УДК 004.056

МІНІМІЗАЦІЯ РИЗИКУ ЗЛОМУ ВЕБСАЙТУ НА ОСНОВІ CMS JOOMLA: КОМПЛЕКС ПРАКТИЧНИХ ЗАХОДІВ

О. І. Хорошевський[1], І. О. Хорошевська[2]

[1]Харківський національний університет радіоелектроніки, пр. Науки, 14, Харків, 61166, Україна, e-mail: oleksii.khoroshevskyi@nure.ua
[2]Харківський національний економічний університет імені Семена Кузнеця, пр. Науки, 9а, Харків, 61165, Україна, e-mail: iryna.bondar@hneu.net

У статті розглянуто проблему забезпечення безпеки вебсайтів, створених на базі CMS Joomla. Цільовою спрямованістю статті є визначення та систематизація комплексу практичних заходів, спрямованих на мінімізацію ризику злому вебсайту на основі CMS Joomla. У результаті дослідження сформовано систематизований перелік засобів підвищення рівня безпеки вебресурсів, згрупований за шістьма основними категоріями. Для кожної з категорій наведено опис практичних інструментів та технологічних рішень, які дозволяють підвищити стійкість вебресурсів до несанкціонованого доступу та зовнішніх атак.

Матеріали статті інтегруються у процес викладання освітніх компонентів, спрямованих на формування у здобувачів вищої освіти спеціальності «Видавництво та поліграфія» компетентностей у сфері сайтобудування на базі CMS Joomla з урахуванням заходів безпеки вебресурсів. Крім того, результати дослідження можуть слугувати інформаційним і методичним джерелом для практикуючих веброзробників та адміністраторів вебресурсів, які займаються питаннями підвищення рівня безпеки вебсайтів і мінімізації ризиків їхнього злому.

Ключові слова: *безпека вебсайтів, злом вебсайтів, CMS Joomla, уразливості, оновлення розширень, резервне копіювання.*

Постановка проблеми. На даний час вебсайти стали невід'ємним інструментом представлення у мережі Інтернет для організацій, освітніх установ, бізнесу, видавництв, громадських ініціатив, особистих проєктів тощо. Значна частина таких вебресурсів створюється на основі систем керування вмістом (CMS), серед яких однією з найпоширеніших є CMS Joomla. Відкритий програмний код, розвинена екосистема розширень та широке використання цієї платформи зумовлюють як її популярність, так і підвищену увагу з боку зловмисників.

Загалом, проблема забезпечення інформаційної безпеки вебсайтів на базі CMS полягає в тому, що значна частина успішних атак пов'язана не стільки з критичними вразливостями ядра системи, скільки з некоректною конфігурацією, несвоєчасним оновленням, використанням ненадійних розширень, слабкою політикою доступу та недостатнім контролем серверного середовища.

Уразливості можуть призводити до викрадення особистих даних (як-от логіни, паролі, електронні адреси тощо), модифікації або видалення інформації, зміни зовнішнього вигляду вебсторінок, розповсюдження шкідливого програмного забезпечення, блокування вебресурсу чи використання його для фішингових атак тощо [1–3]. Наслідками цього є фінансові втрати, репутаційні ризики та порушення цілісності вебсайтів. Особливої актуальності проблема набуває в умовах зростання кількості автоматизованих атак (brute force, SQL-ін'єкцій, XSS), які масово сканують вебсайти на наявність типових уразливостей. Для CMS із відкритим кодом, зокрема Joomla, ризик підвищується через доступність інформації про структуру системи та типові помилки конфігурації.

Таким чином, постає науково-практична проблема виокремлення комплексу заходів, спрямованих на мінімізацію ризику злому вебсайту на основі CMS Joomla. Необхідність узагальнення практичних заходів та їх наукового осмислення зумовлює доцільність проведення дослідження.

Аналіз останніх досліджень і публікацій, присвячених питанню безпеки вебсайтів показав, що в роботах [1 – 5] приділялась увага опису різних типів атак, як-от SQL-ін'єкції, XSS, DDoS та DoS атаки, brute force атаки, фішинг, Cross-Site Request Forgery, Malware, атаки на відомі вразливість конкретних CMS та вразливості API, Man-in-the-Middle атаки, Zero-day атаки, Backdoor атаки, Clickjacking, Social Engineering атаки, Remote File Inclusion атаки тощо.

В розділі монографії «Технологія розробки вебсайтів на базі CMS Joomla» [6, с. 30] зауважується, що забезпечення безпеки вебсайту включає в себе «захист від шкідливих атак, таких як хакерські атаки, DDoS-атаки та віруси, шляхом встановлення та оновлення програмного та апаратного забезпечення захисту, моніторингу активності на вебсайті та реагування на потенційні загрози».

