УДК 004.056.5
Коцюба А.Ю. к.т.н., доц., Лавренчук С.В., Яворський П.М.
Луцький національний технічний університет

ОГЛЯД ЗАСОБІВ БЕЗПЕКИ ВЕБ-САЙТІВ ТА ЇХ ПОРІВНЯННЯ У ПОПУЛЯРНИХ CMS

Коцюба А.Ю., Лавренчук С.В., Яворський П.М. Огляд засобів безпеки веб-сайтів та їх порівняння у популярних CMS. У статті розглянуто основні засоби забезпечення безпеки веб-сайтів, що написані на популярних CMS – Drupal, Wordpress, Joomla, OpenCart. Також було проведено порівняльний аналіз засобів безпеки (модулів) вищезазначених систем керування контентом та зроблено висновки.
Ключові слова: безпека, веб-сайт, CMS, Drupal, Joomla, SQL-ін'єкція, Wordpress

Коцюба А.Ю., Лавренчук С.В., Яворский П.М. Обзор средств безопасности веб-сайтов и их сравнение в популярных CMS. В статье рассмотрены основные средства обеспечения безопасности веб-сайтов, написанных с помощью популярных CMS – Drupal, Wordpress, Joomla, OpenCart. Также был проведен сравнительный анализ средств безопасности (модулей) вышеупомянутых систем управления контентом и сделаны выводы.
Ключевые слова: безопасность, веб-сайт, CMS, Drupal, Joomla, SQL-инъекция, Wordpress

Kotsyuba A.Yu, Lavrenchuk S.V., Jaworski P.M. Review of safety websites and compare them with popular CMS. The basic security arrangements websites that are written on popular CMS – Drupal, Wordpress, Joomla, OpenCart are considered in the article. A comparative analysis of security tools (modules) above content management systems and conclusions was also done.
Keywords: security, website, CMS, Drupal, Joomla, SQL-injection, Wordpress

Постановка проблеми. Для систем, розрахованих на постійну роботу (наприклад, для серверів), безпека відіграє велику роль. Веб-сервери – це основа Інтернету. Вони відповідають за функціонування мільярдів веб-сайтів по всьому світу, в результаті перетворюючись в сховище персональних даних своїх відвідувачів. Забезпечення захисту серверів від атаки ззовні – це одне з найважливіших завдань будь-якої організації.

В останні роки число атак, спрямованих на веб-сервери, значно зросло [5]. Географічне розташування сервера ролі при цьому не відіграє. Загроза, що приймає міжнародний характер, тепер виникає від організованих злочинних співтовариств, що займаються масовим збиранням паролів, фінансових відомостей та іншої інформації. У більшості випадків атака відбувається з мінімальним втручанням, а шкідливі програми, що розміщується на серверах і сайтах, розраховані на зараження максимальної кількості користувачів.

Кожен 16-й сайт в Україні містить шкідливе ПЗ, в той же час у всій мережі інтернет в середньому заражений приблизно кожен 54-й ресурс [6], тому тема статті є **досить актуальною**.

Веб-сервери особливо вразливі через свою відкритість – за своєю природою вони розраховані на обмін інформацією з користувачами. Зловмисник може внести модифікації в код HTTP сервера або сервера бази даних, або самих сторінок веб-сайту, помінявши його початкову функціональність.

Тому для захисту веб-серверу потрібні спільні дії адміністраторів веб-сайтів, програмістів і проектувальників; такі речі, як антивірусне програмне забезпечення, операційні системи і права доступу, вимагають постійної уваги.

Викладення основного матеріалу і обґрунтування отриманих результатів. Питанням безпеки сайту потрібно займатися ще на етапі розробки [1]. Перший етап проектування, створення і використання безпечного веб-сайту – це забезпечення максимального рівня безпеки сервера, на якому він розміщується. Веб-сервер формується кількома шарами програмного забезпечення (рис.1), кожен з яких вразливий до різноманітних способів атаки, метою атаки може стати будь-який з блоків.

Рис.1. **Структура програмного забезпечення веб-сервера**

Основа будь-якого сервера – це операційна система. Забезпечити її безпеку порівняно просто: достатньо вчасно встановлювати останні оновлення системи безпеки. Хакери можуть автоматизувати свої атаки, використовуючи шкідливе програмне забезпечення, перебирають один сервер за іншим в пошуках такого, де оновлення не було встановлено. У зв'язку з цим важливо стежити за тим, щоб оновлення встановлювалися вчасно і належним чином; будь-який сервер, на якому встановлені застарілі версії оновлень, може бути атакований.

