DOI: 10.36910/6775-2524-0560-2020-39-23
УДК: 004.42(07)
Костючко Сергій Миколайович, к.т.н.
https://orcid.org/0000-0002-1262-6268
Сахнюк Андрій Анатолійович, магістр
Мельник Катерина Вікторівна, к.т.н., доцент
https://orcid.org/0000-0002-9991-582X
Луцький національний технічний університет

ОБХІД ЗАХИСТУ САЙТІВ ЗА ДОПОМОГОЮ SQL-ІН'ЄКЦІЙ ТА ЗАХИСТ ВІД НИХ

Костючко С.М., Сахнюк А.А., Мельник К.В. Обхід захисту сайтів за допомогою SQL-ін'єкцій та захист від них. У статті розкрито суть несанкціонованого доступу до бази даних сайту; можливості та основи використання SQL-ін'єкцій; способи захисту від несанкціонованого вторгнення; переваги та недоліки POD.
Ключові слова: SQL-ін'єкція, база даних, сайт, PDO.

Костючко С.Н., Сахнюк А.А., Мельник К.В. Обход защиты сайтов с помощью SQL-инъекций и защита от них. В статье раскрыта суть несанкционированного доступа к базе данных сайта; возможности и основы использования SQL-инъекций; способы защиты от несанкционированного вторжения; преимущества и недостатки POD.
Ключевые слова: SQL-инъекция, база данных, сайт, PDO.

Kostiuchko S.M., Sahniuk A.A., Melnyk K.V. Bypass site protection by means SQL injections and protection against them. The article reveals the essence of unauthorized access to the web-site database; possibilities of using SQL-injections; ways to protect against unauthorized intrusion; POD advantages and disadvantages.
Keywords: SQL injection, database, site, PDO.

Постановка проблеми.

Через різні вразливості веб-серверів і нескладні процедури, атаки на веб-сервери зросли, в основному за рахунок впровадження сценаріїв ASP або PHP. Атаки з використанням SQL ін'єкцій стали основним напрямком, в той час як процес компіляції веб-сервера переважає в сценарії, ігноруючи явище тестування безпеки програмного коду, що призводить до великої кількості надання інтерактивних лазівок в роботі на веб-сервері. Зловмисники можуть використовувати сервер, конфігурувати базу даних дефектів і розробляти структуру незаконних запитів за допомогою програм або сценаріїв вторгнення на сервер, отримувати дозволи адміністратора веб-сайту і отримувати відповідний контент баз даних.

Виклад основного матеріалу й обґрунтування отриманих результатів.

Ін'єкція SQL - це вразливість веб-безпеки, яка дозволяє зловмиснику втручатися у запити, які подає програма до своєї бази даних. Як правило, зловмисник дозволяє переглядати дані, які вони зазвичай не можуть отримати. Це може включати дані, що належать іншим користувачам, або будь-які інші дані, до яких сама програма може отримати доступ. У багатьох випадках зловмисник може змінювати або видаляти ці дані, викликаючи постійні зміни у змісті чи поведінці програми.

У деяких ситуаціях існує можливість застосувати атаку ін'єкцій SQL, щоб поставити під загрозу базовий сервер чи іншу інфраструктуру.

Успішна атака ін'єкцій SQL може призвести до несанкціонованого доступу до конфіденційних даних, таких як паролі, дані кредитної картки або особиста інформація користувача.

В дослідженні було перевірено роботу підриву логіки програми за допомогою ін'єкцій SQL, де є можливість змінити запит, задля перешкоджання логіки програми, але також існує велика різноманітність і інших вразливих ситуацій, атак та методів ін'єкцій SQL, які виникають у різних ситуаціях. Деякі поширені приклади введення SQL включають:

- Отримання прихованих даних, де можна змінити SQL-запит, щоб повернути додаткові результати.
- UNION-атаки, де можливо отримати дані з різних таблиць баз даних.
- Вивчення бази даних, де можливо отримати інформацію про версію та структуру бази даних.
- Сліпа ін'єкція SQL, коли результати запиту, яким керує зловмисник, не відповідають результатам програми.

