УДК 65:591.524.1

БЕЗПЕКА WEB-САЙТІВ ЕКОНОМІЧНИХ ОБ'ЄКТІВ

Анатолій Бєгун,
к.е.н.
ДВНЗ «Київський національний економічний університет імені Вадима Гетьмана»

Анотація. Мета статті полягає в аналізі існуючих методів захисту Web-сайтів економічних об'єктів та визначення їх особливостей та підходів до впровадження економічно-ефективних заходів підвищення рівня їх захищеності. Вчасне виявлення потенційних загроз, виправлення помилок у налаштуванні програмного забезпечення серверу, Web-серверу та Web-додатків створює умови мінімізації можливостей зламу Web-сайтів економічних об'єктів. Це дозволить обрати найкращу виважену стратегію ефективного захисту і знизити витрати на їх підтримку.

Інформаційна безпека господарчого об'єкту надзвичайно важлива в будь-який час незалежно від форм власності. При правильному дотриманні усіх її правил і виконанні необхідних заходів ймовірність того, що цілісність інформаційної системи об'єкту буде набагато вища ніж у випадках, коли ігноруються всі або деякі з цих заходів та правил. Особливо це стосується захисту Web–додатків. Так, дані [1] свідчать, що серед задокументованих нових уразливостей в програмних продуктах 80% призначені Web-додаткам.

Захист Web-сайтів економічних об'єктів є безумовно актуальною темою дослідження багатьох вчених [1-3]. Відомо, що дуже часто виникають ситуації, в яких сайти зламують, потім перекручують, спотворюють або повністю знищують змістову інформацію. Це призводить до величезних матеріальних і моральних збитків із відповідними наслідками. Перш за все не існує повного захисту, так як інформаційні системи економічних об'єктів надзвичайно складні. По-друге, при створенні більшості програм залишається велика кількість помилок, які можуть використовувати зловмисники для нанесення значної шкоди.

У зв'язку із зростанням популярності глобальної мережі Інтернет багато фірм використовують її в цілях реклами своїх товарів та послуг. З розвитком Web-технологій організації почали переносити свою діяльність з продажу товарів та послуг в мережу Інтернет.

Web-сайти, що використовуються в економічній діяльності умовно можна поділити на такі категорії:

- сайт «візитна картка» – малобюджетний сайт ціна якого, як правило, не перевищує 2 – 3 тисячі умовних одиниць. Об'єм інформації сайтів такого виду коливається в проміжку від 1 до 25 Web-сторінок. Задача такого сайту полягає в тому, щоб надати клієнтам основну інформацію про фірму, про її товари та послуги. В деяких випадках сайт такого роду дозволяє фірмі скоротити витрати на рекламу в ЗМІ;

- корпоративний сайт – найбільш розповсюджена група для сайтів комерційних фірм. Сайти такого рівня складності замовляють більшість фірм, що намагаються вести свій бізнес з використанням Інтернет. Об'єм інформації корпоративного сайту зазвичай становить від 25 до 200 Web-сторінок. Професійно розроблений корпоративний сайт позитивно впливає на імідж фірми;

- Інтернет-портал – великий тематичний сайт, що містить від 200 до 600, а інколи і більшу кількість Web -сторінок. Такого роду сайти дозволяють майже повністю відмовитись від інших джерел реклами крім Інтернет;

- Інтернет-«стартап» – сайти дуже великого об'єму, що містять від 1000 Web-сторінок і більше. В залежності від того, який ступінь охвату тематики такі сайти поділяють на тематичні та загального користування. Ці сайти заміняють такі традиційні джерела інформації як ЗМІ.

Викладення основного матеріалу. Прийоми програмування Web-додатків в умовах дотримання політики інформаційної безпеки пов'язані з особливістю перевірки формату даних, що надсилають користувачі з форм, які можуть містити всіляку некоректну інформацію: код PHP, JavaScript, SSI, хакерські скрипти та інше [4]. Тому перше, що необхідно зробити – це уважно виконати фільтрацію усіх даних, які надсилаються користувачем.

