УДК 004.056.5

МОДЕЛІ І МЕТОДИ ОЦІНКИ СТІЙКОСТІ ЛОКАЛІЗОВАНИХ CMS НА ПРИКЛАДІ ВЕБРЕСУРСІВ РФ

П. Ю. Гоголєва[1,a]

[1] Навчально-науковий Фізико-технічний інститут

Анотація

У статті розглянуто підходи до виявлення вразливостей у популярних контент-менеджмент системах (CMS), які використовуються на російських вебресурсах. Запропоновано архітектуру інструменту для оцінки стійкості вебресурсів на базі автоматизованого збору інформації, визначення версії CMS та подальшого аналізу потенційних вразливих компонентів. Окрема увага приділяється модульній структурі сканування, агрегованому пошуку відомостей про вразливості (CVE) та ранжуванню ризиків з використанням відкритих джерел. Розглянуто зовнішні інструменти, що полегшують процес автоматизації аудиту.

Ключові слова: інформаційна безпека, CMS, вразливості, CVE, аналіз, автоматизація, аудит

Вступ

Сучасні вебресурси здебільшого будуються на основі систем керування контентом (CMS), що значно спрощує їх розробку й обслуговування. Водночас широке використання однакових платформ, таких як WordPress, Joomla, Drupal, MODX, Bitrix тощо, зумовлює високий рівень повторюваності помилок конфігурації та використання вразливих компонентів. Особливо актуальним є дослідження стійкості таких систем у випадках, коли вони активно використовуються на рівні державних або інфраструктурно важливих сайтів, як-от у РФ, з якою Україна знаходиться у стані гібридної війни. Згідно з останніми звітами, CMS дедалі частіше стають інструментом для watering hole-атак, особливо проти фахівців критичної інфраструктури, де середовище CMS може використовуватись як одна з платформ для розміщення підроблених оновлень або шкідливого коду, як це спостерігається в аналізі watering hole-кампаній [1].Це також підтверджується у дослідженні [2], де системи керування контентом розглядаються як критична частина цифрової інфраструктури з високим рівнем ризику уразливості через відкриту архітектуру. Під «стійкістю» у цьому контексті розуміється здатність CMS протистояти відомим технічним уразливостям, що мають публічні ідентифікатори CVE (Common Vulnerabilities and Exposures), а також залишатися безпечною в умовах типових сценаріїв загроз. CVE – це уніфіковані записи, які містять опис конкретної вразливості, версії програмного забезпечення, до якої вона застосовується, та іноді – методи її зловживання.

[a]polgog-ipt25@lll.kpi.ua

1. Типові ризики, пов'язані з CMS, та їх наслідки

1.1. Організаційні ризики

Низький рівень інформаційної гігієни у процесі встановлення та обслуговування CMS часто призводить до наявності критичних вразливостей. До поширених організаційних помилок відносяться:

- Використання стандартних облікових даних, такі як логін admin і прості паролі. У ряді випадків ці облікові дані не змінюються, що робить систему легкою мішенню для підбору пароля.
- Нехтування оновленнями. Уразливості в ядрах CMS та плагінах регулярно публікуються у базах даних вразливостей, що надає зловмисникам легку можливість експлуатувати вже відому вразливість.
- Встановлення плагінів з невідомих джерел, зокрема у WordPress, часто містять методи обходу стандартних процедур автентифікації (бекдори) або містять вразливості через відсутність аудитів коду. Дослідження «Over 100 Bugs in a Row: Security Analysis of the Top-Rated Joomla Extensions» [3] демонструє, що навіть найрейтинговіші розширення для Joomla можуть містити десятки вразливостей, що підтверджує необхідність автоматичного аудиту.
- Недостатній контроль прав доступу і некоректне налаштування дозволів призводить до можливості редагування критичних файлів незареєстрованими користувачами.
- Системи на базі CMS можуть зберігати метаінформацію або відкриті API, які дозволяють деанонімізувати користувачів або адміністраторів через OSINT-аналіз [4, 5].

1.2. Технічні проблеми

З технічної точки зору, CMS, як складні програмні системи, містять ряд ризиків:

- Недоступність ресурсу (DoS). Атаки на погано оптимізовані компоненти можуть вивести CMS з ладу через споживання ресурсів.
- XSS (міжсайтове виконання скриптів). Наявність таких вразливостей дозволяє зловмиснику виконувати JavaScript у браузері жертви від імені адміністратора CMS.
- SQL-ін'єкції. Неправильна фільтрація введення або її відсутність дозволяє виконувати довільні запити до бази даних. Наслідки — витік інформації або повний контроль над сайтом.
- Завантаження шкідливих файлів. У багатьох CMS некоректна перевірка типу або розширення файлу дозволяє завантажити .php-файл на сервер, що відкриває доступ до командної оболонки.

