Технічні науки

УДК 004.056.[53+57]

**Стах Т. В.**
магістрант
Львівський національний університет
"Львівська політехніка"
**Грицюк Ю. І.**
доктор технічних наук,
професор кафедри програмного забезпечення
Львівський національний університет
"Львівська політехніка"

**Стах Т. В.**
студентка
Львовский национальный университет
"Львовская политехника"
**Грыцюк Ю. И.**
доктор технических наук,
профессор кафедры программного обеспечения
Львовский национальный университет
"Львовская политехника"

**Stakh T. V.**
student
Lviv National University "Lviv Polytechnic"
**Hrytsiuk Yu. I.**
Doctor of Engineering, Professor,
Professor of Software Department
Lviv National University "Lviv Polytechnic"

# НАЙПОШИРЕНІШІ СУЧАСНІ АТАКИ НА ВЕБ-ДОДАТКИ
# САМЫЕ РАСПРОСТРАНЁННЫЕ АТАКИ НА ВЕБ-ПРИЛОЖЕНИЯ
# THE MOST COMMON MODERN ATTACKS ON WEB APPLIC TIONS

***Анотація:*** *Висвітлено сучасні проблеми хакерських нападів на веб-додатки. Описано такі атаки, як небезпечні перенаправлення та посилання, атаки, пов'язані з маніпуляцією сесіями користувача, та SQL*

*ін'єкції. Наведені приклади містять реалізацію атак такими мовами програмування як Java, PHP, C# та Ruby.*

***Ключові слова:*** *мережа Інтернет, хакер, захист додатків, програмне забезпечення, браузер, хост, небезпечне перенаправлення, посилання, SQL ін'єкція, бази даних.*

***Аннотация:*** *Освещены современные проблемы хакерских нападений на веб-приложения. Описаны такие атаки, как опасные перенаправления и ссылки, атаки, связанные с манипуляцией сессиями пользователя и SQL инъекции. Примеры содержат реализацию на таких языках программирования как Java, PHP, C# и Ruby.*

***Ключевые слова:*** *сеть Интернет, хакер, защита приложений, программное обеспечение, браузер, хост, опасное перенаправление, ссылки, SQL инъекция, базы данных.*

***Summary:*** *Covering modern problems of hacker attacks on web applications. Giving detailed description of such attacks as dangerous redirections and links, attacks related on the user sessions manipulations and SQL injections. Examples include implementation on such programming languages as Java, PHP, C# and Ruby.*

***Key words:*** *Internet network, hacker, application security, software, browser, host dangerous redirect, links, SQL injection, database.*

## Вступ

На сьогодні розробка веб-додатків стає все більш популярною. Веб хости постачають нас всім необхідним: Google Docs, онлайн калькулятори, пошта, хмарні сервіси з необмеженим обсягом пам'яті, карти, погода, новини, тощо. Сьогодні мобільні телефони втрачають величезну частину своєї функціональності без мережі Інтернет, адже більшість мобільних додатків вимагають з'єднання з мережею. Навіть домашні девайси зараз приєднуються до мережі платформами Internet of Things [1, с. 16]. Наприклад, Wink дає змогу користувачам керувати світлом будинку через смартфон.

При побудові різних аплікацій розробники веб-додатків мають поставити собі такі запитання: чи довіряємо ми цілісності запиту, який надходить від браузера користувача? чи впевнені ми в тому, що

завантажені дані є чистими і безпечними? чи впевнені ми в тому, що з'єднання між нашим додатком і браузером користувача не може бути підробленим? чи довіряємо ми сервісам та базам даних, від яких залежимо? І очевидно, що відповідь на всі ці питання має бути "Ні" або ж "Не зовсім".

Різні веб-додатки мають різні функції та призначення, однак всі вони часто стають мішенню для зловмисників. Всі вразливості в коді додатку є специфічними для кожної програми і ніколи не були відомі раніше. Досвідчений зловмисник може легко знайти ці вразливості і використовувати в своїх цілях, не будучи виявленим. То що ж їх може так приваблювати в цих додатках?