Також слід вчасно оновлювати все програмне забезпечення, яке працює на веб-сервері. Будь-яке програмне забезпечення, яке не належить до необхідних компонентів (наприклад, DNS-сервер або засоби віддаленого адміністрування на зразок VNC або служб віддалених робочих столів), слід відключити або видалити. Якщо засоби віддаленого адміністрування все ж необхідні, стежте за тим, щоб не використовувалися паролі за замовчуванням або паролі, які можна легко відгадати. Це стосується не тільки до засобів віддаленого адміністрування, а й до облікових записів користувачів, комутаторів, маршрутизаторів і т.д.

Використання антивірусного програмного забезпечення є обов'язковою вимогою для будь-якого веб-сервера незалежно від операційної системи. У поєднанні з гнучким файєрволом антивірусне програмне забезпечення стає одним з найефективніших способів захисту. Коли веб-сервер стає метою атаки, зловмисник одразу ж намагається завантажити інструменти злому або шкідливі програми, щоб встигнути використати вразливість системи безпеки до того, як вона буде закрита. При відсутності якісного антивірусного пакета вразливість системи безпеки може довгий час залишатися непоміченою. Незважаючи на те, що перевірка файлів при зверненні може позначитися на продуктивності сервера, її переваги для безпеки значно перевершують будь-яке зниження пропускної здатності сервера. Деякі ділянки системи (наприклад, каталог, в якому зберігаються лог-файли HTTP-сервера) можна виключити з області перевірки, що також дозволить знизити вплив на систему.

Наступний за важливістю компонент програмного забезпечення – сам HTTP-сервер; найпопулярнішими альтернативами тут є IIS і Apache.

IIS – це компонент Microsoft Windows, популярний і поширений в силу простоти конфігурації веб-сервер.

Проте при його розгортанні потрібно пам'ятати про наступне:

- вимикати служби, що встановлюються за замовчуванням (наприклад, FTP або SMTP);
- відключити функцію перегляду каталогів, якщо вона не є необхідною, оскільки вона дозволяє відвідувачам бачити, які файли використовуються системою.

Необхідно відключити всі невикористовувані серверні розширення FrontPage. Також слід своєчасно встановлювати всі оновлення IIS – домогтися цього можна, включивши автоматичне оновлення за допомогою панелі управління Windows.

Apache – це веб-сервер з відкритим вихідним кодом, що відрізняється високими можливостями конфігурації і належним рівнем підтримки. Для його успішного розгортання потрібно більш детальне налаштування, але це в той же час забезпечує більшу ступінь контрольованості веб-сервера. Зазвичай сервери Apache працюють під управлінням Linux або BSD, але вони також можуть працювати і під Windows.

При налаштуванні Apache слід скористатися наступними рекомендаціями:

- відключити доступ до ресурсів за замовчуванням, включивши тільки необхідну функціональність ресурсів;
- вести журнал всіх звернень – це допоможе у виявленні підозрілої активності;
- підписка на розсилку Apache Server Announcement дозволяє своєчасно отримувати оновлення та виправлення для системи безпеки.

Якщо для веб-сайтів потрібно розширена функціональність, HTTP-сервер часто доповнюється серверним інтерпретатором PHP або ASP, або працюють за допомогою інтерфейсу CGI.

PHP є однією з найбільш поширених серверних скриптових мов. Вона відрізняється величезною базою функціонального коду, простим синтаксисом, адаптованим кодом і, що найбільш важливо, можливістю взаємодії з різними базами даних. **MySQL** – це одна з найбільш популярних СУБД (систем управління базами даних), які використовуються в поєднанні з PHP; причина полягає в її ефективності, багатій функціональності, а також простоті настройки і використання.

Мова PHP часто критикується за недостатній рівень безпеки, оскільки з часом в ній було виявлено безліч потенційних вразливостей. Проте, вона стабільно розвивається, а більшість вразливостей можна компенсувати за рахунок належної конфігурації або підвищення рівня безпеки розроблюваного коду.

Ось кілька порад по налаштуванню, що стосуються змінних у файлі *«php.ini»*:

- встановити змінну register_globals в значення off;
- встановити змінну safe_mode в значення on;
- в змінній open_basedir слід вказати базовий каталог веб-сайту;
- встановити змінну display_errors в значення off;
- встановити змінну log_errors в значення on;
- встановити змінну allow_url_fopen в значення off.

