системою. Joomla використовує підготовлені запити для запобігання SQL-ін'єкціям, а всі критичні дані, зокрема паролі, зберігаються в зашифрованому вигляді за допомогою надійних хеш-функцій. Завантаження файлів контролюється на предмет відповідності типу, розміру та вмісту, щоб не допустити виконання шкідливого коду.

Останнім, але не менш важливим елементом захисту є система логування та моніторингу. Joomla веде облік подій входу в систему, змін у файлах, дій користувачів та спроб доступу. Ці журнали можуть бути розширені додатковими засобами моніторингу, які автоматично реагують на підозрілу активність, блокують IP-адреси, надсилають сповіщення адміністратору або обмежують функціональність сайту до з'ясування обставин.

Уся ця система працює узгоджено. Запит користувача спочатку проходить через фільтрацію на рівні веб-сервера та .htaccess, потім обробляється ядром Joomla із застосуванням механізмів перевірки сесій, токенів і прав доступу, після чого — за потреби — передається до плагінів чи компонентів. Якщо запит стосується збереження або завантаження даних, відбувається безпечна взаємодія з базою даних або файловим сховищем. Усі дії супроводжуються журналюванням, що дозволяє виявити й проаналізувати загрози.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

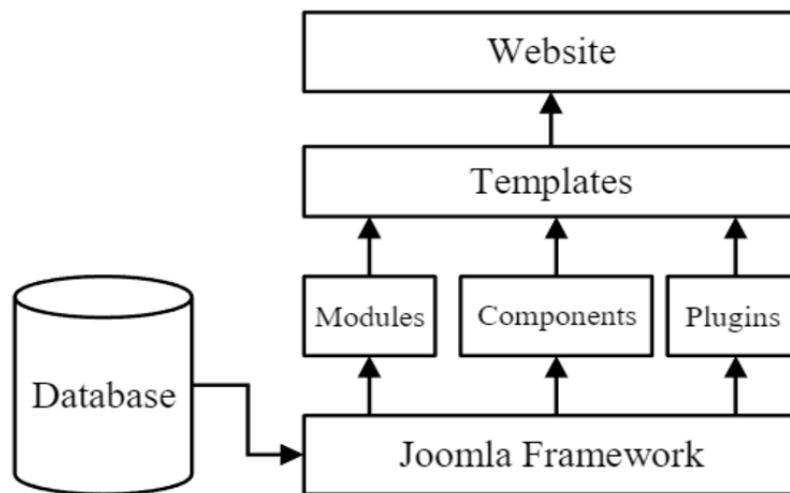


Рис. 2.2 Архітектура Joomla

На рисунку 2.2. показано архітектурні рівні компонентів та їх взаємодію в загальній узагальненій системі. Основу архітектури становить **Joomla Framework**, який забезпечує ядро функціонування всієї системи. Цей рівень відповідає за маршрутизацію, безпеку, взаємодію з базою даних, перевірку прав доступу та обробку сесій користувачів. Усі інші компоненти працюють через API та події, що надаються фреймворком.

Над фреймворком розміщуються **модулі (Modules)**, **компоненти (Components)** та **плагіни (Plugins)**, які є основними засобами реалізації функціональності сайту. Компоненти — це великі функціональні блоки, які відповідають за основну логіку, наприклад, виведення статей або керування користувачами. Модулі використовуються для виведення додаткової інформації у визначених зонах шаблону, наприклад, меню або новин. Плагіни — це розширення, які реагують на певні події системи, наприклад, під час авторизації або збереження матеріалу.

Наступним рівнем є **шаблони (Templates)**, які визначають зовнішній вигляд сайту та форматують дані, що надходять з розширень. Шаблони забезпечують інтеграцію візуального стилю з функціональністю, наданою модулями, компонентами й плагінами. Це дозволяє відокремити логіку від подання, що є важливим принципом архітектури MVC, на якому базується Joomla.

На верхньому рівні перебуває **вебсайт (Website)** — готовий до взаємодії з користувачем результат, який формується після проходження запиту через усі

попередні рівні. Кожен користувацький запит проходить шлях від шаблону до компонента, який у свою чергу взаємодіє з фреймворком і, за потреби, з базою даних. Отримані дані знову проходять крізь шаблон і виводяться кінцевому користувачеві у вигляді HTML-сторінки.

База даних (Database) є джерелом зберігання всіх динамічних даних Joomla: контенту, користувачів, налаштувань, розширень та іншої інформації. Всі запити до бази даних проходять через фреймворк, де вони обробляються з урахуванням безпеки, зокрема через підготовлені SQL-запити.

Уся система працює згідно з принципами взаємної ізоляції, де кожен рівень виконує свою роль і залежить від нижчого шару. Це дозволяє легко змінювати шаблони без впливу на логіку компонентів або оновлювати плагіни без зміни шаблону. Взаємодія між рівнями є стандартизованою, що підвищує безпеку та масштабованість системи.

У межах цієї архітектури легко реалізувати нові функції за допомогою створення власних плагінів або компонентів. Захист реалізується на кожному рівні: фреймворк фільтрує запити, компоненти перевіряють права доступу, а шаблони можуть запобігати XSS-атакам шляхом коректної обробки виводу. Така побудова дозволяє створювати масштабовані, стабільні та безпечні вебресурси на базі Joomla.

2.3 Вибір програмного забезпечення, хостингово середовища та захисних засобів

Ефективність підсистеми захисту web-додатку значною мірою залежить від правильного вибору програмного забезпечення, типу хостингу та засобів безпеки, які забезпечують захист як на рівні сервера, так і безпосередньо в межах CMS Joomla. У цьому підрозділі розглядається обґрунтований вибір компонентів, необхідних для розгортання захищеного середовища.

Для даного проєкту було обрано Joomla 4.4.13 так як ця версія забезпечує підвищену безпеку, розширену підтримку розширень та сумісність з сучасними

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				39
Змн.	Арк.	№ докум.	Підпис	Дата		

технологіями. Проєкт створюється з використанням вбудованого конструктора сайтів Joomla або сторонніх візуальних редакторів, таких як SP Page Builder чи YOOtheme Pro, які дають змогу швидко реалізовувати структуру сайту без глибоких технічних знань.

Цей підхід дозволяє зосередитися на конфігурації захисту через встановлення відповідних плагінів без прямої модифікації коду. Серед ключових плагінів, які забезпечують захист:

RSFirewall! — це потужний плагін безпеки для Joomla, що забезпечує захист адміністративної панелі, веб-фаєрвол (WAF), сканування системи та контроль доступу.



RSFirewall! — це професійне розширення для CMS Joomla, створене компанією **RSJoomla!**, яке забезпечує всебічний захист сайту від мережевих атак, несанкціонованого доступу, шкідливого коду та помилок конфігурації. Плагін дозволяє постійно контролювати стан безпеки системи, здійснювати моніторинг активності користувачів, обмежувати доступ і захищати адміністративну частину сайту. Завдяки сумісності з Joomla 3, 4 і 5, RSFirewall! залишається одним із найнадійніших і найбільш підтримуваних плагінів безпеки.

Основні можливості RSFirewall!:

- Плагін виконує повне сканування системи на наявність змін у файлах ядра Joomla та сторонніх розширеннях.
- Інструмент проактивного захисту (Active Scanner) виявляє шкідливий код, підозрілі файли, скрипти та потенційні бекдори.
- RSFirewall! дозволяє обмежити вхід до адміністративної панелі за IP-адресою або за спеціальним секретним URL-ключем.
- Він веде журнал усіх підозрілих дій, включаючи спроби входу, зміни файлів, доступ до системних ресурсів та SQL-запити.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

- Існує можливість встановити правила фаєрволу для блокування відомих типів атак — SQL-ін'єкцій, XSS, LFI, RFI та Brute Force.
- Плагін надає захист від ботів, дозволяє виявляти та блокувати автоматизовані скрипти, які сканують сайт.
- Доступна функція зміни стандартного URL входу в адмінку Joomla для запобігання автоматичним атакам.
- Можна встановити обмеження на кількість неправильних спроб входу та заблокувати IP на певний час.
- RSFirewall! підтримує автоматичне оновлення своєї сигнатурної бази для виявлення нових загроз.
- Доступна опція резервного копіювання конфігураційних файлів системи безпеки для швидкого відновлення налаштувань.

Переваги RSFirewall!:

- Плагін виконує повне сканування системи на наявність змін у файлах ядра Joomla та сторонніх розширеннях.
- Інструмент проактивного захисту (Active Scanner) виявляє шкідливий код, підозрілі файли, скрипти та потенційні бекдори.
- RSFirewall! дозволяє обмежити вхід до адміністративної панелі за IP-адресою або за спеціальним секретним URL-ключем.
- Він веде журнал усіх підозрілих дій, включаючи спроби входу, зміни файлів, доступ до системних ресурсів та SQL-запити.
- Існує можливість встановити правила фаєрволу для блокування відомих типів атак — SQL-ін'єкцій, XSS, LFI, RFI та Brute Force.
- Плагін надає захист від ботів, дозволяє виявляти та блокувати автоматизовані скрипти, які сканують сайт.
- Доступна функція зміни стандартного URL входу в адмінку Joomla для запобігання автоматичним атакам.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		41

- Можна встановити обмеження на кількість неправильних спроб входу та заблокувати IP на певний час.
- RSFirewall! підтримує автоматичне оновлення своєї сигнатурної бази для виявлення нових загроз.
- Доступна опція резервного копіювання конфігураційних файлів системи безпеки для швидкого відновлення налаштувань.

Недоліки плагіна RSFirewall!

- Плагін є **платним** — безкоштовної версії не існує, тому не всі користувачі можуть одразу спробувати його функціонал.
- Може конфліктувати з іншими плагінами безпеки або розширеннями, що активно змінюють файли ядра.
- Деякі функції вимагають ручного налаштування, що може бути складно для новачків.
- При агресивному конфігуруванні можливе блокування легітимних користувачів або адміністраторів, особливо при використанні IP-фільтрації.

RSFirewall! є потужним інструментом превентивного захисту, який дозволяє виявити шкідливу активність ще до того, як вона спричинить шкоду. У поєднанні з регулярними оновленнями Joomla, складними паролями, резервними копіями (наприклад, за допомогою Akeeba Backup) та налаштованим сервером, плагін створює **багаторівневу систему безпеки**, здатну ефективно протистояти більшості сучасних загроз.

Висновок до другого розділу

Joomla-сайти, попри свою гнучкість, постійно перебувають під прицілом кібератак, включаючи SQL-ін'єкції, XSS, RCE та атаки на сесії. Ці вразливості часто виникають через застарілі компоненти, неправильні налаштування або людський фактор. Автоматизовані боти невпинно сканують сайти, а фішингові атаки

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		42

спрямовані на облікові дані адміністраторів, підкреслюючи нагальну потребу в комплексній та багаторівневій системі захисту.

Ефективний захист Joomla базується на стратегії "оборони в глибину", що починається на рівні сервера. Використання Web Application Firewall (WAF), такого як mod_security, та ретельні налаштування файлу .htaccess є першим ешелonom. Вони фільтрують шкідливі HTTP-запити до досягнення CMS, обмежують доступ до конфіденційних файлів, регулюють права доступу та запобігають індексації вмісту, мінімізуючи вектори для атак.

Захист на рівні сервера є першим бар'єром проти атак. Реалізація Web Application Firewall (WAF), наприклад, за допомогою mod_security, дає змогу блокувати шкідливі запити ще до їх обробки CMS. Важливо також правильно налаштувати .htaccess — цей файл використовується для обмеження доступу до службових файлів, встановлення безпечних HTTP-заголовків і контролю параметрів запитів. Такі заходи знижують ризик експлуатації вразливостей, пов'язаних із невірною обробкою даних ще до взаємодії з ядром Joomla.

На рівні самої CMS захист реалізується через її ядро, систему керування доступом (ACL), фільтрацію вхідних даних і інтеграцію захисних механізмів, таких як CAPTCHA та CSRF-токени. Підготовлені SQL-запити запобігають ін'єкціям, а шифрування чутливої інформації — витоку даних. Joomla також дозволяє вести детальний журнал подій, що суттєво спрощує моніторинг активності, виявлення аномалій і аудит безпеки. Розширення, які дотримуються стандартів API, дозволяють безпечно розширювати функціональність, не порушуючи захисні механізми системи.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПІДСИСТЕМИ ЗАХИСТУ

3.1 Налаштування середовища та виконання Joomla

Першим кроком з налаштування середовища потрібно з встановлення локального серверу ХАМРР. **ХАМРР** — це безкоштовний, зручний у використанні пакет програмного забезпечення, який дозволяє створити локальне серверне середовище для розробки веб-сайтів і веб-додатків. Назва ХАМРР розшифровується як:

- **Х** — кросплатформенний (Windows, Linux, macOS),
- **А** — Apache (веб-сервер),
- **М** — MySQL або MariaDB (система управління базами даних),
- **Р** — PHP (мова програмування),
- **Р** — Perl (ще одна мова програмування, менш часто використовується).

Завантажити програмне забезпечення можна завантажити як з офіційного сайту, та і з будь-яких інших іших ресурсів.

Краще все таки завантажувати з офіційного сайту а ніж з неофіційних сайтів. Оскільки в майбутньому можуть виникати проблеми, пов'язані з зараженням вірусами такими як скрін-локер і додатково встановленим програмами.

