УДК 004.056

МОДЕЛЬ ДАНИХ ДЛЯ УДОСКОНАЛЕННЯ КІБЕРБЕЗПЕКИ CONTENT MANAGEMENT SYSTEM

DOI 10.36994/2788-5518-2021-02-02-15

Гнатюк В.О., *к.т.н., доц. Національний авіаційний університет, Київ, Україна, viktorgnatyuk@ukr.net.*
Зозуля С.В., *Національний авіаційний університет, Київ, Україна, sergeyzoz@ukr.net*

Анотація. *Сьогодні жодна сучасна компанія, установа чи підприємство не може існувати без використання у своїй діяльності інформаційних технологій, зокрема веб-сайтів. Головне призначення веб-сайту полягає в тому, щоб надати інформацію користувачам про цю компанію, установу чи підприємство, про переваги її перед конкурентами, а також про товари чи послуги, які нею пропонуються. Проте, використовуючи веб-сайти важливо подбати про забезпечення необхідного рівня кібербезпеки, оскільки не виконання цього аспекту може призвести до виникнення кіберінцидентів, що в свою чергу може призвести до великих фінансових та іміджевих втрат, тому розробка і дослідження нових ефективних методів удосконалення кібербезпеки веб-сайтів є актуальною науковою задачею. Оскільки сучасні веб-сайти розробляються і підтримуються з використанням систем керування контентом (CMS), мета роботи полягає у тому, щоб унеможливити реалізацію зловмисниками кіберінцидентів в CMS, для досягнення якої, розроблено модель даних для удосконалення кібербезпеки CMS, яка за рахунок визначення видів вразливостей для CMS та підвищення рівня кібербезпеки CMS, дозволяє ідентифікувати можливі види вразливостей для CMS та виконавши превентивні дії підвищити рівень кібербезпеки CMS. Розроблена модель даних та сформовані на її основі засоби будуть корисними, насамперед, для адміністраторів веб-сайтів, а також для фахівців з кібербезпеки у складі команд реагування на кіберінциденти типу CERT/CSIRT на які покладаються обов'язки щодо захисту ІТС в межах компаній, установ чи підприємств.*

Ключові слова: *Content Management System, кіберінцидент, WordPress, веб-сайт, плагін.*

MODEL OF DATA FOR IMPROVING CYBERSECURITY CONTENT MANAGEMENT SYSTEM

Viktor Gnatyuk, *Ph.D., Ass. Prof. National Aviation University, Kyiv, Ukraine. viktorgnatyuk@ukr.net*
Sergey Zozulja. *National Aviation University, Kyiv, Ukraine. sergeyzoz@ukr.net.*

© Infocommunication and computer technologies, № 2 (02), 2021
© Гнатюк В.О., Зозуля С.В.

Abstract. *Today, every company, institution or enterprise uses information technology in its activities, including websites. The main purpose of the website is to provide users with information about this company, institution or enterprise, its advantages over competitors, as well as the goods or services it offers. However, when using websites, it is important to ensure the required level of cybersecurity, as failure to comply with this aspect can lead to cyberincidents, which in turn can lead to large financial and image losses, so the development and research of new effective methods to improve cybersecurity of websites is an urgent scientific task. As modern websites are developed and maintained using content management systems (CMS), the aim of the work is to prevent the implementation of cyberincidents by attackers in the CMS, to achieve which, developed a data model to improve cybersecurity CMS, which by identifying types of vulnerabilities for CMS and increase the level of cyber security CMS, allows you to identify possible types of vulnerabilities for CMS and perform preventive actions to increase the level of cyber security CMS. The developed data model and the tools formed on its basis will be useful, first of all, for webmasters, and also for cybersecurity experts as a part of teams of response to cyber incidents like CERT / CSIRT who are responsible for ITS security within the companies, establishments or enterprises.*

Keywords: *Content Management System, cyber incident, WordPress, website, plugin.*

Вступ

Сьогодні жодна сучасна компанія, установа чи підприємство не може існувати без використання у своїй діяльності інформаційних технологій, зокрема веб-сайтів. Головне призначення веб-сайту полягає в тому, щоб надати інформацію користувачам про цю компанію, установу чи підприємство, про переваги її перед конкурентами, а також про товари чи послуги, які нею пропонуються. Проте, використовуючи веб-сайти важливо подбати про забезпечення необхідного рівня кібербезпеки, оскільки не виконання цього аспекту може призвести до виникнення кіберінцидентів [1], що в свою чергу може призвести до великих фінансових та іміджевих втрат, тому розробка і дослідження нових ефективних методів удосконалення кібербезпеки веб-сайтів є актуальною науковою задачею.

Наразі на ринку представлено велику кількість рішень для розробки та підтрики веб-сайтів. Задля спрощення процесу організації веб-сайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах використовуються системи керування контентом (Content Management System, CMS). Виходячи зі статистики частки ринку [2, 3], найпопулярнішою системою управління контентом є WordPress, яка використовується 40,4% усіх веб-сайтів в Інтернеті, за якою слідує Shopify і Joomla. Згідно дослідження компанії iTrack [4] за березень 2021 року всього було опитано 4960632 доменів та побудовано загальний рейтинг CMS (рис. 1).

