Research Paper | SQL injection attacks

What threats are small and medium businesses facing in an event of SQL injection attack? and how can they defend against such attacks?

Author:	Karam Ebrahim | 500844969
Learning route:	Cyber Security
University:	Amsterdam University of Applied Sciences
Course:	When Murphy Strikes
Lecturer:	David Bos
Date:	03-02-2023
Project type:	Research Paper | SQL injection attacks
Version:	1.0 | Final

Amsterdam University of Applied Sciences

Executive summary

This research paper examines SQL injection attacks and their risks to small and medium businesses (SMBs). The goal is to suggest the best cyber security measures in order to protect these businesses from SQL injection attacks.

The main question of this research is:

What threats are small and medium businesses facing in an event of SQL injection attack? and how can they defend against such attacks?

To answer the main question it's important to understand what SQL injection is and how these attacks work.

SQL Injection (SQLi) is a type of injection attack that allows malicious SQL statements to be executed. These statements manage a database server that is hidden behind a web application. SQL Injection vulnerabilities allow attackers to bypass application security measures. They can bypass authentication and authorization of a web page or web application and retrieve the entire SQL database's content.

SQL injection can pose a significant risk to SMBs if mitigating controls are not implemented, affecting data confidentiality and integrity, as well as authentication and authorization. A hacker can steal sensitive information such as user data or banking transactions. To avoid such risks, it is very important to consider the best security controls and countermeasures.

In order to protect SMBs from such attacks, the following defense techniques should be considered:

- Input validation
- Prepared statements with parameterized queries
- Using stored procedures
- Allowlist input validation
- Adapting least privilege principle
- Adapting the latest technologies

This research paper also suggests the best actions to deal recover from an SQL injection attack.

To learn more about these types of attacks, their dangers to SMBs and the best security measures they should take to stop them, keep reading this interesting and in-depth research paper.

Table of Contents

1. Introduction

Small and medium-sized businesses (SMBs) are increasingly becoming targets of cyber attacks, with SQL injection attacks being one of the most common methods employed by hackers. These attacks take advantage of flaws in a website's database, granting the attacker unauthorized access to sensitive information such as customer data, financial records, and other confidential business information. A successful SQL injection attack can have devastating consequences, resulting in financial losses, reputational damage, and loss of customer trust.

Given the significant risks posed by SQL injection attacks, it is critical that SMBs take precautions to protect themselves from these types of cyber threats. SMBs may not have the same level of security measures in place as larger companies, making it easier for attackers to find vulnerabilities. Many SMBs, however, lack the resources and expertise to effectively defend themselves against SQL injection attacks.

The main question of this research is:

What threats are small and medium businesses facing in an event of SQL injection attack? and how can they defend against such attacks?

To answer this question, this research will investigate the following sub questions:

- How does an SQL injection attack work?
- What are the threats of a successful SQL injection attack?
- What are the best security practices that small and medium businesses can take to defend against SQL injection attacks?

Because this type of attack poses a significant risk to SMBs, it is critical that they take it seriously. This research paper investigates SQL injection attacks, discusses the risks they pose, and suggests possible defense techniques that can assist SMBs in protecting their systems from these attacks. This research paper is intended for SMBs and their IT (Cyber Security) departments who need to start properly defending their facilities against all cyber threats and the specific type of attack discussed in this research paper.

2. SQL injection

SQL Injection (SQLi) is a type of injection attack that allows malicious SQL statements to be executed. These statements manage a database server that is hidden behind a web application. SQL Injection vulnerabilities allow attackers to bypass application security measures. They can bypass authentication and authorization of a web page or web application and retrieve the entire SQL database's content. SQL Injection can also be used to add, modify, and delete records in the database.

Any website or web application that uses a SQL database, such as MySQL, Oracle, SQL Server, or others, may be vulnerable to SQL Injection. Criminals may exploit it to gain unauthorized access to sensitive data such as customer information, personal information, trade secrets, intellectual property, and so on. SQL injection attacks are one of the most common, widespread, and dangerous web application vulnerabilities.