***Популярність.*** Якщо у вас є популярний сайт, ви отримуєте велику кількість відвідувань кожну секунду. Продуктивність і доступність вашого веб-сайту є однією з головних переваг, яку ви маєте над усіма іншими. Ймовірно, у вас також є багато шанувальників і конкурентів, тому пошкодження вашого додатку може дуже потішити їх.

***Протест / політика.*** Це такі види атак, як анонімні напади на урядові, релігійні та корпоративні веб-сайти, відбуваються задля для свого задоволення або щоб зробити якусь важливу заяву до влади чи суспільства, кинути комусь виклик.

***Незадоволені співробітники***. Не всі атаки відбуваються ззовні, найчастіше вони організовуються за допомогою когось зсередини, тобто інсайдера. Наприклад, в 2014 році атака SQL ін'єкції [2, с. 3] була звинувачена у 8,1 % всіх веб-атак.

Однак, не завжди в навчальних посібниках, чи навіть в мережі Інтернет можна знайти достатньо інформації стосовно таких поширених видів атак на веб-додатки, як небезпечні перенаправлення, атаки на сесії та міжсайтовий скриптинг чи SQL ін'єкції. Тому *метою цієї роботи* є аналіз найпопулярніших сучасних атак на веб-додатки в мережі Інтернет,

встановлення їх негативних наслідків, а також подання можливих варіантів їх запобігання.

**Виклад основного матеріалу**

На сьогодні відомі такі найпопулярніші види нападів на веб-додатки, як небезпечні перенаправлення, атака на сесії та міжсайтовий скриптинг та SQL ін'єкція. Проаналізуємо кожну з них дещо детальніше.

***Небезпечні перенаправлення.*** Ця категорія вразливостей веб-додатків використовується в атаках, в яких жертва шляхом обману перенаправляється на шкідливий сайт. Зловмисники можуть маніпулювати URL-адресами сайту для перенаправлення на сторінки небажаних ресурсів.

Небезпечні перенаправлення та посилання є можливими тоді, коли веб-додаток приймає при введенні небезпечні дані, які посилаються на веб-додаток, який перенаправить запит на адресу, що міститься в цих небезпечних даних. Змінюючи безпечну адресу на шкідливий сайт, зловмисник може успішно почати атаку, однією з яких є крадіжка облікових даних користувачів. Оскільки ім'я сервера в модифікованому посиланні збігається з іменем вихідного сайту, спроби атаки можуть мати більш надійний вигляд.

Для безпечного перенаправлення користувача на іншу сторінку можна використати таку реалізацію:

```
Java: response.sendRedirect ( "http://www.examplesite.com");
PHP:
<? PHP
/ * Перенаправлення браузера * /
header("Location: http://www. examplesite.com/");
?>
ASP.NET:
Response.Redirect("~/folder/Login.aspx")
Rails:
 redirect_to login_path
```

У наведеному вище прикладі адреси є явно оголошені в коді і зловмисник не може ними маніпулювати.

Дещо інакше виглядає код небезпечних перенаправлень. Наприклад, код Java отримує адресу від 'URL' параметра GET і перенаправляє на цю адресу: response.sendRedirect (request.getParameter ( "URL"));

Код PHP отримує адресу з рядка запиту, а потім перенаправляє користувача на цю адресу:

```
$redirect_url = $_GET['url']; header("Location: " . $redirect_url);
```

Аналогічно мовою C # .NET:

```
string url = request.QueryString["url"]; Response.Redirect(url);
```

І в Rails: redirect_to params[:url]

Наведений вище код є вразливим для атаки, якщо ніякої перевірки або додаткового методу контролю не застосовуються. Ця уразливість може бути використана як частина атаки шляхом перенаправлення користувачів на шкідливий сайт. Якщо перевірка не застосовується та зловмисник може створити гіперпосилання, щоб перенаправити користувачів на неперевірений шкідливий веб-сайт, наприклад:

```
http://example.com/example.php?url=http://site.example.com
```

Користувач бачить посилання на потрібний сайт (example.com) і не усвідомлює, що стає жертвою атаки.

***Атаки, пов'язані з маніпуляцією сесіями.*** Цей вид атаки складається з експлуатації механізму управління веб-сесіями, яка, зазвичай, управляється токеном сеансу.