© Костючко С.М., Сахнюк А.А., Мельник К.В.

Підрив логіки програми

Розглянемо програму, яка дозволяє користувачам входити з іменем користувача та паролем. Якщо користувач подає ім'я користувача Andrew та пароль 123456, програма перевіряє облікові дані, виконуючи наступний SQL-запит:

SELECT * FROM accounts WHERE username = 'andrew' AND password = '123456'

Якщо запит повертає реквізити користувача, то реєстрація успішна. В іншому випадку реєстрація не відбудеться.

Якщо, наприклад, в поле паролю Ввести знак “ ' “ може виникнути помилка, яка надасть нам потрібну інформацію про базу даних чи про файли сайту. Наприклад, як зображено на рисунку нижче.

Error: Failure is always an option and this situation proves it	
Line	49
Code	0
File	/var/www/mutillidae/process-login-attempt.php
Message	Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''''' at line 1
Trace	#0 /var/www/mutillidae/index.php(96): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='andrew' AND password='''

Рис. 1 – Помилка при відмові в реєстрації користувача

Для перевірки можливості використання SQL коду в полі реєстрації можна в полі password ввести 123456' and 1=1#. Після вводу в разі незахищеності системи реєстрація пройде успішно. Вид запиту після вводу пароля:

SELECT * FROM accounts WHERE username = 'andrew' AND password = '123456' and 1=1#'

Щоб перейти на відповідний аккаунт наприклад andrew, але пароль нам не відомий можна використати or замість and після чого запит буде мати вигляд:

SELECT * FROM accounts WHERE username = 'andrew' OR password = '123456' or 1=1#'

Також існує можливість ввійти, як будь-який користувач, без пароля просто за допомогою поля username. Наприклад, введення імені користувача admin'# та порожнього пароля призводить до наступного запиту:

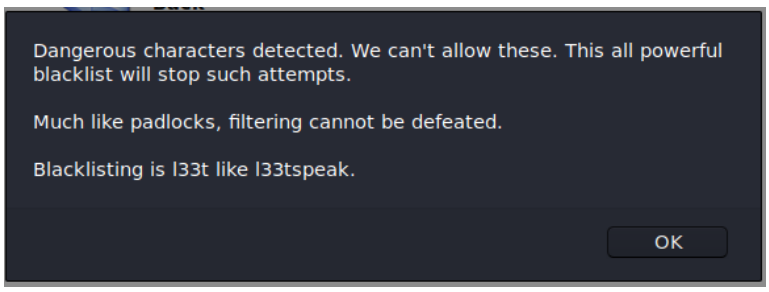
SELECT * FROM accounts WHERE username = 'admin'# AND password = 'aaaaaa'

Цей запит повертає користувача, ім'я користувача якого є, admin і успішно входить в систему. Так як після 'admin' стоїть знак коментаря #, то все, що йде після нього, буде ігноруватись. Тому запит в дійсності буде таким:

```
SELECT * FROM accounts WHERE username = 'admin'#
```

Підрив логіки програми з більш складнішим захистом.

Візьмем більш складну систему яка блокує різні символи крім букв. Тоді при вводі будь-якого знаку буде видаватись така помилка:



© Костючко С.М., Сахнюк А.А., Мельник К.В.

Рис. 2 – Помилка при вводі на стороні клієнта

Дана помилка відбувається тільки на стороні клієнта, і до серверу запит не відправляється, тому даним способом доступ до бази не буде отримано. Але, якщо будуть введені коректні символи, то зв'язок з базою відбудеться хоча б, щоб перевірити параметри авторизації. На даному етапі можна скористатися програмою Burp Suite та переглянути які пакети надсилаються:

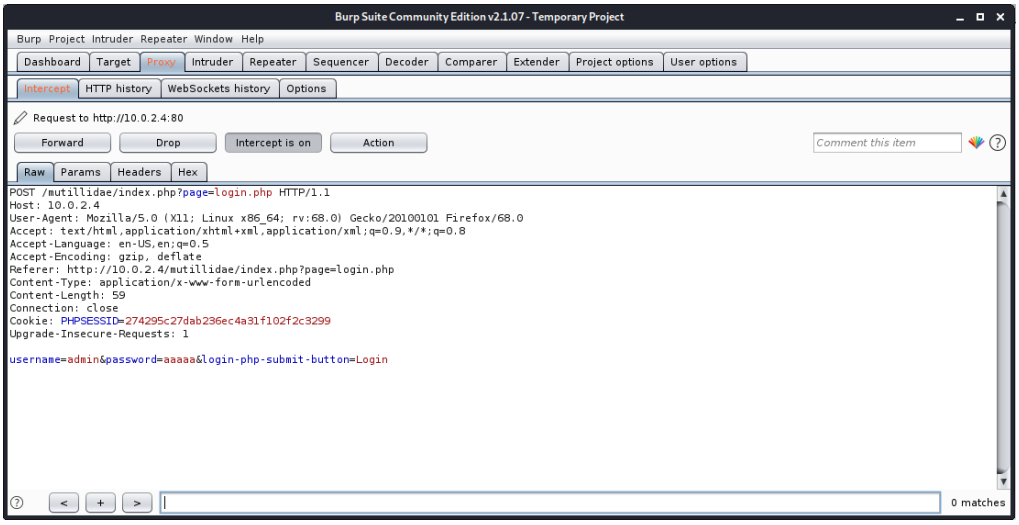


Рис. 3 – Інтерфейс програми Burp Suite

Дана програма може "притримати" пакети, які вже вийшли з клієнта, а попередня помилка не буде завадою для отримання доступу до бази. В цій програмі також можливо дещо підправити відправлені дані, а саме переправити пароль на вже відомий код 123456' or 1=1# .

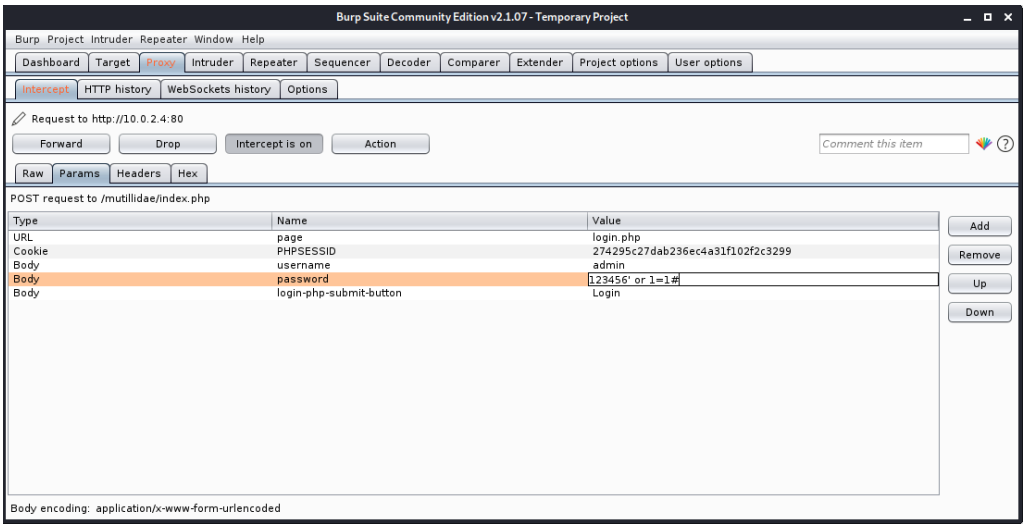


Рис. 4 – Деякі можливості програмного продукту Burp Suite

Виявлення вразливості ін'єкцій SQL

Більшість уразливостей ін'єкцій SQL можна швидко та надійно знайти за допомогою веб-сканера вразливості Burp Suite .

© Костючко С.М., Сахнюк А.А., Мельник К.В.

Ін'єкцію SQL можна виявити вручну за допомогою систематичного набору тестів проти кожної точки входу в програму. Зазвичай це стосується:

– Подання символу єдиної цитати та пошуку помилок чи інших аномалій.