What are SQL queries?
A SQL query is a request to perform some action on an application database. Queries can also be used to execute commands from the operating system. When a user runs a query, a set of parameters ensures that only the desired records are returned. Attackers take advantage of this during a SQL injection by injecting malicious code into the query's input form.

2.1 How an SQL injection attack works

The first step in a SQL injection attack is to investigate how the targeted database works. This is accomplished by entering a variety of random values into the query and watching how the server responds.

Attackers then use their knowledge of the database to create a query that the server will interpret and execute as a SQL command. A database, for example, may store information about customers who have made a purchase using customer ID numbers. An attacker may enter "CustomerID = 1000 OR 1=1" into the input field instead of searching for a specific customer ID. Because the statement 1=1 is always true, the SQL query would return all available customer IDs as well as any associated data. This enables the attacker to bypass authentication and gain administrative access (Hanna, 2021).

SQL attacks can be written to delete an entire database, bypass the need for credentials, remove records, or add unwanted data, in addition to returning unauthorized information.

2.2 Most common types of SQL injection attacks

SQL injections are generally classified into three types: in-band SQLi (Classic), inferential SQLi (Blind), and out-of-band SQLi. SQL injections can be classified based on how they access backend data and how much damage they can cause.

1. In-band SQL Injection

In-band SQL injection is the most common type of attack. In this type of SQL injection attack, a malicious user uses the same communication channel for the attack and to collect results. The following techniques are the most common types of in-band SQL injection attacks:

- Error-based SQL injection
  With this technique, the attacker performs actions that cause the database to produce error messages. The attacker can potentially use the data from these error messages to gather information about the structure of the database. Error messages are useful when developing a web application or web page, but they can be a vulnerability later on because they expose database information (Crowdstrike, 2022).

- Union-based SQL injection
  With this technique, the UNION SQL operator is used by attackers to combine multiple select statements and return a single HTTP response. This response may contain data that can be leveraged by the attacker. This technique can be used by an attacker to extract information from a database. This is the most common type of SQL injection technique, and it requires more security measures to combat than error-based SQL injection (Crowdstrike, 2022).

2. Inferential SQL Injection

To learn more about the server's structure, the attacker sends data payloads to it and observes its response and behavior. Because the data is not transferred from the website database to the attacker, the attacker is unable to see information about the attack in-band.
Blind SQL injections rely on the server's response and behavioral patterns, so they are typically slower to execute but just as dangerous. Blind SQL injections can be divided into two types:

- Boolean injection
  The attacker sends a SQL query to the database, requesting that it return a result. The outcome depends on whether the query is true or false. The information in the HTTP response will change or remain unchanged depending on the outcome. The attacker can then determine whether the message produced a true or false result (Imperva, 2022).

- Time-based injection
  The attacker sends a SQL query to the database, which causes it to wait (a specific number of seconds) before responding. The attacker can determine whether a query is true or false based on the time it takes the database to respond. Based on the outcome, an HTTP response will be generated either immediately or after a short delay. The attacker can thus determine whether the message they used returned true or false without relying on database data (Imperva, 2022).

3. Out-of-Band SQL Injection

This type of attack is only possible if certain features on the database server used by the web application are enabled. This type of attack is typically used as an alternative to the in-band and inferential SQL Injection techniques. Out-of-band SQL Injection is used when the attacker is unable to use the same channel to launch the attack and gather information, or when a server is too slow or unstable to perform these actions. These techniques rely on the server's ability to generate DNS or HTTP requests in order to transfer data to an attacker (Imperva, 2022).

2.3 Sub-conclusion

SQL injection attacks are a serious threat to SMBs as they allow attackers to gain unauthorized access to sensitive information such as customer data, financial records, and other confidential business information. These attacks exploit vulnerabilities in a website's database and can lead to financial losses, reputational damage, and loss of customer trust. To defend against SQL injection attacks, SMBs should be aware of the different types of SQL injection attacks, such as in-band and inferential SQL injection, and the techniques used in these attacks, such as error-based and union-based SQL injection.

