УДК 681.5

О.Ф. Балашов, Ю.І. Скорін, М.Ю. Лосєв

Харківський національний економічний університет, Харків

СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ ТА БЕЗПЕКА WEB-САЙТУ ФАН-КЛУБУ «ІНЖЕК-МЕТАЛІСТ»

Розглядається актуальність проблеми забезпечення безпеки системи управління контентом Web-сайту фан-клубу «Інжек-Металіст». Проводиться аналіз та розглядаються вимоги до системи управління контентом з точки зору зручності користування та безпеки, а також подальший розвиток такої системи, виправляючи недоліки. Розглядається декілька можливих атак та вразливостей системи управління контентом та засоби їх запобігання. Описано платформу розробки, технологію проектування і архітектуру системи управління контентом Web-сайту фан-клубу «Інжек-Металіст».

Ключові слова: *Web-сайт, система управління контентом (CMS), технологія проектування, безпека.*

Вступ

Глобальна мережа Internet стала невід'ємною частиною життя величезної кількості людей. З кожним днем з'являється все більше нових Web-служб, які по функціональним можливостям та інтерфейсу не поступаються звичайним Windows-програмам.

З кожним роком Web-сайти стають все більш складними і інтерактивними, а інформація, розміщена на них, – повнішою і якіснішою. У простому випадку сайт є сукупністю статичних документів HTML (Hyper Text Markup Language). Інформація, розміщена на такому сайті, як правило, постійна.

Цей підхід в розробці є виправданим лише в тих випадках, якщо не потрібно отримувати інформацію від користувача або генерувати електронні документи автоматично.

Сучасні вимоги до Web-вузлів зобов'язали використовувати новий підхід до розробки. Інформація, що розміщується на Web-вузлах, повинна швидко оновлюватися.

Для вирішення цієї проблеми створюються системи управління вмістом (CMS), які дозволяють легко змінювати наповнення сайту через інтуїтивно зрозумілий інтерфейс.

"Система управління контентом", або CMS – останнім часом один з найпоширеніших способів адміністрування Web-порталу.

Система управління контентом – інформаційна система або комп'ютерна програма, яка використовується для забезпечення та організації спільного процесу створення, редагування та управління контентом.

Головною метою такої системи є можливість збирати в одне ціле та об'єднати на основі ролей та задач усі різнотипні джерела знань та інформації доступні, як в середині організації, так і з зовні, а також можливість забезпечення взаємодії робітників, робочих груп та проектів зі створеними базами знань, інформацією та даними так, щоб їх легко можна було знайти та не однократно використати звичним для користувача способом.

За допомогою розроблюваного сайту користувачі мають можливість дізнаватися про останні новини стосовно ФК «Металіст», фан-клубу «ІНЖЕК-МЕАЛІСТ», ЄВРО 2012, ХНЄУ, волонтерство та інше. Після реєстрації користувачі мають можливість спілкуватися особистими повідомленнями, замовляти атрибутику та квитки на футбольні матчі.

Адміністрування сайту може здійснювати користувач з правами адміністратора, в рамках даної дипломної роботи задача відноситься до задач начальника прес-центру. Адміністратор сайту має можливість повністю керувати контентом та профілями зареєстрованих користувачів.

Існують наступні способи роботи CMS:

Генерація сторінок по запиту. Системи такого типу працюють на основі зв'язки «модуль редагування → база даних → модуль представлення».

Модуль представлення генерує сторінку з контентом при запиті на основі інформації з бази даних.

Інформація в БД змінюється за допомогою модуля редагування. Сторінки заново створюються сервером при кожному запиті, а це створює навантаження на сервер. Але це навантаження може бути багатократно зменшено при використанні методів гешування, які маються в сучасних Web-серверах.

Генерація сторінок при редагуванні.

Системи цього типу при редагуванні сторінок вносять зміну у вміст сайту та створюють набір статичних сторінок. При такому способі втрачається інтерактивність між відвідувачами сайтів та контентом даного сайту.

Змішаний тип. Як зрозуміло із назви, цей тип поєднує в собі переваги перших двох.

Може бути реалізований шляхом гешування – модуль представлення генерує сторінку один раз, надалі вона по проходженню деякого часу буде в декілька разів швидше завантажуватися із кешу.

Другий підхід – збереження визначених інформаційних блоків на етапі редагування сайту і збирання сторінок з цих блоків при запиті відповідної сторінки користувачем [1].

Платформа розробки і архітектура системи управління контентом

Система управління контентом реалізується за допомогою мови програмування PHP. Середовищем розробки є NetBeans IDE, допоміжні технології CSS, JavaScript.