Наразі існують сотні, а може, навіть й тисячі доступних CMS, завдяки їх функціональності ці системи можна використовувати в різних компаніях. Перші CMS були розроблені у великих корпораціях для організації роботи з документацією. У 1995-му від компанії CNET відокремилася окрема компанія Vignette, яка започаткувала ринок для комерційних CMS. З часом діапазон

продукції розширювався і дедалі більше інтегрувався у сучасні мережеві рішення аж до популярних веб-порталів. Багато сучасних CMS поширюються як безкоштовні і легкі у встановленні (інсталяції) програми, які розробляються під ліцензією GNU/GPL групами ентузіастів. Системи управління веб-сайтом часто розраховані на роботу у певному програмному середовищі. Наприклад, система MediaWiki, під управлінням якої працює Вікіпедія, написана мовою програмування PHP і зберігає вміст і налаштування у базі даних типу MySQL або PostgreSQL; тому для її роботи потрібно, щоб на сервері, де вона розміщена, були встановлені веб-сервер (Apache, IIS чи інший), підтримка PHP та системи керування базами даних MySQL або PostgreSQL, а також, в разі необхідності, додаткові програми для обробки зображень чи математичних формул. Такі вимоги є досить типовими для відкритих CMS. Існує два типи встановлення CMS: локальна та хмарна. Локальне встановлення означає, що програмне забезпечення CMS можна встановити на сервері. До такого підходу зазвичай вдаються підприємства, які хочуть гнучкості у створенні. Відомі CMS, які можна встановити локально, - це Wordpress.org, Drupal , Joomla , ModX та інші. Хмарна CMS розміщена у середовищі постачальника. За такого підходу програмне забезпечення CMS не може бути змінено для клієнта. Прикладами відомих хмарних CMS є SquareSpace, Wordpress.com та WIX.

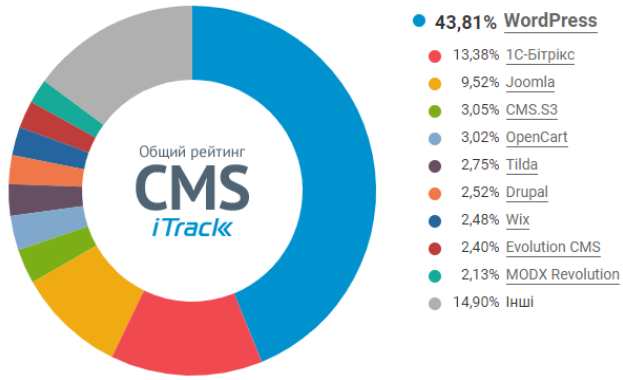

Рис. 1. Загальний рейтинг CMS

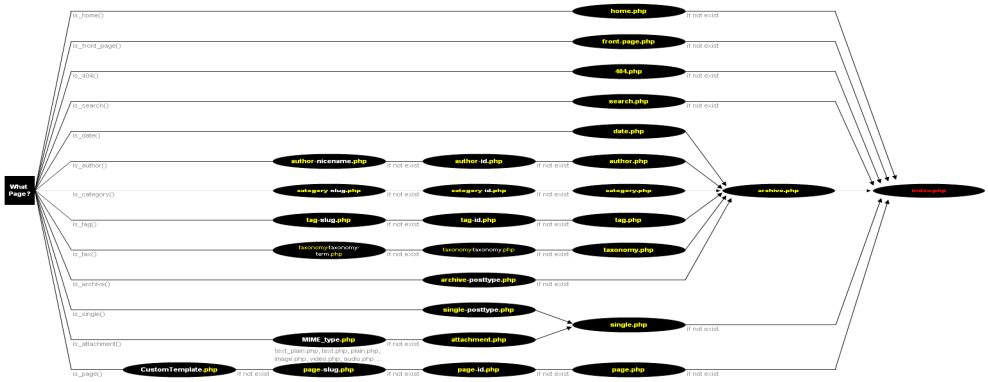

Рис. 2. Структура теми Wordpress

Структура теми CMS Wordpress представлена на рисунку 2, а список основних файлів шаблонів CMS Wordpress включає обов’язкові файли та додаткові файли (табл. 2). Використовуючи CMS Wordpress важливо подбати про забезпечення необхідного рівня кібербезпеки, оскільки не виконання цього аспекту може нести з собою великі фінансові та іміджеві втрати.

Таблиця 1.
Список основних файлів шаблонів Wordpress

Ім'я файлу	Опис	Обов'язковий
style.css	головна таблиця стилів; основна інформація про шаблон	Так
index.php	головна сторінка шаблона	Так
functions.php	набір функцій теми Wordpress	Ні
header.php	шаблон «шапки» сайту; використовується при запиті	Ні
footer.php	шаблон підвалу сайту; використовується при запиті	Ні
sidebar.php	шаблон бокової колонки сайту; використовується при запиті	Ні
comments.php	шаблон коментарів; якщо файлу не існує, використовується шаблон із теми за замовчуванням	Ні
single.php	шаблон сторінки з єдиним записом; якщо файлу не існує, використовується index.php	Ні
page.php	шаблон окремої сторінки (для записів типу «Page»)	Ні
category.php	шаблон категорії; використовується при запиті категорії	Ні
author.php	шаблон автора блогу; використовується при запиті автора блогу	Ні
date.php	шаблон виводу дати; використовується при запиті виводу дати та часу	Ні
archive.php	шаблон виводу архіву блогу	Ні
search.php	шаблон виводу пошуку	Ні
404.php	шаблон для виводу повідомлення помилки 404	Ні

У разі використання CMS Wordpress варто подумати про його безпеку, оскільки за статистикою, більшість зломів CMS Wordpress сайтів можна було б уникнути, якби їх власники використовували хоча б мінімальні рекомендації з безпеки CMS Wordpress. Тому, мета даної роботи полягає у тому, щоб унеможливити реалізацію зловмисниками кіберінцидентів в CMS, для досягнення якої, розроблено модель даних для удосконалення кібербезпеки CMS, яка за рахунок визначення видів вразливостей для CMS та підвищення рівня кібербезпеки CMS, дозволяє ідентифікувати можливі види вразливостей для CMS та виконавши превентивні дії підвищити рівень кібербезпеки CMS.