У роботі [7] автори провели емпіричне дослідження безпеки WordPress та Joomla за допомогою сканерів OWASP ZAP, Vega, Detectify та Skipfish. Для цього були створені тестові вебсайти на обох CMS, які сканувалися з подальшим групуванням результатів за категоріями вразливостей (XSS, SQLi, directory listing тощо) та рівнем загрози (високий, середній, низький). Отримані результати підтвердили наявність типових проблем у стандартних конфігураціях CMS та засвідчили потребу у додаткових заходах захисту.

Авторами роботи [8] проведено аналіз безпеки розширень Joomla з офіційного каталогу з рейтингом топ-10. Дослідження виявило уразливості типу XSS та SQLi, що підтвердило ризик використання розширень як точки входу для атак. Результати сканування 7,797 вебсайтів показали системну проблему – відсутність ізоляції розширень від ядра CMS. Автори даної роботи рекомендують регулярні оновлення, статичний та динамічний аналіз коду, а також застосування політик безпеки на рівні браузера (зокрема CSP).

У статті [9] запропоновано підхід до оцінювання безпеки вебсайтів на WordPress, Joomla та Drupal. Автори підкреслюють, що значна частина вразливостей пов'язана з використанням розширень (модулів, плагінів, шаблонів), застарілих версій CMS та некоректних конфігурацій. Дослідження показує, що навіть базове («статичне»)

сканування вебсайтів на цих платформах може надати зловмисникам суттєвий обсяг інформації для атак. Отримані результати підтверджують необхідність регулярного аудиту конфігурацій, контролю оновлень розширень та моніторингу відомих уразливостей для мінімізації ризиків злому.

В роботі [10] зауважується на важливості встановлення оновлень із офіційного каталогу розширень Joomla та систематичності цього процесу. Підкреслюється, що затримка оновлень є критичним чинником ризику: 42% вебзломів у середовищах CMS з відкритим вихідним кодом відбуваються саме через несвоєчасні оновлення.

У статті [11] описано етапи оновлення вебсайтів на CMS (Joomla, WordPress, Drupal, Opencart), акцентуючи на усуненні вразливостей безпеки, прискоренні завантаження та зменшенні навантаження на хостинг. Зазначено переваги оновлень (виправлення помилок, закриття уразливостей) та ризики (несумісність розширень, втрата функціоналу). Запропоновано систематизовану послідовність з 18 етапів для оновлення ядра та розширень Joomla, що дозволяє мінімізувати ризики виникнення негативних наслідків після оновлення вебсайту.

В роботах [1, 10 – 15] запропоновано заходи, що надають можливість підвищити рівень безпеки вебсайту, серед яких: використання HTTPS, SSL-сертифіката, WAF, контроль доступу користувачів (уникнення загальних груп дозволів, використання двофакторної аутентифікації тощо), оновлення CMS та використовуваних розширень, застосування патчів безпеки протягом 48 годин після випуску оновлень, обмеження доступу до чутливих зон (налаштування часу очікування сеансу, блокування доступу до конфіденційних шляхів, таких як /administrator для неавторизованих IP-адрес й ін.), резервне копіювання тощо.

Наведений аналіз свідчить про значну увагу науковців та практикуючих веброзробників до окремих аспектів безпеки вебсайтів на CMS: типів атак, емпіричного виявлення вразливостей ядра та розширень Joomla, важливості оновлень тощо. Водночас у наявних працях переважно розглядаються або теоретичні моделі загроз, або часткові практичні заходи (HTTPS/WAF/оновлення) без їхньої систематизації. Питання комплексного підходу до мінімізації ризику злому саме для CMS Joomla, не отримало належного відображення. Відсутність узагальнення практичних рекомендацій у єдину методологію з акцентом на специфіку Joomla зумовлює актуальність та практичну цінність цього дослідження.

Мета статті – визначення та систематизація комплексу практичних заходів, спрямованих на мінімізацію ризику злому вебсайту, створеного на базі CMS Joomla.

Виклад основного матеріалу дослідження. В основі систематизованого переліку заходів, що надають можливість підвищити рівень безпеки та мінімізувати вірогідність злому вебсайту покладено багаторічний практичний досвід авторів статті в сфері веброзробки на CMS Joomla та результати власних досліджень [1, 6, 11, 16]. Це дозволило визначити доцільні заходи та систематизувати їх за категоріями: 1) захист передавання та зберігання даних; 2) керування оновленнями та програмним середовищем; 3) аутентифікація та керування обліковими записами; 4) контроль доступу та мережевий захист; 5) захист від автоматизованих атак,

аудит безпеки та антивірусне сканування; 6) резервування та відновлення вебсайту. Розглянемо їх змістовне навантаження.

