Описані вище рішення щодо захисту від загроз безпеки хмарних обчислень неодноразово були застосовані системними інтеграторами в проектах побудови приватних хмар. Після застосування цих рішень кількість інцидентів, що сталися, істотно знизилася. Але багато проблем, пов'язаних із захистом хмар ще вимагають ретельного аналізу і вирішення.

Список бібліографічних посилань

1. Kobzev I. V., Melnikov O. F., Orlov O. V. Cloud technology and e-government // Current scientific research: Collection of scientific articles. Publishing house «BREEZE», Montreal, Canada, 2017. P. 159-163.

2. Проект Закону про внесення змін до деяких законодавчих актів України щодо обробки інформації в системах хмарних обчислень : законопроект № 4302 від 24.03.2016 // БД «Законопроекти» / Верховна Ради України. URL: http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_1?pf3511= 58527 (дата звернення: 09.10.2018)

3. Греба Р. Електронне врядування-2017: досягнення і перспективи // Українська правда. 09 січня 2018 09:30. URL: https://www.pravda.com.ua/ columns/2018/01/9/7167821/ (дата звернення: 09.10.2018).

4. Threats of cloud computings and methods was more their than protection // Developers Club geek daily blog. June 2013. URL: http://developers-club.com/ posts/183168/ (дата звернення: 09.10.2018).

Одержано 10.10.2018

УДК 004.658

Олександр Олександрович КОВАЛЬОВ,
аспірант 2 курсу
Ужгородського національного університету

ЗАХИСТ ВІД СЛІПИХ SQL-ІН'ЄКЦІЙ ЗА ДОПОМОГОЮ ПРАВИЛЬНОЇ ОБРОБКИ ПОМИЛОК НА СТОРОНІ АРІ

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

Впровадження SQL, залежно від типу СКБД та умов впровадження, може дати можливість атакуючому виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання та/або запису локальних файлів та виконання довільних команд на сервері.

Основна небезпека до якої може вести вдало проведена атака SQL-ін'єкцією – це повна втрата інформації з БД. Добре якщо там немає нічого цінного, але, в залежності від обставин, частіше за все в БД зберігається багато досить критичних даних: облікові записи користувачів (включаючи паролі), номери телефонів, адреси

електронної пошти, а нерідко номери карт, їх терміни дії та інші супутні відомості.

Будь-яка процедура, яка створює інструкції SQL, повинна бути перевірена на наявність вразливостей впровадження, оскільки SQL Server виконає всі синтаксично допустимі запити, які він отримує. Навіть параметризовані дані можуть оброблятися досвідченого зловмисника.

Основна форма впровадження SQL складається з прямої вставки коду в змінні користувача введення, які об'єднуються з командами SQL і виконуються.

Менш пряма атака впроваджує шкідливий код в рядки, призначені для зберігання в таблиці, або у вигляді метаданих. Коли збережені рядки згодом об'єднуються в динамічну команду SQL, виконується шкідливий код.

Процес впровадження працює шляхом передчасного завершення текстового рядка і додавання нової команди. Оскільки перед виконанням доданої команди до неї можуть бути додані додаткові рядки, зловмисник завершує введений рядок за допомогою коментаря"--". Наступний текст ігнорується під час виконання.

Наступний сценарій показує приклад SQL ін'єкції. Сценарій створює SQL-запит, об'єднуючи жорстко закодовані рядки разом з рядком який був введеним користувачем:

var TakeCity;

TakeCity = Request.form ("TakeCity");

*var sql = "select * from OrdersTable where TakeCity = '" + TakeCity + "'";*

Сліпа ін'єкція SQL (Structured Query Language) - це тип атаки SQL-ін'єкції, яка задає базі даних істинні або хибні питання і визначає відповідь на основі відповіді додатків. Ця атака часто використовується, коли веб-додаток налаштовано для відображення загальних повідомлень про помилки, але не пом'якшило код, вразливий для впровадження SQL.

Коли зловмисник використовує SQL-ін'єкцію, іноді веб-додаток відображає повідомлення про помилки з бази даних зі скаргами на неправильний синтаксис SQL-запиту. Сліпа SQL-ін'єкція майже ідентична звичайній SQL-ін'єкції, з тією лише різницею, що дані витягуються з бази даних. Якщо база даних не виводить дані на веб-сторінку, зловмисник змушений вкрасти дані, задавши базі даних ряд істинних або помилкових питань. Це робить використання вразливості SQL-ін'єкції більш складним, але не неможливим.

Використання підготовлених операторів з прив'язкою змінних - це те, як всі розробники повинні спочатку навчитися писати запити до бази даних.

Вони прості в написанні і більш зрозумілі, ніж динамічні запити. Параметризрвані запити змушують розробника спочатку визначити весь код SQL, а лише потім передати кожен параметр запит. Цей стиль

кодування дозволяє базі даних розрізняти код і дані, незалежно від того, що саме ввів користувач.

Підготовлені інструкції гарантують, що зловмисник не зможе змінити призначення запиту, навіть якщо команди SQL вставлені зловмисником. У безпечному прикладі, якщо зловмисник повинен був ввести ідентифікатор користувача *tom' or '1'= '1*, параметризований запит не був би вразливий і замість цього шукав би ім'я користувача, яке буквально відповідало всій рядку *tom' or '1'= '1*.

Розробникам подобається підхід, заснований на підготовлених операторах, оскільки весь SQL-код залишається в додатку. Це робить додаток відносно незалежним від бази даних.

Список бібліографічних посилань

1. Gregg Keizer (April 25, 2008). "Huge Web hack attack infects 500,000 pages". Archived from the original on October 19, 2015. Retrieved October 16, 2015.

2. Stephen W., Keromytis D. (2010), SQLrand: Preventing SQL Injection Attacks, Department of Computer Science Columbia University.

3. Michael Zino (May 1, 2008). "ASCII Encoded/Binary String Automated SQL Injection Attack". Archived from the original on June 1, 2008.

4. Justin C. "Using SQLBrute to brute force data from a blind SQL injection point". Archived from the original on June 14, 2008. Retrieved October 18, 2008.

5. Ronald L. (2008), "SQL Injection", Hong Kong Computer Emergency Response Team Coordination Centre.

Одержано 31.10.2018

УДК 004.056

Тетяна Петрівна КОЛІСНИК,
кандидат педагогічних наук, доцент,
доцент кафедри інформаційних технологій факультету № 4 (кіберполіції)
Харківського національного університету внутрішніх справ;
ORCID: https://orcid.org/0000-0002-7442-8136;
Володимир Володимирович ТУЛУПОВ,
кандидат технічних наук, доцент,
доцент кафедри інформаційних технологій факультету № 4 (кіберполіції)
Харківського національного університету внутрішніх справ;
ORCID: https://orcid.org/0000-0003-4794-743X

ОСОБЛИВІ ПИТАННЯ ЗАХИСТУ СИСТЕМ IP-ВІДЕОСПОСТЕРЕЖЕННЯ

На теперішній час як у системі Міністерства внутрішніх справ так і на державному та регіональних рівнях впроваджуються різні програми та заходи безпеки. Так, наприклад у березні 2018 року на сесії Харківської обласної ради була прийнята «Програма забезпечення публічної безпеки і порядку та протидії злочинності на території Харківської області на 2018-2019 роки», відповідно до якої