Заходимо на офіційний сайт збірки www.apachefriends.org. Після чого потрібно вибрати операційну систему, виходячи з того в якій працюємо та завантажити (рис. 3.1)

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		44



Рис. 3.1 Вибір версії XAMPP

Далі слід вибрати основні компоненти, які будуть використовуватись у роботі, зокрема: Apache, MySQL, FileZilla FTP Server, PHP та phpMyAdmin. Решту компонентів можна встановити додатково — це спростить майбутню роботу та зекономить час на налаштування необхідного середовища. Після цього натискаємо “Next”, чекаємо завершення інсталяції локального сервера XAMPP, запускаємо його з правами адміністратора та надаємо мережевий доступ через брандмауер.

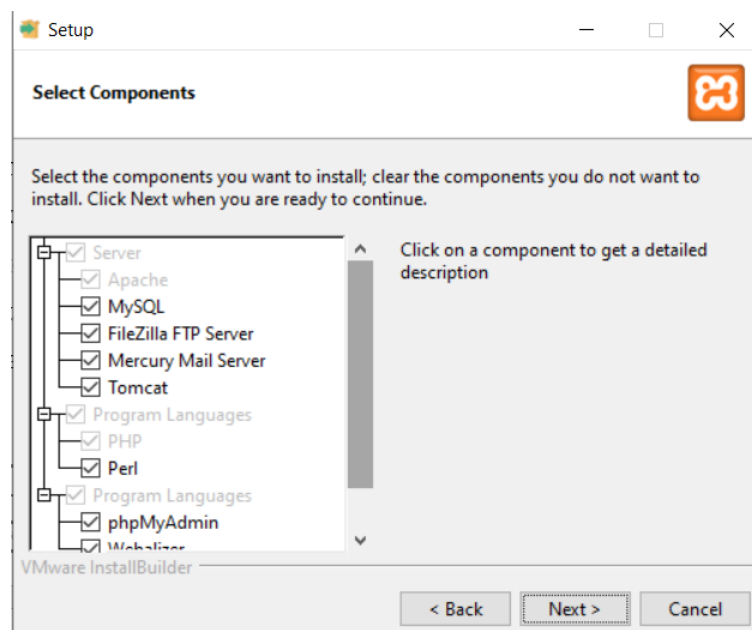


Рис. 3.2 Компоненти XAMPP

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

Після встановлення XAMPP рекомендовано перевірити працездатність компонентів та модулів. Після запуску сервера відкриваємо панель керування XAMPP і по черзі запускаємо служби MySQL та Apache. Для перевірки роботи служб необхідно запустити програму з правами адміністратора.

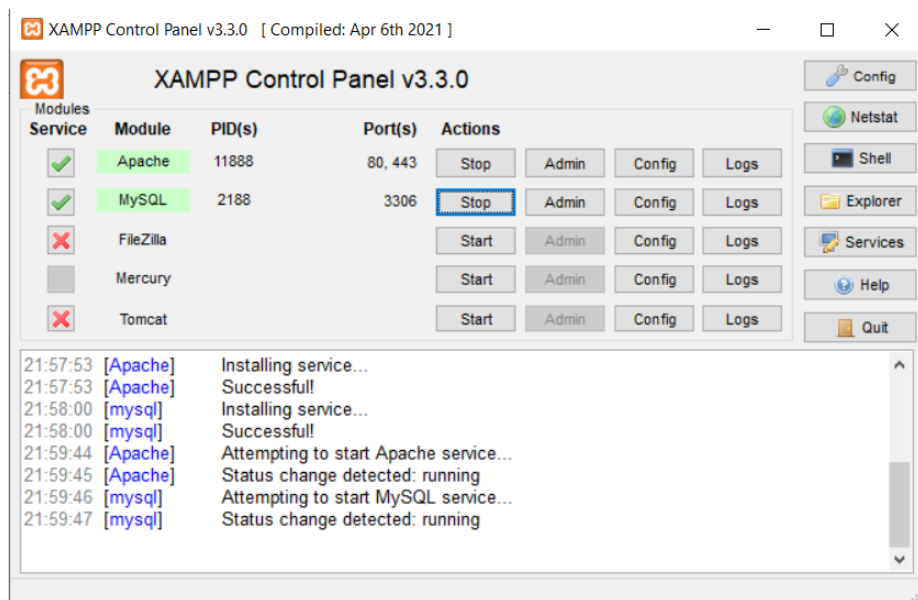


Рис. 3.3 Панель керування

Натиснувши кнопку “Admin” навпроти MySQL, переходимо до інтерфейсу керування базами даних — PhpMyAdmin. На рисунку 3.4 представлено головну сторінку PhpMyAdmin. Спочатку створимо нову базу даних, яка використовуватиметься для майбутнього сайту.

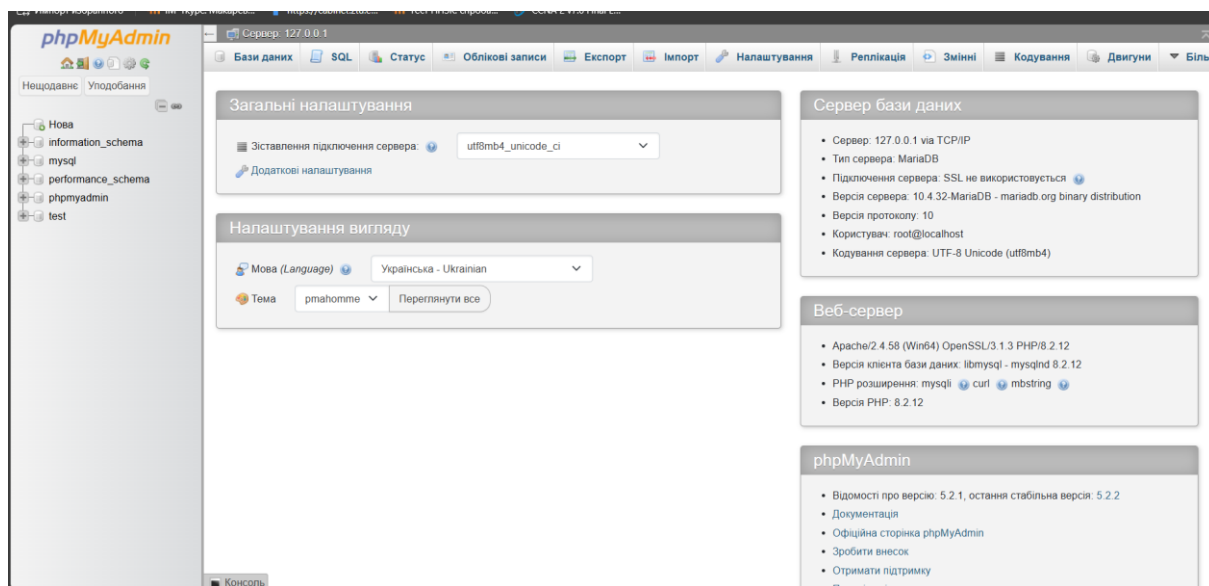


Рис. 3.4 Головне вікно PhpMyAdmin

Переходимо до розділу створення бази даних та натискаємо “Нова”, надаємо ім’я для бази даних “ test_base”. Після цього ми вибираємо алгоритм сортування utf8mb4_unicode_general_ci. Цей алгоритм є спрощеним набором правил сортування, основною метою якого є максимальне підвищення швидкості за рахунок використання різноманітних скорочень і спрощень. Після цього натискаємо кнопку «Створити».

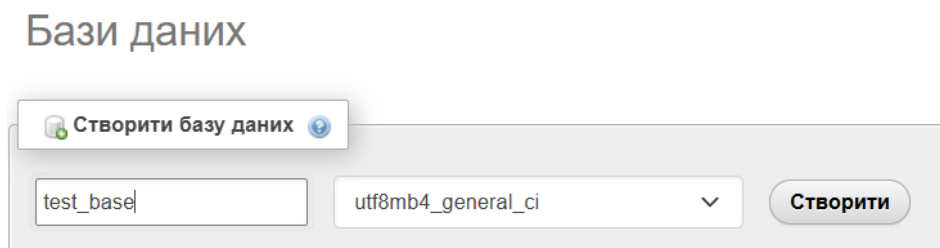


Рис. 3.5 Вікно створення бази даних

Встановлення Joomla

Joomla - це відкрита універсальна система керування контентом, призначена для публікації інформації в мережі Інтернет. Вона підходить для розробки як невеликих, так і масштабних корпоративних сайтів, веб-порталів, інтернет-магазинів, спільнот користувачів та особистих веб-сторінок.

Щоб встановити CMS потрібно спочатку перейти на офіційний сайт, вибрати потрібну версію та завантажити. Після завантаження потрібно створити папку в директорії ХАММР що знаходиться в папці “htdocs” під назвою “diploma” та розпакувати архів в цю папку. Далі переходимо по адресі **/localhost/joomla/** у будь-якому браузері який для вас кращий, де “joomla” це назва вашого проєкту, що створений у директорії ХАМРР. Після переходу по адресі в нас з’являється вікно встановлення Joomla. На екрані з’явиться вікно для вибору мови Joomla, обираємо ту яка буде зручніша для роботи в інтерфейсі. Потім у вікні “назва сайту” вводимо назву нашого сайту, у нашому випадку назва “Joomla 5”, натискаємо “налаштування даних для входу”.

Далі потрібно ввести основні данні для встановлення Joomla. Встановлюємо логін для облікового запису, логін матиме назву “admin”. Наступним кроком буде потрібно задати ім’я користувача “Serhii”, пароль та e-mail які надалі використовуватимуться для входу в систему. Після заповнення форми, наведеної

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		47

на рис 3.6, наступним кроком є налаштування параметрів підключення до бази даних.

Рис . 3.6 Заповнення форма.

На етапі налаштування бази даних необхідно вказати ключові параметри для встановлення з'єднання. Тип бази даних обирається як **MySQLi**, що є рекомендованим варіантом для Joomla. У полі **ім'я хоста** вказується значення **localhost**, оскільки база даних працює на локальному сервері. Далі вводиться **ім'я користувача** — за замовчуванням у XAMPP це **root**. Поле **пароль** зазвичай залишають порожнім, якщо пароль не був встановлений вручну. Останнім кроком потрібно ввести **назву бази даних**, яку було створено раніше, наприклад, **joomla_db**.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		48


Встановлення Joomla! Joomla! 4.4.13


Конфігурація бази даних

Оберіть тип бази даних. *

MySQLi

Це, як правило, "localhost", або назва, надана вашим хостингом. *

localhost

Ім'я користувача повинно бути створено вами або надано хостом. *

root

Для безпеки сайту використання паролю для облікового запису MySQL є обов'язковим

Деякі хостинг-провайдери дозволяють тільки певне ім'я бази даних для кожного сайту. Використовуйте префікс таблиць у цьому випадку для окремих сайтів на Joomla! *

joomla_db

Введіть префікс таблиць або скористайтесь автоматичним заповненням. *

dsmj0_

If you are using an existing database with tables with the same prefix, Joomla will rename those existing tables by adding the prefix "bak_".

Шифрування з'єднання *

За налаштуванням (підконтрольне серверу)

Встановити Joomla >

Рис . 3.7 Ключові параметри

Після правильного проходження всіх кроків повинно з'явитися вікно Рис 3.8, яке свідчить про успішну інсталяцію Joomla.


Встановлення Joomla! Joomla! 4.4.13


Вітаємо!

Вітаємо! Joomla встановлена!

Встановити додаткові мови >

Ці налаштування рекомендовані для PHP, з тим, щоб забезпечити повну сумісність з Joomla. Однак, Joomla! буде, як і раніше, працювати, якщо налаштування не зовсім відповідають рекомендованим.

Директиви	Рекомендовано	Поточне
Показувати помилки	Викл.	Вкл.
Буферизація виводу (Output Buffering)	Викл.	Вкл.
Вбудована підтримка ZIP	Вкл.	Викл.
Доступно intl	Вкл.	Викл.

Open Site

Open Administrator

Рис . 3.8 Успішне встановлення Joomla

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступним етапом буде встановлення плагінів в Joomla. Щоб встановити плагін у Joomla, спочатку потрібно увійти в адміністративну панель сайту.