При установці MySQL створюється база даних «test», яка використовується за умовчанням, і відкрита обліковий запис «root» без пароля. Даному обліковому запису автоматично надається повний доступ до всіх інших баз даних на сервері. У зв'язку з цим необхідно виконати наступне:

- відразу змінити пароль облікового запису «root»;
- створити новий обліковий запис mysql і надати йому мінімально необхідні права;
- видалити базу даних «test» і відповідних користувачів.

SQL-ін'єкція використовується для атаки веб-сайтів, що працюють з базами даних. Можливість впровадження SQL-коду виникає, якщо в SQL-запитах використовуються невідфільтровані дані, що вводяться користувачами.

SQL-запити використовуються для добування інформації з бази даних, додавання інформації в базу даних, а також для зміни і видалення даних в базі. Багато сучасних веб-сайтів використовують скрипти і SQL для динамічного формування вмісту сторінки. У SQL-запитах часто використовуються дані, що вводяться користувачами; це може привести до загрози безпеки, оскільки хакери можуть спробувати впровадити у вхідні дані шкідливий SQL-код. Без належних заходів захисту такої код може бути успішно виконаний на сервері.

Розглянемо наступний PHP-код:

```
$ Firstname = $ _POST ["firstname"]; mysql_query ( "SELECT * FROM users WHERE first_name = '$ firstname'");
```

Після того, як користувач введе своє ім'я в веб-формі, SQL-запит поверне список всіх користувачів з тим же ім'ям. Якщо вказати в формі ім'я «Іван», то SQL-запит матиме такий вигляд:

```
"SELECT * FROM users WHERE first_name = 'Іван'"
```

Це допустима конструкція, яка спрацює так, як очікується. Але що трапиться, якщо замість імені ввести, наприклад, « '; drop table; # »? Тоді конструкція буде виглядати наступним чином:

```
"SELECT * FROM users WHERE first_name = '';DROP TABLE users; # '"
```

Крапка з комою дозволяє виконувати кілька наступних команд одна за одною. В результаті проста SQL-команда перетворюється в складну конструкцію:

```
SELECT * FROM users WHERE first_name = '';
```

Це призведе до того, що в базі даних буде цілком видалена відповідна таблиця, а символ «#», що стоїть у третьому рядку, призведе до того, що MySQL проігнорує решту рядка.

Наведена в прикладі вразливість особливо небезпечна, оскільки її можна використовувати для виведення закритих даних, зміни окремих полів або видалення інформації. Деякі СУБД також дозволяють виконувати системні команди за допомогою SQL.

У PHP є спеціальна функція *mysql_real_escape_string*, що видаляє з рядка потенційний код SQL-ін'єкції. Її слід використовувати для фільтрації всіх даних, які впроваджуються в SQL-інструкції.

Атаки веб-серверів можна розділити на дві категорії:

- *Локальні* атаки зазвичай спрямовані на крадіжку інформації або перехоплення управління на окремому веб-сервері.
- *Глобальні* атаки зазвичай спрямовані на кілька веб-сайтів і ставлять собі за мету зараження всіх їх відвідувачів.

Сервери також можуть заражатися через *локальну* мережу. Наприклад, сімейство червів *Fujacks* здатне заражати HTML-, PHP- і ASP-файли, розташовані на мережевих дисках.

Яндекс у 2014 році проаналізував ТОП-10 000 популярних заражених сайтів [3]. Виявилося, що переважна більшість використовують систему DLE (50%), також популярні WordPress (19%) і Joomla (19%). Найбільш незахищеними версіями виявилися: WordPress 3.2.1, 3.1.3 та 2.9.2 і CMS Joomla версії 1.5 (рис.2).

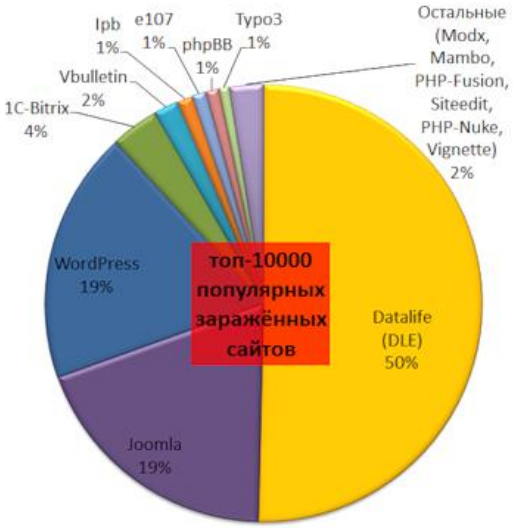

Рис.2. **Найбільш уразливі** CMS

Рекомендації, як запобігти злому CMS, досить прості:

1. Регулярно оновлювати CMS.
2. Не вказувати її тип і версію в коді сторінки.
3. Відмовитися від контрафактних версій систем.
4. Стежити за всіма даними, які користувач може ввести на сторінках сайту або передати серверним скриптам за допомогою запитів.
5. Скоротити число використовуваних сторонніх скриптів, модулів і розширень.
6. Не забувати про правила безпечної роботи в Інтернеті.
7. Перед установкою на веб-сервер ПО перевірити його уразливості і дізнатися про їх усунення за допомогою **The Open Source Vulnerability Database** (Уразливість бази даних з відкритим вихідним кодом**).**

Аналіз засобів захисту у популярних CMS.

Wordpress

У цієї CMS є багато модулів, які відповідають за безпеку [2]. Але немає такого, який би охоплював усі області, які потребують захисту. Розглянемо десять найбільш популярних WP-плагінів[4]:

1. Simple Backup – WordPress-плагін було розроблено *для створення і завантаження безкоштовних резервних копій* WordPress-сайта.

Вимоги: PHP 5.2 або вищі, WordPress 3.3 або новіше, Linux Style Server, mysqldump (для бекапа баз даних) і tar, zip, gzip, або bzip.

2. Ask Apache Password Protect – дуже незвичайний плагін для забезпечення безпеки. На відміну від аналогічних плагінів, він працює не на рівні додатку, а на рівні мережі, не використовуючи php, щоб запобігти можливим атакам. Ask Apache Password Protect був розроблений для зупинки атак ще до того, як ці атаки торкнуться конкретно вашого блогу.

Вимоги: веб-сервер Apache і підтримка хостингом файлів .htaccess.

3. Login Dongle – захищає всю інформацію про авторизацію за допомогою захисних питань і додаткового шару захисту.

Вимоги: WordPress 1.0 чи вище.

4. Sideways8 Custom Login and Registration. Плагін було розроблено так, щоб користувачі не бачили, як працює вбудована опція авторизації, реєстрації чи скидання паролю для входу на сайт.

Вимоги: WordPress 3.3 чи новіше.

5. Exploit Scanner. Цей плагін веде пошук по файлах WordPress і по БД для пошуку будь-яких ознак підозрілої шкідливої активності, а також перевіряє імена файлів. При цьому він не видаляє все підряд – право вибору лишає за адміністратором.

Вимоги: WordPress 3.3 або новіше.

6. WordPress AntiVirus. Простий у використанні плагін для автоматичного регулярного моніторингу всіх підозрілих "ін'єкцій". Він здатний попередити про можливу загрозу хакерського злому. Крім того, в ньому є підтримка різних мов.

Вимоги: PHP 5.1 і WordPress 2.8 і новіше.

7. Acunetix WP Security. Ще один безкоштовний плагін для забезпечення переліку захисних дій. Так, він сканує блог на помилки безпеки, пропонує рішення для знайдених вразливостей, приховує версію «движка», видаляє мета-теги з коду ядра і т.д.

Вимоги: WordPress 3.0 або вище, PHP5.

8. WordPress HTTPS (SSL). Цей плагін був створений як пакетне рішення (що включає 'private' і 'shared' SSL, примусову захищену авторизацію, панель адміністрування безпеки, пошук часткових помилок) для WordPress SSL.

Вимоги: WordPress 3.0 або вище.

9. Anti-spam plugin. Цей плагін блокує спам в коментарях в автоматичному режимі. Основні переваги – у відсутності "капчі", перевірочних питань або складних налаштувань.

Вимоги: WordPress 3.0 або новіше.

10. Theme Authenticity Checker. Цей плагін сканує всі файли теми і повідомляє, якщо знайде якийсь шкідливий код. Відмінний інструмент для боротьби з рекламою.

Вимоги: WordPress 3.0 або більш**е.**

Drupal

В цілому уразливості Друпал пов'язані з PHP-скриптами. А виявлення вразливих місць пов'язано з тим, що всі хакери знають цю мову програмування і готові вивчати коди CMS Drupal. Отже, в питанні безпеки результату досягає той, хто першим виявляє вразливість: зловмисник – щоб зламати, розробник – щоб поставити патч, виправити код, захистити ненадійне місце.