Системні вимоги для серверу: підтримка PHP 5.x з підтримкою бази даних MySQL 4.1.x 5.x Перевагою такої зв'язки (Apache + PHP + MYSQL) є:

можливість безкоштовного використання;

легке налаштування;

висока продуктивність;

достатній рівень безпеки.

Система управління вмістом у багатьох випадках є причиною діставання несанкціонованого доступу до Web-серверу.

Постійно публікуються все нові вразливості популярних CMS, можливість експлуатації яких ставить під загрозу безпеку всього серверу.

Захист системи управління вмістом дозволить значно підвищити захищеність серверу від зовнішніх погроз.

Всі звернення до бази даних і запити проходять через модуль безпеки запитів. Тим самим підвищується рівень безпеки системи, проте є і ряд недоліків цієї схеми.

Оскільки ще досить багато уразливих модулів елементів CMS, таких як модулі новин і каталоги, система безпеки повинна повністю контролювати кожен з модулів і процесів.

Існує декілька типів атак на системи управління контентом.

Той, що атакує може здійснити модифікацію рядка запитів так, щоб викликати SQL-injection або PHP-including. Майже для всіх CMS реалізацій PHP-including повністю неможливе.

Проте реалізація SQL-injection можлива у багатьох випадках. В кожній CMS є модуль, що дозволяє залишити будь-яку інформацію на сайті.

Тут можливо обійти перевірку введеної інформації і опублікувати на сайті спеціальний код, тим самим реалізувавши XSS атаку. Також існують вразливості, пов'язані з розмежуванням прав адміністраторів різного рівня доступу.

Впровадження SQL-коду (англ. SQL injection) – один з розповсюджених засобів взламу сайтів, що працюють з базами даних. Засіб засновано на впровадженні в запит довільного SQL-коду.

В залежності від типу СУБД, що використовується і умов впровадження, цей засіб може дати можливість атакуючому виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), дістати можливість читання і (або) запису локальних файлів та довільних команд на сервері.

Атака типу SQL injection може бути можливою з причини некоректної обробки вхідних даних, які використовуються в SQL-запитах [3].

Головним недоліком CMS є динамічна адресація. Така адресація сторінок дозволяє тому, що атакує легко змінювати значення переданих змінних, що ставить під загрозу всю систему безпеки.

Також використання динамічної адресації є небажаним для реєстрації сайту пошуковими системами. В CMS для вирішення цієї проблеми використовується mod_rewrite, проте не завжди підтримується компаніями, що надають послуги хостингу. Проте використання mod_rewrite не дозволяє захистити CMS від передачі модифікованих змінних.

В цьому випадку вся обробка і виявлення атак лягає на систему управління. Це допускає можливість використання досить широкого спектру атак, які реалізують SQL-injection в такий спосіб.

XSS (Cross Site Scripting) – тип вразливості інтерактивних інформаційних систем. XSS виникає, коли в сторінки, котрі генерує сервер, з якоїсь причини потрапляють скрипти користувачів.

Специфіка подібних атак полягає в тому, що замість безпосередньої атаки на сервер вони використовують вразливий сервер в якості засобу атаки на клієнта [4].

Вразливість, типа XSS, виникає в ситуаціях, коли дані, що були введені користувачем, виводяться без належної фільтрації в тексті html документа, що згенерував сервер.

Наприклад, може бути ситуація, коли дані, відправлені одним користувачем без фільтрації виводяться іншим користувачам. Типовою системою такого роду є чати, форуми.

Другим варіантом уразливості, є ситуація, коли частина HTTP GET запиту виводиться на цій же html сторінці тому ж користувачеві без належної фільтрації. Як правило – це ситуації, коли без належної фільтрації виводиться ідентифікатор сесії або інші GET параметри.

В першу чергу мається на увазі SQL-ін'єкція з можливістю впровадження benchmark() функції, таким чином, що один запит сильно навантажить вразливий сервер.

Про вразливість типа SQL-ін'єкція і у тому числі про використання функції benchmark в SQL запиті для проведення DOS атаки, розказано в цій статті [5].

Будь-яка система управління містить вразливості і досить часто адміністратори забувають про оновлення системи управління. Це може служити причиною злому сайту і всього сервера.

Оновлення системи управління є досить складною процедурою. Більшість систем управління не дозволяють здійснити оновлення повністю автоматично.

Потрібне їх доопрацювання руками адміністратора, що викликає боязнь оновлень системи управління. Цю проблему можливо вирішити лише за допомогою системи активних оновлень.

CMS здійснюють неповний аналіз параметрів, що передаються.

Саме на цьому рівні можна захистити систему управління від атак SQL-injection і PHP-including.