Припустимо, що у Web-додатку сайту економічного об'єкту існує три форми введення: ім'я користувача, його e–mail та тіло повідомлення. Насамперед, обмежимо кількість даних, які передаються з форм введення таким чином

```
<input type=text name=username
maxlength=20>.
```

Дійсно, така дія не може претендувати на роль гарантованого захисту, але таким чином можливе обмеження користувача від випадкового введення значень форми довших ніж 20 символів. А для того, щоб у користувача не виникло бажання завантажити документ із формами введення й підправити параметр maxlength, встановимо з самого початку Web-додатку, що обробляє дані, перевірку змінної оточення Web–сервера *HTTP–REFERER:*

```
<?
$referer=getenv("HTTP_REFERER");
if (!ereg("^http://
www.myserver.com")) {
echo "hacker? he-he...\n";
exit;
}
?>
```

Дійсно, якщо дані передані не з форм документа, що перебуває на сервері *www.myserver.com*, то користувачу буде видане повідомлення про помилку та запропоновано вихід із програми. Насправді, і це теж не може слугувати гарантією того, що дані дійсно передані з документа. Зрештою, змінна *HTTP_REFERER* формується браузером, і ніхто не може перешкодити хакеру підправити код браузера, або просто «зайти телнетом» на 80-й порт і сформувати свій запит. Але подібний захист притаманний тільки від неосвічених хакерів. Втім, за власними спостереженнями, близько 80 % зловмисників на цьому етапі зупиняються й далі не діють. Іншим способом може бути винесення цього фрагменту коду в окремий файл і виклик його з усіх додатків, в яких потрібно робити таку перевірку.

Наступним етапом заходів захисту Web -сайтів економічних об'єктів є «тверда» фільтрація переданих даних. Насамперед, не будемо довіряти змінній `maxlength` у формах введення, а засобами функції `substr` «поріжемо» рядок

```
$username=substr($username,0,20);
```

і заборонимо використовувати порожні поля введення

```
if (empty($username)) {
echo "ім'я користувача не
може бути порожнім!";
exit;
},
```

крім букв кирилиці й латинського алфавіту, знака "_" (підкреслення), пропуску й цифр

```
if (preg_match("/[^(\w)|(\x7F-
\xFF) |(\s)]/",$username)) {
echo "невірний формат імені
користувача.";
exit;
}
```

У більш складних ситуаціях доцільно використовувати Perl–сумісні регулярні вирази (Perl–compatible Regular Expressions). Те ж саме можна зробити з використанням стандартні функції мови PHP ereg() та eregi() [].

Для поля введення адреси e-mail додамо в список дозволених символів знаки «@» і «.», інакше користувач

не зможе коректно ввести адресу. Проте, необхідно заборонити використання кирилиці, букви «й» та пропуску:

```
if (preg_match("/
[^(\w)|(\@)|(\.)]/
",$usermail)) {
echo "невірний формат адреси e-mail";
exit;
}
```

Поле введення тексту не будемо піддавати таким складним перевіркам – перебирати всі знаки, які можна використати, попросту немає необхідності, тому обмежимося використанням функцій nl2br() і htmlspecialchars() – це не дасть недобросовісному користувачу вставити в текст повідомлення html–теги. Є розробники, що вважають за необхідне використовувати деякі безпечні теги. Якщо є така необхідність – можна зробити якісь шаблони, що будуть замінятись на потрібні теги. Але ніколи не слід дозволяти користувачам використання тегів, що припускають підключення зовнішніх ресурсів – від тривіального <img> до небезпечного <bgsound>.

При тестуванні одного з Web-додатків першою поміченою помилкою був саме дозвіл на вставку картинок. Через декілька хвилин було знайдено файл, у якому перелічувалися IP–адреси, імена та паролі всіх присутніх у цей момент часу користувачів. Було послано тег <img src=http://myserver.com/myscript.pl>, у результаті чого браузери всіх користувачів, що були присутнім у той момент на сайті, викликали додаток myscript.pl з хосту myserver.com. А скрипт, перед тим як видати location на картинку, записував в лог-файл половину змінних оточення – зокрема QUERY_STRING, REMOTE_ADDR та інших. Для кожного користувача з вищезгаданим результатом не дозволяється вставка HTML–тегів у Web -додатках.

Для примітивних Web –сайтів, які перераховані вище, засобів вистачає, щоб зробити його більш-менш складним для атак зловмисників. Однак, для зручності, додатки звичайно містять деякі можливості для моделювання – на кшталт: можливість видалення некоректно введених користувачем повідомлень, які дозволені вузькому (або не дуже) колу осіб.

Припустимо, вся система адміністрування Web -додатку також складається із двох частин – сторінки зі списком повідомлень, де можна відзначати підлягаючі видаленню повідомлення, і безпосередньо Web -додатку, що видаляє повідомлення. Назвемо їх відповідно admin1.php та admin2.php.

Найпростіший і найбільш надійний метод аутентифікації користувача – розміщення скриптів у директорії, що захищена файлом «.htaccess». Для подолання такого захисту потрібно вже не додаток «зламати», а мати справу із Web -сервером. Однак, не завжди такий спосіб придатний до використання – іноді виникає необхідність проводити авторизацію засобами самого додатка.