Визначення видів вразливостей для CMS

Задамо множину видів вразливостей V, які існують під час функціонування CMS:

$$V = \left\{ \bigcup_{i=1}^{n} V_i \right\} = \{V_1, V_2, \dots, V_n\}, \left(i = \overline{1,n}\right), \quad (1)$$

де n – кількість можливих видів вразливостей.

Наприклад, в результаті проведення аналізу існуючих видів вразливостей [5-8] сформуємо таблицю для CMS Wordpress табл. 2.

Таблиця 2.
Види вразливосткй для CMS Wordpress

№	Код	Назва	Опис
1	OS	Використання застарілого ПЗ	У разі використання застарілої версії CMS Wordpress, плагінів або тем, є ймовірність, що це ПЗ містить відомі уразливості, опис яких знаходиться у відкритому доступі. Хакботи аналізують тисячі вебсайтів в годину і визначають версії використовуваного софта. Якщо у відповідях знаходиться застаріла версія ПЗ з відомою вразливістю, бот зламує такий сайт за вже відомою методикою. Це найпопулярніший метод злому сайтів. Оновлення софта - це не тільки новий функціонал або виправлення багів, але і виправлення безпеки в знайдені вразливості. Остання версія ПЗ не містить відомих вразливостей. Оновлюйте CMS Wordpress, плагіни і теми вручну або автоматично. Це гарантує, що ви отримаєте останні виправлення безпеки, які захистять вебсайт від вразливостей, до того ж автоматичне оновлення скорочує час, який витрачається на обслуговування вебсайту.

№	Код	Назва	Опис
2	BFA	Логін і пароль (brute force attack)	Слабкий логін і пароль - другий за популярністю спосіб злому сайтів. Спочатку хакери заходять на сторінку авторизації, тому що тут знаходиться найпростіший спосіб потрапити в адміністративну панель сайту. У разі використання стандартного логіну Admin, Administrator, Root, ім'я домена і т.д., або той же логін, що і ім'я користувача, який можна знайти в мета-тегах автора на сторінках або постах, то хакери вже знають половину інформації для входу на вебсайт. Друга половина - підібрати пароль. В мережі Інтернет є списки найпопулярніших паролів. Наприклад, існує список з 10.000 найпопулярніших паролів містить 91% всіх паролів, якими користуються Інтернет-користувачі. За статистикою, на CMS Wordpress працює більше 41% всіх вебсайтів в Інтернеті. З огляду на таку кількість сайтів і вже готові логіни і паролі, хакери створюють ботів, які обходять сайти і пробують підібрати правильну комбінацію логіна і пароля, це називається brute force attack, тобто атака грубою силою, або атака методом перебору паролів. Крім сторінки авторизації, хакбот може спробувати підібрати правильну комбінацію через шлюз xmlrpc.php, або «зламати» пошту і змінити пароль через сторінку відновлення пароля. Використовуйте унікальний складний пароль, що складається з цифр, букв і символів довжиною не менше 12 символів. Складні логіни і паролі - не єдине, що ви можете зробити для посилення безпеки входу в адміністративну панель. Варто також обмежити кількість невдалих спроб входу, перенести сторінку входу за стандартною адресою ...wp-login.php на іншу, застосувати двох-факторну авторизацію тощо.
3	H	Вразливість хостингу	Хакери зламують не тільки сайти, але і хостинги. Якщо при виборі хостингу орієнтуватися тільки на ціну, то в підсумку це може обійтися дорожче через злом хостингу, а

№	Код	Назва	Опис
			потім вебсайту. Більшість хостингів досить безпечні, але деякі хостинги приділяють недостатньо уваги безпеці, наприклад, не поділяють облікові записи користувачів. Хостинг повинен постійно оновлювати апаратне та програмне забезпечення, застосовувати останні виправлення безпеки і стежити за захистом сайтів і файлової системи. Обирати хостинг варто з високим рейтингом і великою кількістю позитивних відгуків. Хороший хостинг - це основа продуктивності і безпеки сайту.
4	HP	Вразливість PHP	PHP вразливість - це великий розділ різних способів «злому», тому що PHP - це основна технологія, на якій працює CMS Wordpress. Згідно з дослідженням WP WhiteSecurity, 54% вразливостей виявлено в плагінах CMS Wordpress, 31,5% вразливостей в ядрі CMS Wordpress і 14,3% вразливостей - в темах CMS Wordpress. Хороший розробник оперативно усуває знайдену уразливість в своєму ПЗ. Видаліть всі невикористовувані плагіни і теми, і не встановлюйте занедбаний своїми розробниками ПЗ. Якщо плагін або тема давно не оновлювалися, це не означає, що вони занедбані. Можливо, вони працюють як потрібно, і оновляться, коли буде потрібно сумісність з останньою версією CMS Wordpress або PHP.
5	SUS	Використання ПЗ з ненадійних джерел	Завжди використовуйте ПЗ тільки з перевірених джерел. Встановлюйте ПЗ з офіційного сховища CMS Wordpress, добре відомих комерційних репозитаріїв або з сайту розробника. Не встановлюйте «зламані» («nulled») версії тем і плагінів, тому що вони можуть містити «шкідливий» код. Навіть якщо використовується кращий захист на вебсайті, «nulled» версія теми або плагіна може привести до «злому» вебсайту.
6	SSL	SSL підключення	Коли відвідувач переходить на вебсайт, починається підключення між його пристроєм і сервером, де розташований веб-сайт, після

№	Код	Назва	Опис
			цього встановлюється з'єднання і інформація проходить кілька вузлів мережі перед доставкою в пункт призначення. Коли на вебсайті авторизується або робить покупку відвідувач, він передає дані про свій рахунок або банківську карту через це з'єднання, яке за замовчуванням не захищене. Тобто будь-який, хто може переглядати інформацію, може отримати її в готовому незахищеному вигляді. Щоб зашифрувати цю інформацію, потрібно використовувати захищене SSL підключення. Більшість хостингів пропонує послугу додавання платного чи безкоштовного SSL-сертифіката на сайт. З 2017 року встановлений SSL сертифікат на сайті став одним з факторів ранжування Google. Встановіть SSL сертифікат, це посилить безпеку вебсайту.