Виходячи з того, де можуть розташовуватися вразливі місця, можна виділити наступні причини виникнення вразливостей[2]:

- занадто простий пароль адміністратора – найчастіше злом в Інтернеті відбувається простим перебором паролів;
- неправильні атрибути файлів для *nix-систем, тобто зазвичай, завантажуючи сайт на хостинг, треба виставляти атрибути 644 для файлів і 755 папок (тобто без дозволу виконання для файлів і без можливості запису, редагування файлів);
- розробник сайту написав небезпечний модуль або PHP-скрипт (який приймає дані з форми і використовує їх без належної перевірки, або скрипт, який звертається до системних функцій, але може викликатися простими користувачами без обмеження прав);
- неправильне налаштування рівнів доступу на багатокористувацькі сайти;
- неправильне використання або налаштування модулів;
- дозвіл завантаження файлів з неправильними розширеннями, наприклад, .php.

Модуль **Security Review** (рис.3) автоматизує процес перевірки помилок, які можуть бути допущені під час налаштування сайту, роблить сайт небезпечним, схильним до відомих уразливостей і нестійким до спроб атак. Якщо результати перевірок потрібно записувати в журнал, то додатково потрібно включити модуль *Dblog*. Основною метою модуля є підвищення обізнаності адміністратора сайту про ті параметри, які впливають на безпеку. Результати перевірок можуть бути некоректними і залежати від унікальних чинників. Результати, які видає цей модуль, слід використовувати як додаткове джерело інформації про можливі помилки. Якщо будь-які пункти перевіряти не потрібно, то натисніть навпроти пункту посилання «Пропустити». Якщо потрібно виконати перевірку ще раз, то розгорніть форму «Запустити», яка знаходиться на сторінці модуля, і натисніть кнопку «Запустити перевірку». Натиснувши посилання «Деталі», напроти пункту, можна подивитися детальну інформацію про даному пункті і (або) знайдені помилки.

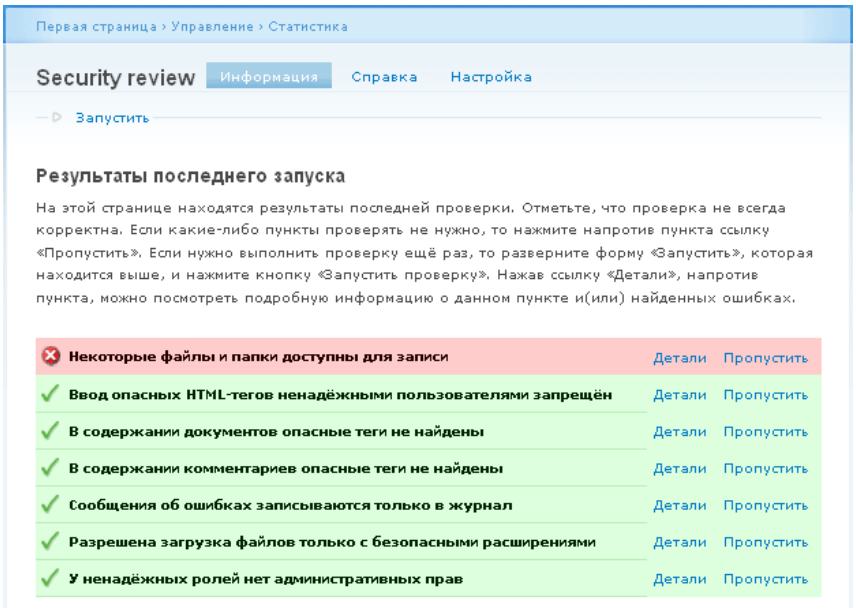

Рис.3. **Сторінка модуля** Security Review

Joomla

Плагін **jHackGuard** [7] призначений для захисту сайтів Joomla від хакерських атак і злому. Цей плагін був успішно протестований багатьма користувачами протягом декількох років і показав свою ефективність.

За словами розробників плагіна, jHackGuard є засобом захисту практично від усіх видів хакерських атак, а саме:

1. *Sql-injection (її називають SQL-ін'єкція)* – це атака, при якій проводиться вставка шкідливого коду в рядки, а потім цей код передається на SQL-Server для виконання.
2. *Віддалений URL / Включення файлів* – це віддалене виконання коду або виконання коду з підтримкою віддалених файлів.
3. *XSS атака* – це атака на вразливість, яка існує на сервері і дозволяє впровадити в HTML-сторінку довільний код, в якому може бути все що завгодно.