2. Приклади вразливостей із бази CVE

Для ілюстрації рівня ризику наведемо кілька відомих випадків, що мали широкий розголос:

1. **CVE-2018-7600** (відомий як *Drupalgeddon2*) дозволяє неавторизованому користувачу виконувати довільний PHP-код на сервері через уразливість в обробці запитів у Drupal 7 та 8. Відноситься до класу RCE (Remote Code Execution).
2. **CVE-2021-24284** стосується плагіна *Kaswara Modern VC Addons* для WordPress. Відсутність перевірки дозволяла завантажити довільний файл через AJAX-функцію, що відкривало можливість RCE.
3. **CVE-2020-25213** виявлена у плагіні *File Manager* для WordPress. Невірна логіка авторизації дозволяла неавторизованим користувачам виконувати довільні команди через вебінтерфейс плагіна, що відкривало можливість повного компрометаційного доступу до сайту.

3. Мета та задачі дослідження

Метою роботи є розробка моделі оцінки стійкості CMS, які використовуються на території РФ з подальшим аналізом наявних уразливостей та компонентів ризику. Для досягнення цієї мети було поставлено наступні задачі:

1. Проаналізувати найбільш розповсюджені CMS на доменах .ru та .рф
2. Розробити модуль автоматичного визначення CMS та її версії.
3. Інтегрувати сканери компонентів для визначення модулів, тем і плагінів.
4. Реалізувати автоматизований пошук CVE для кожної одиниці.
5. Забезпечити фільтрацію за відповідністю версії CMS.
6. Додати механізм ранжування знайдених вразливостей.
7. Побудувати загальну оцінку стійкості ресурсу.

4. Розподіл CMS у російському доменному просторі

Вивчення поширеності CMS у російських доменних зонах дозволяє визначити пріоритети для подальшого аналізу вразливостей та оцінки стійкості. За статистичними даними 2020 року абсолютним лідером серед CMS у доменних зонах .RU та .РФ є WordPress. Її частка у зоні .RU становить 51% (понад 526000 сайтів), а в зоні .РФ – 42% (близько 45000 сайтів). На другому місці за популярністю перебуває 1С-Бітрікс: у зоні .RU її частка становить 13,4% (138000 сайтів), а в зоні .РФ – 13,6% (14700 сайтів). Ця комерційна CMS демонструє стабільне зростання, зокрема завдяки інтеграції з екосистемою програмного забезпечення 1С, яке активно використовується в російському бізнес-середовищі. Сайти, що базуються на 1С-Бітрікс, зазвичай одразу сумісні з системами документообігу, обліку товарів тощо. Третє місце займає Joomla, яка ще у 2018 році була другою за популярністю, однак протягом останніх років втратила частку. У доменній зоні .RU її частка дорівнює 12% (122600 сайтів), а в зоні .РФ – 13,5% (14500 сайтів). Попри наявність великої кількості плагінів та активну спільноту розробників, Joomla дещо поступається WordPress у зручності адміністрування та поширеності. На четвертій позиції розташувалася CMS MODX з часткою 5,3% у .RU (54500 сайтів) та 5,5% у .РФ (6000 сайтів). Замикає п'ятірку Drupal, яка має 5,3% у .RU (34000 сайтів) і 3,5% у .РФ (3800 сайтів).

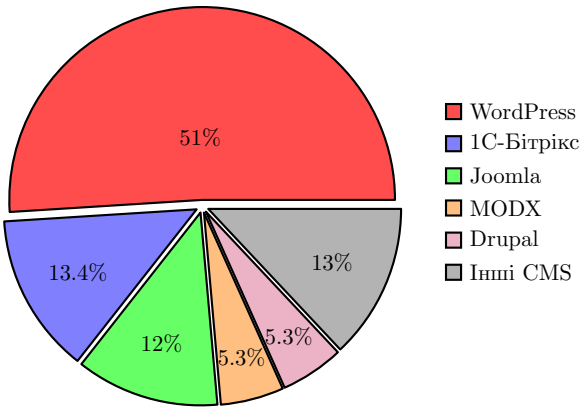

Рис. 1. Діаграма розподілу використання CMS у доменних зонах РФ

5. Методика реалізації

Для реалізації поставлених задач було створено консольний Python-додаток із модульною архітектурою. Кожен етап роботи (визначення CMS, збір метаінформації, CVE-пошук, аналіз, формування звіту) реалізується як окремий підмодуль, що дозволяє легко адаптувати під різні CMS. У межах розробки фреймворку для оцінки стійкості CMS було інтегровано низку інструментів з відкритим кодом, кожен з

яких орієнтований на певну підсистему — визначення CMS, збір компонентів, аналіз конфігурацій або пошук відомих вразливостей. Вибрані рішення належать до категорії традиційних інструментів технічного аудиту безпеки, які активно використовуються в процесі тестування на проникнення, моніторингу вразливостей і розвідки кіберзагроз.