Отже, використовуючи вираз (1) та дані з табл.2, при $n = 6$ отримаємо:

$$V_{CMS\ Wordpress} = \left\{\bigcup_{i=1}^{6} V_i\right\} = \{V_1, V_2, V_3, V_4, V_5, V_6\} = \\ = \{V_{OS}, V_{BFA}, V_H, V_{HP}, V_{SUS}, V_{SSL}\} = \{OS, BFA, H, HP, SUS, SSL\}, \quad (2)$$

де V_1= V_{OS}= OS, V_2= V_{BFA}= BFA,…, V_6= V_{SSL}= SSL – види вразливостей для CMS Wordpress.

Підвищення рівня кібербезпеки CMS

Задамо множину дій A для підвищення рівня кібербезпеки CMS:

$$A = \left\{\bigcup_{j=1}^{m} A_j\right\} = \{A_1, A_2, \dots, A_m\}, \quad (3)$$

де $A_j \subseteq A$, $\left(j = \overline{1,m}\right)$, m – кількість дій для підвищення рівня рівня кібербезпеки CMS.

Наприклад, розглянемо дії для підвищення рівня кібербезпеки CMS Wordpress [5-8] табл. 3.

Таблиця 3.

Дії для підвищення рівня кібербезпеки CMS Wordpress

№	Код	Назва	Опис
1	CSLP	Зміна сторінки входу на сайт	За замовчуванням сторінка авторизації знаходиться за

№	Код	Назва	Опис
			адресою ваш-сайт.ua/wp-login.php. Боти заходять на цю сторінку і намагаються підібрати логін та пароль. Щоб зменшити активність цих ботів на своєму сайті, ви можете змінити адресу сторінки авторизації на іншу, наприклад, ваш-сайт.ua/page1 або ваш-сайт.ua/lost.php.
2	RS	Обмеження доступу до wp-login.php та wp-admin	Сторінка авторизації та панель адміністратора – важливі сторінки, які дають доступ до сайту, тому доступ до цих сторінок можна обмежити. У разі використання статичного IP, можна заборонити доступ до сторінки входу або панелі адміністратора: # Limit login and admin by IP ErrorDocument 401 /шлях до сайту/index.php?error=404 ErrorDocument 403 /шлях до сайту/index.php?error=404 <IfModule mod_rewrite.c> RewriteEngine on RewriteCond %{REQUEST_URI} ^(.*)?wp-login\.php(.*)$ [OR] RewriteCond %{REQUEST_URI} ^(.*)?wp-admin$ RewriteCond %{REMOTE_ADDR} !^IP Адреса 1$ RewriteCond %{REMOTE_ADDR} !^IP Адреса 2$ RewriteCond %{REMOTE_ADDR} !^IP Адреса 3$ RewriteRule ^(.*)$ - [R=403,L] </IfModule> Перші два рядки перенаправляють відвідувача з неавторизованого IP на сторінку з помилкою 404. Це правило не викликатиме циклічні редиректи, тому сайт не виглядатиме завислим.

№	Код	Назва	Опис
			Замініть /шлях до сайту/ у двох перших рядках на свою адресу. Замініть IP Адреса 1, IP Адреса 2і IP Адреса 3 на ті IP-адреси, з яких хочете мати доступ до сторінок wp-login.php і wp-admin. Якщо потрібна лише одна IP-адреса, видаліть рядки 9 і 10, якщо потрібно більше адрес, додайте потрібну кількість рядків. Якщо користувач має динамічні IP (або Мультисайт), використовуйте це правило: # Limit login and admin by manual query ErrorDocument 401 /шлях до сайту/index.php?error=404 ErrorDocument 403 /шлях до сайту/index.php?error=404 <IfModule mod_rewrite.c> RewriteEngine on RewriteCond %{REQUEST_METHOD} POST RewriteCond %{HTTP_REFERER} !^http://(.*)?ваш-сайт.ua [NC] RewriteCond %{REQUEST_URI} ^(.*)?wp-login\.php(.*) $ [OR] RewriteCond %{REQUEST_URI} ^(.*)?wp-admin$ RewriteRule ^(.*)$ - [F] </IfModule> Замініть /шлях до сайту/ і / сайт.ua/ своя адреса. Це правило визначає, що тільки ті користувачі, які набрали вручну wp-login.php або wp-admin в браузері, отримають доступ до цих сторінок.
3	CFSS	Зміна файлової структури сайту	CMS Wordpress - система з відкритим вихідним кодом, стандартна структура файлів і папок якої знаходиться у відкритому доступі, тому будь-хто,