3. The threats of SQL injection attacks

SQL injection can pose a significant risk to SMBs if mitigating controls are not implemented, affecting data confidentiality and integrity, as well as authentication and authorization. A hacker can steal sensitive information such as user data or banking transactions from databases used by vulnerable programs or applications. SQL injection vulnerabilities should never be left unpatched, they must be addressed in all circumstances. If an application's authentication or authorization are compromised, an attacker can log in as any other user, such as an administrator, and increase his privileges.

3.1 The impact of a successful SQL injection attack

SQL injection attacks can have a significant negative impact on SMBs. SQL injection attacks frequently target sensitive company data and private customer information that SMBs have access to. When a malicious user successfully completes a SQL injection attack, the following consequences may occur:

- **Exposes sensitive company data**
  SQL injection allows attackers to retrieve and alter data, potentially exposing sensitive company data stored on the SQL server (Crowdstrike, 2022).

- **Compromise clients privacy**
  An attack could expose private user data, such as credit card numbers, depending on the data stored on the SQL server (Crowdstrike, 2022).

- **Give an attacker administrative access to the system**
  If a database user has administrative privileges, an attacker can use malicious code to gain access to the system (Crowdstrike, 2022).

- **Give an attacker general access to the system**
  An attacker could gain access to the system without knowing a user's credentials if weak SQL commands are used to check user names and passwords. An attacker with general access to the system can cause additional damage by accessing and manipulating sensitive information (Crowdstrike, 2022).

- **Compromise the Integrity of the data**
  SQL injection allows attackers to modify or delete data from the system (Crowdstrike, 2022).

- **Reputation damage**
  A successful SQL injection attack can result in unauthorized transactions and the theft of sensitive financial information, causing the company to suffer significant financial losses.

- **Financial losses**
  Customers may lose trust in a company if their personal information is compromised, which can severely harm its reputation.

3.2 Risk management

Risks that are considered high priority are typically those with large consequences and a high probability of occurrence. SQL injection attacks are considered as a high priority risks that should be quickly addressed.

SQL injection attacks

Likelihood: High

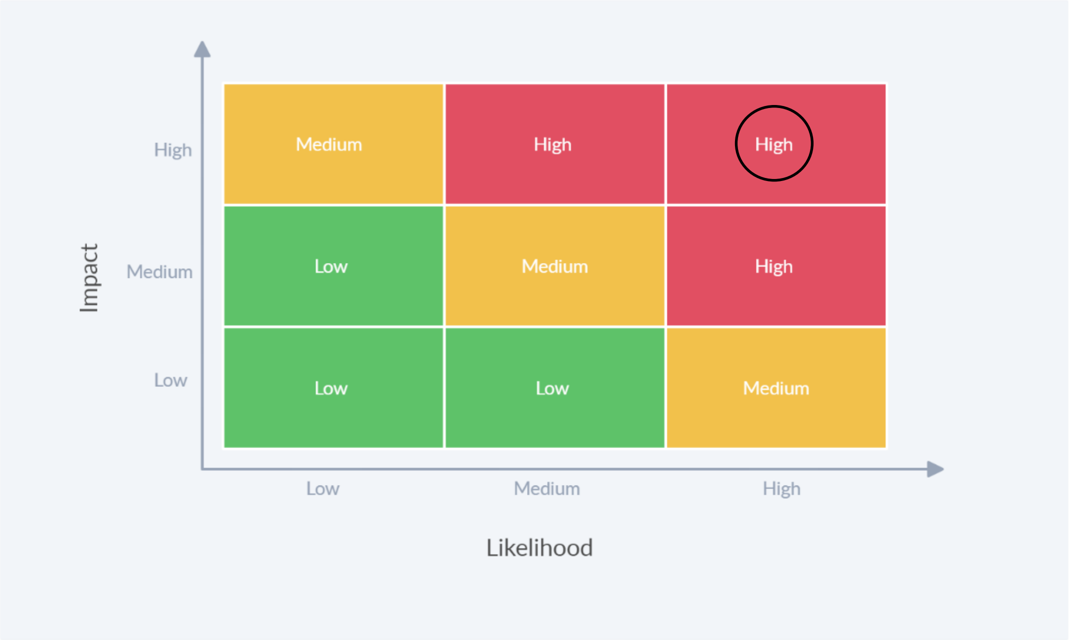
SMBs depend heavily on their online (IT) services for day-to-day operations. However, with the rise in cyber attacks, SMBs are vulnerable to one of the most common and subversive attacks (SQL injection attacks). This type of attack can have a significant impact on SMBs' entire IT systems and websites.