Перший, найпростіший спосіб – авторизація засобами HTTP – через код 401. Побачивши такого коду повернення, будь-який браузер висвітить вікно авторизації та запросить ввести логін і пароль. А надалі браузер при отриманні коду 401 буде намагатися підставити Web -серверу поточні для даного сеансу логін та пароль, і тільки у випадку невдачі зажадає повторної авторизації. Приклад коду для такої авторизації існує в документації:

```
if (!isset($PHP_AUTH_USER)) {
Header("WWW-Authenticate:
Basic realm=\"My Realm\");
Header("HTTP/1.0 401
Unauthorized");
exit;
}.
```

Розмістимо цей фрагмент коду на початку Web-додатку admin1.php. Після виконання додатку виникнуть дві встановлені змінні – $PHP_AUTH_USER і PHP_AUTH_PW,

у яких відповідно будуть міститися ім'я й пароль, що введені користувачем. Наприклад, їх можна, перевірити за SQL-базою даних:

```
$sql_statement="select password from peoples where name="$PHP_AUTH_USER"";
$result = mysql($dbname, $sql_ statement);
```

// Перенаправити користувача в зону авторизації, якщо такого користувача не існує

```
if (mysql_numrows($result) != 1) {
Header("HTTP/1.0 401 Auth Required");
Header("WWW-authenticate: basic realm=\"My Realm\"");
exit;
}
$rpassword = mysql_result ($result,0, "password");
$sql_statement = "select password (`$PHP_AUTH_PW`)";
$result = mysql($dbname, $sql_ statement);
$password = mysql_result($result,0);
if ($password != $rpassword) {
Header("HTTP/1.0 401 Auth Required");
Header("WWW-authenticate: basic realm=\"My Realm\"");
exit;
}
```

Тобто – перевірка наявності одного і тільки одного користувача в базі. Точно таку ж перевірку на авторизацію варто вмонтувати й у скрипт admin2.php. Якщо користувач «продвинутий», то він приходить до admin2.php через admin1.php, і вже є авторизованим до ніяким повторним запитам на введення імені й паролю не буде – браузер сам передасть пароль. На протилежне – виконується вивід повідомлення про помилку та відбувається вихід з програми. На жаль, не завжди вдається скористатися алгоритмом авторизації через код 401 і доводиться виконувати її тільки засобами додатка. У загальному випадку модель такої авторизації може бути наступною:

- користувач один раз авторизується за допомогою форми у Web -додатку, що перевіряє правильність імені й пароля;

- інші додатки захищеної частини яким-небудь чином перевіряють факт авторизації користувача.

Така модель має назву сесійної – після проходження авторизації відкривається так названа «сесія», протягом якої користувач має доступ до захищеної частини системи. Якщо сесія закінчилася – доступ закривається. На цьому принципі, зокрема, будується більшість різноманітних Web-додатків: користувач може одержати доступ тільки після того, як пройде процедуру входу. Основна складність даної схеми полягає в тому, що усі додатки захищеної частини якимось чином повинні знати про те, що користувач, що посилає дані, успішно авторизувався.

Розглянемо кілька варіантів розв'язання цієї складності:

1. Після авторизації усі додатки захищеної частини викликаються з якимось параметром, наприклад adminmode=1. Але кожний, кому відомий цей параметр «adminmode», може сам сформувати URL і увійти в режимі адміністрування. Крім того – немає можливості відрізнити одного користувача від іншого.

2. Скрипт авторизації може якимнебудь чином передати ім'я користувача іншим Web-додаткам. Поширено в багатьох Web-додатках – для того, щоб відрізнити, де чиє повідомлення йде, поруч із формою типу text для введення повідомлення, прилаштовується поле типу hidden, де вказується ім'я користувача. Але цей спосіб теж не надійний, тому що хакер може скачати документ із формою до себе на диск і поміняти значення форми hidden. Деяку користь тут може принести вищезгадана перевірка HTTP_REFERER – але, як я вже говорив, ніяких гарантій вона не дає.

3. Визначення користувача по IP–адресі. У цьому випадку, після проходження авторизації, де-небудь у локальній базі даних (sql, dbm, або в звичайному текстовому файлі) зберігається поточний IP користувача, а всі додатки захищеної частини дивляться в змінну REMOTE_ADDR і перевіряють, є чи така адреса в базі. Якщо є – виходить, авторизація була, якщо немає – виводиться повідомлення про помилку і вихід із програми. Це більш надійний спосіб – не пройти авторизацію й одержати доступ удається лише в тому випадку, якщо з того ж IP працює інший користувач, що успішно авторизувався. Однак, з огляду на поширеність проксі-серверів і технології IP–Masquerading – це цілком реально.