№	Код	Назва	Опис
			включаючи хакерів, знає, звідки почати атаку на сайт. Щоб протистояти таким атакам можна реорганізувати стандартну файлову структуру CMS Wordpress. Ви можете перенести всю установку CMS Wordpress в нову папку, змінити стандартне розташування папок wp-content, plugins і uploads та створити нову папку для тем.
4	MWPC	Переміщення wp-config.php	У файлі wp-config.php зберігаються паролі, ключі безпеки та інша важлива інформація, тому краще перемістити цей файл у безпечне місце. Якщо ви перемістите файл у нове місце, хакерам спочатку доведеться його знайти, після чого спробувати зламати. Перемістіть файл wp-config на один рівень. Якщо початкове розташування файлу було /home/public_html/wp-config.php, перенесіть файл в /home/wp-config.php Якщо розташування файлу було: /home/public_html/abcd-xyz/wp-config.php, то перенесіть файл у: /home/public_html/wp-config.php Не переносіть файл, якщо на одному рівні вище вже є файл wp-config.php, або якщо сайт встановлений у підпапці або субдомені. Якщо хочете перемістити wp-config.php в інше місце, перенесіть його в нове місце, на старому місці створіть ще один wp-config.php і додайте до нього код: define('ABSPATH', dirname(__FILE__) . '/');

№	Код	Назва	Опис
			require_once(ABSPATH . '../шлях/до/wp-config.php'); ?> Замініть /шлях до/wp-config.php на нову адресу файлу wp-config.
5	RMSA	Обмеження дозволів MySQL облікового запису	Якщо хакер отримає доступ до файлу wp-config і дізнається логін та пароль до бази даних, він отримає доступ до бази даних, і зможе робити все, що дозволено робити цьому обліковому запису. Щоб у зломщика було менше шансів отримати контроль над сайтом, дайте цьому обліковому запису лише необхідний мінімум привілеїв. Для звичайного використання сайту рекомендовані привілеї SELECT, INSERT, UPDATE, DELETE та ALTER для оновлення CMS Wordpress. Зайдіть у phpMyAdmin, вкладка SQL: Перегляд всіх користувачів: select user,host from mysql.user; Показати привілеї: show grants for 'user'@'host'; Замініть user і host на ім'я користувача, яке ви можете дізнатися у файлі wp-config і хост, зазвичай localhost. Щоб змінити привілеї, спочатку потрібно скасувати поточні привілеї: revoke all privileges on *.* from 'user'@'host'; Щоб дати нові привілеї, використовуйте цей запит: grant SELECT,INSERT,UPDATE,DELETE ON `db`.* TO 'user'@'host';

№	Код	Назва	Опис
			Застосувати: flush privileges;
6	DPHPD	Вимкнення директиви php.ini	За замовчуванням у CMS Wordpress немає файлу php.ini. Якщо немає файлу, потрібно створити його в кореневій папці сайту. Потрібно додати expose_php = off, щоб приховати версію PHP, і allow_url_include = off щоб вимкнути можливість для php коду використовувати цю директиву, яку хакери використовують для XSS атак.
7	SPIFF	Встановлення паролю на важливі файли та папки	Для встановлення паролю створюється файл із паролями (генератор .htpasswd), розміщується файл у недоступній з Інтернету папці (наприклад, на один рівень вище кореневої папки) і додається запис до .htaccess з адресою цього файлу, або можна скористатися плагіном AskApache Password Protect.

8	DES	Вимкнення показу помилок на сайті	Наступний код вимикає помилки на сторінках сайту. Його необхідно додати до файлу wp-config.php, що знаходиться в корені вашого сайту. Найпростіше знайти в цьому файлі текст
			define ('WP_DEBUG ", false);
			і замість нього додати:
			error_reporting(0); // вимикаємо виведення інформації про помилки
			ini_set('display_errors', 0); // вимикаємо виведення інформації про помилки на екран
			define('WP_DEBUG', false);
			define('WP_DEBUG_DISPLAY', false);
			Перед редагуванням файлів PHP на сайті потрібно робити їхню

№	Код	Назва	Опис
			резервну копію, це дозволить швидко повернутися на крок назад у разі якихось механічних помилок.
9	RAIF	Заборона доступу до важливих файлів	Забороніть доступ до важливих файлів Ви можете закрити доступ до важливих файлів wp-config.php, htaccess, php.ini та логів помилок. Додайте це правило з Кодексу CMS Wordpress: <FilesMatch "^.*(error_log|wp-config\.php|php.ini|\.[hH][tT][aApP].*)$"> Order deny,allow Deny from all </FilesMatch> Якщо у вас є php5.ini або php7.ini замість php.ini, замініть php.ini в першому рядку на ваш файл.
10	SLDF	Установлення обмеження на розмір файлів, що завантажуються	WordPress має вбудований завантажувач медіафайлів. Це робить простим завантаження файлів на сайт, немає необхідності використовувати FTP-клієнти або менеджери файлів. Всі медіафайли, які можуть знадобитися у сайті, можуть бути легко завантажені в кілька кліків миші. Однак варто відзначити, що медіа завантажувач використовує PHP для завантаження файлів на сервер і, на жаль, має ліміт на максимальний розмір файлу, що завантажується. Перевірити максимальний розмір файлу в WordPress, можна зайшовши в Панель адміністратора→ Медіафайли→ кнопка Додати новий. Обмежте максимальний розмір файлів, що завантажуються 10Мб:

№	Код	Назва	Опис
			# Limit file upload size 10MB LimitRequestBody 10240000
11	RAPHPF	Заборона доступу до файлів PHP	Обмежте доступ до php файлів тем і плагінів, оскільки хакери можуть впровадити в них шкідливий код. # Block access to PHP files RewriteCond %{REQUEST_URI} !^/wp-content/plugins/file/to/exclude\.php RewriteCond %{REQUEST_URI} !^/wp-content/plugins/directory/to/exclude/ RewriteRule wp-content/plugins/(.*\.php)$ - [R=404,L] RewriteCond %{REQUEST_URI} !^/wp-content/themes/file/to/exclude\.php RewriteCond %{REQUEST_URI} !^/wp-content/themes/directory/to/exclude/ RewriteRule wp-content/themes/(.*\.php)$ - [R=404,L]
12	PEPHPF	Заборона виконання PHP файлів у папці Uploads	Зазвичай хакери завантажують шкідливий код у папку /wp-content/uploads/. Додайте це правило, щоб вимкнути можливість виконувати php файли в папці uploads: # Disallow php execution in Uploads <Directory "/var/www/wp-content/uploads/"> <Files "*.php"> Order Deny,Allow Deny from All </Files> </Directory> Замініть /var/www/ у рядку 2 на свою адресу. Якщо не виходить, проблема швидше за все на адресі. Спробуйте різні варіанти адреси: повний шлях, відносний шлях з першим слешем, відносний шлях без першого слеша.

