8. Имитационное моделирование экономических систем : [учебное пособие] / [Ю.Г. Лысенко, Г.С. Овечко, А.В. Овечко и др.]. – 1-е изд. ; ред. Ю.Г. Лысенко. – Донецк : ООО «Юго-Восток, Лтд», 2006. – 259 с.
9. Концева В.В. Фінансові потоки в логістичних системах / В.В. Концева, С.С. Костенко // Вісник Національного транспортного університету. – 2009. – № 19. – С. 361–364.
10. Поддєрьогін А.М. Ефективність управління грошовими потоками підприємства / А.М. Поддєрьогін, Я.І. Невмержицький // Фінанси України. – 2011. – № 6. – С. 119–127.
11. Покараева Н.Г. Финансовая логистика: вчера, сегодня, завтра / Н.Г. Покараева // РИСК: Ресурсы, информация, снабжение, конкуренция. – 2009. – № 2. – С. 127–130.
12. Сваталова Ю.С. Особенности управления финансовыми потоками логистической системы холдинга / Ю.С. Сваталова, Т.А. Козенкова // Экономический рост и конкурентоспособность России: тенденции, проблемы и стратегические приоритеты. – 2011. – № 4. – С. 147–152.
13. Supply chain logistics management / [D.J. Bowersox, D.J. Closs, M.B. Cooper]. – Vol. 2. – New York, NY : McGraw-Hill, 2002. – 402 p.
14. Fundamentals of logistics management / [D.B. Grant, D.M. Lambert, J.R. Stock, L.M. Ellram]. – McGraw-Hill/Irwin, 2005. – 512 p.
15. Supply Chain System Financial Logistics Funds Balance of Supply and Demand Based on the System Dynamics / [Y. Wang, D.L. Yu, Y.L. Peng, H.Y. Hao] // Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on October, 2008. – P. 1–5.

УДК 004.852

Нємкова О.А.,
кандидат фізико-математичних наук, доцент,
доцент кафедри економіки,
Львівський інститут
Університету банківської справи
Смолинець Ю.А.,
аспірант кафедри економіки,
Львівський інститут
Університету банківської справи

МЕТОДИ ПОПЕРЕДЖЕННЯ XSS-АТАК НА УРЯДОВИХ НАВЧАЛЬНИХ РЕСУРСАХ

Нємкова О.А., Смолинець Ю.А. Методи попередження XSS-атак на урядових навчальних ресурсах. У статті визначено фактори, що пояснюють причини XSS-атак, а також методи їх попередження на урядових навчальних ресурсах. Проаналізовано статистику кількості здійснених атак за 2016–2017 роки. Наведено приклади шкідливого коду для здійснення XSS-атак. Запропоновано систему заходів для уникнення XSS-атак.

Ключові слова: XSS, Cross-site-scripting, методи атак, урядові навчальні ресурси, попередження атак.

Немкова О.А., Смолинець Ю.А. Методы предотвращения XSS-атак на государственных обучающих ресурсах. В статье определены факторы, которые объясняют причины XSS-атак, а также методы их предотвращения на государственных обучающих ресурсах. Проанализирована статистика количества произведенных атак за 2016–2017 годы. Приведены примеры вредного кода для осуществления XSS-атак. Предложена система мероприятий для устранения XSS-атак.

Ключевые слова: XSS, Cross-site-scripting, методы атак, государственные обучающие ресурсы, предотвращение атак.

Nemkova O.A., Smolynets Y.A. Methods of preventing XSS-attacks on government educational web-resources. In the article were defined factors, which can explain reasons of XSS-attacks and methods of preventing them on government educational resources. The statistics of the number of attacks carried out for 2016–2017 years was analyzed. Examples of malicious code are provided for performing XSS attacks. System of events for preventing XSS-attacks was suggested.

Key words: XSS, Cross-site-scripting, attack methods, government educational web-resources, preventing attacks.

Постановка проблеми. Сьогодні, у період стрімкого розвитку інформаційних технологій та мережі Інтернет, все більше створюється безліч веб-ресурсів різного призначення (від комерційних до навчальних).