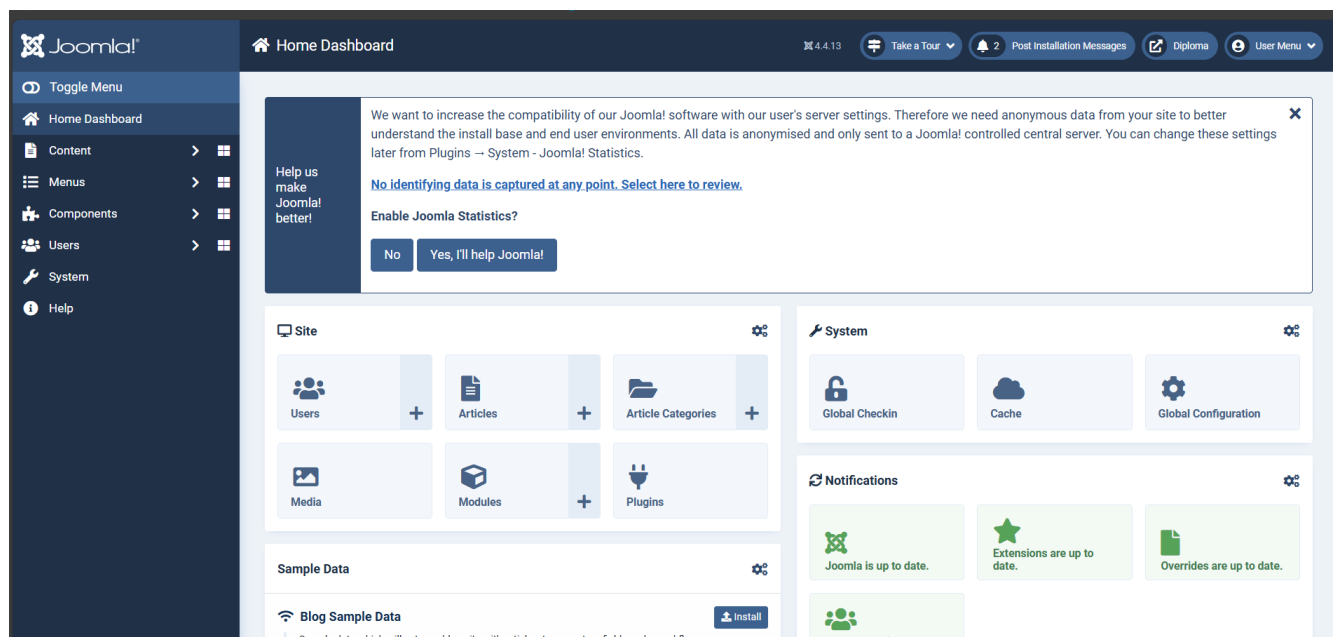


Рис. 3.9 Адміністративна панель Joomla

Далі у верхньому меню обираємо пункт «Система» → «Керування» → «розширення». На сторінці, що відкриється, натискаємо кнопку для завантаження пакета з файлу, вибираємо потрібний ZIP-архів плагіна з комп'ютера та натискаємо «Завантажити та встановити». Після завершення встановлення переходимо в «Розширення» → «Плагіни», знаходимо встановлений плагін у списку й активуємо його. Першим із рекомендованих засобів безпеки, який буде інтегровано в рамках цього проєкту, є RSFirewall!, оскільки він є потужним інструментом для захисту адміністративної панелі Joomla, виявлення шкідливого коду та моніторингу системної активності.

Плагін забезпечує глибоке сканування файлів ядра та розширень на наявність змін, що дає змогу швидко виявляти загрози. RSFirewall! також дозволяє обмежити доступ до адмін-панелі за IP-адресою або через секретний ключ у URL, що значно знижує ризик несанкціонованого доступу.

Окрім цього, він включає інструменти проактивного захисту, які виявляють підозрілі дії, шкідливі скрипти та потенційні бекдори. Плагін підтримує захист від найбільш поширених атак, таких як SQL-ін'єкції, XSS, RFI, LFI та Brute Force, а

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				50
Змн.	Арк.	№ докум.	Підпис	Дата		

також веде журнали активності користувачів і дозволяє налаштовувати обмеження на кількість невдалих спроб входу.

Функціональність RSFirewall! доповнюється автоматичним оновленням сигнатур загроз, можливістю змінити URL входу в адмінку та захистити конфігураційні файли, що забезпечує комплексний багаторівневий захист Joomla-сайту. На рис. 3.10 показаний конструктор RSFirewall.

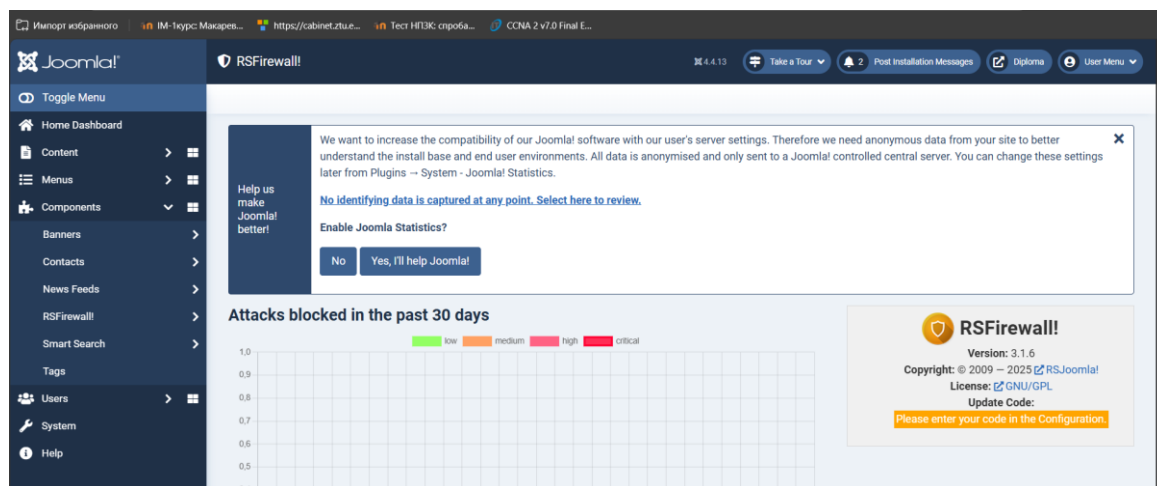


Рис . 3.10 Конструктор RSFirewall

Налаштування локального середовища з використанням XAMPP та встановлення CMS Joomla створює надійну платформу для розробки і тестування веб-сайтів. Інтеграція плагіна RSFirewall! одразу після встановлення Joomla дозволяє забезпечити базовий та розширений рівень захисту ще на початковому етапі роботи. Завдяки цьому створюється стабільне та безпечне середовище, яке відповідає сучасним вимогам до веб-безпеки, зокрема захисту від SQL-ін'єкцій, brute force атак та інших типових загроз. Це формує міцну основу для подальшої розробки та розгортання повнофункціонального сайту.

3.2 Налаштування та тестування плагіна RSFirewall

Першим кроком для налаштування безпеки сайту потрібно встановити та активувати самий плагін RSFirewall!. Цей етап є ключовим, оскільки RSFirewall! це потужний інструмент захисту, розроблений спеціально для Joomla, який забезпечує

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		51

всебічний захист вебресурсу. Його встановлення є відправною точкою для подальшого підвищення рівня безпеки.

Перш за все, необхідно завантажити дистрибутив плагіна з офіційного сайту розробника (RSJoomla!). Для цього потрібен активний акаунт та наявність ліцензії. Після завантаження ZIP-архіву переходимо у панель керування Joomla до розділу Система → Вставлення → Розширення →, вибираємо файл архіву та натисніть Завантажити та встановити.

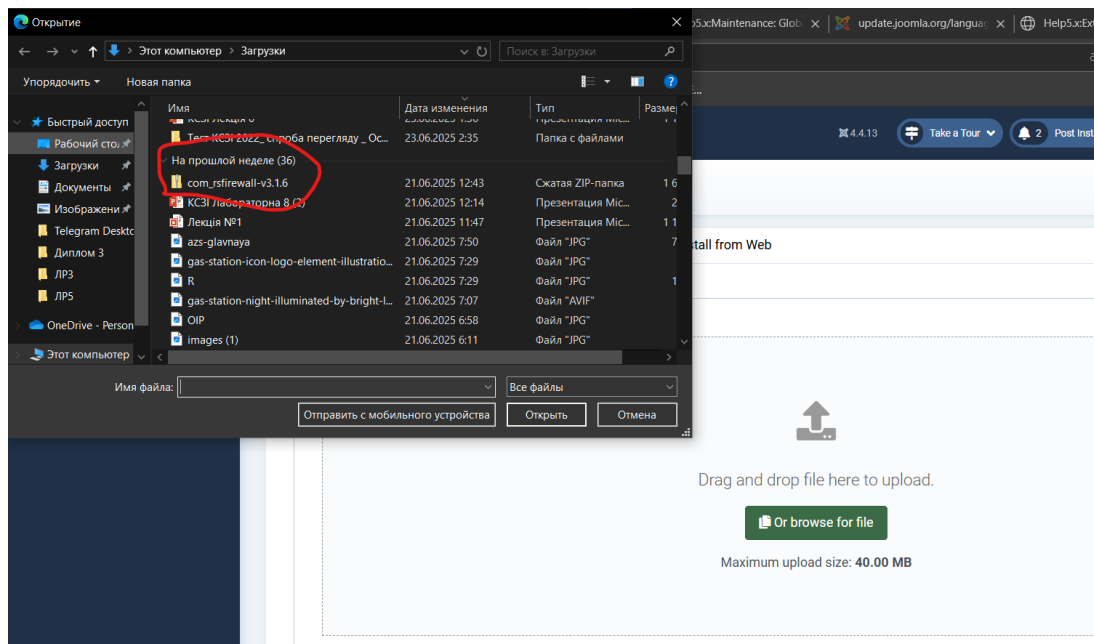


Рис 3.13 Встановлення плагіну

Після встановлення плагіна необхідно переконаватися, що він успішно доданий до системи. Для цього потрібно зайти в адміністративну панель Joomla і перейти до меню «Компоненти». Якщо все було встановлено правильно, у списку з'явиться новий пункт — RSFirewall!. Це означає, що компонент активний і готовий до використання. Також бажано перевірити, чи увімкнено системний плагін RSFirewall! у розділі «Розширення» → «Плагіни», і переконаватися, що його статус — «увімкнено». На рис 3.14 показано що компонент RSFirewall! з'явився в меню, а отже — плагін встановлено коректно і він інтегрувався в систему.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				52
Змн.	Арк.	№ докум.	Підпис	Дата		

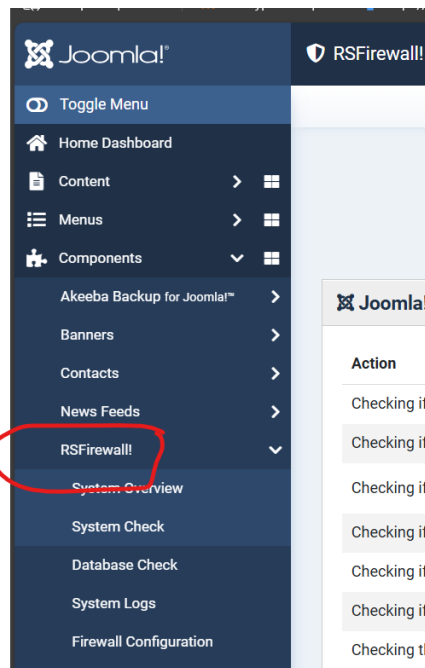


Рис 3.14 Зовнішній вигляд меню «Компоненти» після успішного встановлення RSFirewall!

Після виконання цих кроків користувач може ознайомитися з головним меню плагіна RSFirewall! та його основними розділами. Для цього потрібно натиснути на пункт RSFirewall! у меню «Компоненти». Відкриється головна сторінка плагіна, яка містить кілька важливих інструментів для забезпечення безпеки сайту.

Першим відкривається розділ Control Panel — це головна панель керування, де відображається загальна інформація про стан захисту сайту, попередження та швидкий доступ до інших функцій. Далі йде System Check — інструмент для повного сканування системи, який допомагає виявити підозрілі або змінені файли, перевірити права доступу, налаштування PHP та інші параметри, що впливають на безпеку.

У розділі Firewall Configuration знаходяться налаштування фаєрвола. Тут можна увімкнути захист від типових атак, таких як Brute Force чи SQL-ін'єкції, обмежити доступ до адміністративної частини сайту за IP-адресою, увімкнути блокування підозрілих запитів та інші функції.

Розділ Logs містить журнали подій — це зведення всіх важливих дій, що відбувалися на сайті, включаючи спроби входу, зміни файлів та виявлені загрози. У Blacklisted IPs зберігається список IP-адрес, яким заборонено доступ до сайту. Це

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		53

можуть бути адреси, які неодноразово намагалися зламати сайт або виконували підозрілу активність.

У вкладці Exceptions можна вказати файли чи розширення, які плагін не повинен перевіряти під час сканування. Це корисно, якщо певні файли постійно змінюються, але не становлять загрози.

Також доступний розділ Updates, у якому можна вручну перевірити та встановити останні оновлення сигнатур атак, щоб система реагувала на нові загрози. Якщо потрібно скинути всі налаштування до початкових, це можна зробити у розділі Restore Defaults.

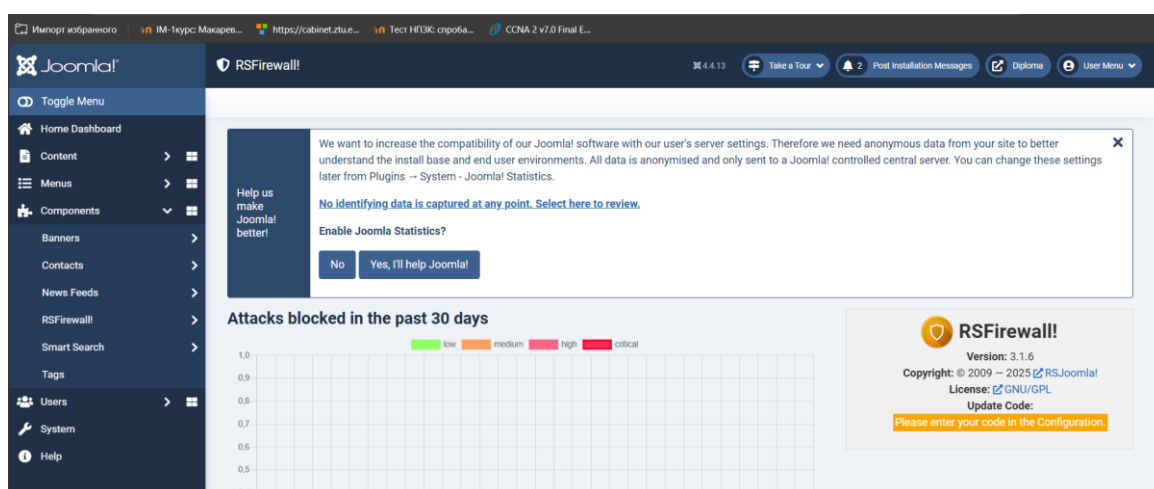


Рис 3.15 Головна сторінка плагіну RSFirewall!