1. Захист передавання та зберігання даних. Для вебресурсів, побудованих на CMS Joomla, рекомендується забезпечувати роботу як клієнтської, так і адміністративної частини шляхом використання протоколу HTTPS, що передбачає встановлення SSL/TLS-сертифіката на сервері та активацію параметрів примусового HTTPS у глобальних налаштуваннях CMS Joomla [17]. Такий підхід гарантує шифрування даних, які передаються між браузером користувача та компонентами Joomla (форми авторизації, реєстрації, форми зворотного зв'язку тощо), і знижує ризик їх перехоплення. Чутливі дані, що зберігаються у базі даних Joomla, мають бути захищені за рахунок використання надійних алгоритмів хешування паролів та додаткового шифрування полів, які містять персональну інформацію користувачів.

Окремим аспектом захисту даних під час роботи з файловою структурою вебсайту на основі CMS Joomla є використання захищених протоколів передавання файлів. Для адміністрування файлової системи сервера доцільно відмовитися від використання незахищеного протоколу FTP на користь SSH-доступу (SFTP) / FTPS, які забезпечують шифрування каналу передавання даних між клієнтом та сервером і, відповідно, знижують ризик перехоплення облікових даних адміністратора чи модифікації файлів під час їх завантаження [18]. Відмітимо, що SFTP використовує безпечне шифрування як для відправлення, так і для отримання файлів [19].

2. Керування оновленнями та програмним середовищем. Підтримання актуального стану CMS Joomla охоплює систематичне оновлення ядра через вбудований менеджер оновлень, а також своєчасне оновлення встановлених розширень (шаблонів, компонентів, модулів, плагінів) [11]. Невикористовувані розширення слід повністю вилучати з системи, адже їхній застарілий код може слугувати вектором для зловмисних дій. Ті самі принципи застосовуються до серверної інфраструктури (PHP, вебсерверів, бібліотек), яка забезпечує роботу Joomla. Актуальні версії PHP і СУБД (MySQL/MariaDB) є невід'ємною частиною захисту, оскільки саме в них регулярно усуваються виявлені недоліки безпеки.

До важливих аспектів безпеки серверного оточення належить, також, коректне налаштування PHP (відключення небезпечних функцій типу register_globals чи allow_url_fopen, обмеження ресурсів через php.ini), що ускладнює реалізацію типових атак на Joomla.

3. Аутентифікація та керування обліковими записами. Для облікових записів користувачів, зокрема адміністративних, у Joomla необхідно уникати передбачуваних шаблонних конструкцій для паролів, встановлювати політику складності паролів (як-от комбінація великих і малих літер, цифр і спеціальних символів) та обмеження щодо їхньої мінімальної довжини. Так, рекомендованою є довжина пароля не менше 12 символів [20].

Важливим механізмом безпеки є використання двофакторної аутентифікації (2FA), підтримка якої реалізована у ядрі Joomla та може бути увімкнена для облікових записів через систему керування користувачами або спеціальні плагіни 2FA (як-от Two Factor Authentication 2FA for Joomla [21]). Це дозволяє значно знизити

ризик компрометації адміністраторських облікових записів під час атак перебору паролів на вебсторінці та в адміністративній панелі вебсайту.

4. Контроль доступу та мережевий захист. Для вебресурсів на Joomla доцільно застосовувати веб-брандмауери (на рівні хостингу або у вигляді зовнішніх сервісів WAF), які фільтрують HTTP-трафік і блокують типові шаблони шкідливих запитів, характерних для DDoS, DoS, SQL-ін'єкцій, XSS та інших атак на CMS.

Додатково рекомендується обмежувати доступ до адміністративної панелі Joomla: застосовувати обмеження за IP-адресами на рівні вебсерверу чи файлу .htaccess, змінювати типові маршрути доступу до панелі керування, а також надавати права адміністрування лише вузькому колу довірених користувачів. Додатковим локальним засобом захисту є налаштування файлу .htaccess для блокування доступу до конфігураційних файлів, панелі керування та каталогів, а також примусового перенаправлення на HTTPS із фільтрацією підозрілих запитів та забороною на виконання/відкриття всіх PHP-файлів напряму з браузера (за виключенням декількох) (рис. 1).

```
<Filesmatch ".(php)$">
order allow,deny
deny from all
</Filesmatch>

<Files ~ ".(php)$">
Deny from all
</Files>

<Filesmatch "^index.php">
order allow,deny
allow from all
</Filesmatch>

<Filesmatch "^index2.php">
order allow,deny
allow from all
</Filesmatch>

<Filesmatch "^index3.php">
order allow,deny
allow from all
</Filesmatch>
```

Рис. 1. Налаштування .htaccess для підвищення захищеності вебсайту

Використання виділеної IP-адреси для вебсайту на Joomla розглядається як один із заходів мережевого захисту, адже воно мінімізує ризики блокування, пов'язані з іншими вебсайтами на shared IP, та забезпечує більш ефективне впровадження політик обмежень доступу і моніторинг трафіку.