Однак разом із розвитком технологій розвиваються і загрози, що можуть звести нанівець усю працю творців веб-ресурсів. Саме тому під час створення та підтримки веб-ресурсів важливим елементом його архі-

тектури є попередження та захист від хакерських атак. У статті розглянутий один із найнебезпечніших видів атак на інформаційні ресурси, а саме XSS-атаки.

Аналіз останніх досліджень і публікацій. Згідно з даними "Akamai's State of the Internet Security Report" кількість XSS-атак у першій чверті 2017 року зросла на 39% порівняно з 2016 роком, а частка таких атак на інформаційні ресурси становить 10% усіх хакерських атак у світі [1].

Результати цього дослідження свідчать про те, що під час розроблення великої кількості веб-ресурсів не враховується можливість хакерських атак, зокрема XSS. У Національний базі даних вразливостей ("National Vulnerabilities Database"), створеній за сприяння "NIST", кількість XSS вразливостей зросла на понад ніж 200% порівняно з 2016 роком [2]. Ці дані ще раз демонструють нам, що безпека веб-ресурсів повинна бути на першому місці, особливо якщо це стосується урядових навчальних ресурсів, що можуть містити конфіденційну інформацію.

Формулювання цілей статті. Метою статті є аналіз одного з найпопулярніших видів хакерських атак та визначення методів протидії хакерським атакам, зокрема XSS.

Виклад основного матеріалу. XSS є найбільш поширеною атакою на веб-ресурси.

Атаки XSS виникають, коли додаток включає дані, що надаються користувачем, у сторінку, яка відправляється в браузер без належної перевірки або фільтрування змісту. Існують три відомі типи атак XSS:

1) збережені;

2) відображені;

3) атаки XSS, основані на об'єктній моделі документа (DOM).

Виявити більшість атак XSS досить легко за допомогою тестування або аналізу коду.

Основною метою XSS-атак є отримання контролю над віддаленою системою або з метою впровадження певного коду, або з метою отримання із сайту будь-яких конфіденційних даних (баз даних, номерів кредиток, адрес електронних скриньок тощо). Ця проблема зараз є дуже поширеною, адже найбільшою цінністю у сучасному світі є інформація. Інформація, що міститься на урядових навчальних ресурсах, є надзвичайно важливою, адже зазвичай під час реєстрації на навчальних ресурсах використовуються ті самі облікові записи користувачів, які необхідні для доступу до робочої інформації та кореспонденції, іншими словами, ті самі логін та пароль, які необхідні для доступу до робочого середовища держслужбовців. Саме за цією інформацією і полюють хакери, адже завдяки ним вони можуть отримати доступ до урядових даних. Безпека цих даних є задачею № 1 під час створення та розгортання веб-ресурсів, особливо урядових.

XSS-атака є одним з найулюбленіших видів атак у хакерів. Абревіатура "XSS" розшифровується як "Cross Site Scripting", або «міжсайтовий скриптинг». Першу літеру "C" замінили на "X", оскільки абревіатура CSS вже зайнята, позначає «Каскадні таблиці стилів» і застосовується у веб-програмуванні.

XSS-атака – це атака на вразливість, яка існує на сервері, що дає змогу впровадити в згенеровану сервером HTML-сторінку якийсь довільний код, в якому може бути взагалі все, що завгодно, а також передавати цей код як значення змінної, фільтрація по якій не працює, тобто сервер не перевіряє цю змінну на наявність в ній заборонених знаків -, <,>, ‘, ". Значення цієї змінної передається від згенерованої HTML-сторінки на сервер в скрипт шляхом відправлення запиту.

На наступному кроці PHP-скрипт у відповідь на цей запит генерує HTML-сторінку, в якій відображаються значення потрібних зловмиснику змінних, а також відправляє цю сторінку на браузер зловмисника.

Тобто, кажучи простіше, XSS-атака – це атака за допомогою вразливостей на сервері на комп'ютери клієнтів.