Після успішного встановлення плагіна RSFirewall! для Joomla необхідно активувати базовий захист сайту. Для цього потрібно перейти до контрольної панелі компонента, яка знаходиться за шляхом Components → RSFirewall!. У відкритому вікні користувачу буде запропоновано скористатися функцією автоматичної перевірки стану системи.

Кнопка “Perform the system check” розташована безпосередньо в головному інтерфейсі RSFirewall! і служить для ініціації повного аудиту конфігурації Joomla-сайту. Після її натискання плагін виконує сканування системи на предмет типових вразливостей, неправильних налаштувань або відсутніх захисних механізмів. Зокрема, перевіряється наявність файлів .htaccess, коректність прав доступу до файлів і каталогів, налаштування безпеки PHP, доступність адміністративної панелі без обмежень, наявність шкідливих файлів або змін у системних елементах.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		54

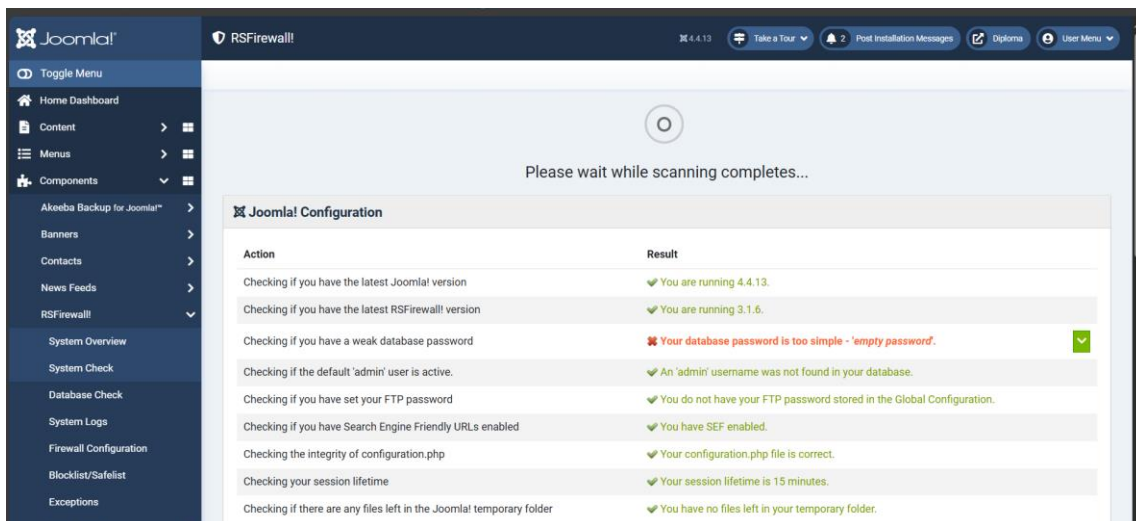


Рис 3.15 Сканування системи на предмет вразливостей

Після завершення перевірки користувач отримує звіт із детальним списком виявлених проблем. Кожна проблема супроводжується рекомендацією та кнопкою “Fix it”, яка дозволяє автоматично застосувати рекомендоване виправлення. Якщо деякі проблеми потребують ручного втручання, плагін надає точні інструкції для їх вирішення.

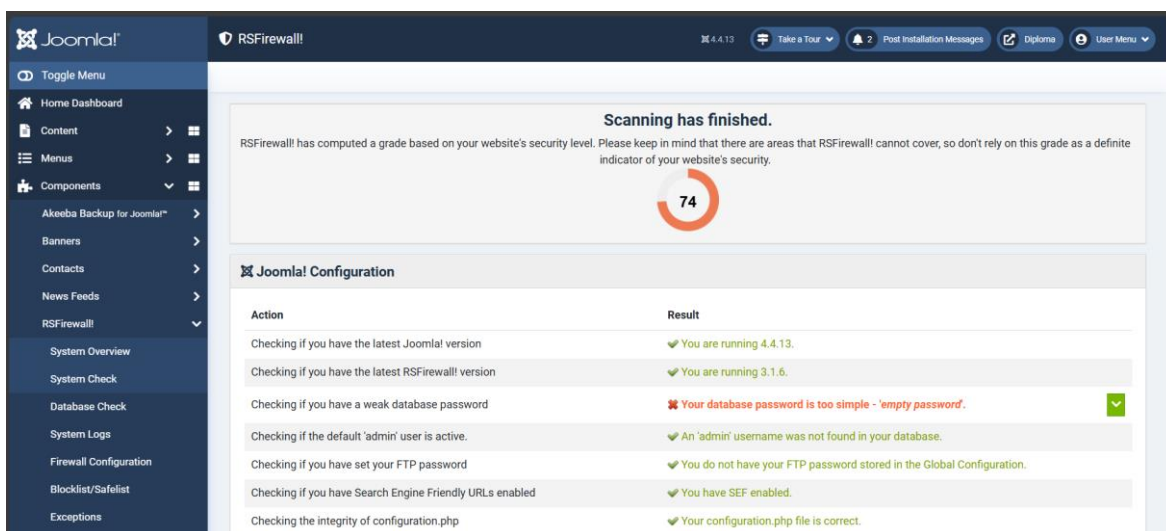


Рис 3.16 Звіт з перевірки

Серед типових дій, які виконує користувач після перевірки: створення або редагування .htaccess файлів, зміна прав доступу до конфігураційних файлів, обмеження доступу до адміністративної частини сайту, вимкнення небезпечних РНР-функцій, оновлення застарілих компонентів або модулів.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

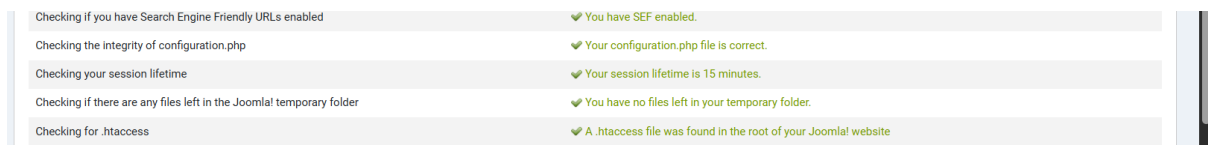


Рис 3.17 Результат редагування .htaccess

Налаштування брандмауера (Firewall) у RSFirewall! дозволяє значно підвищити безпеку сайту. Для активації основного захисту сайту в RSFirewall! необхідно увімкнути **Active Scanner**, який виконує роль брандмауера й аналізує всі вхідні запити на наявність небезпечної активності. У параметрах сканера можна вказати, як саме обробляти загрози: блокувати їх із записом до журналу або повністю забороняти доступ. Важливо також активувати захист від **brute-force атак**, що дозволяє обмежити кількість спроб входу в адміністративну панель.

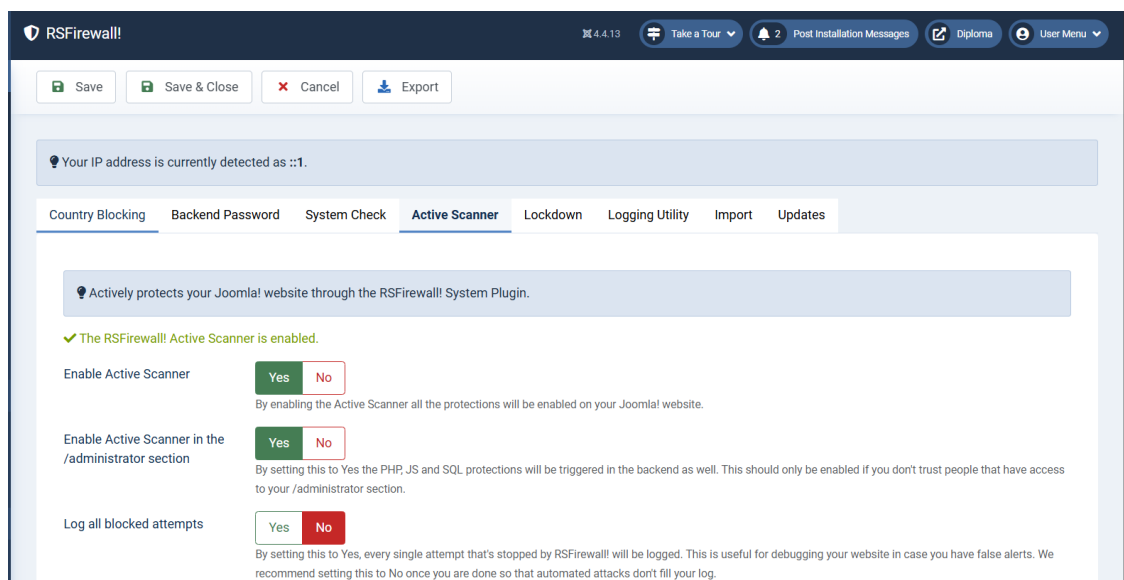


Рис 3.18 Активація Active Scanner

Щоб уникнути випадкового блокування доступу до адміністративної частини сайту, слід додати власну IP-адресу до списку дозволених у розділі **Safelist** компонента RSFirewall!. Це гарантує, що система безпеки не обмежуватиме вхід із цієї адреси навіть у разі спрацювання захисних механізмів. За необхідності можна також увімкнути режим **System Lockdown**, який тимчасово блокує всі зміни на сайті. Ця функція доцільна у випадках виявлення критичних загроз або перед публікацією сайту в мережі.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

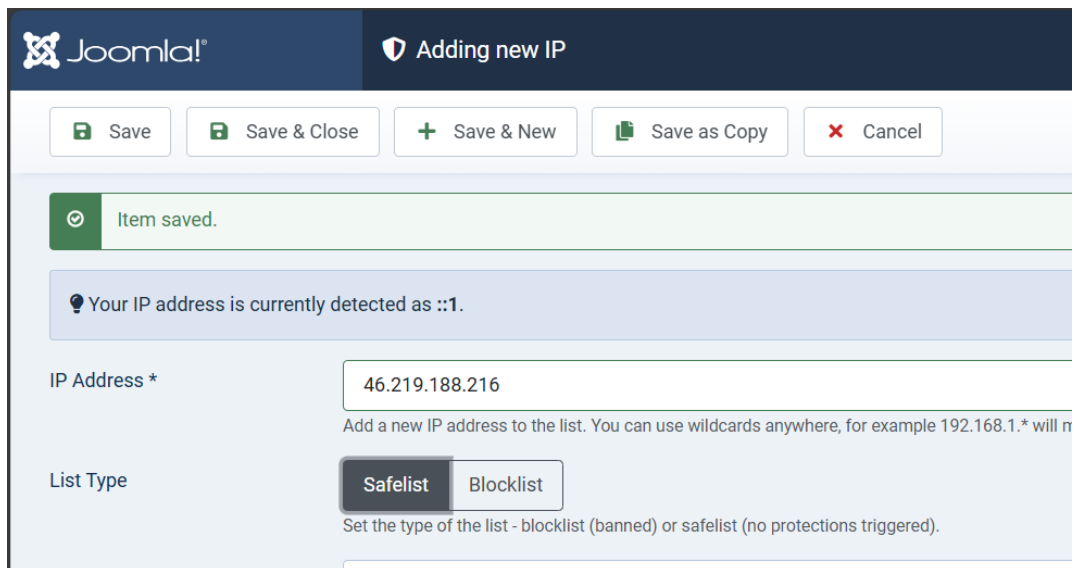


Рис 3.19 Додавання власної IP-адреси у список дозволених

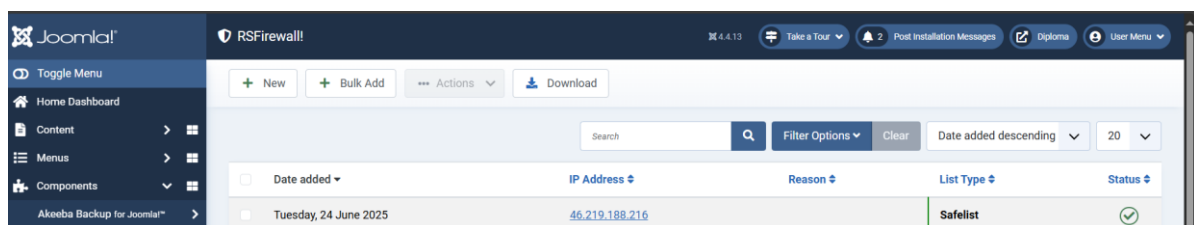


Рис 3.20 Результат успішного додавання власної IP-адреси

У системі безпеки RSFirewall активовані такі правила захисту від найпоширеніших типів атак:

- SQL Injection (sqli) – увімкнено перевірку GET-запитів на наявність SQL-ін'єкцій, що дозволяє блокувати спроби несанкціонованого доступу або маніпуляції базою даних.
- XSS: Cross Site Scripting (xss) – активовано фільтрацію JavaScript у POST-формах, що забезпечує захист від впровадження шкідливого скрипту у форми зворотного зв'язку та інші елементи взаємодії з користувачем.
- LFI/RFI: Local/Remote File Inclusion – включено захист від підключення локальних та віддалених файлів через URL-параметри, що запобігає доступу до конфіденційних файлів і завантаженню шкідливих скриптів.
- DoS-захист – реалізовано блокування запитів від підозрілих користувацьких агентів (curl, perl, Java), а також активовано перевірку IP-адрес на наявність

у чорних списках (Tornevall, Spamhaus, JustSpam.org), що знижує ризик масових автоматизованих атак.