№	Код	Назва	Опис
			Також, можна створити новий файл .htaccess і помістіть його в папку /wp-content/uploads/. Додайте до нього такий код: # Disallow php execution in Uploads <Files *.php> deny from all </Files>
13	DAWPF	Заборона доступу до папки wp-includes	У папці wp-includes є важливі файли, які можуть бути використані для «злому» сайту. Додайте це правило, щоб захистити /wp-includes/: # Block wp-includes folder and files <IfModule mod_rewrite.c> RewriteEngine On RewriteBase / RewriteRule ^wp-admin/includes/ - [F,L] RewriteRule !^wp-includes/ - [S=3] RewriteRule ^wp-includes/[^/]+\.php$ - [F,L] RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L] RewriteRule ^wp-includes/theme-compat/ - [F,L] </IfModule>
14	SSIMS	Захист сайту від впровадження шкідливих скриптів	Забороніть використання коду в php файлах. Додайте це правило: # Disallow change GLOBALS and _REQUEST Options +FollowSymLinks RewriteEngine On RewriteCond %{QUERY_STRING} (<|%3C).*script.*(>|%3E) [NC,OR] RewriteCond %{QUERY_STRING} GLOBALS(=|[|%[0-9A-Z]{0,2}) [OR] RewriteCond %{QUERY_STRING} _REQUEST(=|[|%[0-9A-Z]{0,2}) RewriteRule ^(.*)$ index.php [F,L]
15	EELS	Увімкнення логів подій на сервері	Логи знаходяться на хостинг панелі, залежно від софту, що використовується, вони можуть

№	Код	Назва	Опис
			називатися по-різному: «Логи», «Журнал подій», «Метрики» тощо. Зазвичай у цьому розділі знаходяться 2 журнали: Логи помилок і Логи доступу. Ці логи містять інформацію про успішні спроби доступу. Якщо щось трапиться, ви зможете знайти сліди хакера та спробувати відновити сайт. Логи зберігаються на сервері кілька днів, після чого автоматично видаляються. Якщо сайт був зламаний, скопіюйте логи на комп'ютер, поки вони не були видалені з сервера.
16	UTFA	Використання двофакторної авторизації	Двофакторна авторизація використовує механізм, коли для входу на сайт вам потрібно не тільки ввести логін і пароль, але також токен безпеки, який унікальний для кожного користувача. Термін дії цього токена обмежений, зазвичай, це 1 хвилина. Токени генеруються автоматично і являють собою 6-значний пароль, який можна отримати на свій смартфон. Токен безпеки є унікальним для кожного користувача і має обмежений час існування. Навіть якщо хтось знає логін та пароль до сайту, він не зможе увійти, тому що він не матиме потрібного токена. У репозитарії CMS Wordpress знаходиться багато плагінів для додавання цього способу захисту, наприклад: Google Authenticator, Duo Two-Factor Authentication, Two-Factor, Rublon Two-Factor Authentication тощо.

№	Код	Назва	Опис
17	CKS	Зміна ключів та солей	Ключі безпеки в CMS Wordpress - це набір випадкових символів, які використовуються CMS Wordpress для шифрування паролів, які зберігаються в cookies браузера. «Солі» — інший набір випадкових символів, який використовується для посилення ключів. Для підтримки безпеки змінюйте ключі та «солі» раз на кілька місяців. Згенеруйте нові ключі та вставте їх у wp-config.php.

18	DUN	Вимкнення нумерації користувачів	Якщо зловмисник введе в рядок адреси ваш сайт.ua/?author=1, він буде перенаправлений на сторінку користувача з ID = 1.
			У цьому випадку хакер знатиме ім'я користувача, і йому залишиться тільки дізнатися пароль.
			Навіть якщо користувачі використовують складні паролі, зловмиснику краще не знати ID користувачів. Додайте цей код до .htaccess:
			RewriteCond %{QUERY_STRING} author=d
			RewriteRule ^ /? [L,R=301]
			або:
			RewriteCond %{REQUEST_URI} !^/wp-admin [NC]
			RewriteCond %{QUERY_STRING} ^author=\d+ [NC,OR]
			RewriteCond %{QUERY_STRING} ^author=\{num
			RewriteRule ^ - [L,R=403]
			Або додайте цей код до functions.php:
			/* Redirect to Home page */
			add_action('template_redirect', 'tb_template_redirect');
			function tb_template_redirect()
			{if (is_author()){

№	Код	Назва	Опис
			wp_redirect(home_url()); exit;}}
19	DSSVI	Вимкнення інформації про версію програмного забезпечення сервера	Підпис Сервера — це службова інформація, в якій йдеться про те, що сайт використовує, наприклад, сервер Apache певної версії та ОС Ubuntu певної версії. Щоб вимкнути показ версій програмного забезпечення на вашому сервері, додайте цей код .htaccess: <IfModule mod_headers.c> Header unset X-Powered-By Header unset Server </IfModule> або: ServerSignature Off
20	DUID1	Видалення користувача з ID 1	При установці CMS Wordpress за замовчуванням створюється користувач із правами адміністратора з ID=1. Ви вже знаєте про зміну імені користувача, але також можна змінити ID адміністратора сайту. Створіть нового користувача з правами адміністратора, зайдіть на сайт під новим обліковим записом і видаліть старого адміністратора з ID 1. З іншого боку, деяким плагінам потрібні додаткові права для своєї роботи, яких у них може не бути, якщо створити нового користувача з правами адміністратора.
21	CETLS	Зміна тексту помилки під час входу на сайт	За замовчуванням CMS Wordpress показує стандартне повідомлення на сторінці авторизації, коли відвідувач вводить неправильний логін або пароль. У цьому повідомленні йдеться, що саме було введено неправильно, — логін чи пароль. Коли хакер вводить неправильний пароль до правильного імені користувача, CMS Wordpress