Також зазначимо, що на рівні хостингу доцільно активувати засоби захисту від ботів (rate limiting, Bot Fight Mode, фільтрацію за user-agent), які зменшують навантаження від автоматизованого сканування та спроб брутфорсу на Joomla.

5. Захист від автоматизованих атак, аудит безпеки та антивірусне сканування. Для зменшення впливу автоматизованих атак (ботів), які здійснюють масові спроби авторизації чи реєстрації в Joomla, варто інтегрувати механізми CAPTCHA у форми реєстрації, коментарів, зворотного зв'язку. Сучасні версії

Joomla підтримують інтеграцію з різними сервісами CAPTCHA через плагіни, що дозволяє ускладнити автоматизований перебір облікових даних і зменшити кількість спам-реєстрацій.

Застосування спеціалізованих розширень Joomla для захисту вебсайту від вторгнень і хакерських атак дає змогу реалізувати багаторівневу автентифікацію, фільтрацію шкідливого трафіку та моніторинг безпеки, що істотно знижує ризики несанкціонованого доступу. Прикладами таких розширень є Admin Tools Professional, Securitycheck, RSFirewall! Для застосування цих розширень треба, щоб вони були суміжні з версією Joomla, на який розроблено вебсайт.

Так, використання RSFirewall! дозволяє не лише зменшити ризики зовнішніх атак, але й забезпечити постійний моніторинг стану системи та оперативне реагування на інциденти. Його функціонал охоплює багаторівневий захист від зовнішніх атак, моніторинг системних змін, контроль доступу до адміністративної панелі та підтримку цілісності даних [22]. Приклад-фрагмент процесу перевірки вебсайту засобами RSFirewall! подано на рис. 2.

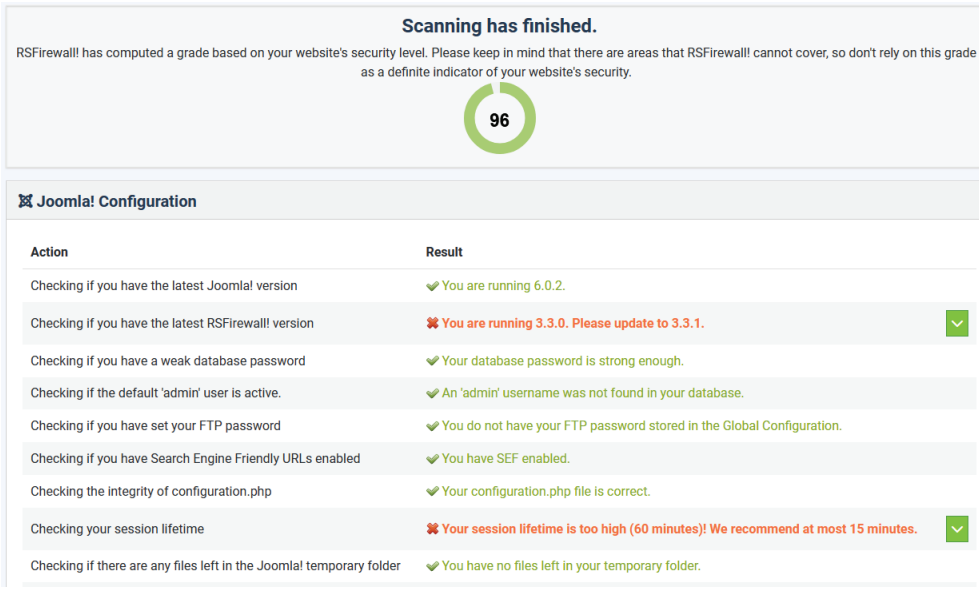

Рис. 2. Приклад-фрагмент перевірки вебсайту засобами RSFirewall!

Під час перевірки оцінюється відповідність ключовим критеріям безпеки: актуальність версії Joomla та RSFirewall!, надійність пароля для бази даних, наявність користувача зі стандартним ім`ям облікового запису, наявність пароля від FTP, збереженого у налаштуваннях Joomla, активація SEF-адрес та ін., а також застосування спеціальних налаштувань сервера, що мінімізують ризик злому тощо. На основі результатів аналізу приймається рішення щодо необхідних виправлень.

Для вебсайтів на основі Joomla важливо організувати збір і аналіз журналів подій як на рівні CMS, так і на рівні вебсервера та розширень безпеки. Це дає змогу своєчасно виявляти ознаки злому – спроби входу до адміністративної частини,

підозрілі запити до системних файлів чи зміни у файловій структурі. Додатково в RSFirewall! можна налаштувати пароль адміністративної частини (Backend Password) із секретним словом: у разі входу з іншого браузера чи IP-адреси без його введення доступ блокується, що значно знижує ризик несанкціонованого доступу. Приклад налаштування пароля адміністративної частини з використанням секретного слова та пароля наведено на рис. 3.