Impact: High

If SMB websites and systems are unavailable for any reason, it can have a significant impact on daily work processes. As a result, it is critical to prioritize database security and IT system security in general in order to detect and prevent similar attacks in the future.

Risk	Risk Description	Likelihood	Impact	Risk Owner	Actions& mitigations
SQL injection	Cyber-attack aims to damage and disrupt databases.	High	High	IT dep, CISO	Implement proper detect and prevent measures.

Risk Matrix – DDoS attacks



	Low	Medium	High
High	Medium	High	High
Medium	Low	Medium	High
Low	Low	Low	Medium

3.3 Sub-conclusion

SQL injection attacks pose a significant threat to SMBs, potentially compromising data confidentiality and integrity, as well as authentication and authorization. The impact of a successful attack can include exposure of sensitive company data, compromise of client privacy, and financial losses. It is crucial for SMBs to prioritize and address SQL injection vulnerabilities through proper detection and prevention measures to mitigate these risks. Additionally, by including SQL injection attacks in their risk management strategy and assigning a specific risk owner, SMBs can take steps to effectively address and prevent these types of attacks.

4. Best security practices to defend against SQL injection attacks

SQL injection attacks, which exploit vulnerabilities in a website's database to gain unauthorized access to sensitive information, are a major threat to SMBs. SMBs must implement a variety of security practices in order to effectively defend against these types of attacks. This chapter will go over some of the best security practices for preventing SQL injection attacks, such as input validation, the use of prepared statements and regular monitoring and updates. This chapter also discuss the significance of employee education and awareness in preventing SQL injection attacks as well. SMBs can protect their websites and databases from these types of attacks and reduce the risk of data breaches and financial losses by implementing these best practices.

4.1 Detecting SQL injection attacks

SQL injection detection methods range from checking server logs to monitoring database errors but the majority of SQL injection vulnerabilities can be quickly and reliably discovered using a web vulnerability scanner. SQL injection can be manually detected by running a set of tests against every entry point in the application (Portswigger, 2022). This usually entails:

- Submitting the single quote character (') and checking for errors or other anomalies.
- Submitting some SQL-specific syntax that evaluates to the entry point's base (original) value and another value, and looking for systematic differences in the resulting application responses.
- Submitting Boolean conditions such as OR 1=1 and OR 1=2, and examining the application's responses for differences.
- Submitting payloads designed to cause time delays when executed within a SQL query, and examining differences in response time.
- Submitting Out-of-band application security testing (OAST) payloads that, when executed within a SQL query, cause an out-of-band network interaction, and monitoring for any interactions that occur.

4.2 Preventing SQL injection attacks

To prevent a SQL injection attack from occurring, SMBs can follow these practices:

- **Input Validation**
  the validation process is designed to verify that the type of input submitted by a user is allowed. Input validation ensures that the type, length, format, and so on are accepted. Only the value that passes validation can be processed. It helps counter any commands inserted into the input string. It is similar to seeing who knocks before opening the door (Klein, 2021).

- **Prepared Statements with Parameterized Queries**
  Prepared statements are used to ensure that no dynamic variables in a query can escape their position. The core query is defined first, followed by the arguments and their types. Because the query knows what type of data is expected, such as a string or a number, they know exactly how to incorporate it into the query without causing problems (Sucuri, 2022).

- **Using Stored Procedures**
  Stored procedures are common SQL operations that are saved on the database and differ only in their arguments. Because stored procedures cannot be dynamically inserted within queries, they make it much more difficult for attackers to execute malicious SQL (Sucuri, 2022).