The screenshot displays the Joomla! Security Settings interface. At the top, a note states: "If any of the following files will be changed, you will be alerted by email and an entry will be posted in the System Log." Below this, several settings are listed:

- Use IP Address as reported by Joomla!**: Set to "Yes". Description: "Set this to 'Yes' to grab the IP as reported by the Joomla! framework. This will take into account the 'Behind Load Balancer' option under Global Configuration. Set it to 'No' and select the PHP headers that forward the real IP address from the 'Grab IP from Proxy Headers' field below."
- PHP Protections**:
 - Local file inclusion**: Set to "Yes". Description: "This disallows directory traversal techniques (such as controller=.././etc/passwd) that might allow an attacker to read sensitive files by exploiting poorly coded extensions."
 - Remote file inclusion**: Set to "Yes". Description: "This disallows access to URLs (such as controller=http://www.malicious-site.com/exploit.txt) that might allow an attacker to download and run malicious scripts by exploiting poorly coded extensions."
 - Enable protections for**: A dropdown menu showing "URL data (GET)". Description: "Form data (POST) enables filtering for information submitted through forms (eg. article editing, user registration etc). URL data (GET) enables filtering for variables that are located in the URL (eg. http://www.yoursite.com/index.php?option=com_test¶meter1=value1)".
- SQL Protections**:
 - Enable protections for**: A dropdown menu showing "URL data (GET)". Description: "Form data (POST) enables filtering for information submitted through forms (eg. article editing, user registration etc). URL data (GET) enables filtering for variables that are located in the URL (eg. http://www.yoursite.com/index.php?option=com_test¶meter1=value1)".

At the bottom of the screenshot, there are buttons for "Save", "Save & Close", "Cancel", and "Export". Below these, more settings are visible:

- JS Protections**:
 - Filter Javascript**: Set to "Yes". Description: "By setting this to Yes, the Javascript will be filtered instead of the connection being dropped."
 - Enable protections for**: A dropdown menu showing "Form data (POST)". Description: "Form data (POST) enables filtering for information submitted through forms (eg. article editing, user registration etc). URL data (GET) enables filtering for variables that are located in the URL (eg. http://www.yoursite.com/index.php?option=com_test¶meter1=value1)".
- Denial of Service**:
 - Deny access to the following User Agents**: A list box containing "perl", "cURL", and "Java". Description: "The following User Agents are usually automated requests to your website and should not be allowed. empty User Agents are usually DoS attack attempts or automated connections to your website. perl scripts are used for automated connections to your website. cURL is used for automated connections to your website. Java performs automated connections as well. Mozilla Impersonators are automated scripts trying to impersonate a legitimate browser."
 - Protect forms from abusive IPs**: Set to "Yes". Description: "Enabling this option will protect your forms from abusive IPs by checking if they exist in a PBL list."
 - PBL lists**: A list box containing "Tornevall", "The Spamhaus Project", and "JustSpam.org". Description: "Search the following PBL lists (public blocklists) when checking for abusive IPs."

Рис 3.21 Увімкнення систем захисту

Наступним етапом є налаштування захисту від атак типу Brute Force – одного з ключових і найнеобхідніших елементів безпеки для системи управління сайтом Joomla.

Даний тип захисту спрямований на запобігання несанкціонованим спробам входу до адміністративної панелі, які здійснюються шляхом багаторазового

підбору логіну та пароля. Зловмисники використовують автоматизовані скрипти, які можуть здійснювати тисячі спроб за короткий проміжок часу, намагаючись підібрати вірні облікові дані.

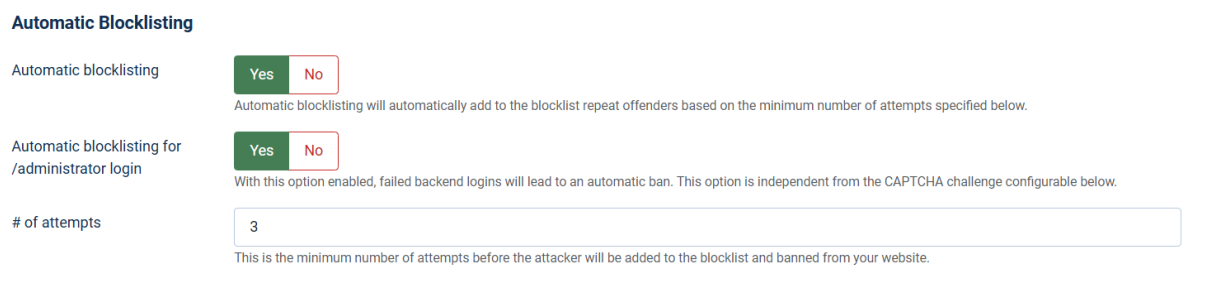


Рис 3.22 Встановлення обмежень

У цьому розділі налаштовується захист від автоматизованих спроб несанкціонованого доступу до сайту шляхом підбору паролів. Для забезпечення максимальної ефективності системи рекомендується активувати всі ключові параметри безпеки, які допомагають виявляти та блокувати підозрілу активність. Зокрема, слід увімкнути захист від Brute Force атак із встановленням чітких обмежень — наприклад, блокування IP-адреси після трьох невдалих спроб входу або трьох хибних запитів на відновлення паролю.

Процес створення нового облікового запису адміністратора є ключовим для надання іншому користувачеві повного доступу до панелі керування сайтом. Під час цього необхідно уважно вказати всі обов'язкові дані: ім'я користувача, електронну пошту, надійний пароль, а також призначити роль "Administrator".

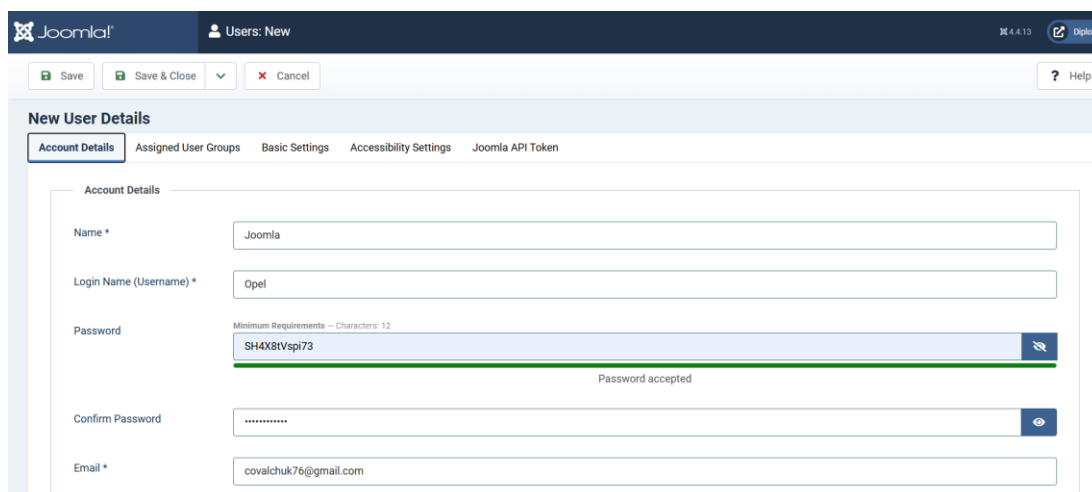


Рис 3.23 Створення користувача

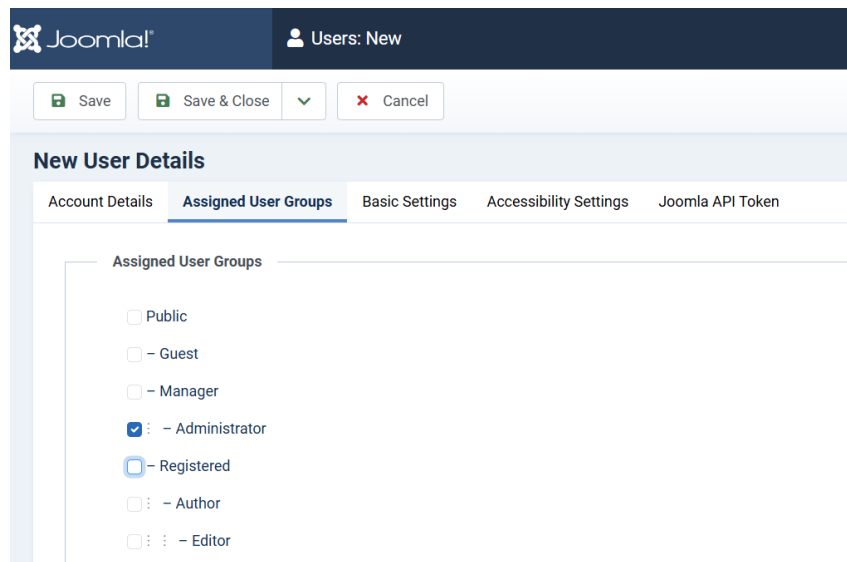


Рис 3.24 Надання ролі адміністратора

Якщо всі дані були введені правильно, новий користувач має з'явитися в списку у розділі «Користувачі» на панелі керування.

Двофакторна автентифікація (2FA) є важливою складовою системи безпеки, яка суттєво підсилює захист облікових записів на сайті. У Joomla ця функція доступна у відповідному розділі налаштувань безпеки та дозволяє додати додатковий рівень автентифікації при вході в систему. Після активації 2FA, окрім пароля, користувач повинен буде ввести одноразовий код із мобільного додатку, наприклад Google Authenticator або іншої сумісної програми.

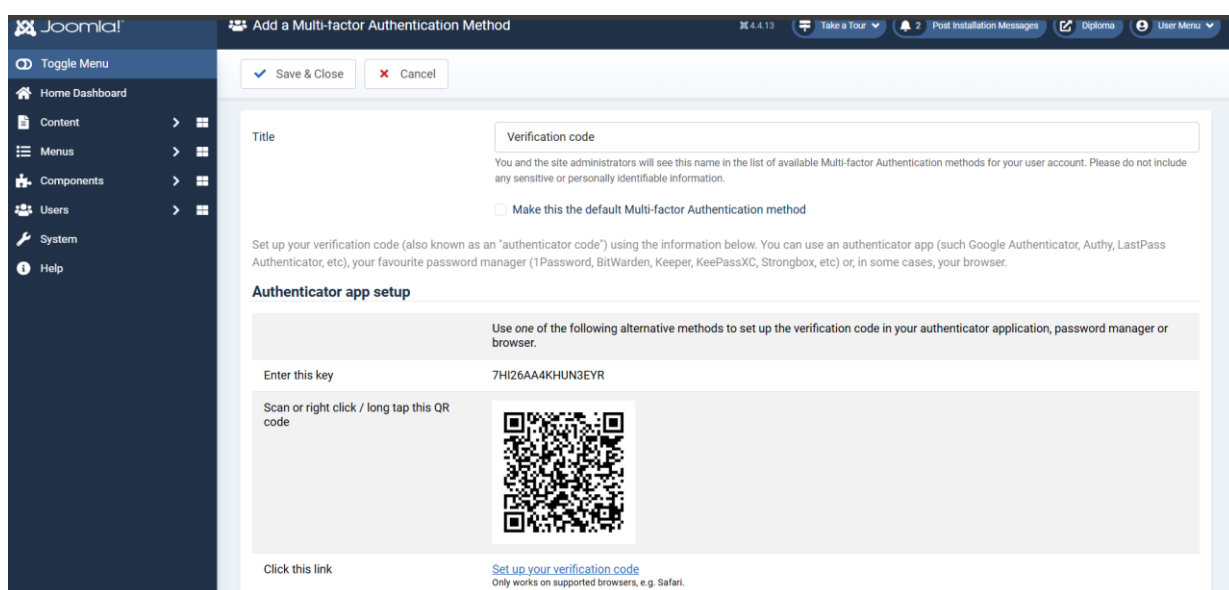


Рис 3.25 Налаштування двофакторної автентифікації

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				60
Змн.	Арк.	№ докум.	Підпис	Дата		

На скріншоті відображається сторінка налаштувань двофакторної автентифікації в Joomla. Інтерфейс розділено на дві основні частини. У верхній частині користувачу пропонується відсканувати QR-код або вручну ввести секретний ключ у застосунку для автентифікації (наприклад, Google Authenticator чи Authy). Це дозволяє підключити 2FA-захист до облікового запису адміністратора Joomla.