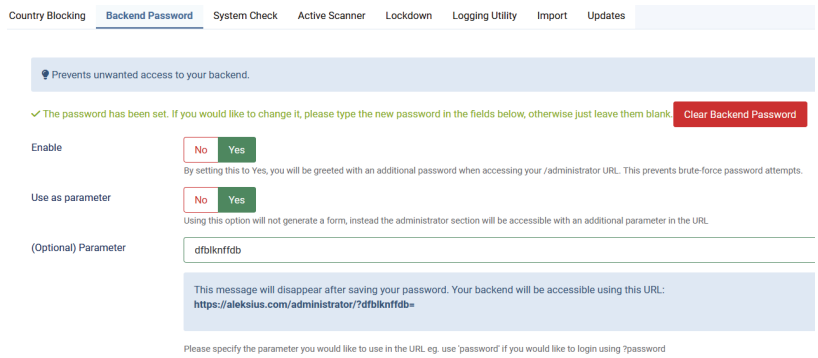

Рис. 3. Приклад налаштування пароля адміністративної частини

Важливо зазначити, що аналіз вмісту системного журналу (System Logs) RSFirewall! дозволяє визначити, з яких IP-адрес і коли здійснювалися атаки, а також їхній тип. Наприклад, на рис. 4 наведено подію з описом «Remote file inclusion attempted», що свідчить про зафіксовану спробу атаки типу Remote File Inclusion.

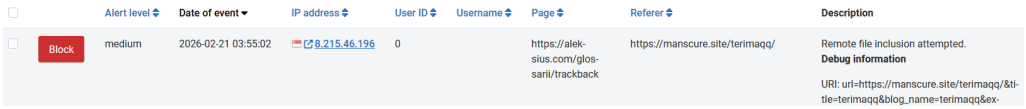

Рис. 4. Приклад виявлення спроби включення зовнішнього файлу

Така ін’єкція полягає у включенні зовнішнього файлу у вебсторінку через вразливість у параметрах URL, що може надати зловмиснику можливість виконання шкідливого коду.

У разі виявлення протиправних дій IP-адресу можна додати до списку блокувань. Таким чином, аналіз логів забезпечує не лише фіксацію факту підозрілої активності, а і уточнення її часових рамок та напряму атаки, що є важливим для подальшого вдосконалення політики безпеки вебсайту.

Безумовно, не можна оминати важливість питання регулярного зовнішнього сканування Joomla-сайту спеціалізованими сервісами на наявність шкідливого коду, SEO-спаму, black-list статусу тощо. Серед корисних сервісів для зовнішнього сканування зазначимо наступні: Sucuri SiteCheck, HackerTarget Joomla Security Scanner та Website Antivirus Scanner.

З метою внутрішнього (серверного) сканування файлів і бази даних на предмет шкідливих скриптів, бекдорів, вебшелів, з використанням антивірусів хостингрівня

або розширень доречно використовувати такі сервіси: Website Antivirus Scanner for Joomla, Sucuri ServerSide Scanner, ClamAV, Imunify360.

6. Резервування та відновлення вебсайту. Для мінімізації наслідків можливих атак на вебсайти необхідно організувати регулярне резервне копіювання вебресурсу на окремому сервері або у хмарному сховищі. У практиці супроводу Joomla широко застосовуються спеціалізовані розширення, зокрема компонент Akeeba Backup [23], який дозволяє створювати повні резервні копії вебсайту, бази даних та важливих директорій (зображення, вкладення). Використання копій забезпечує швидке відновлення вебсайту до стану на момент створення копії. Приклад налаштувань Akeeba Backup (рис. 5) демонструє можливість збереження бази даних не лише на хостингу, а й у зовнішньому хмарному сховищі Google Drive.

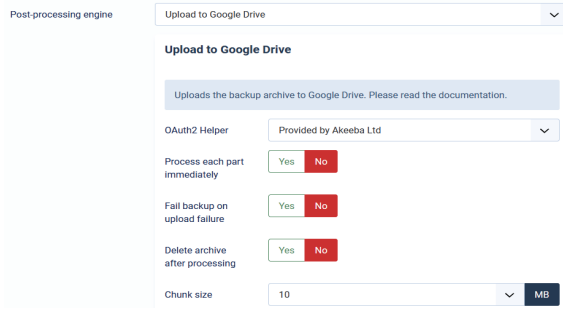

Рис. 5. Налаштування Akeeba Backup для завантаження бази даних на Google Drive

Наявність актуальних резервних копій забезпечує можливість оперативного відновлення вебсайту у разі злому, технічних збоїв, людських помилок тощо.