Для здійснення надійної фільтрації необхідно відкидати все спеціальний символи і залишати лише букви латинського алфавіту і арабські цифри. Тим самим можна гарантувати неможливість здійснення некоректних запитів SQL ще на рівні ядра системи управління.

Проведений аналіз системи управління контенту і механізмів реалізації атак дозволяє виробити вимоги, яким повинна задовольняти безпечна CMS.

Перш за все система управління має бути повністю захищена від модифікації рядка запиту. Це дозволить повністю уникнути досить широкого класу атак SQL-injection.

Дуже часто системи управління не дозволяють використовувати повністю статичні адреси і використовують пряму передачу значення змінних в запиті. Цю проблему необхідно повністю вирішити, використовуючи обробку помилки 404.

Для реалізації аналізу рядка запиту повинно бути реалізоване наступне:

1. Повне видалення спеціальних символів. В результаті виконання подібного фільтру повинні залишатися лише символи латинського алфавіту в нижньому регістрі, арабські цифри і “/”.

2. Далі має бути проведений логічний аналіз подібного запиту на предмет існування вказаної сторінки, директорії або модуля.

3. В разі, якщо вказана сторінка існує, відбувається генерація її коду на підставі інформації, отриманою з бази даних.

Висновок

Кількість модулів, що здійснюють публікацію інформації, що була отримана від відвідувачів сайту, має бути зведена до мінімуму.

Перевірка публікованої інформації повинна вироблятися найякісніше. Необхідно повністю унеможливити розміщення javascript і різних файлів із зовнішніх серверів. Це дасть практично повний захист від реалізації XSS атак.

Запропонований захист реалізується при створенні web-сайту на власній CMS фан-клубу «Інжек-Металіст».

Список літератури

1. Система керування вмістом. Вікіпедія — вільна енциклопедія. [Електроний ресурс]. – Режим доступу URL: http://uk.wikipedia.org/wiki/CMS.

2. Каталог CMS. CMS Magazine [Електроний ресурс]. – Режим доступу URL: http://www.cmsmagazine.ru/catalogue.

3. Внедрение SQL-кода. Википедия — свободная энциклопедии [Електроний ресурс]. – Режим доступу URL: http://ru.wikipedia.org/wiki/SQL_injection.

4. Межсайтовый скриптинг. Википедия — свободная энциклопедии. [Електроний ресурс]. – Режим доступу URL: http://ru.wikipedia.org/wiki/Xss.

5. Phoenix. SQL инъекция в MySQL сервере третей версии [Електроний ресурс]. – Режим доступу URL: http://www.securitylab.ru/contest/212101.php.

6. Балашов О.Ф. Система управління наповненням контентом сайту. Міжнародна науково-практична конференція молодих вчених, аспірантів та студентів «Актуальні проблеми науки та освіти молоді: теорія, практика, сучасні рішення», 21-22 квітня 2011р.: тези доповідей. Том І. – Х. : ХНЕУ, 2011. – С. 14 – 15.

Надійшла до редколегії 23.03.2012

Рецензент: д-р техн. наук, доц. К.О. Метєшкін, Харківська національна академія міського господарства, Харків.

СИСТЕМА УПРАВЛЕНИЯ КОНТЕНТОМ И БЕЗОПАСНОСТЬ WEB-САЙТА ФАН-КЛУБА «ИНЖЭК-МЕТАЛЛИСТ»

А.Ф. Балашов, Ю.И. Скорин, М.Ю. Лосев

Рассматривается актуальность проблемы обеспечения безопасности системы управления контентом Web-сайта фан-клуба «Инжэк-Металлист». Проводится анализ и рассматриваются требования к системе управления контентом с точки зрения удобства пользования и безопасности, а также дальнейшее развитии етакой системы, исправляя недостатки. Рассматривается несколько возможных атак и уязвимостей системы управления контентом и средства их предотвращения. Описаны платформу разработки, технологию проектирования и архитектуры системы управления контентом Web-сайта фан-клуба «Инжэк-Металлист».

Ключевые слова: *Web-сайт, система управления контентом (CMS), технология проектирования, безопасность.*

CONTENT MANAGEMENT SYSTEM AND SECURITY WEB-SITE FAN CLUB "INGEC-METALIST"

A.F. Balashov, Yu.I. Skorin, M.Yu. Losev

We consider the relevance of security content management system Web-site fan club "Ingec-Metalist". The analysis and reviewed the requirements for content management systems in terms of usability and security as well as further development of the system, correcting deficiencies. We consider several possible attacks and vulnerabilities content management system and means of prevention. We describe the development platform, technology architecture design and content management system Web-site fan-club "Ingec-Metalist".

Keywords: *Web, control the system by content (CMS), planning technology, safety.*