XSS-атака найчастіше використовується для крадіжки Cookies. В них зберігається інформація про сесії перебування користувача на сайтах, що й потрібно хакерам для перехоплення управління особистими даними користувача на сайті в межах, поки сесія не буде закрита сервером, на якому розміщено сайт. В Cookies зберігається зашифрований пароль, під яким користувач входить на даний сайт. За наявності необхідних утиліт хакери розшифровують даний пароль і отримують постійний і нічим необмежений доступ до акаунтів користувача на різних ресурсах.

Інші можливості XSS-атак.

1) Під час відкривання сторінки відкривається велика кількість непотрібних вікон.

2) Переадресація на інший сайт (наприклад, на сайт конкурента).

3) Завантаження на комп'ютер користувача скрипта з довільним кодом (навіть шкідливого) шляхом втілення посилання на виконуваний скрипт зі стороннього сервера.

4) Крадіжка особистої інформації з комп'ютера користувача, наприклад Cookies, інформації про відвідані сайти, версії браузера та операційної системи, що встановлено на комп'ютері користувача, IP-адреси комп'ютера користувача.

5) XSS-атака може бути проведена не лише через сайт, але й через вразливості в браузері чи інших клієнтах. Тому рекомендується частіше оновлювати використовуване програмне забезпечення.

6) Проведення XSS-атак через використання SQL-коду.

Можливостей у XSS-атак досить багато, зловмисник може опанувати особистою інформацією, а це дуже неприємно. До того ж XSS-атака завдає шкоди виключно клієнтським машинам, залишаючи сервер в повністю робочому стані, і в адміністрації різних серверів часом мало стимулів встановлювати захист від цього виду атак.

Розрізняють XSS-атаки двох видів: активні і пасивні. За першого виду атаки шкідливий скрипт зберігається на сервері і починає свою діяльність під час завантаження сторінки сайту в браузері клієнта. За другого виду атак скрипт не зберігається на сервері, а шкідлива дія починає виконуватися тільки в разі певної дії користувача, наприклад під час натискання на сформоване посилання [3].

Розглянемо приклади шкідливого коду на мові програмування PHP.

Наприклад, поле для вводу логіна користувача має такий вигляд:

<input name="username" value="<? echo $_GET [‘username’] ?>">

Під час підтвердження входу на сайт ім'я користувача передається за допомогою параметрів $_GET. Це

означає, що коли користувач натискає на кнопку входу на сайті, то формується відкритий запит, за якого всі параметри запиту є доступними (вони відображені в адресній стрічці браузера). Ця стрічка має такий вигляд:

http://www.server.com/index.php?username="username"

Цей спосіб запитів на сервер є досить небезпечним, адже для того, щоб отримати, наприклад, дані COOKIE, зловмиснику достатньо написати такий код в адресній стрічці:

http://www.server.com/index.php?username="><script> alert(document.cookie)</script>

Після цього HTML-код сторінки міститиме шкідливий код і виглядатиме таким чином:

<input name="username" value=""><script>alert (document.cookie)</script>">

Цей код не нестиме жодної загрози користувачу, лише відображатиме COOKIE користувача, однак цей приклад чітко демонструє можливості XSS-атак. Варто зауважити, що код може бути значно небезпечніше, і, наприклад, будучи вставленим на сторінку Гостьової книги сайту, він відображатиме COOKIE усім користувачам, які відвідають цю сторінку.

Щоб забезпечити захист користувачів веб-ресурсу від такого типу вразливостей, достатньо фільтрувати запити на сервер, використовуючи синтаксис регулярних виразів:

```
<?php
 // Видаляємо всі символи крім букв та цифр
  $username = preg_replace("/[a-z0-9]/i", "", $_GET [‘username’]); ?>
```

Також можна загострити увагу користувачів, додавши вивід повідомлення про заборонені символи у полі:

```
<?php // Перевіряємо всі символи на літери та цифри if(!(preg_match("/^([a-z0-9]*)$/i",$_GET[‘username’]))) { header("plz_die.php"); } ?>
```

Під час передачі даних на сервер варто приділяти увагу ВСІМ змінним, що передаються від користувача, а саме GET, POST, COOKIE, адже в будь-яку з них може буде вбудований шкідливий код. Особливу увагу варто надавати полям вводу паролів, оскільки в них не прийнято вводити різного виду обмеження.