Визначений комплекс заходів застосовується у навчальному процесі за спеціальністю «Видавництво та поліграфія» в Харківському національному університеті радіоелектроніки (освітні компоненти: «Web-технології», «Основи Web-технологій», «Проєктування та розробка web-систем», «Web-аналітика та пошукова оптимізація»), а також у Харківському національному економічному університеті імені Семена Кузнеця (освітній компонент «Системи керування вмістом (CMS)»). Це забезпечує опанування здобувачами знань і навичок щодо підвищення безпеки вебсайтів, створюваних на базі CMS Joomla.

Висновки. У результаті дослідження визначено та систематизовано комплекс практичних заходів, спрямованих на мінімізацію ризику злому вебсайтів на основі CMS Joomla. Запропоновані заходи згруповано за шістьма основними категоріями: захист передавання та зберігання даних; керування оновленнями та програмним середовищем; аутентифікація та керування обліковими записами; контроль доступу та мережевий захист; захист від автоматизованих атак, аудит безпеки та антивірусне сканування; резервування та відновлення вебсайту. Наведено опис вмісту заходів, що застосовуються у межах кожної з категорій для здійснення процесу мінімізації ризику злому вебсайту, створеного засобами CMS Joomla.

Наведені рекомендації інтегруються у навчальний процес за спеціальністю «Видавництво та поліграфія» в Харківському національному університеті радіо-

електроніки та Харківському національному економічному університеті імені Семена Кузнеця. Вони використовуються у межах освітніх компонентів, присвячених сайтобудуванню на CMS Joomla з акцентом на реалізації заходів безпеки вебресурсів.

Перспективами подальших досліджень є розроблення методики комплексного оцінювання рівня безпеки вебсайтів, створених на базі CMS Joomla, з урахуванням визначених у дослідженні груп заходів захисту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хорошевський О. Відеолекція «Створення вебсайту» (частина 6). Захист від злому. URL: https://aleksius.com/uk/cms-joomla/zakhyst-vid-zlomu.
2. Fedorenko O.H., Velychko S.V., Kaidan Y.V. Investigating vulnerabilities of personal data on financial websites. *CEUR Workshop Proceedings*. 2024. Vol-3917. P. 451–458.
3. Albalawi N., Alamrani N., Aloufi R., Albalawi M., Aljaedi A., Alharbi A.R. The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities. *Electronics*. 2023. Vol. 12. Issue 12, 2664. https://doi.org/10.3390/electronics12122664.
4. Kirsten S. Cross Site Scripting (XSS). URL: https://owasp.org/www-community/attacks/xss/.
5. Alla Rud. Безпека CMS: як захистити свій сайт від загроз. URL: https://hyperhost.ua/info/uk/bezpeka-cms-yak-zaxistiti-svii-sait-vid-zagroz.
6. Хорошевський О. І. Технологія розробки вебсайтів на базі CMS Joomla. *Поліграфічні, мультимедійні та web-технології. Інновації та розвиток: монографія*. Харків: ТОВ Друкарня Мадрид, 2024. С. 5–33.
7. Zamościński, P., Kozieł, G. Analysis of Security CMS Platforms by Vulnerability Scanners. *J. Comput. Sci. Inst*. 2020. Vol. 16. P. 261-268. https://doi.org/10.35784/jcsi.2020.
8. Niemietz M., Korth M., Mainka C., Somorovsky, J. Over 100 Bugs in a Row: Security Analysis of the Top-Rated Joomla Extensions. 2021. ArXiv. https://doi.org/10.48550/arXiv.2102.03131.
9. Bose S., Narayanan A. K. Security Analysis of CMS based Websites through CMSPY. *ICRRD Journal*. 2023. Vol. 4. Issue 4. P. 162-174. URL: https://icrrd.com/public/media/05122023025501-CMS-Security-Analysis.pdf.
10. Essential Joomla Security Protocols - Best Practices Every Developer Should Follow. URL: https://moldstud.com/articles/p-essential-joomla-security-protocols-best-practices-every-developer-should-follow.
11. Хорошевський, О. І., Хорошевська, І. О., Назаров, Д. Л. Етапи процесу оновлення вебсайту, створеного засобами CMS. *Таврійський науковий вісник. Серія: Технічні науки*. 2025. №3. С. 167-175. https://doi.org/10.32782/tnv-tech.2025.3.18.
12. CMS Security Best Practices 2026: Protect Your Website from Vulnerabilities. URL: https://uzera.com/blog/cms-security-best-practices-saas-teams.
13. Притула Б. Як забезпечити безпеку сайту: ТОП-7 перевірених способів. URL: https://brander.ua/blog/yak-zabezpechyty-bezpeku-saytu-top-7-perevirenykh-sposobiv.
14. Найкращі методи створення безпечних веб-сайтів. URL: https://whileweb.com/uk/blog/najkrashi-metodi-stvorennya-bezpechnih-veb-sajtiv/.
15. Топ-10 правил безпеки сайту. URL: https://realhost.pro/blog/top-10-web-security.