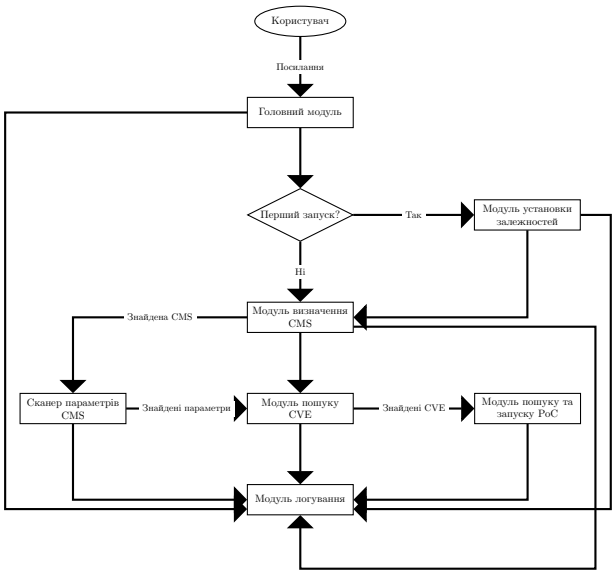

Рис. 2. Блок-схема логіки роботи інструменту

5.1. Визначення CMS та їх структури

- CMSeeK – універсальний сканер для виявлення CMS з відкритим кодом. Працює на основі аналізу HTTP-заголовків, структури URL, вмісту HTML і характерних директорій. Підтримує понад 100 різних CMS. Надає базову інформацію про тип системи, версію та деякі плагіни. CMSeeK працює на принципах сигнатурного аналізу.
- WPScan – спеціалізований інструмент для WordPress, який використовує власну базу вразливостей та дозволяє визначати ядро CMS, плагіни, теми, облікові записи з відкритим доступом. WPScan базується на HTTP-аналізі, перевірці типових шляхів та взаємодії з API. Окрім збору компонентів, може попередньо ідентифікувати CVE, що робить його ефективним для швидкої діагностики.
- JoomScan – інструмент від OWASP, призначений для сканування сайтів на Joomla. Проводить загальну перевірку конфігурації, визначає версію CMS, відомі плагіни та виводить рекомендації щодо посилення безпеки. JoomScan застосовує методи URL-брутфорсу та аналізу відповідей сервера.
- Droopescan – сканер, орієнтований на Drupal, здатний визначати модулі, теми та версію системи. Його основна перевага – збір точного списку компонентів, які потім можна зіставити з CVE.
- check_bitrix – утиліта, адаптована для роботи з CMS 1С-Бітрікс. Визначає наявність Bitrix на сервері через аналіз сигнатур, JS-фреймворків, специфічних директорій (/bitrix/admin/, /upload/ тощо). Дозволяє виявити ознаки застарілих версій.

5.2. Пошук і ранжування вразливостей

- SearchSploit – консольна утиліта, яка взаємодіє з локальною базою ExploitDB. Дозволяє знаходити відомі публічні експлойти, фільтрувати їх за типом програмного забезпечення, версією, ключовими словами. Основа методу – пошук сигнатур у назвах та описах скриптів.
- CVEMap – інструмент для ранжування вразливостей за ризиком. Працює на базі локальної або оновлюваної бази CVE, яку додатково збагачує інформацією про CVSS-оцінку, EPSS (імовірність експлуатації), наявність шаблонів для Nuclei, згадки в open-source репозиторіях. Це дозволяє визначити, які з виявлених уразливостей є пріоритетними для аналізу або усунення.
- GitHub Search – застосовується для пошуку PoC (Proof of Concept) через GitHub API. Метод базується на побудові Python-запитів, з подальшою фільтрацією за репутаційними ознаками (зірки, дата оновлення, відповідність опису).

Ці інструменти становлять ядро системи оцінки стійкості CMS, дозволяючи забезпечити збір релевантної інформації про цільову платформу, наявні уразливості та потенційні ризики без активного втручання в структуру вебресурсу.

