Journal of the Korea Convergence Society
Vol. 10. No. 10, pp. 21-26, 2019

ISSN 2233-4890
https://doi.org/10.15207/JKCS.2019.10.10.021

Research on Countermeasure of SQL Injection Attack

Sunghyuck Hong
Professor, Division of ICT, Information Security Major, Baekseok University

SQL Injection 공격을 효율적으로 방어하는 대응책 연구

홍성혁
백석대학교 ICT학부 정보보호전공 교수

Abstract At present, it is indispensable to utilize data as an information society. Therefore, the database is used to manage large amounts of data. In real life, most of the data in a database is the personal information of a group of members. Because personal information is sensitive data, the role of the database administrator who manages personal information is important. However, there is a growing number of attacks on databases to use this personal information in a malicious way. SQL Injection is one of the most known and old hacking techniques. SQL Injection attacks are known as an easy technique, but countermeasures are easy, but a lot of efforts are made to avoid SQL attacks on web pages that require a lot of logins, but some sites are still vulnerable to SQL attacks. Therefore, this study suggests effective defense measures through analysis of SQL hacking technology cases and contributes to preventing web hacking and providing a secure information communication environment.

Key Words : SQL Injection, Filtering Bypass, Stored Procedure, Error Message, Prepared Statement

요 약 현재 사회는 정보화 사회로 데이터를 활용하는 것이 필수불가결하다. 따라서 데이터베이스를 활용하여 방대한 양의 데이터를 관리하고 있다. 실생활에서 데이터베이스에 들어있는 데이터들은 대부분 한 그룹의 회원들의 개인정보들이다. 개인정보는 민감한 데이터이기 때문에 개인정보를 관리하는 데이터베이스 관리자의 역할이 중요하다. 하지만 이런 개인정보를 악의적으로 사용하기 위해 데이터베이스를 공격하는 행위가 늘고 있다. SQL Injection은 가장 많이 알려져 있고 오래된 해킹기법 중에 하나이다. SQL Injection 공격은 공격하기 쉬운 기법으로 알려져 있으나 대응방안 또한 쉽지만 많은 로그인을 요구하는 웹페이지에서 SQL 공격을 피하기 위한 노력을 많이 하지만 일부 사이트는 여전히 SQL 공격에 취약하다. 따라서 본 연구에서 SQL해킹 기술 사례 분석을 통하여 효과적인 방어책을 제시하여 웹 해킹을 막고 안전한 정보통신 환경을 제공하는 데 기여한다.

주제어 : SQL Injection, 필터링 우회, Stored Procedure, 오류메시지, Prepared Statement

1. Introduction

As the information society develops, almost all the information is being dataized and managed. Therefore, paper data became unnecessary and data management became easier. However, as much of the data came into the database, it was threatened by hacking. Fortunately, as security technology has been developed to prevent hacking, data has been less leaked, but with the

*This paper was supported by 2019 Baekseok University Fund.
*Corresponding Author : Sunghyuck Hong(sunghyuck.hong@gmail.com)
Received August 30, 2019 Revised September 26, 2019
Accepted October 20, 2019 Published October 28, 2019

development of hacking technology, security has become even more important today. The oldest and most well known hacking technique is SQL Injection. SQL Injection is a hacking technique that is easy to break but also easy to crack. In Chapter 2, we describe SQL Injection and its damage cases. In Chapter 3, we discuss kinds of SQL Injection. In Chapter 4, we discuss the corresponding countermeasures. In Chapter 5, we try to finish the report.

2. SQL Injection

2.1 SQL Injection

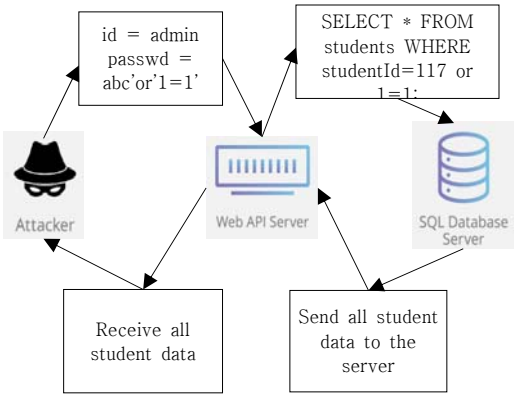

Fig. 1. SQL Injection Principle

SQL Injection is a code injection attack method that manipulates a database abnormally by intentionally exploiting loopholes in application security to execute malicious SQL statements[1].