16. Хорошевський О. І., Хорошевська І. О. Визначення комплексу показників, що впливають на якість вебсайту. *Таврійський науковий вісник. Серія: Технічні науки*. 2025. Т. 2. С. 216-224. https://doi.org/10.32782/tnv-tech.2025.2.22.
17. Joomla! Documentation: Enabling HTTPS on your site. URL: https://docs.joomla.org/Enabling_HTTPS_on_your_site.
18. What Is Secure File Transfer? Understanding SSH, FTPS, and SFTP. URL: https://hosting.com/blog/what-is-secure-file-transfer/.
19. Brian McHugh. Secure File Transfers: Best Practices, Protocols And Tools. URL: https://www.advsyscon.com/blog/secure-file-transfers/.
20. Безпечне поводження з паролями. Двоетапна перевірка. URL: https://chatovi.online/articles/Bezpechne%20povodzhennia%20z%20paroliami.%20Dvoetapna%20perevirka.
21. Joomla! Extensions Directory: Two Factor Authentication 2FA for Joomla. URL: https://extensions.joomla.org/extension/two-factor-authentication-2fa-for-joomla/.
22. Joomla! Extensions Directory: RSFirewall! URL: https://extensions.joomla.org/extension/rsfirewall.
23. Joomla! Extensions Directory: Akeeba Backup. URL: https://extensions.joomla.org/extension/akeeba-backup/.

REFERENCES

1. Khoroshevskyi O. Videolektsiia «Stvorennia vebsaitu» (chastyna 6). Zakhyst vid zlomu. Retrieved from: https://aleksius.com/uk/cms-joomla/zakhyst-vid-zlomu (in Ukrainian).
2. Fedorenko O.H., Velychko S.V., Kaidan Y.V. (2024). Investigating vulnerabilities of personal data on financial websites. *CEUR Workshop Proceedings*, 3917, 451–458 (in English).
3. Albalawi, N., Alamrani, N., Aloufi, R., Albalawi, M., Aljaedi, A., & Alharbi, A. R. (2023). The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities. *Electronics*, 12(12), 2664. https://doi.org/10.3390/electronics12122664 (in English).
4. Kirsten S. Cross Site Scripting (XSS). Retrieved from: https://owasp.org/www-community/attacks/xss/ (in English).
5. Alla Rud. Bezpeka CMS: yak zakhystyty svii sait vid zahroz. Retrieved from: https://hyperhost.ua/info/uk/bezpeka-cms-yak-zaxistiti-svii-sait-vid-zagroz (in Ukrainian).
6. Khoroshevskyi O. I. Tekhnolohiia rozrobky vebsaitiv na bazi CMS Joomla [Website development technology based on CMS Joomla]. *Polihrafichni, multymediini ta web-tekhnolohii. Innovatsii ta rozvytok : monohrafiia – Printing, multimedia and web technologies. Innovations and development: a monograph*. Kharkiv: Printing House Madrid, 2024, pp. 5–33 (in Ukrainian).
7. Zamościński, P., & Kozieł, G. (2020). Analysis of security CMS platforms by vulnerability scanners. *Journal of Computer Sciences Institute*, 16, 261–268. https://doi.org/10.35784/jcsi.2020 (in Polish).
8. Niemietz, M., Korth, M., Mainka, C., & Somorovsky, J. (2021). Over 100 Bugs in a Row: Security Analysis of the Top-Rated Joomla Extensions. ArXiv. https://doi.org/10.48550/arXiv.2102.03131 (in English).
9. Bose S., Narayanan A. K. (2023). Security Analysis of CMS based Websites through CMSPY. *ICRRD Journal*, 4(4), 162-174. Retrieved from: https://icrrd.com/public/media/05122023025501-CMS-Security-Analysis.pdf (in English).