Мова програмування PHP містить декілька функцій, які допомагають забезпечити захист веб-додатків. Однією з таких функцій є htmlspecialchars(), яка гарантує, що будь-який введений користувачем код (javascript, php) буде відображатись, але не буде виконуватись.

Синтаксис функції має такий вигляд:

string htmlspecialchars (string str [, int quote_style [, string charset]])

Через перший обов’язковий параметр str функції передається текст для обробки, який повертається після обробки як результат роботи.

Другий необов’язковий параметр quote_style задає режим обробки одинарних та подвійних лапок. За замовчуванням даний параметр відповідає константі ENT_COMPAT, тобто в цьому режимі подвійні лапки замінюються символом """, при цьому одинарні лапки залишаються без змін. Крім цього параметр може приймати два інші значення: ENT_QUOTES та ENT_NOQUOTES. В першому випадку крім подвійних лапок обробці піддаються також одинарні лапки, які замінюються символом "'". Значення параметра ENT_NOQUOTES задає режим, в якому жоден з видів лапок не замінюється, тобто не потрапляє під обробку.

Останній параметр charset задає кодування, наприклад "cp1251" чи "KOI8-R".

Ця функція призначена для відображення коду та HTML-розмітки на Web-сторінці, але введені користувачем дані варто пропускати через неї для запобігання неприємностей.

Іншою корисною функцією є stripslashes(), яка призначена для видалення зворотних слешів (бекслеші) та має такий вигляд:

string stripslashes (string str)

Функція приймає єдиний параметр str зі стрічкою для обробки. Результатом роботи функції є стрічка str, в якій видаляються екрануючі бекслеші (‘ перетворюються в ‘, подвійні бекслеші (\) перетворюються в одиночні () тощо). [4]

Висновки. Сучасні методи атак на урядові веб-ресурси є основним елементом інформаційної зброї, за допомогою якої розгортається інформаційна війна у цифровому світі.

Атаки, що базуються на методі використання вразливостей коду, можна завжди попередити, якщо приділяти під час розроблення веб-ресурсу значну увагу безпеці та правильності написання коду. Як правило, такі методи атак на веб-ресурси мають успіх, якщо система захисту не була чітко спланована та реалізована з боку розробників.

Результати аналізу свідчать про те, що велика кількість веб-ресурсів, зокрема урядових, є незахищеними від XSS-атак, оскільки ця вразливість ігнорується на етапі розробки веб-ресурсу або під час планування робіт. Для того щоб запобігти XSS-атакам, розробникам слід виконати такі прості кроки.

1) Заборонити включення безпосередньо параметрів $_GET, $_POST, $_COOKIE в згенеровану HTML-сторінку. Рекомендується використовувати альтернативні функції і параметри.

2) Заборонити завантаження довільних файлів на сервер, щоб уникнути завантаження шкідливих скриптів. Зокрема, рекомендується заборонити завантаження на сервер файлів різних типів скриптів і HTML-сторінок.

3) Всі завантажені файли зберігати в базі даних, а не в файловій системі. Структури даних це не порушує (навіть навпаки), а проблем може бути значно менше.

Під час розширення функціональності сайту зростає можливість проведення XSS-атак, тому розширення слід проводити з обережністю і регулярним тестуванням [3].

Список використаних джерел:

1. Akamai’s State of the Internet Security Report, Akamai [Електронний ресурс]. – Режим доступу : https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report.
2. The National Vulnerabilities Database, compiled by NIST [Електронний ресурс]. – Режим доступу : https://nvd.nist.gov.
3. Хакерська атака на сайт [Електронний ресурс]. – Режим доступу : http://thelocalhost.ru/xakerskaya-ataka-na-sajt.
4. Безпечне програмування PHP [Електронний ресурс]. – Режим доступу : http://www.php.su/articles/?cat=security&page=012.