This occurs mainly when the user does not properly filter and escape the data input. The attacker is easy to attack, but the destructive power is so great that it is the first thing a developer learns[2].

Table 1, 2 & 3 shows OWASP TOP 10 since 2013. SQL Injection is a high-risk method of attack, which has been the top of all Open Web Application Security Project (OWASP) TOP 10, which is updated every four years, in 2010, 2013 and 2017[3,4].

Table 1. OWASP TOP 10(2010)

T1	Injection
T2	Cross-Site Scripting(XSS)
T3	Broken Authentication and Session Management
T4	Insecure Direct Object References
T5	Cross-Site Request Forgery(CSRF)
T6	Security Misconfiguration
T7	Insecure Cryptographic Storage
T8	Failure to Restrict URL Access
T9	Insufficient Transport Layer Protection
T10	Invalidated Redirects and Forwards

Table 2. OWASP TOP 10(2013)

T1	Injection
T2	Broken Authentication and Session Management
T3	Cross-Site Scripting(XSS)
T4	Insecure Direct Object References
T5	Security Misconfiguration
T6	Sensitive Data Exposure
T7	Missing Function level Access Control
T8	Cross-Site Request Forgery(CSRF)
T9	Using Components with Known Vulnerabilities
T10	Invalidated Redirects and Forwards

Table 3. OWASP TOP 10(2017)

T1	Injection
T2	Broken Authentication
T3	Sensitive Data Exposure(XXE)
T4	XML External Entities
T5	Broken Access Control
T6	Security Misconfiguration
T7	Cross-Site Scripting(XSS)
T8	Insecure Deserialization
T9	Using Components with Known Vulnerabilities
T10	Insufficient Logging & Monitoring

2.2 Damage cases

There is a case of SQL Injection attack in Korea. Immediate Stay O2O service "How are you here?" In this attack, customer information has

been leaked and customers have been sent an unpleasant text message. Although it is a common technique, it does not recognize the fact of the hacking and it is revealed through the hacking technique.

In another example, the British Information Board announced that Worldview, a famous travel website, announced that a credit card number was hacked by an SQL injection attack and warned against it. The British Intelligence Commission has charged World View with a penalty of £ 7,500. Worldview said that this attack hacked card information of about 4,000 customers[5].

In addition to cases of personal information leakage, it is possible to control the system through web attack. Grasping a system is more dangerous because it is more likely that a secondary attack is more likely to occur than if the victim ends up harming the system. If you add administrator privileges to a new user, the attacker may become an administrator.

3. Types of SQL Injection

3.1 Bypass filtering

The filtering bypass technique is a general attack method. It is an attack that is targeted at a login page that normally inputs an ID and a password. It uses an error of logical operation of the T / F of the query statement to cause the login authentication query to be unconditionally TRUE, It is a way to neutralize. As a typical pattern, input an abnormal SQL query as 'admin' and password 'OR'1' = '1' in the login field of the web server.

Table 4. Filtering Bypass

```
SELECT *
FROM tables
WHERE id= 'admin' AND passwd=''OR'1'=''1';
```

Table 4 shows a filtering bypass sample. Passwd can succeed in login because 'OR'1' = '1 syntax is always TRUE even if ID and password are not matched correctly[6].

3.2 Stored Procedure

Stored Procedure SQL Injection using stored procedures. The stored procedure is a form of SQL set up for operational convenience. Especially, xp_cmdshell, which can be used in MS SQL, is one of the frequently exploited stored procedures[7] because it provides a role to execute Windows commands.