- **Allowlist Input Validation**
  As a general rule, do not trust on user-submitted data. Allowlist validation can be used to compare user input to a predefined set of known, approved, and defined input. When data is received that does not match the assigned values, it is rejected, thereby protecting the application or website from malicious SQL injections (Sucuri, 2022).

- **Least Privilege**
  Reduce the privileges assigned to each database account in the environment to mitigate the potential impact of a successful SQL injection attack. Consider creating a view that only allows access to a subset of a table and giving the account access to the view rather than the entire table if an account only needs access to a subset of a table. Also, unless absolutely necessary, do not grant accounts the ability to create or delete. Setting up appropriate privilege controls can help to limit the amount of access an attacker has when they compromise an account (Threat Intelligence, 2022).

- **Adopt the latest technologies**
  SQL injection protection is not available in older web development technologies. Use therefore the most recent version of the development environment and language, as well as any associated technologies (Acunetix, 2022).

- **Conduct Regular Penetration Tests**
  Regular database penetration testing can reveal threats such as XSS, injections, insecure passwords, and unpatched vulnerabilities. It can also determine how effective the defenses are against various types of attacks, such as SQL injections. Furthermore, regularly auditing the database for suspicious activity can improve security (Threat Intelligence, 2022).

- **Train and maintain awareness**
  To ensure the security of the business web application, everyone involved in its development must be aware of the risks associated with SQL Injections. All developers, QA staff, DevOps and SysAdmins should receive appropriate security training (Acunetix, 2022).

4.3 Recovering from SQL injection attack

In the event of a SQL injection, the following steps may be taken to address the situation:

- **Locate the Vulnerable Code:** The first step in recovery is to identify where the vulnerability is located. This can be done through manual testing or by using automated SQL injection attack tools (Sucuri, 2022).
- **Remove Injected Content and Backdoors:** Once the location of the malware has been identified, it is necessary to remove any malicious injections and restore the database to a clean state. Additionally, it is important to check for backdoors on the rest of the website and file systems (Sucuri, 2022).
- **Patch the Vulnerability:** Vulnerabilities in databases, applications and third-party components are frequently exploited by hackers. Once identified, patches and updates should be applied to the vulnerable code and any other out-of-date components (Sucuri, 2022).
- **Update Data:** After a compromise, it is important to change all passwords and application secrets as soon as the vulnerability is patched. Additionally, data should be cleaned to ensure that there are no rogue admin users or backdoors present in the database (Sucuri, 2022).
- **Monitor SQL Statements:** A monitor can be set up to identify any rogue SQL statements to the database. A tool that uses behavioral analysis and/or machine learning can help detect indicators of compromise to the website (Sucuri, 2022).
- **Set Up a WAF:** A web application firewall can be set up to filter malicious requests to the website. This can provide protection against new vulnerabilities before patches are made available (Sucuri, 2022).

4.4 Sub-conclusion

SQL injection attacks are a major threat to SMBs and require a variety of security practices to defend against. By implementing best practices such as input validation, prepared statements with parameterized queries, using stored procedures, allowlist input validation, and least privilege, SMBs can protect their websites and databases from these types of attacks and reduce the risk of data breaches and financial losses. Additionally, regular monitoring and updates, as well as employee education and awareness are crucial in preventing SQL injection attacks.

5. Conclusion

SQL injection attacks are a serious threat to SMBs as they allow attackers to gain unauthorized access to sensitive and confidential business information by exploiting vulnerabilities in a website's database. To defend against SQL injection attacks, SMBs should be aware of the different types of SQL injection attacks, such as in-band and inferential SQL injection, and the techniques used in these attacks, such as error-based and union-based SQL injection.

SQL injection attacks pose a significant threat to SMBs, potentially compromising data confidentiality and integrity, as well as authentication and authorization. The impact of a successful attack can include exposure of sensitive company data, compromise of client privacy, and financial losses. It is crucial for SMBs to prioritize and address SQL injection vulnerabilities through proper detection and prevention measures to mitigate these risks. Additionally, by including SQL injection attacks in their risk management strategy and assigning a specific risk owner, SMBs can take steps to effectively address and prevent these types of attacks.

