УДК 681.5

О.Ф. Лановий, І.В. Кобзев, С.В. Калякін

Харківський національний університет внутрішніх справ, Харків

СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ І БЕЗПЕКА WEB-САЙТІВ

Розглядаються вимоги до систем управління контентом з точки зору безпеки та несанкціоновану доступу до Web-сайтів, які створені за допомогою таких систем. Завдання створення максимально захищеної системи управління стоїть сьогодні особливо гостро. Описано платформи розробки і архітектури сучасних систем управління контентом, їх недоліки і вразливості з точки зору безпеки, а також запропонована концепція захищеної системи управління змістом Web-сайту.

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

Вступ

За останні декілька років глобальна мережа Internet стала невід'ємною частиною життя величезної кількості людей. Зараз практично в кожній компанії є свій Web-сайт, можливості якого дозволяють користувачеві здійснювати замовлення через Internet, або просто отримувати необхідну інформацію; більшість комп'ютерів мають можливість виходу до Internet. 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, а особливо на тих які представляють навчальні заклади, повинна швидко оновлюватися. Для вирішення цієї проблеми створюються системи управління вмістом (CMS), які дозволяють легко змінювати наповнення сайту через інтуїтивно зрозумілий інтерфейс. "Система управління сайтом", або CMS – останнім часом один з найпоширеніших способів адміністрування Web-порталу.

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 кешування, 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 кешування – модуль представлення генерує сторінку один раз, надалі вона по проходженню деякого часу буде в декілька разів швидше завантажуватися із кеша. Другий підхід – збереження визначених інформаційних блоків на етапі редагування сайту і збирання сторінок з цих блоків при запиті відповідної сторінки користувачем [1].

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.

Все існуючі CMS не гарантують безпеки Web-сервера. Ці проблеми пов'язані з вразливістю системи управління вмістом. Потрібно постійно оновлювати систему управління, стежити за правильністю функціонування і працездатністю. Для підтримки системи управління постійно потрібен системний адміністратор, що значно збільшує витрати на експлуатацію і не гарантує повної безпеки. Величезне значення має людський чинник, оскільки адміністратор завжди може забути встановити необхідне оновлення. Таким чином завдання розробки максимально захищеної системи управління стоїть сього-

дні особливо гостро.

Головними недоліками безкоштовної системи управління є повна відсутність підтримки і велика затримка виходу оновлень, що робить систему управління небезпечною. Проте щодня значна кількість сайтів створюється на безкоштовних системах управління вмістом. Перш за все це пов'язано з нерозумінням всіх масштабів погроз для компанії.

Використовуючи комерційну систему управління вмістом, можливо підвищити рівень безпеки корпоративного Web-сервера, проте далеко не всі виробники CMS піклуються про безпеку належним чином. Це пов'язано з тим, що основні зусилля компаній направлені на розробку інтерфейсу користувача. Дійсно, майже всі компанії, купуючи CMS, оцінюють її лише з точки зору зручності інтерфейсу, абсолютно забуваючи про безпеку.

Серед усіх систем управління, що використовуються в Україні можна виділити найбільш популярні і безкоштовні (Joomla, PHPNuke, e107, Wordpress, Joostina, DLE, Kasseler, Limbo, Sawanna,) та комерційні системи управління (Amiro CMS, NetCat, Bitrix, Twilight CMS, CMS СЕКУНДА) [2].

Слід пам'ятати, що високий рівень безпеки для CMS є необхідним, але важливим чинником все одно залишається інтерфейс користувача. Тому оцінку систем управління слід проводити по різних параметрах що стосується безпеки, призначеного для користувача інтерфейсу і вартості.

Міркуючи про системи управління вмістом, слід враховувати, що часто від користувачів цього класу програмного забезпечення не вимагається спеціальної технічної кваліфікації. Втім, будь-який користувач, будучи живою людиною, не застрахований від здійснення помилкових дій в повсякденній роботі. Предмет дослідження – те, як різні CMS реагують на помилкові дії користувачів. Здебільше всі програми - учасники списку в різних ситуаціях або не допускають помилкових дій, або повідомляють про них.

Оцінка безпеки комерційних систем управління вмістом є дуже трудомістким завданням. Перш за все це пов'язано з відсутністю документації по архітектурі систем управління і закритими вихідними кодами, оскільки всі комерційні CMS написані на PHP і закодовані за допомогою Zend Optimizer.