3.3 Incorrect Queries

This query is based on an error message in the database. It is an attack that causes an error to get the database name, the table, the column, and ultimately the data based on the error information. This technique is used as a step for collecting preliminary information and is not a direct attack because it only acquires data without modifying it[7].

3.4 Union Queries

This query uses the Union operator to add the query that the attacker intended to the parameter, thereby cutting out the desired internal information. Table 5 shows a simple query for SQL. The Union operator is used when combining two query statements and returning them, which is mainly used where data such as postal codes or bulletin boards can be easily identified[8].

Table 5. Basic Union query structure

```
SELECT * FROM A
UNION
SELECT * FROM B;
```

3.5 Blind Queries

Blind queries are used when the web is vulnerable to SQL Injection or database messages are not visible to attackers. Unlike ordinary queries, however, it is true and false that the data is retrieved only by the server's response when the query is true and false. Therefore, it combines the results of the iterative process to obtain the desired information, thereby causing the attack to occur. Because many comparisons are needed, attackers attack using automated tools[9].

4. Countermeasures of SQL Injection attack

4.1 Introduction of Web Firewall (WAF)

4.1.1 Physical Web Firewall

We recommend a physical dedicated WAF for environments where the size of the enterprise is large, where security is important, or where the budget is sufficient. You can use appliances or deploy cloud-based solutions.

4.1.2 Logical Web Firewall

If you do not have the capacity to deploy dedicated web firewall appliances, you should consider a public web firewall. Most public Web firewalls act as a web firewall in a logical configuration, not a device.

4.2 Secure Coding

4.2.1 Validating input values

In all web security domains, there is a competition for external input values. All external input values should not be trusted. The same is true for SQL Injection. Do not trust any input from outside. The external input value may be a value directly typed by the user, but it may include an external value that can modulate the value using a proxy tool such as the Burp Suite even if the user does not type it directly. The most basic strategy of SQL Injection is to validate the input value. There are two ways to validate input values. The first is the blacklist method, which restricts the characters or keywords that change the structure of the SQL query. The other is a white list method, which is a much stronger security method than the black list method.

In the blacklist method, all but the prohibited characters are allowed, but whitelisting is prohibited except for the permitted characters. Because it only accepts the specified characters, you need to keep the whitelist differently depending on the functionality of your web application. When defining a whitelist, categorizing / patterning using regular expressions is more advantageous for maintenance than defining individual characters one at a time.

4.2.2 Dynamic Query Usage Restrictions

You do not need to worry about SQL injection if your web application only uses static queries when working with DBs. But realistically you will not be able to use dynamic queries. Using dynamic queries is almost a necessity because all modern web applications interact with users and are based on dynamic functionality. Therefore, it is a good idea to use dynamic queries as static queries by using parameterize queries.

4.2.3 Limiting Error Message Output

DB error information should not be exposed to the user as it is. DB error information often includes internal information to help developers easily debug it. Based on this information, the hacker will try to figure out the DB structure and attempt to leak data. Therefore, DB error should not be exposed to the user as it is. You also need to be careful with too much guidance messages. It is better to notify the message when the login fails, the ID is wrong, the password is wrong, and the login information is inconsistent.

4.2.4 DB Security

4.2.4.1 Isolating DB Accounts

The DB account used by the administrator and the DB account accessed by the web application must be separated.

4.2.4.2 Restrict DB Account Permissions

The web application creates a dedicated account to access the DB and assigns only those privileges that are essential to this account based on the principle of least privilege.

4.2.4.3 Removing default / committed stored procedures

When the database is installed, it is a good idea to remove the default included procedures unless absolutely necessary.

4.3 Vulnerability Check and Monitoring

4.3.1 Continuous Vulnerability Check

You should regularly check for SQL Injection vulnerabilities. It is recommended to periodically check for vulnerabilities by attempting to simulate hacking or using a Web vulnerability checker. Keep in mind that even a simple event page can expose you to SQL vulnerabilities whenever you interact with the database[10].