10. Essential Joomla Security Protocols - Best Practices Every Developer Should Follow. Retrieved from: https://moldstud.com/articles/p-essential-joomla-security-protocols-best-practices-every-developer-should-follow (in English).
11. Khoroshevskyi, O. I., Khoroshevska, I. O. & Nazarov, D. L. (2025). Etapy protsesu onovlennia vebsaitu, stvorenoho zasobamy CMS [Stages of the process of updating a website created with CMS tools]. *Taurida Scientific Herald. Series: Technical Sciences*, (3), 167-175. https://doi.org/10.32782/tnv-tech.2025.3.18 (in Ukrainian).
12. CMS Security Best Practices 2026: Protect Your Website from Vulnerabilities. Retrieved from: https://uzera.com/blog/cms-security-best-practices-saas-teams (in English).
13. Prytula B. Yak zabezpechyty bezpeku saitu: TOP-7 perevirenykh sposobiv. Retrieved from: https://brander.ua/blog/yak-zabezpechyty-bezpeku-saytu-top-7-perevirenykh-sposobiv (in Ukrainian).
14. Naikrashchi metody stvorennia bezpechnykh veb-saitiv. Retrieved from: https://whileweb.com/uk/blog/najkrashi-metodi-stvorennya-bezpechnih-veb-sajtiv (in Ukrainian).
15. Top-10 pravyl bezpeky saitu. Retrieved from: https://realhost.pro/blog/top-10-web-security (in Ukrainian).
16. Khoroshevskyi, O. I., & Khoroshevska, I. O. (2025). Vyznachennia kompleksu pokaznykiv, shcho vplyvaiut na yakist vebsaitu [Determining a set of indicators that affect the quality of the website]. *Taurida Scientific Herald. Series: Technical Sciences,* (2), 216-224. https://doi.org/10.32782/tnv-tech.2025.2.22 (in Ukrainian).
17. Joomla! Documentation: Enabling HTTPS on your site. Retrieved from: https://docs.joomla.org/Enabling_HTTPS_on_your_site (in English).
18. What Is Secure File Transfer? Understanding SSH, FTPS, and SFTP. Retrieved from: https://hosting.com/blog/what-is-secure-file-transfer (in English).
19. Brian McHugh. Secure File Transfers: Best Practices, Protocols And Tools. Retrieved from: https://www.advsyscon.com/blog/secure-file-transfers/ (in English).
20. Bezpechne povodzhennia z paroliamy. Dvoetapna perevirka. Retrieved from: https://chatovi.online/articles/Bezpechne%20povodzhennia%20z%20paroliami.%20Dvoetapna%20perevirka (in Ukrainian).
21. Joomla! Extensions Directory: Two Factor Authentication 2FA for Joomla. Retrieved from: https://extensions.joomla.org/extension/two-factor-authentication-2fa-for-joomla/ (in English).
22. Joomla! Extensions Directory: RSFirewall! Retrieved from: https://extensions.joomla.org/extension/rsfirewall (in English).
23. Joomla! Extensions Directory: Akeeba Backup. Retrieved from: https://extensions.joomla.org/extension/akeeba-backup/ (in English).

doi: 10.32403/1998-6912-2026-1-72-73-84

MINIMISING THE LIKELIHOOD OF WEBSITE HACKING: PRACTICAL MEASURES BASED ON CMS JOOMLA

O. I. Khoroshevskyi[1], I. O. Khoroshevska[2]

[1]*Kharkiv National University of Radio Electronics,*
Nauky Ave. 14, Kharkiv, 61166, Ukraine,e-mail: oleksii.khoroshevskyi@nure.ua
[2]*Simon Kuznets Kharkiv National University of Economics,*
Nauky Ave. 9a, Kharkiv, 61165, Ukraine,e-mail: iryna.bondar@hneu.net

The article discusses the problem of ensuring the security of websites built on content management systems, in particular, the CMS Joomla. The study's relevance stems from the growing number of cyberattacks on web resources and the prevalence of automated website scanning to identify common vulnerabilities. It is noted that a significant part of successful attacks is associated not only with flaws in CMS software code but also with incorrect system configuration, outdated software versions, unreliable extensions, and insufficient control over the server environment.

The paper analyses scientific publications on the security issues of CMS-based websites, enabling identification of the main threat types and approaches to their minimisation. It has been established that most studies focus on individual aspects of web resource protection, whereas systematising practical security measures for the CMS Joomla requires further generalisation.

The purpose of this article is to identify and systematise a set of practical measures to minimise the likelihood of hacking a website running the CMS Joomla. As a result of the study, a systematic list of measures to improve the security of web resources has been compiled, grouped into six main categories: protection of data transmission and storage; management of updates and the software environment; authentication and account management; access control and network protection; protection against automated attacks, security auditing and antivirus scanning; website backup and recovery. For each category, there is a description of practical tools and technological solutions that can increase the resilience of web resources to unauthorised access and external attacks.

The article presents materials integrated into the teaching process of educational components aimed at developing the competencies of higher education students for a degree in "Publishing and Printing" in website development using the CMS Joomla, taking into account web resource security measures. In addition, the study's results can serve as a source of information and methodology for practising web developers and web resource administrators involved in improving website security and minimising the risk of hacking.

Keywords: *website security, website hacking, CMS Joomla, vulnerabilities, update extensions, backup.*

Стаття надійшла до редакції 18.03.2026.
Submitted: 18.03.2026.
Прийнято до друку: 14.04.2026.
Accepted: 14.04.2026.
Опубліковано: 30.05.2026.
Published: 30.05.2026.