Оскільки http зв'язок використовує безліч різних з'єднань TCP, веб-серверу необхідний спосіб розпізнавати з'єднання кожного користувача. Найбільш корисний метод залежить від токену, тобто, веб-сервер відправляє в браузер клієнта після його успішної аутентифікації. Токен сеансу, як правило, складається з рядка змінної ширини і його можна використовувати по-різному, як в адресі, в заголовку http, так і в його тілі [3, с. 3].

Атака сесії компрометує токен сеансу шляхом крадіжки або передбачення правильного токену сеансу для отримання несанкціонованого доступу до веб-сервера.

У прикладі, який наведено на рис. 1, спочатку зловмисник використовує перехоплювач, щоб захопити дійсний токен сеансу з ім'ям "ID сеансу", потім він використовує справжній токен для отримання несанкціонованого доступу до веб-сервера.

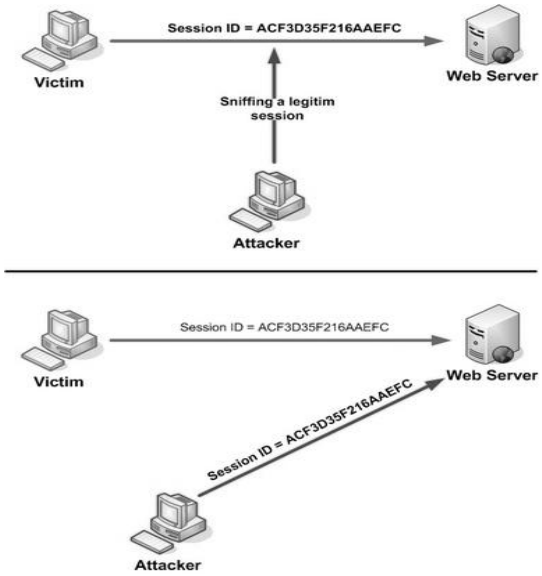


Рис. 1. Захоплення сеансу [4]

Зловмисник може поставити під загрозу токен сеансу використанням шкідливого коду або програми, що працюють на стороні клієнта. Приклад (рис. 2) показує, як зловмисник може використовувати атаку XSS (Cross-site Scripting) [5, с. 3], щоб вкрасти токен. Якщо зловмисник надішле оброблене посилання на шкідливий код JavaScript, то, коли жертва натискає на посилання, JavaScript почне працювати і виконувати інструкції, зроблені нападником. У цьому прикладі на рис. 2 використовується атака XSS, яка показує значення cookie поточного сеансу. Використовуючи цю ж техніку, можна створити спеціальний код JavaScript, який буде відправляти cookie зловмиснику, а саме: <SCRIPT>alert(document.cookie);</SCRIPT>

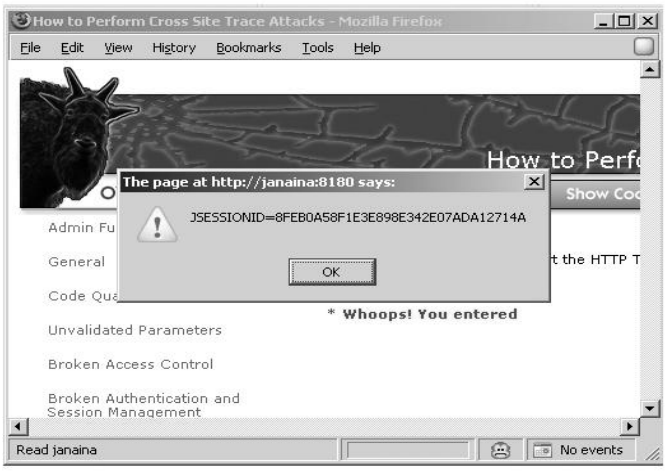


Рис. 2. Ін'єкція коду [4]

***SQL ін'єкція.*** Це атака, в якій шкідливий код вставляється в рядки, які потім пройшли до примірника SQL Server для синтаксичного аналізу і виконання. Будь-яка процедура, яка будує оператори SQL, має бути перевірена на наявність вразливостей для ін'єкцій, оскільки SQL Server виконує всі синтаксично правильні запити, які він отримує [6, с. 89].