4.3.2 Logging and Monitoring

SQL Injection is often attacked by causing an error or by repeatedly calling the same page or function. Therefore, careful attention should be paid to the subject of caution when a lot of errors occur or when the same page is repeatedly called by the same IP[11-15].

5. Conclusion

SQL Injection is known to be the most dangerous attack of web hacking technique although it is long since it was published. There are so many ways to deal with this, but if you rely on only one thing, there will be many loopholes. Therefore, it is necessary to utilize the attacks so far and their countermeasures to prevent them before they are attacked. However, as technology continues to evolve, hacking technology will also evolve, so constant research and study will be needed to develop countermeasures.

REFERENCES

[1] J. S. Park. (2016). A Data Driven Index for Convergence Sensor Networks. *Journal of the Korea Convergence Society*, *7(6),* 43-48.

[2] D. Wetter (2012). OWASP Top 10: Zwei *Jahre danach. Datenschutz Und Datensicherheit-DuD, 36(11),* 810-813. DOI : 10.1007/s11623-012-0277-1

[3] K. Fowler. (2012). Confirming and Recovering from SQL Injection Attacks. *SQL Injection Attacks and Defense*, 443-484.
DOI : 10.1016/b978-1-59-749963-7.00010-4[5]

[4] S. D. Curation. (2018). *Digital Curation Centre Template v1 (protocols.io.srwed7e).* Protocols.io.
DOI : 10.17504/protocols.io.srwed7e

[5] A. Pomeroy & Q. Tan. (2011). Effective SQL Injection Attack Reconstruction Using Network Recording. *2011 IEEE 11th International Conference on Computer and Information Technology*.
DOI : 10.1109/cit.2011.103

[6] G. Koziel, B. Krawczynski, J. Marucha, P. Wojcicki & S Skulimowski (2018). Application To Examine Sql Injection Vulnerabilities As A Tool In Computer Science Education. *INTED 2018 Proceedings*.
DOI : 10.21125/inted.2018.1739

[7] J. Halde. (2008.). SQL Injection analysis, Detection and Prevention.
DOI : 10.31979/etd.mnyq-9gq5

[8] J. Y. Choi. (2017). Development of educational programs for managing medical information utilizing medical data generation and analysis techniques. *Journal of Digital Convergence, 15(10),* 377-386.

[9] S. Hong (2013). XSS Attack and Countermeasure: Survey. *Journal of Digital Convergence*, *11(12),* 327-332.

[10] B. R. Kim, B. R. Yoo & S. Y. Jung. (2012). Philippine Learning Management System Design and Implementation. *Journal of the Korea Convergence Society, 3(2),* 1-5.

[11] S. S. Shin, J. I. Kim & J. J. Youn. (2015). Vulnerability Analysis of the Creativity and Personality Education based on Digital Convergence Curation System.

Journal of the Korea Convergence Society, 6(4), 225-234.

[12] J. S. Park. (2016). A Data Driven Index for Convergence Sensor Networks. *Journal of the Korea Convergence Society, 7(6),* 43-48.

[13] S. Hong. (2014). Research on Wireless Sensor Networks Security Attack and Countermeasures : Survey. *Journal of Convergence for Information Technology, 4(4),* 1-6.

[14] P. S. Shin & J. M. Kim. (2014). Security and Hacking on Wireless Networking for Small and Medium Business : Survey. *Journal of Convergence for Information Technology, 4(3),* 15-20.

[15] H. J. Yoon. (2018). Classification of Normal and Abnormal Heart Sounds Using Neural Network. *Journal of Convergence for Information Technology, 8(5),* 131-135.

홍 성 혁(Sunghyuck Hong) 정회원

· 2007년 8월 : Texas Tech University, Computer Science (공학박사)
· 2012년 3월 ~ 현재 : 백석대학교 ICT 학부 정보보호 전공 부교수
· 관심분야 : 블록체인, 사물인터넷 보안, 경량보안프로토콜

· E-Mail : sunghyuck.hong@gmail.com