– Подання деякого специфічного для SQL синтаксису, який оцінює базове (вихідне) значення точки введення та інше значення та шукає систематичні відмінності в отриманих відповідях додатків.

– Посилають логічні умови, такі, як OR 1=1 і OR 1=2, і шукаємо відмінності у відповідях додатка чи сайту.

– Подання корисних навантажень, призначених для запуску затримок у часі при виконанні в межах SQL-запиту, та пошуку відмінностей у часі, необхідному для відповіді.

– Подання корисних навантажень OAST, призначених для запуску позадіапазонної мережевої взаємодії при виконанні в рамках SQL-запиту, а також для моніторингу будь-яких результуючих взаємодій.

Введення SQL в різні частини запиту

Більшість уразливостей для введення SQL виникає в WHERE пункті SELECT запиту. Цей тип ін'єкції SQL, як правило, добре зрозумілий досвідченим тестерам.

Але вразливості введення SQL можуть в принципі виникати в будь-якому місці в межах запиту та в різних типах запитів. Найбільш поширені інші місця, де виникає ін'єкція SQL, це:

– У UPDATE висловлюваннях, в межах оновлених значень або WHERE пункту.

– У INSERT висловлюваннях у межах вставлених значень.

– У SELECT висловлюваннях, в межах назви таблиці чи стовпця.

– У SELECT заявах, у межах ORDER BY.

Захист сайту від SQL-ін'єкцій

Так як ми вже визначили, що SQL-ін'єкції дуже небезпечні тому потрібно мати надійний захист від даних атак. Тому одним з виходів використовувати PHP Data Objects або PDO. PDO (PHP Data Objects) - рівень абстракції для запитів вашої бази даних і є приголомшливою альтернативою MySQLi, оскільки він підтримує 12 різних драйверів баз даних. MySQL - це найпопулярніша база даних. Вона також дуже добре поєднується з PHP, саме тому ця пара технологій добре підтримується у світі PHP.

Якщо фактично знаєте, що єдиною базою даних SQL, якими ви користуєтесь, є або MySQL, або MariaDB, слід вибрати PDO.

Існує думка, що головна перевага PDO полягає в тому, що він переноситься з бази даних в базу даних але насправді це не є так. Насправді, дуже рідко приймається рішення щодо переключення баз даних на конкретний проект, а саме для компаній середнього рівня дана перевага є недоречною. Незважаючи на це, зазвичай, як правило, вважають за краще дотримуватися поточної технології, яка використовується, якщо тільки немає обґрунтованих причин втрачати значну кількість часу та грошей для переносу.

Справжня перевага PDO полягає в тому, що використовується практично подібний API для будь-якої з безлічі баз даних, які він підтримує, тому вам не потрібно вивчати нову для кожної. Названі параметри також, безсумнівно, є величезною перевагою для PDO, оскільки ви можете повторно використовувати однакові значення в різних місцях запитів. На жаль, ви не можете використовувати одні і ті ж названі параметри не один раз із вимкненим режимом емуляції.

Робота підготовлених заявок PDO

Для простішого пояснення, підготовлені заяви PDO працюють так:

1. Підготуйте SQL-запит із порожніми значеннями, як заповнювачі або знаком питання, або ім'ям змінної з двокрапкою, що передує йому, для кожного значення.
2. Прив'яжіть значення або змінні до заповнювачів
3. Виконувати запит одночасно.

Недоліки PDO

Однак при використанні визначених виразів спільно з PDO необхідно знати деякі нюанси, щоб уникнути неприємних ситуацій. Наприклад, в MySQL клієнта деякі запити, складені за допомогою визначених виразів, тому не можуть бути виконані, а так само вони не використовують кеш, що може уповільнити роботу вашого web-додатка.

© Костючко С.М., Сахнюк А.А., Мельник К.В.

Гарантована безпека при використанні визначених виразів звучить добре, але розробники не повинні приймати PDO і інші види абстракції, зумовлені вираженням, як абсолютний захист від злому. Будь-які вхідні дані повинні перевірятися, PDO - додаткова лінія оборони. Це розширення не закриває все безліч вразливостей, за допомогою яких може бути завдано шкоди вашій інформації, але в той же час, PDO непогано справляється з питанням запобігання SQL ін'єкцій.