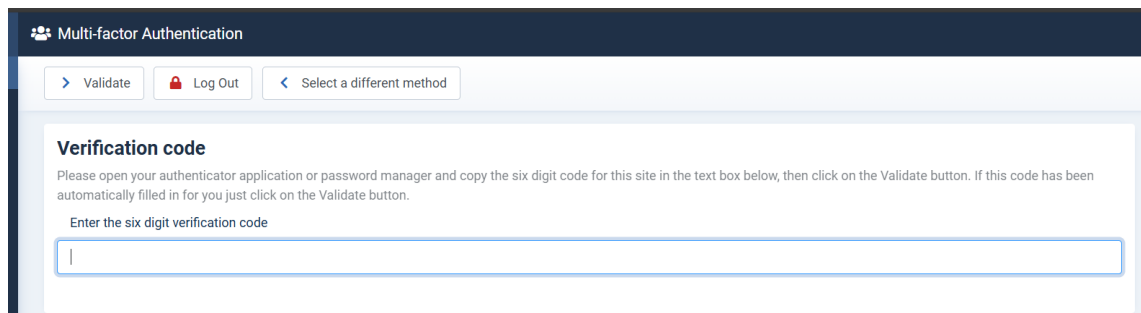


Рис 3.26 Запит на верифікаційний код аутентифікатору.

Після проведених налаштувань при вході на сторінку адміністратора видає вікно підтвердження входу за допомогою двофакторної автентифікації. На сторінці є поле для введення шестизначного коду, який потрібно отримати з додатку-автентифікатора. Над полем розташовані кнопки: "Validate" (Підтвердити), "Log Out" (Вийти) та "Select a different method" (Вибрати інший метод).

Після цього було відкрито розділ reCAPTCHA у плагіні RSFirewall, який відповідає за захист форм входу та реєстрації на сайті. У цьому розділі адміністратор може активувати Google reCAPTCHA v2 — це класична версія CAPTCHA, яка додає на форму просту перевірку "Я не робот".

Щоб її увімкнути, потрібно ввести Site key і Secret key, які попередньо генеруються на офіційному сайті Google reCAPTCHA. Після збереження налаштувань ця перевірка з'являється на формах входу або реєстрації. Користувачам потрібно буде поставити галочку "Я не робот" або пройти коротку візуальну перевірку (наприклад, вибрати всі зображення зі світлофорами).

Це значно ускладнює автоматизованим ботам спроби входу чи реєстрації, тим самим підвищуючи загальний рівень безпеки сайту.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		61

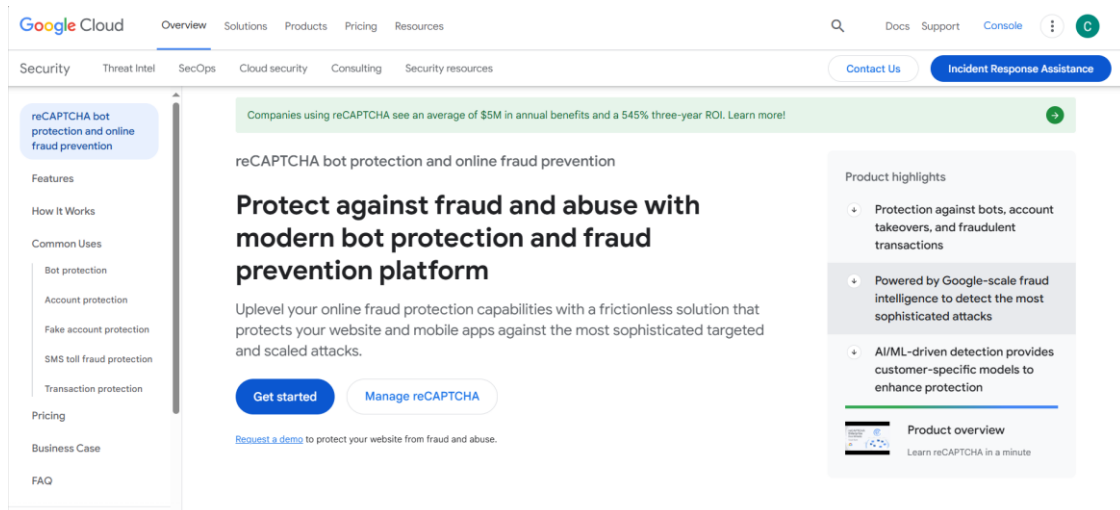


Рис 3.27 Головна сторінка сайту

Було здійснено реєстрацію сайту в адміністративній панелі Google reCAPTCHA, де у відповідній формі було вказано назву ресурсу (Label), вибрано тип захисту — reCAPTCHA v2, а також додано доменне ім'я сайту та контактну електронну адресу адміністратора. Після прийняття умов використання сервісу, Google автоматично згенерував два унікальні ключі — публічний (Site Key) та секретний (Secret Key).

Ці ключі є необхідними для інтеграції reCAPTCHA з Joomla. Вони використовуються для зв'язку між сайтом і сервісом Google: публічний ключ вбудовується у форми на сайті, а секретний зберігається на сервері та використовується для перевірки достовірності відповідей користувачів. Після отримання ключів можна переходити до налаштування плагіну reCAPTCHA безпосередньо в Joomla або RSFirewall.

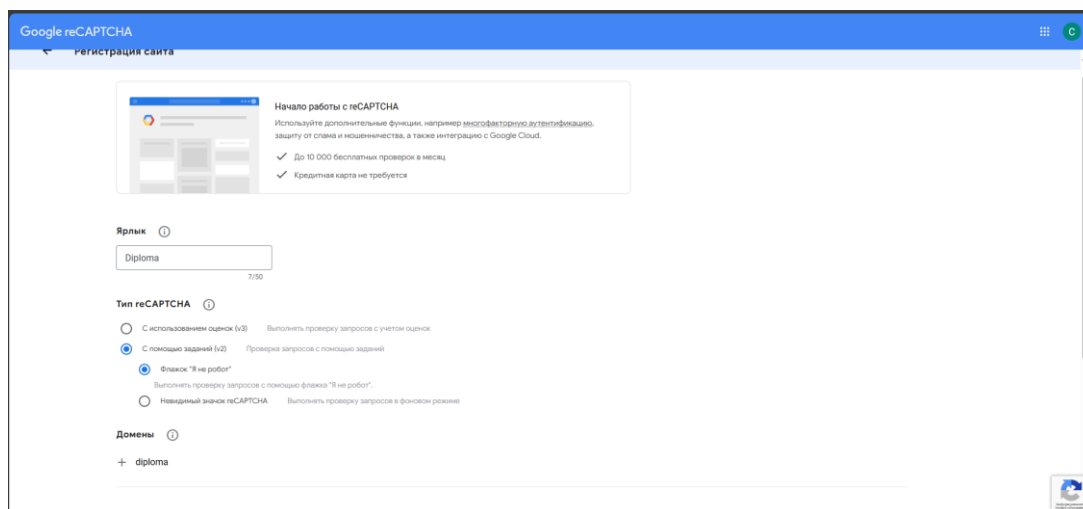


Рис 3.28 Вибір версії для нашого плагіну.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

Используйте этот ключ в HTML-коде, который ваш сайт передает на устройства пользователей. [Информация об интеграции на стороне клиента](#)

СКОПИРОВАТЬ
КЛЮЧ САЙТА

6Lc-vmwrAAAAIoY6fjDon1TavL56Ti-mPZIS_Ng

Используйте этот секретный ключ для обмена данными между сайтом и сервисом reCAPTCHA. [Информация об интеграции на стороне сервера](#)

СКОПИРОВАТЬ
СЕКРЕТНЫЙ КЛЮЧ

6Lc-vmwrAAAAAGJMUzQ0soVlg9HTkvRgGXfJDrKf

[ОТКРЫТЬ НАСТРОЙКИ](#)

[ПЕРЕЙТИ В GOOGLE АНАЛИТИКУ](#)

Рис 3.29 Ключі для інтеграції reCAPTCHA v2

Після завершення реєстрації сайту в Google reCAPTCHA система автоматично надає два унікальні ключі — Site Key та Secret Key. Ці ключі є обов'язковими для інтеграції захисту reCAPTCHA з вебсайтом на Joomla.

Наступним кроком є перехід до відповідного плагіна у Joomla (наприклад, RSFirewall або CAPTCHA - reCAPTCHA), де отримані ключі потрібно вставити у відповідні поля. Site Key використовується для відображення CAPTCHA на сторінках сайту, тоді як Secret Key дозволяє серверу перевіряти правильність виконаної перевірки.

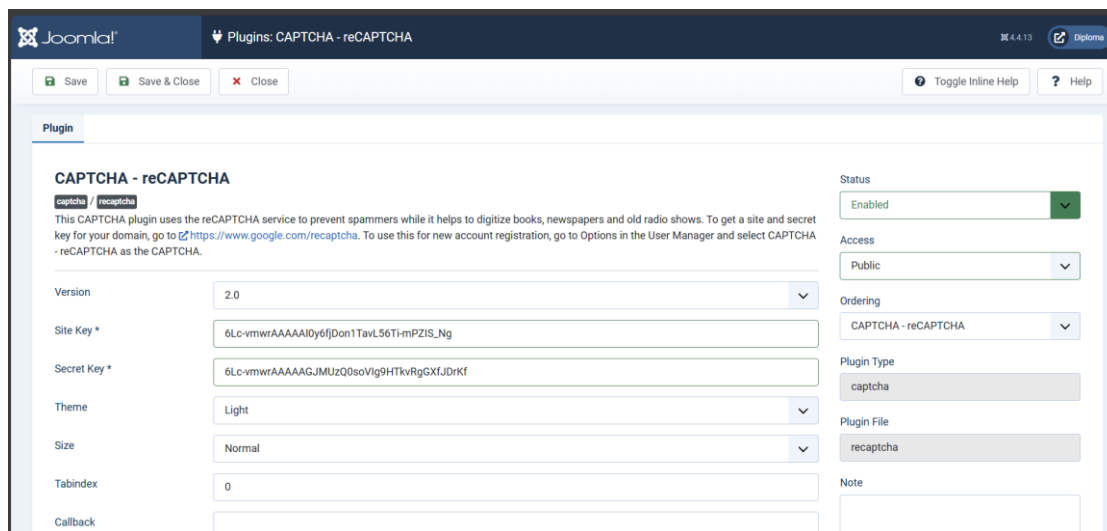


Рис 3.30 Ведення ключів у самий плагін.

Після заповнення полів і збереження налаштувань необхідно активувати опцію "Enable". Ця функція забезпечує автоматичне додавання reCAPTCHA v2 до форм входу та реєстрації, тим самим запобігаючи спробам автоматизованого входу, масової реєстрації ботів та іншим типам зловмисних дій.

Розділ Logging Utility у плагіні RSFirewall! призначений для налаштування системи автоматичного оповіщення про події, пов'язані з безпекою сайту на Joomla. Це дає змогу адміністраторам швидко реагувати на загрози, отримуючи відповідні повідомлення електронною поштою. Завдяки цьому розділу можна не лише виявляти спроби несанкціонованого доступу або змін до системних файлів, а й здійснювати постійний моніторинг стану безпеки. Гнучкі параметри конфігурації дозволяють обрати, які саме типи подій варто реєструвати, як довго зберігати логи, та з якою частотою надсилати повідомлення. Крім того, передбачена можливість інтеграції з зовнішніми сервісами для швидкого отримання інформації про IP-адреси, що робить аналіз загроз більш ефективним. У сукупності, Logging Utility є важливим інструментом для проактивного захисту вебресурсу та своєчасного реагування на потенційні вразливості.

The screenshot displays the 'Logging Utility' configuration page within the RSFirewall! interface. At the top, there are buttons for 'Save', 'Save & Close', 'Cancel', and 'Export'. Below these, a status message indicates the user's IP address is detected as ::1. The main configuration area includes tabs for 'Country Blocking', 'Backend Password', 'System Check', 'Active Scanner', 'Lockdown', 'Logging Utility' (selected), 'Import', and 'Updates'. A blue box at the top of the configuration area states: 'Logs any events that trigger RSFirewall! so that you can review them.' The configuration fields include: 'Email address(es)' with a text input and a note to specify every address on a new line; 'Include these alert levels' with radio buttons for 'low' (selected) and 'medium'; '# days to keep log history' set to 30, with a note about automatic deletion; '# emails per hour' set to 50, with a note about spamming; '# events to show' set to 5, with a note about the System Overview; 'IPv4 Whois Service' set to 'http://whois.domaintools.com/{ip}', with a note about the placeholder; and 'IPv6 Whois Service' set to 'http://whois.domaintools.com/{ip}', with a note about the placeholder.

Рис 3.31 Налаштування розділу Logging Utility

На скріншоті видно, що в налаштуваннях розділу Logging Utility плагіна RSFirewall. Рівень оповіщення встановлено на low, тобто система надсилатиме повідомлення про всі події — навіть незначні. Це корисно для максимально детального моніторингу, але може створити великий потік повідомлень на пошту.

Історія журналу подій зберігатиметься протягом 30 днів, після чого записи автоматично видалятимуться, щоб база не перевантажувалась. Обмеження на кількість листів на годину встановлено на 50, що дозволяє уникати перевантаження поштового сервера або потрапляння повідомлень у спам. У системному огляді буде відображено останні 5 подій, що дозволяє швидко побачити поточну активність без перевантаження інтерфейсу.