4. Єдиним, відомим простим і досить надійним способом верифікації особи користувача є авторизація за допомогою випадкового ідентифікатора користувача (uid). Розглянемо його більш докладно.

Після авторизації користувача скрипт, що проводить авторизацію, генерує досить довге випадкове число:

```
mt_srand((double)microtime()*1000000);
$uid=mt_rand(1,1000000);
```

З цим числом Web –додаток виконує наступні дії:

а) заносить у локальний список користувачів, що авторизувалися;

б) видає користувачу.

Користувач при кожному запиті, крім іншої інформації відправляє серверу ще свій ідентифікатор. При цьому в документі з формами введення буде присутній поряд з іншими формами тег виду:

```
<input type=hidden name=uid
value=1234567890>
```

Тип поля uid – невидиме для користувача, але він передається скрипту захищеної частини Web -додатку. Той порівнює переданий ідентифікатор користувача з тим, що зберігається в локальній базі й або виконує свою функцію, або видає повідомлення про помилку і завершує роботу.

Єдине, що необхідно зробити при такій організації захисту– періодично чистити локальний список ідентифікаторів і/або зробити для користувача кнопку «вихід», при натисканні на яку локальний ідентифікатор користувача зникне з бази даних на сервері – сесія закрита.

Деякі програмісти використають у якості ідентифікаторів (uid) не «одноразове» число, що динамічно генерується, а пароль користувача. Це припустимо, але це неправильно, оскільки пароль користувача звичайно не міняється від сесії до сесії, а це означає, що хакер зможе сам відкривати сесії.

Така модель може бути використана взагалі скрізь, де потрібна ідентифікація користувача.

На закінчення варто згадати й про таку корисну подію, як ведення логів. Якщо в кожну з описаних процедур вмонтувати можливість занесення події в лог-файл із вказівкою IP–адреси потенційного зловмисника, то у випадку реальної атаки визначити хакера буде набагато простіше, оскільки хакери звичайно обирають типи атак, які поступово ускладнюються. Для визначення IP–адреси бажано використовувати не тільки стандартну змінну REMOTE_ADDR, але й менш відому HTTP_X_FORWARDED_FOR, що дозволяє визначити IP-адресу користувача, яка перебуває за проксі–сервером і якщо проксі-сервер дозволяє таку дію.

При веденні лог-файлів, необхідно пам'ятати, що доступ до них повинен бути тільки у адміністратора. Найкраще, якщо лог-файли будуть розташовані за межами дерева каталогів, яке доступне через WWW. У випадку відсутності такої можливості необхідно створити окремий каталог для лог-файлів і заборонити до нього доступ за допомогою .htaccess (Deny from all).

ВИСНОВКИ

На сьогоднішній день не існує єдиного підходу до оцінки якості впровадження систем захисту Web -сайтів економічних об'єктів. Все залежить від

цілей, що ставляться перед системами такого класу. Можливо привести тільки загальні рекомендації, що допоможуть вищому керівництву в прийнятті рішення щодо впровадження системи захисту:

- по-перше, витрати на забезпечення інформаційної безпеки не мають перевищувати вартість самого Web -сайту економічного об'єкту;

- по-друге, витрати на забезпечення інформаційної безпеки не мають перевищувати величину збитків, що можуть бути нанесені в випадку атаки на Web-сайт економічного об'єкту.

Основна проблема тут полягає в визначенні розміру збитків, які можуть бути нанесені зловмисниками Web-сайту економічного об'єкту.

ЛІТЕРАТУРА

1. *Андрианов В.В.* Обеспечение информационной безопасности бизнеса/ В.В. Андрианов, С.А. Зефиров, В.Б. Голованов, Н.А. Голдуев. – М.: ЦИПС и Р: Альшина Паблишерз, 2011. – 373 с.

2. *Бєгун А.В.* Аналіз загроз інформації Web -порталу через атаки на Web-додатки. - Зб. Моделювання та інформаційні технології в економіці, №80, 2009. – С. 101-107.

3. *Дериева Е.* Symantec: безопасных браузеров нет//Компьютерное обозрение. – 2006. -№38.- С. 54-55.

4. *Томсон Лаура, Веллинг Дюк.* Разработка Web-приложений на PHP и MySQL. – Киев: «ДиаСофт», 2001 г., 672 с.