№	Код	Назва	Опис

			показує повідомлення, що пароль неправильний. Це дає йому зрозуміти, що ім'я користувача є вірним.
			Щоб змінити текст повідомлення, додайте ці рядки до functions.php.
			/* Return custom message on login */
			function no_wordpress_errors(){
			return 'Невірний Логін або Пароль';}
			add_filter('login_errors', 'no_wordpress_errors');
			Замініть повідомлення у рядку 2 на своє повідомлення.

22	DXML	Вимкнення XML-RPC	XML-RPC — це API інтерфейс, який використовується для доступу до сайту через мобільні програми, для трекбеків та пінгбеків та використовується плагіном Jetpack. Якщо ви користуєтесь чимось із цього, то залиште цю функцію увімкненою або частково ввімкненою. З іншого боку, XML-RPC може бути використаний хакерами для атак із перебором логінів та паролів. Замість того, щоб намагатися підібрати логін і пароль на звичайній сторінці авторизації, де ви можете встановити обмеження на кількість спроб входу, бот може спробувати необмежену кількість комбінацій через інтерфейс xml-rpc. Щоб протистояти таким атакам, використовуйте складні логіни та паролі. Інша проблема в тому, що атаки брут-форс витрачають велику кількість ресурсів сервера. Якщо сайт знаходиться на недорогому хостингу, в результаті такої атаки сайт може зависнути

№	Код	Назва	Опис
			через використання всіх ресурсів сервера. Щоб повністю вимкнути XML-RPC, додайте це правило до .htaccess: # Block XML-RPC requests <Files xmlrpc.php> Order Allow,Deny Deny from all </Files>
23	SSR	Регулярне сканування сайту	Після встановлення плагіну безпеки налаштуйте регулярне сканування. Автоматичне сканування за розкладом знаходиться в платних плагінах, але в деяких безкоштовних воно також є. Наприклад, Sucuri Security. Без регулярного сканування вразливість може пройти непоміченою, і це може спричинити злом сайту. Ви про це можете дізнатися, коли пошукові системи позначать ваш сайт як «що містить шкідливий код» і знизять його в пошуковій видачі. Встановіть частоту сканування трохи менше, ніж частота бекапу. Якщо сайт зламають, буде від чого його відновити.
24	VEL	Перегляд логів подій	Коли ви встановлюєте сайт на хостингу, у цей час починають вестися логи подій. Десь на вашому обліковому записі повинні зберігатися логи помилок та логи доступу. Зазвичай називаються Логи, якщо ви не можете їх знайти, запитайте у техпідтримки. Логи подій - це записи про те, як хтось отримував доступ до сайту, переглядав важливі файли або намагався отримати доступ до них. Якщо ви час від часу переглядатиме логи, ви можете звернути увагу на якусь незвичайну активність.

№	Код	Назва	Опис
			Наприклад, якщо звичайні відвідувачі намагалися отримати доступ до .htaccess або wp-config.php. Якщо ви бачите, що якийсь відвідувач отримав доступ до цих файлів, то це може означати, що ваш сайт зламали. Якщо ви час від часу переглядатиме логи подій і помітите щось дивне, ви зможете вчасно відреагувати на зміни.
25	IFS	Установлення фаєрволу на сервері	Ви можете встановити фаєрвол на сервері, це захистить сайт і сервер від хакерів ще на підході. Не плутайте фаєрвол на сервері з WAF (web application firewall), фаєрволом на сайті, який знаходиться в плагіні Wordfence. Якщо хакер вже потрапив на сайт, зупинити його буде важче, тому краще зупинити його на підході до сайту, тобто на рівні сервера. Ви можете спробувати встановити безкоштовний фаєрвол ConfigServer Security & Firewall. Перед інсталяцією поговоріть з техпідтримкою хостингу, у вас може не бути прав на інсталяцію ПЗ, або, можливо, якийсь файрвол вже встановлено на сервері.

Отже, використовуючи вираз (3) та дані з табл. 3, при $n = 25$ отримаємо:

$$A_{CMS\ Wordpress} = \{\bigcup_{j=1}^{25} A_j\} = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10},$$
$$A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}, A_{17}, A_{18}, A_{19}, A_{20}, A_{21}, A_{22}, A_{23}, A_{24}, A_{25}\} =$$
$$= \{A_{CSLP}, A_{RS}, A_{CFSS}, A_{MWPC}, A_{RMSA}, A_{DPHPD}, A_{SPIFF}, A_{DES}, A_{RAIF}, A_{SLDF}, A_{RAPHPF}, A_{PEPHPF},$$
$$A_{DAWPF}, A_{SSIMS}, A_{EELS}, A_{UTFA}, A_{CKS}, A_{DUN}, A_{DSSVI}, A_{DUID1}, A_{CETLS}, A_{DXML}, A_{SSR}, A_{VEL}, A_{IFS}\} =$$
$$\{CSLP, RS, CFSS, MWPC, RMSA, DPHPD, SPIFF, DES, RAIF, SLDF, RAPHPF, PEPHPF,$$
$$DAWPF, SSIMS, EELS, UTFA, CKS, DUN, DSSVI, DUID1, CETLS, DXML, SSR, VEL, IFS\}, \quad (4)$$

де A_1=A_{CSLP}=$CSLP$, A_2=A_{RS}=RS, …, A_{25}=A_{IFS}=IFS – дії для підвищення рівня кібербезпеки CMS Wordpress.