6. Приклад роботи та візуалізація результатів

Для тестування використовувалася віртуальна машина DC-2, що вразлива за своїм дизайном. Нижче вибірково наведено частину логів після виконання сканування вище зазначеної машини.

```
[INFO] Resolved target IP: 192.168.110.153
[INFO] Detected CMS: WordPress
[INFO] CMS version updated: 4.7.10
[INFO] Searching for CVEs via NVD API...
[NVD Matches for]: WordPress 4.7
1. [CVE-2017-1001000]
[INFO]Metasploit Modules Found (by CVE):
auxiliary/scanner/http/wordpress
      \_content\_injection
[CVEMap Summary Table]:
```

7. Висновки

У ході дослідження було розроблено фреймворк для автоматизованого аналізу стійкості CMS, що використовуються на вебресурсах, популярних у доменних зонах РФ, з якою Україна знаходиться у стані гібридної війни. Запропонована система поєднує модулі визначення CMS, виявлення компонентів, пошуку публічних вразливостей (CVE) та їх аналітичної оцінки з точки зору рівня ризику.

Таблиця 1. Ранжування знайденої вразливості за допомогою CVEMap

ID	CVE-2017-1001000
CVSS	7.5
SEVERITY	HIGH
EPSS	0.80371
PRODUCT	wordpress
AGE	2951
TEMPLATE	–

Фреймворк було протестовано у контрольованому середовищі з відомими вразливостями. В результаті підтверджено ефективну інтеграцію таких зовнішніх засобів, як WPScan, JoomScan, Droopescan, CMSeeK, check_bitrix, SearchSploit, CVEMap та ін. Інструмент виводить структуровані звіти, підтримує аналіз компонентів і формує підсумкову оцінку стійкості ресурсу. По-друге, зібрана статистика засвідчує домінування WordPress, Bitrix та Joomla в російському сегменті Інтернету. Це дозволяє чітко визначити пріоритети аналізу та тестування. Методика може використовуватись для вивчення ймовірних векторів атак, прогнозування ризиків та створення моделей інформаційної протидії. У сучасних умовах гібридної війни важливо мати інструментарій, здатний ідентифікувати слабкі місця інформаційної інфраструктури супротивника. Запропоноване рішення дозволяє досліджувати поширені вразливості CMS без безпосереднього втручання в цільову систему, спираючись на відкриті дані та репозиторії.Зокрема, результати останніх досліджень підтверджують доцільність включення аналізу CMS до комплексної оцінки кіберстійкості об'єктів критичної інфраструктури, оскільки вебінтерфейси керування можуть виступати опосередкованими точками контролю в умовах прихованого впливу на IT-системи [6]. У перспективі можливий розвиток у таких напрямках:

- розширення підтримки менш популярних CMS;
- додавання інтерфейсу для OSINT-модулів;
- побудова динамічного графа залежностей між плагінами та уразливостями;
- розробка функціоналу групового аудиту для кластерів сайтів. Таким чином, фреймворк може стати ефективним інструментом інформаційного аналізу в академічних, дослідницьких та оборонних цілях.

Перелік використаних джерел

1. *Threat Response Unit eSentire.* Popular Info-Stealing Malware SolarMarker is Using Watering Hole Attacks and Fake Chrome Browser Updates to Infect Business Professionals, Warns eSentire. — 2024. — URL: https://www.esentire.com/blog/popular-info-stealing-malware-solarmarker-is-using-watering-hole-attacks-and-fake-chrome-browser-updates-to-infect-business-professionals-warns-esentire ; Accessed: 2025-05-03.
2. *Akintola S., James A.* Privacy and Security in Open-Source CMS Platforms: Evaluating Risks and Implementing Best Practices // ResearchGate. — 2025. — URL: https://www.researchgate.net/publication/388961111_Privacy_and_Security_in_Open-Source_CMS_Platforms_Evaluating_Risks_and_Implementing_Best_Practices ; Accessed: 2025-05-03.
3. Over 100 Bugs in a Row: Security Analysis of the Top-Rated Joomla Extensions / M. Niemietz, L. Nguyen, T. Müller, T. Holz // arXiv preprint arXiv:2102.03131. — 2021. — URL: https://arxiv.org/abs/2102.03131 ; Accessed: 2025-05-03.
4. *Chalyi O., Stopochkina I.* Information retrieval and deanonymization in the tasks of early detection of potential attacks on critical infrastructure // Кібербезпека: освіта, наука, техніка. — 2024. — Т. 2, № 26. — С. 305—322. — DOI: https://doi.org/10.28925/2663-4023.2024.26.694.
5. *Чалий О., Стьопочкіна І., Кривуля Є.* Методи інформаційного пошуку в задачах деанонімізації зловмисника // XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених "Теоретичні і прикладні проблеми фізики, математики та інформатики". — Київ, Україна, 2024. — С. 182—185.
6. Визначення параметрів непомітних кібератак на системи керування об'єктів критичної інфраструктури / О. Новіков, І. Стьопочкіна, М. Ільїн, М. Овчарук // Наукові вісті КПІ. — 2024. — № 1. — С. 69—75. — DOI: 10.20535/kpisn.2025.1.322905.