Основна форма SQL ін'єкції складається з прямої вставки коду у вхідні змінні, призначені для користувача, які з'єднуються командами SQL і виконуються. Менш пряма атака вписує шкідливий код в рядки, які призначені для зберігання в таблиці або в вигляді метаданих. Коли збережені рядки потім об'єднуються динамічною командою SQL, шкідливий код виконується.

Процес вставляння коду здійснюється шляхом передчасного завершення текстового рядка і додавання нової команди. Оскільки вставлена команда може мати додаткові рядки, перш ніж вона буде виконана, зловмисник завершує доданий рядок з коментарем у вигляді знака "-". Такий текст ігнорується під час виконання.

Наведений нижче сценарій показує просту SQL-ін'єкцію. Сценарій будує запит SQL шляхом конкатенації жорстко-закодованих рядків разом з рядком, введеним користувачем: var City; City = Request.form ("City"); var sql = "select * from Orders where City = '" + City + "'";

Користувачеві буде запропоновано ввести назву міста. Якщо він введе Lviv, то запит виглядатиме приблизно так: SELECT * FROM Orders WHERE City = 'Lviv'

Проте, припустимо, що користувач вводить таке: Lviv '; drop table Orders--. В цьому випадку, такий запит буде виконано: SELECT * FROM Orders WHERE City = 'Lviv';drop table Orders--‘

Крапкою з комою (;) позначається кінець одного запиту і початок іншого. Подвійний дефіс (-) вказує на те, що інша частина поточного рядка є коментарем і має бути проігнорована. Якщо змінений код синтаксично правильний, то він буде виконаний сервером.

Отже, з розвитком корисних технологій невід'ємно розвиваються й шкідливі її прояви. Час не стоїть на місці, а отже зловмисники стають здібнішими і винахідливішими. Тож варто постійно тримати руку на пульсі, вивчаючи нові види загроз аби мати можливість протистояти їм.

**Висновки**

Іноді веб-ресурси містять не лише корисну інформацію, але й шкідливий код. Відвідування цих сайтів може призвести до зараження комп'ютера вірусами, а також до псування або крадіжки ваших даних, тому важливо пам'ятати про такий вид атак, як небезпечні пере направлення та посилання.

XSS (англ. Cross Site Scripting – "міжсайтовий скриптінг") – тип вразливості інтерактивних інформаційних систем у вебі. XSS виникає, коли на сторінки, які були згенеровані сервером, з якоїсь причини потрапляють користувацькі скрипти. Специфіка подібних атак полягає в тому, що замість безпосередньої атаки сервера зловмисники використовують вразливий сервер для атаки на користувача.

SQL ін'єкція – один з поширених способів злому сайтів та програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду. Атака типу впровадження SQL може бути можлива

за некоректної обробки вхідних даних, що використовуються в SQL-запитах.

Розробники додатків, що працюють з базами даних, має знати про таку уразливість і вживати заходів протидії впровадженню SQL.

Безпека веб-додатків є одним з найкритичніших питань в циклі життя додатку, оскільки його процес відбувається у мережі, як правило відкритій для решти світу. Окрім цього, якщо додаток відкритий, запит на нього надіслати може будь-хто. Тому правильна обробка і фільтрація вхідних запитів є ключовою в питанні захисту веб-додатків.

**Література:**


1. *Vermesan O. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems / O. Vermesan, P. Fries // River Publishers. – 2013. – Pp. 16-39.*
2. *Clarke J. SQL Injection Attacks and Defense / J. Clarke // Syngress. – 2012. – № 2. – Pp. 5-20.*
3. *Boneh D. Session Management and User Authentication / D. Boneh // CriptoStanford. – 2011. – Pp. 3–15.*
4. *Material from the site "Owasp.org" [Electronic resource]. - Access: https://www.owasp.org/index.php/Session_hijacking_attack5. Grossman J. XSS Attacks: Cross-site Scripting Exploits and Defense / J. Grossman // Syngress. – 2007. – Pp. 3-12.*
5. *Hartley D. SQL Injection Attacks and Defense / D. Hartley // Elsevie.r – 2012. – № 2. – Pp. 89-138.*