Основні параметри плагіна jHackGuard складаються з різних фільтрів, за якими плагіном відстежується все, що вводять користувачі, тим самим забезпечуючи безпеку сайту. За замовчуванням включені всі фільтри, але якщо потрібно, можна змінити налаштування плагіна (рис.4).

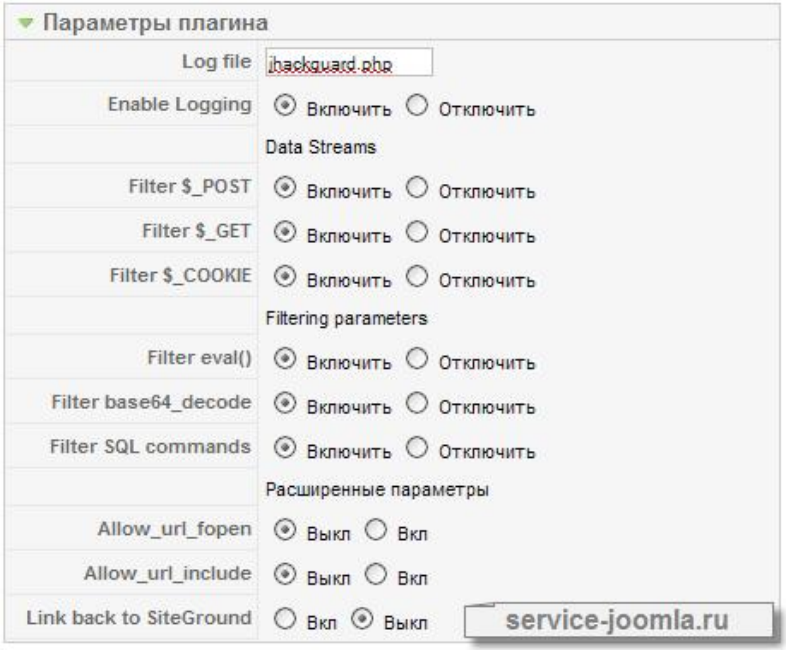

Рис.4. **Параметри плагіна** jHackGuard

Висновки. У роботі розглянуто основні відомості щодо безпеки веб-сайтів, а також проведено порівняльний аналіз модулів захисту трьох популярних CMS: Wordpress, Drupal та Joomla. На його результатах зроблено висновки:

1) захищеність веб-ресурсу залежить в основному від уважності та професіоналізму його розробника та адміністратора;

2) якщо питання стосується захисту CMS, то тут є кілька аспектів безпеки: захист коду, захист авторизації користувачів, захист файлів, і т.д.;

3) на основі зробленого аналізу можна сказати, що найбільш оптимальним у плані захисту є захист сайтів на Drupal.

1. Безопасность и защита сайтов [Електронний ресурс]. – Режим доступу: http://www.infosecurity.ru/iprotect/websec/ . – Назва з екрану.
2. Безопасность сайта на Друпал. Базовые понятия [Електронний ресурс]. – Режим доступу: http://tlito.ru/bezopasnost-sayta-na-drupal-bazovye-ponyatiya-video-doklad-nikolaya-shapovalova. – Назва з екрану.
3. Безопасность CMS [Електронний ресурс]. – Режим доступу: http://webportnoy.ru/articles/133/. – Назва з екрану.
4. 10 популярных плагинов для безопасности WordPress [Електронний ресурс]. – Режим доступу: https://wpcafe.org/plugins/10-populyarnyih-plaginov-dlya-bezopasnosti-wordpress/. – Назва з екрану.
5. Исследование безопасности сайтов на различных CMS [Електронний ресурс]. – Режим доступу: https://habrahabr.ru/company/ruward/blog/209950/. – Назва з екрану.
6. Україна стала світовим лідером за кількістю шкідливих сайтів [Електронний ресурс]. – Режим доступу: http://ua.korrespondent.net/tech/technews/3366180-ukraina-stala-svitovym-liderom-za-kilkistui-shkidlyvykh-saitiv . – Назва з екрану.
7. HackGuard – плагин, который защитит Joomla от хакерских атак и взлома [Електронний ресурс]. – Режим доступу: http://service-joomla.ru/plagini/item/44-jhackguard.html. – Назва з екрану.