Таким чином, виконавши дії зазначені у табл. 3 ми значно підвищимо рівень кібербезпеки CMS Wordpress. Для успішної реалізації кібератаки, за таких налаштувань, потрібна дуже висока кваліфікація зловмисників та значні матеріально-технічні затрати.

STATUS	TEST DESCRIPTION	TEST RESULTS	
PASSED	Check if WordPress core is up to date.	You are using the latest version of WordPress.	Details & Tips
PASSED	Check if automatic WordPress core updates are enabled.	Core updates are configured optimally.	Details & Tips
PASSED	Check if plugins are up to date.	All plugins are up to date.	Details & Tips
FAILED	Check if there are deactivated plugins.	There are 3 deactivated plugins.	Details, Tips & Auto Fixer
PASSED	Check if active plugins have been updated in the last 12 months.	All active plugins have been updated in the last 12 months.	Details & Tips
PASSED	Check if active plugins are compatible with your version of WP.	All active plugins are compatible with your version of WordPress.	Details & Tips
PASSED	Check if themes are up to date.	All themes are up to date.	Details & Tips
PASSED	Check if there are deactivated themes.	There are no deactivated themes.	Details & Tips

Рис. 3. Фрагмент вікна діагностики безпеки веб-сайту з використанням Security Ninja

Після виконання дій, що зазначені у табл. 3 можна, також перевірити їх ефективність скориставшись плагіном для тестування рівня безпеки веб-сайту (наприклад, Security Ninja [9]) рис. 3, хоча використання подібних тестів перевіряється практично протягом багатьох років, але позитивна оцінка не гарантує, що веб-сайт не «зламають». Також, негативна оцінка тесту не означає, що веб-сайт обов'язково буде «зламано».

Висновки

Таким чином, у цій роботі розроблено модель даних для підвищення кібербезпеки CMS, яка за рахунок визначення видів вразливостей для CMS та підвищення рівня кібербезпеки CMS, дозволяє ідентифікувати можливі види вразливостей для CMS та виконавши превентивні дії підвищити рівень кібербезпеки CMS. Розроблена модель спрямована на те, щоб унеможливити реалізацію зловмисниками кіберінцидентів в CMS. Модель та сформовані на її основі засоби будуть корисними, насамперед, для адміністраторів веб-сайтів, а також для фахівців з кібербезпеки у складі команд реагування на кіберінциденти типу CERT/CSIRT на які покладаються обов’язки щодо захисту ІТС в межах компаній, установ чи підприємств.

Література

1. Гнатюк В.О. Аналіз дефініцій поняття «інцидент» та його інтерпретація у кіберпросторі / В.О.Гнатюк // Безпека інформації. — №3 (19). — 2013. — С. 175-180.
2. «Tech Reports - What CMS?». https://whatcms.org/Tech_Reports December 14, 2020.
3. «W3Techs content management usage». Режим доступу в Інтернет: https://w3techs.com/technologies/overview/content_management. March 1, 2021.
4. Исследование популярности CMS за 2021 год. Режим доступу в Інтернет: https://itrack.ru/research/cmsrate/#!cms-free-tab. Березень 2021 р.
5. Безопасность Вордпресс. Полное руководство. Режим доступу в Інтернет: https://techbear.ru/rukovodstvo-po-bezopasnosti-wordpress/. 01 жовтня 2021 р.
6. Як захистити сайт на WordPress – 17 способів. Режим доступу в Інтернет: https://hostiq.ua/blog/ukr/17-ways-to-secure-wordpress/.
7. Защита WordPress – 12 Советов, чтобы защитить ваш сайт. Режим доступу в Інтернет: https://www.hostinger.ru/rukovodstva/zashchita-wordpress. 03 липня 2019 р.
8. Защита сайта на WordPress от взлома https://hostpro.ua/blog/zashhita-sayta-na-wordpress-ot-vzloma/. 29 жовтня 2020 р.
9. Security Ninja – Secure Firewall & Secure Malware Scanner https://wordpress.org/plugins/security-ninja/.

References

1. Gnatyuk V. Analysis of definitions of the concept of «incident» and its interpretation in cyberspace. Information Security. №3 (19). 2013. P. 175-180.
2. «Tech Reports - What CMS?». https://whatcms.org/Tech_Reports. December 14, 2020.
3. «W3Techs content management usage». Internet access mode: https://w3techs.com/technologies/overview/content_management. March 1, 2021.
4. CMS popularity survey for 2021. Internet access mode: https://itrack.ru/research/cmsrate/#!cms-free-tab. March 2021.
5. WordPress security. Complete guide. Internet access mode: https://techbear.ru/rukovodstvo-po-bezopasnosti-wordpress/. October 1, 2021.
6. How to protect your site on WordPress - 17 ways. Internet access mode: https://hostiq.ua/blog/ukr/17-ways-to-secure-wordpress/.
7. WordPress Protection - 12 Tips to Protect Your Website. Internet access mode: https://www.hostinger.ru/rukovodstva/zashchita-wordpress. July 3, 2019.
8. Security of the WordPress site from hacking. Internet access mode: https://hostpro.ua/blog/zashhita-sayta-na-wordpress-ot-vzloma/. October 29, 2020.
9. Security Ninja - Secure Firewall & Secure Malware Scanner https://wordpress.org/plugins/security-ninja/.