Для аналізу підозрілих дій із конкретних IP-адрес активовано сервіс Whois — для IPv4-адрес використовується посилання <http://whois.domaintools.com/{ip}>, а для IPv6 також зазначено відповідний сервіс. Це дає змогу швидко перевіряти, звідки саме відбувається підозріла активність.

Тестування плагіну

Процес тестування проводиться для оцінки ефективності ключових механізмів захисту Joomla-сайту, зокрема антивірусного сканування, роботи фаєрвола, аудиту змін, двофакторної автентифікації та перевірки через Google reCAPTCHA. На зображенні нижче продемонстровано роботу механізму Google reCAPTCHA, який використовується як додатковий рівень безпеки для захисту форм авторизації. Після введення логіна й пароля система вимагає підтвердження, що користувач не є ботом, запропонувавши пройти візуальну перевірку (наприклад, вибрати всі зображення зі світлофорами чи мостами). Такий захід значно ускладнює реалізацію автоматизованих атак типу brute-force, знижуючи ризик несанкціонованого доступу до облікових записів. Використання reCAPTCHA в поєднанні з іншими засобами безпеки дозволяє створити надійний бар'єр проти ботів і скриптів злоумисників.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

You are here: Home

Home

User Registration

* Required field

Name *

Zafira

Username *

Opel

Password *

Minimum Requirements — Characters: 12

Lq38sqUnP57sMa

Password accepted

Confirm Password *

Email Address *

nosivka19@gmail.com

Captcha *

☒ I'm not a robot

reCAPTCHA

Register

Рис 3.32 Перевірка Google reCAPTCHA.

Оскільки під час тестування не було зафіксовано підозрілого трафіку або спроб несанкціонованого доступу, перевірка ефективності засобів безпеки, зокрема Google reCAPTCHA, відбувалась у звичайному режимі без відображення візуальних тестів (фото). Система автоматично визначила, що користувач не є ботом, тому механізм reCAPTCHA обмежився лише підтвердженням "Я не робот" без подальших візуальних завдань. Така поведінка є типовою в умовах низького ризику, коли не виявлено активності, схожої на автоматизовану. Це свідчить про правильну інтеграцію захисту та коректну роботу механізму в умовах нормального трафіку.

Для підвищення рівня безпеки доступу до адміністративної панелі Joomla було активовано вбудовану CAPTCHA. Після цього, при кожному вході в меню адміністратора, система додатково запитує ввести символи, зображені на картинці. Такий механізм дозволяє захистити сайт від автоматизованих спроб входу (brute-force атак), оскільки лише реальний користувач здатен правильно розпізнати й ввести текст із зображення, що значно ускладнює роботу ботів.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		66

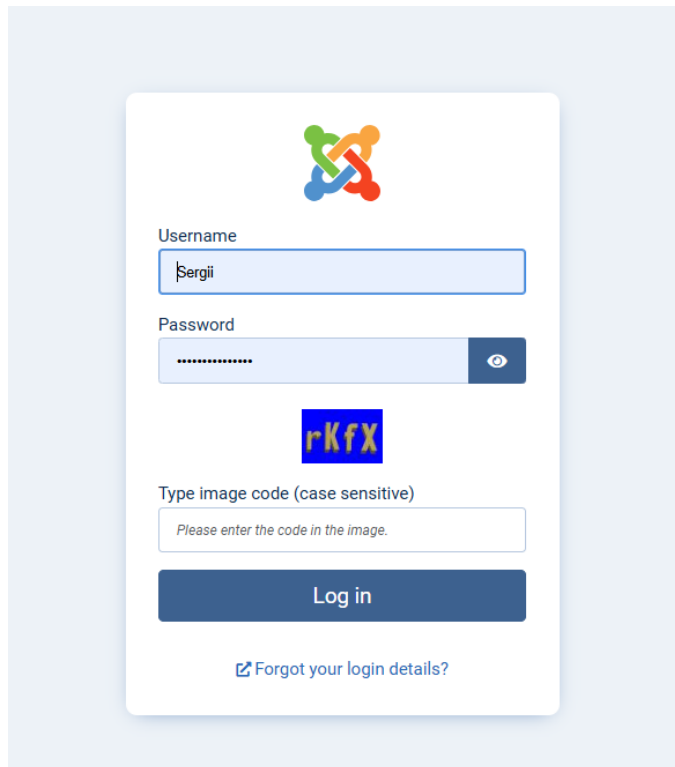


Рис 3.33 Вікно роботи вбудованої САПТСНА у вході адміністратора

Після введення логіна, пароля та проходження вбудованої САПТСНА, система безпеки Joomla! переходить до наступного етапу автентифікації — запиту коду верифікації. Це є частиною механізму двофакторної автентифікації (2FA), який було попередньо активовано в налаштуваннях облікового запису адміністратора. Користувач повинен ввести шестизначний код, згенерований у мобільному додатку для автентифікації Google Authenticator.

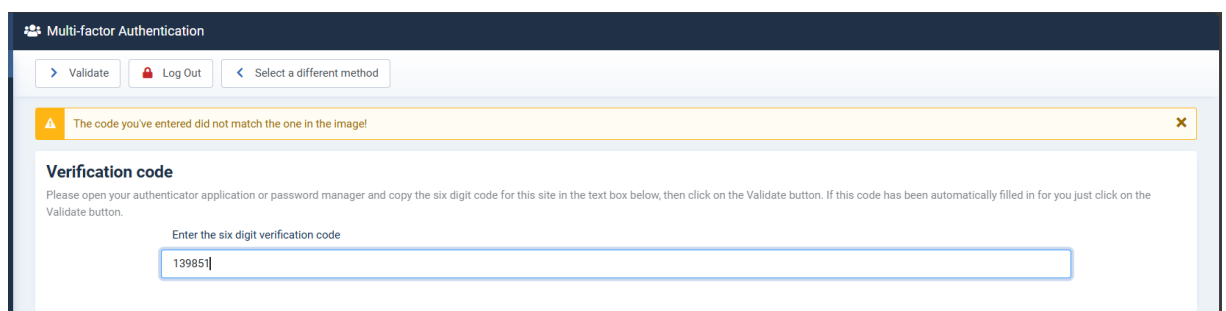


Рис 3.34 Введення коду згенерованого Google Authenticator

Після успішного проходження всіх перевірок безпеки — введення логіна, пароля, САПТСНА та коду двофакторної автентифікації — система надає доступ до головної панелі адміністратора. Це означає, що користувач пройшов повну процедуру автентифікації, і Joomla! розпізнала його як авторизованого та надійного. Головна панель відкриває доступ до всіх функцій керування сайтом, що дозволяє

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

виконувати налаштування, оновлення, встановлення розширень і керування Контентом.

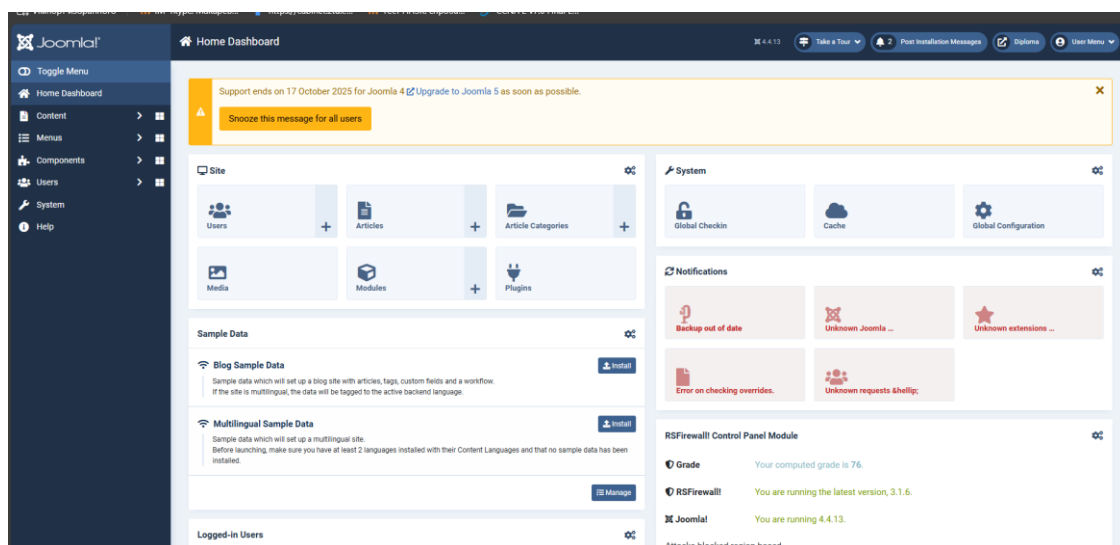


Рис 3.35 Головна панель

Наступним етапом є тестування системи логуювання. Цей процес передбачає перевірку, чи коректно фіксуються всі події, пов'язані з безпекою сайту, зокрема спроби входу, зміни системних файлів, блокування IP-адрес, активація захисних механізмів тощо. У панелі керування плагіном RSFirewall! відкривається розділ Logging Utility, де адміністратор може переглянути останні події, відфільтрувати їх за типом або датою, а також перевірити, чи були надіслані відповідні повідомлення на вказану електронну пошту. Це тестування дозволяє переконатися, що система безпеки не лише реагує на загрози, а й надає повну інформацію для аналізу інцидентів, що є критично важливим для своєчасного виявлення та реагування на атаки.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

RSFirewall!									
+ Add to Blocklist + Add to Safelist X Delete X Empty the log Download									
Search Filter Options Clear Date of event descending 20									
	Alert level	Date of event	IP address	User ID	Username	Page	Referer	Description	
☐	Block	medium	2025-06-25 16:40:06	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using an unknown username.	
☐	Block	medium	2025-06-25 16:39:52	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using an unknown username.	
☐	Block	medium	2025-06-25 16:39:40	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using an unknown username.	
☐	Block	medium	2025-06-25 16:36:31	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using a known username.	
☐	Block	medium	2025-06-25 16:36:15	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using a known username.	
☐	Block	medium	2025-06-25 16:35:53	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/	There was an unsuccessful attempt to login into the backend section of your website using an unknown username.	
☐	Block	medium	2025-06-25 16:32:37	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using an unknown username.	
☐	Block	medium	2025-06-25 16:32:36	0		http://localhost/joomla/	http://localhost/joomla/a	There was an unsuccessful attempt to login into the	

Рис 3.36 Системний журнал Joomla з записами про невдалі логіни.

У таблиці відображені кілька записів про невдалі спроби входу в адміністративну панель сайту. Усі події мають середній рівень загрози (medium) та сталися з локальної IP-адреси ::1 (localhost). Деякі спроби були здійснені з невідомим іменем користувача, а деякі — з відомим, але вхід усе одно не відбувся.

Кожен запис містить дату й час події, сторінку, на яку здійснювався вхід (/administrator/index.php), а також посилання на джерело (referer). Для кожної події доступна кнопка Block, яка дозволяє заблокувати джерело спроби входу.

☐	Block	medium	2025-06-24 14:16:41	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php?option=com_users&task=method.save&id=0&method=totp&user_id=786	There was an unsuccessful attempt to login into the backend section of your website using an unknown username. Debug information Username requested: Serhii Password requested: Lq38s:qUnPS7sMa
☐	Block	medium	2025-06-25 16:11:22	0		http://localhost/joomla/administrator/index.php	http://localhost/joomla/administrator/index.php	There was an unsuccessful attempt to login into the backend section of your website using a known username. Debug information Username requested: Opel Password requested: SH4X8tV

Рис 3.37 Записи про невдалі спроби входу до адміністративної панелі Joomla.

На скріншоті показано журнал подій безпеки в Joomla, де зафіксовано кілька невдалих спроб входу в адміністративну панель сайту. Спочатку були три спроби з невідомими іменами користувачів, після чого відбулася спроба входу з відомим логіном, але з неправильно введеним паролем. Усі події зареєстровані компонентом RSFirewall! і мають середній рівень загрози.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				69
Змн.	Арк.	№ докум.	Підпис	Дата		

Ще одним кроком у забезпеченні безпеки сайту є проведення перевірки системи — System Check у компоненті RSFirewall!.

Цей інструмент дозволяє автоматично проаналізувати поточний стан безпеки Joomla-сайту. Після запуску перевірки RSFirewall! сканує права доступу до файлів і папок, конфігураційні налаштування, версію Joomla, наявність відомих вразливостей та інші критичні параметри.

Результати перевірки відображаються у вигляді списку з рекомендаціями щодо усунення знайдених проблем. Це дає змогу своєчасно виявити та виправити потенційні загрози безпеці.