Проте для оцінки безпеки CMS можна скористатися статистичними даними по знаходженню різних вразливостей. Тут слід розуміти, що більшість розробників ретельно приховують всі знайдені помилки в безпеці, аби не підірвати комерційний успіх продуктів.

Проте загальна статистика серед лідерів ринку така, що менш популярні системи управління мають меншу кількість вразливостей. Перш за все це пов'язано з кількістю витраченого часу на пошук вразливостей. Але, розглядаючи статистику знайдених вразливостей, є можливість спроектувати систему управління сайтом так, щоб повністю унеможливити появу більшості типів вразливостей. Тут важливо правильно вибрати технологію і платформу для розробки системи.

Для безпеки і простоти використання системи управління вмістом велике значення має вибір платформи розробки.

Платформи розробки і архітектури сучасних систем управління вмістом

Для розробки системи управління вмістом необхідне використання 3-х складових:

- Web-сервер (Apache, IIS, NGINX);
- платформа розробки (PHP, PERL, ASP);
- сервер бази даних (MYSQL, PGSQL, MSSQL, ORACLE).

Для вибору платформи розробки і сервера бази даних потрібно виходити з наступного:

- найбільшої поширеності на різних Web серверах;
- показників продуктивності;
- можливостей масштабованості;
- рівня безпеки;
- вартості використання.

Для розробки CMS найбільш раціональним є використання зв'язки Apache + PHP + MYSQL. Основними перевагами такої зв’язки є:

- можливість безкоштовного використання;
- легке налаштування;
- висока продуктивність;
- достатній рівень безпеки.

Майже всі комерційні і безкоштовні системи управління працюють саме на Apache, PHP і MYSQL. Тому аналіз архітектури потрібно проводити з врахуванням того, що система буде розроблена саме на цій платформі.

Багато дорогих систем управління мають версію на більш досконалих платформах, які використовують як базу даних Oracle і MSSQL. Але оскільки спочатку всі системи проектуються так, щоб було можливо використовувати їх на безкоштовних платформах, то вибір більш коштовних платформ веде лише до підвищення продуктивності більшості CMS.

Важливо відзначити те, що 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 вмістом.

По-перше, 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.

По-друге, 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.

Недоліки і вразливості сучасних систем управління вмістом

Атаки SQL-injection.

Впровадження SQL-коду (англ. SQL injection) – один з розповсюджених засобів взлому сайтів і програм, 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 більшості 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.

Майже у всіх 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.

Атаки XSS.

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, різні системи управління.

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].

Концепція захищеної системи управління

Для розробки концепції захищеної системи управління необхідно також розглянути недоліки систем управління, що найбільш часто зустрічаються.

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. Тобто оновлення здійснюється автоматично. У більшості систем управління автоматичні оновлення здійснюються частково по запиту адміністратора з системи управління.

Більшість 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.

Основні вимоги до безпеки сучасних систем управління вмістом

Проведений аналіз сучасних систем управління вмістом 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 кода на підставі інформації, отриманою з бази даних.

Висновки

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. Запропонована концепція захисту реалізується при створенні порталу кафедри інформаційних систем та технологій в діяльності ОВС ХНУВС. Портал реалізується на базі безкоштовної системи управління контентом Joomla версії 1.0.15.

Список літератури

1. *Систе́ма керува́ння 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.*

Надійшла до редколегії 18.03.2010

Рецензент: д-р техн. наук, проф. О.С. Петров, Східноукраїнський національний університет ім. Володимира Даля, Луганськ.

СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ И БЕЗОПАСНОСТЬ WEB-САЙТОВ

О.Ф. Лановой, И.В. Кобзев, С.В. Калякин

Рассматриваются требования к системам управления контентом с точки зрения безопасности и несанкционированного доступа к Web-сайтам, которые созданы с помощью таких систем. Задача создания максимально защищенной системы управления стоит сегодня особенно остро. Описаны платформы разработки и архитектуры современных систем управления контентом, их недостатки и особенности с точки зрения безопасности, а также предложена концепция защищенной системы управления содержанием Web-сайта

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

CONTROL CONTENT SYSTEMS AND SAFETY OF WEB-SITES

O.F. Lanovoy, I.V. Kobzev, S.V. Kalyakin

Examined system requirement management content from point of safety and unauthorized division to the Web-sites which are created by such systems. The task of creation of the maximally protected control system stands today especially sharply. The platforms of development and architecture of the modern control content systems, their failings and features, are described from point of safety, and also conception of the protected control maintenance of Web-site system is offered

Keywords: *Web-site, control content (CMS) system, safety, platform, attack.*