Висновки та перспективи подальшого дослідження.

Отже, SQL-ін'єкції дуже небезпечні для незахищеної системи, так це є найпопулярніший спосіб взлому, і не є дуже складним для розуміння і використання. Тому навіть недосвідчений зловмисник може отримати доступ до бази даних. Байдуже ставлення до захисту від даних атак може призвести до катастрофічних наслідків.

Для надійного захисту використовуйте різні технології захисту, як приклад, технологію PDO, яка захистить від SQL ін'єкцій і дозволить використовувати чистий шар абстракції з бази даних, що забезпечить вам майбутню гнучкість у разі зміни баз даних.

Список бібліографіного опису

1.Addison Berry, Angela Byron, Bruno De Bondt. Using Drupal (2nd Edition). – O'Reilly. 2012. – 500p.
2.Al-Darwish, N.: Page Gen: An Effective Scheme for Ddynamic Generation of Web Pages. Information and Software Technology 45(10), 15 July 2003, Pages 651-662
3.Cynthia McCourt. Drupal: The Guide to Plainning and Building Websites. – Wrox. 2011. – 504 p.
4.Douglas Vernon Denny. Drupal 7 Webform Cookbook. – Packt Publishing. 2012. – 276 p.
5.FabienPotencier. TemplatingEnginesin PHP (переклад), Templatingenginesin PHP – Follow-Up (переклад)
6.Jennifer Hodgdon. A Programmer's Guide to Drupal. – O'Reilly. 2012. – 114p.
7.Ric Shreves, Brice Dunwoodie. Drupal 7 Bible. – Wiley. 2011. – 768 p.
8.Smarty 3.1.29 Released – 2015.
9.Trevor James. Migrating to Drupal 7. – Packt Publishing. 2012. – 158p.
10. http://anton.shevchuk.name/php/php-template-engin/
11. https://drudesk.com.ua/blog/funktsionalni-mozhlyvosti-twig
12. ZhangZhuo, The SQL injection attack technology and preventive measures research, 2-4 2007.01.
13. ZhouWenYu, Based on preventing SQL injection network security technology analysis and application, 43-50 2010.06.
14. Xiaozhu The SQL;1; injection into holes of ASP too mystierious full contact, 2005.01.

References

15. Addison Berry, Angela Byron, Bruno De Bondt. Using Drupal (2nd Edition). – O'Reilly. 2012. – 500p.
16. Al-Darwish, N.: Page Gen: An Effective Scheme for Ddynamic Generation of Web Pages. Information and Software Technology 45(10), 15 July 2003, Pages 651-662
17. Cynthia McCourt. Drupal: The Guide to Plainning and Building Websites. – Wrox. 2011. – 504 p.
18. Douglas Vernon Denny. Drupal 7 Webform Cookbook. – Packt Publishing. 2012. – 276 p.
19. FabienPotencier. TemplatingEnginesin PHP (переклад), Templatingenginesin PHP – Follow-Up (переклад)
20. Jennifer Hodgdon. A Programmer's Guide to Drupal. – O'Reilly. 2012. – 114p.
21. Ric Shreves, Brice Dunwoodie. Drupal 7 Bible. – Wiley. 2011. – 768 p.
22. Smarty 3.1.29 Released – 2015.
23. Trevor James. Migrating to Drupal 7. – Packt Publishing. 2012. – 158p.
24. http://anton.shevchuk.name/php/php-template-engin/
25. https://drudesk.com.ua/blog/funktsionalni-mozhlyvosti-twig
26. ZhangZhuo, The SQL injection attack technology and preventive measures research, 2-4 2007.01.
27. ZhouWenYu, Based on preventing SQL injection network security technology analysis and application, 43-50 2010.06.
28. Xiaozhu The SQL;1; injection into holes of ASP too mystierious full contact, 2005.01.

© Костючко С.М., Сахнюк А.А., Мельник К.В.