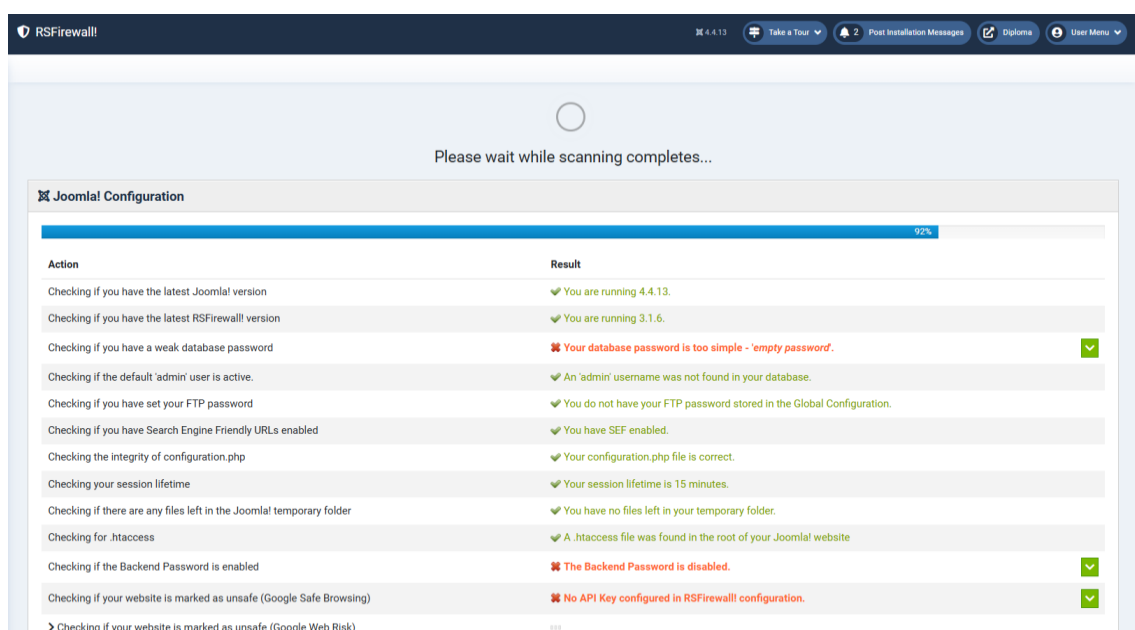


Рис 3.38 Запуск сканера перевірки системи.

Сканування безпеки сайту, виконане за допомогою компонента RSFirewall!, показало поточний рівень захисту — 76 зі 100 балів. Це означає, що загалом сайт захищений добре, однак є кілька моментів, які потребують покращення.

Серед основних проблем виявлено відсутність пароля до бази даних, що становить серйозну вразливість. Також на сайті досі активний обліковий запис з іменем "admin", що не рекомендується, оскільки такі логіни найчастіше атакують автоматизовані системи підбору паролів. Крім цього, вимкнено опцію Backend Password, яка дозволяє встановити додатковий захист для адміністративної панелі. Також не вказано API-ключ для сервісів Google Safe Browsing або Web Risk — ця функція необов'язкова, але могла б допомогти у виявленні потенційних загроз.

Позитивним є те, що на сайті використовується актуальна версія Joomla! та RSFirewall!, увімкнено SEO-дружні посилання, не виявлено шкідливого програмного забезпечення, а також відсутні тимчасові або непотрібні файли. Це свідчить про те, що базова безпека підтримується на належному рівні, проте деякі налаштування варто доопрацювати для підвищення загального рівня захисту.

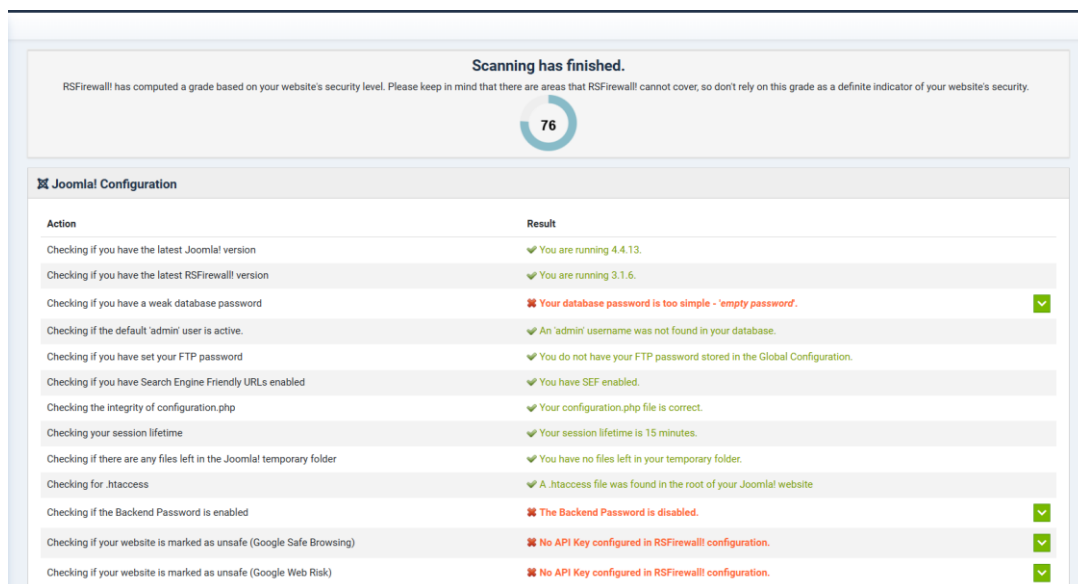


Рис 3.39 Результат сканування.

Використання плагіна RSFirewall! дозволяє суттєво посилити безпеку Joomla-сайту завдяки широкому набору інструментів: антивірусному скануванню, налаштуванням фаєрвола, захисту від brute-force атак, двофакторній автентифікації та інтеграції з Google reCAPTCHA. Проведене налаштування та тестування показали, що основні механізми безпеки працюють коректно. Водночас результати системного сканування виявили кілька критичних моментів, які потребують виправлення, зокрема — відсутній пароль до бази даних та активний обліковий запис “admin”. Усунення цих недоліків дозволить досягти максимально можливого рівня захисту і зменшити ризики зловмисних атак у майбутньому.

Висновок до третього розділу

Налаштування локального середовища з використанням XAMPP у поєднанні з встановленням CMS Joomla створює повноцінну платформу для розробки, тестування та початкового впровадження веб-сайтів. XAMPP забезпечує всі необхідні компоненти серверного оточення, включно з Apache, MySQL і PHP, що дозволяє моделювати роботу сайту в умовах, максимально наближених до реального хостингу. Завдяки простому інтерфейсу та зручному керуванню компонентами, розробник може швидко розгорнути CMS, внести необхідні налаштування та працювати з базою даних через phpMyAdmin.

Одразу після встановлення Joomla доцільно інтегрувати плагін RSFirewall!, який забезпечує базовий і розширений рівень захисту ще на початковому етапі. Цей плагін включає низку потужних інструментів, таких як антивірусне сканування, налаштування фаєрвола, захист від brute-force атак, підтримка двофакторної автентифікації, інтеграція з Google reCAPTCHA та моніторинг дій користувачів. Він також дозволяє обмежити доступ до адміністративної панелі за IP-адресою або через спеціальний ключ, вести журнал активності та сканувати системні файли на наявність змін. Таке багаторівневе захист дозволяє знизити ризики зовнішнього втручання та зловмисних атак на сайт.

У процесі тестування працездатності механізмів безпеки було підтверджено, що RSFirewall! виконує свої функції ефективно. Водночас були виявлені критичні налаштування, які потребують негайного виправлення — зокрема, активний обліковий запис “admin” без зміни імені користувача та відсутній пароль до бази даних. Ці недоліки значно знижують загальний рівень захисту і можуть бути використані зловмисниками. Усунення виявлених проблем дозволить досягти максимально можливого рівня безпеки Joomla-сайту та забезпечить його стабільну роботу в умовах сучасних кіберзагроз.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				72
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах кваліфікаційної роботи було досягнуто всіх поставлених завдань, що дозволило реалізувати комплексний підхід до захисту вебдодатку, побудованого на CMS Joomla. Першим етапом стало налаштування локального середовища за допомогою XAMPP — це створило повноцінну інфраструктуру для розробки, моделювання й тестування веб-сайту в умовах, максимально наближених до реального хостингу. У результаті було успішно встановлено й налаштовано сервер Apache, систему управління базами даних MySQL та середовище PHP. Також було створено базу даних і підключено CMS Joomla, що стало основою для подальших досліджень у сфері інформаційної безпеки.

Наступним важливим завданням стало встановлення та конфігурація системи керування контентом Joomla. Було розгорнуто повноцінний веб-сайт, створено адміністративний обліковий запис, підключено базу даних і виконано базові налаштування. На цьому етапі також було проаналізовано архітектуру CMS Joomla з точки зору її безпекових особливостей, зокрема систему ACL (керування правами доступу), реалізацію фільтрації даних, підтримку CAPTCHA, CSRF-токенів та механізмів захисту від SQL-ін'єкцій. Було виявлено, що навіть базові функції Joomla забезпечують початковий рівень безпеки, однак для захисту від актуальних загроз цього недостатньо.

З метою підвищення рівня захисту було інтегровано плагін RSFirewall!, що є потужним інструментом для виявлення вразливостей, сканування системи, моніторингу активності користувачів, реалізації фаєрволу та захисту від найпоширеніших атак — зокрема brute-force, SQL-ін'єкцій, XSS, RFI та LFI. Було проведено тестування його функціональності, результати якого засвідчили коректну роботу плагіна. Разом з тим системне сканування виявило декілька критичних проблем у налаштуванні безпеки, зокрема активний обліковий запис “admin” із типовим ім'ям та відсутність пароля до бази даних. Це дозволило зробити висновок про важливість не лише встановлення інструментів захисту, а й їхнього правильного налаштування.

		Ковальчук С.В.			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, у результаті виконання кваліфікаційної роботи було не лише успішно реалізовано технічну частину з розгортання CMS Joomla на локальному сервері, але й розроблено базову модель комплексного захисту веб-сайту. Практична реалізація продемонструвала ефективність підходу “оборони в глибину”, що поєднує заходи на рівні сервера, самої CMS і додаткових розширень. Отримані результати підтверджують доцільність використання RSFirewall! як інструмента підвищення безпеки Joomla-сайтів та акцентують увагу на необхідності дотримання сучасних практик кіберзахисту в умовах постійно зростаючих загроз.

.

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		74

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Web Application Security: An Essential Guide. [Електронний ресурс] – Режим доступу: <https://techvify-software.com/web-application-security/>
2. Protect Web Applications. [Електронний ресурс] – Режим доступу: <https://www.fortinet.com/solutions/enterprise-midsize-business/protect-web-apps>
3. What Is The Web Application Security. [Електронний ресурс] – Режим доступу: <https://medium.com/@GuruTechnolabs/what-is-the-web-application-security-5c00636ae6ad>
4. Complete beginner’s guide to web application security. [Електронний ресурс] – Режим доступу: <https://www.invicti.com/blog/web-security/getting-started-web-application-security/>
5. Web Application Security Basics 101: where to start. [Електронний ресурс] – Режим доступу: <https://medium.com/@talhakhali101/devsecops-basics-where-to-start-aa7abee4ac4>
6. Web Application Security: Risks, Technologies, and Best Practices.[Електронний ресурс] – Режим доступу: <https://www.cycognito.com/learn/application-security/web-application-security.php>
7. Безпека веб-сайтів. Загрози безпеці веб-сайтів та поради щодо захисту від шкідливих атак.[Електронний ресурс] – Режим доступу: <https://www.onistudio.com.ua/bezpeka-veb-sajtiv/>
8. Найкращі методи створення безмечних веб-сайтів .[Електронний ресурс] – Режим доступу:
9. Захист веб-додатків.[Електронний ресурс] – Режим доступу: <https://itbiz.ua/statti-ta-obzori/zaxist-veb-dodatktiv-chomu-ce-vazhливо/>
10. Стратегія і основні кроки при розробці web—сайту. [Електронний ресурс] — режим доступу: <http://ruszura.in.ua/neobhidno—znaty/stratehiya—i—osnovni—kroky—pry—rozrobtsi—web—sajta.html>

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				75
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Best Practices to Secure your Joomla Website [Електронний ресурс] — режим доступу: <https://magazine.joomla.org/all-issues/april-2021/best-practices-to-secure-your-joomla-website>
12. Joomla Security Strike Team (JSST). (2024). Joomla! Vulnerable Extensions List and Security Advisories. [Електронний ресурс] – Режим доступу: <https://developer.joomla.org/security.html>
- 13.OWASP Foundation. (2023). OWASP Top Ten Web Application Security Risks – 2021. [Електронний ресурс] – Режим доступу: <https://owasp.org/Top10>
14. Akeeba Ltd. (2024). Admin Tools for Joomla – User Guide & Security Best Practices. [Електронний ресурс]– Режим доступу: <https://www.akeeba.com/products/admin-tools.html>
15. RSJoomla! (2024). RSFirewall! – Advanced Security Plugin for Joomla. [Електронний ресурс] — режим доступу: <https://www.rsjoomla.com>
16. Securitycheck Extensions. (2024). Securitycheck & Securitycheck Pro – Features Overview. [Електронний ресурс] – Режим доступу: <https://securitycheck.protegetuordenador.com>
17. Akeeba Ltd. (2024). Akeeba Backup – Official Documentation. [Електронний ресурс] — режим доступу: <https://www.akeeba.com/documentation.html>
18. CVE Details. (2024). Joomla Security Vulnerabilities [Електронний ресурс] — режим доступу: <https://www.cvedetails.com>
19. National Institute of Standards and Technology (NIST). (2023). National Vulnerability Database (NVD). . [Електронний ресурс] — режим доступу: <https://nvd.nist.gov>
20. Apache Friends. (2024). XAMPP – Free and Open Source Cross-platform Web Server Solution Stack Package. [Електронний ресурс] — режим доступу: <https://www.apachefriends.org>

		Ковальчук С.В			ККІКБ.КР.Б – 125 – 09 – ПЗ	Арк.
		Пірог О.В				76
Змн.	Арк.	№ докум.	Підпис	Дата		