SMBs can start addressing SQL injection attacks by implementing a variety of security measures. By implementing best practices such as input validation, prepared statements with parameterized queries, using stored procedures, allowlist input validation and least privilege, SMBs can protect their websites and databases from these types of attacks and reduce the risk of data breaches and financial losses. Additionally, regular monitoring and updates, as well as employee education and awareness are crucial in preventing SQL injection attacks.

It is important to note that security is an ongoing process and it's crucial to keep up to date with the latest techniques, technologies, and best practices in the field of cybersecurity.

6. Recommendations

Based on the research conducted in this paper, it is clear that SQL injection attacks pose a significant risk to small and medium-sized businesses. To defend against these types of cyber threats, SMBs should implement the following recommendations:

- **Input validation:** SMBs should ensure that all user input is properly validated before being processed. This can help prevent malicious commands from being inserted into the input string and potentially causing harm.

- **Prepared statements with parameterized queries:** SMBs should use prepared statements to ensure that dynamic variables in a query cannot escape their intended position. This can help prevent SQL injection attacks.

- **Use stored procedures:** Stored procedures are common SQL operations that are saved on the database and differ only in their arguments. Because stored procedures cannot be dynamically inserted within queries, they make it much more difficult for attackers to execute malicious SQL.

- **Allowlist input validation:** SMBs should use allowlist validation to compare user input to a predefined set of known, approved, and defined input. This can help protect the application or website from malicious SQL injections.

- **Least Privilege:** SMBs should reduce the privileges assigned to each database account in the environment to mitigate the potential impact of a successful SQL injection attack. This can help limit the amount of access an attacker has when they compromise an account.

- **Adopt the latest technologies:** SMBs should use the most recent version of the development environment and language, as well as any associated technologies, as SQL injection protection is not available in older web development technologies.

- **Conduct regular penetration tests:** SMBs should conduct regular database penetration testing to reveal threats such as XSS, injections, insecure passwords, and unpatched vulnerabilities. This can help determine how effective the defenses are against various types of attacks, such as SQL injections.

- **Train and maintain awareness:** SMBs should ensure that all developers, QA staff, DevOps, and SysAdmins receive appropriate security training to ensure the security of the business web application.

7. References

Acunetix. (2022). *What is SQL Injection (SQLi) and How to Prevent It*. Retrieved from acunetix.com: https://www.acunetix.com/websitesecurity/sql-injection/

Chen, D. (2021). *SQL Injection Attack Detection and Prevention Techniques Using DeepLearning*. Retrieved from researchgate.net: https://www.researchgate.net/publication/349022673_SQL_Injection_Attack_Detection_and_Prevention_Techniques_Using_Deep_Learning

Crowdstrike. (2022, October 10). *crowdstrike.com*. Retrieved from SQL INJECTION (SQLI): HOW TO PROTECT AGAINST SQL INJECTION ATTACKS: https://www.crowdstrike.com/cybersecurity-101/sql-injection/

Hanna, K. T. (2021). *SQL injection*. Retrieved from techtarget.com: https://www.techtarget.com/searchsoftwarequality/definition/SQL-injection

Imperva. (2022). *SQL (Structured query language) Injection*. Retrieved from imperva.com: https://www.imperva.com/learn/application-security/sql-injection-sqli/#:~:text=SQL%20injection%2C%20also%20known%20as,lists%20or%20private%20customer%20details.

Klein, E. (2021). *How to Defend Your Business Against SQL Injections*. Retrieved from logz.io: https://logz.io/blog/defend-against-sql-injections/

Portswigger. (2022). *SQL injection*. Retrieved from portswigger.com: https://portswigger.net/web-security/sql-injection

Sucuri. (2022). *sucuri.net*. Retrieved from What is an SQL Injection Attack?: https://sucuri.net/guides/what-is-sql-injection/

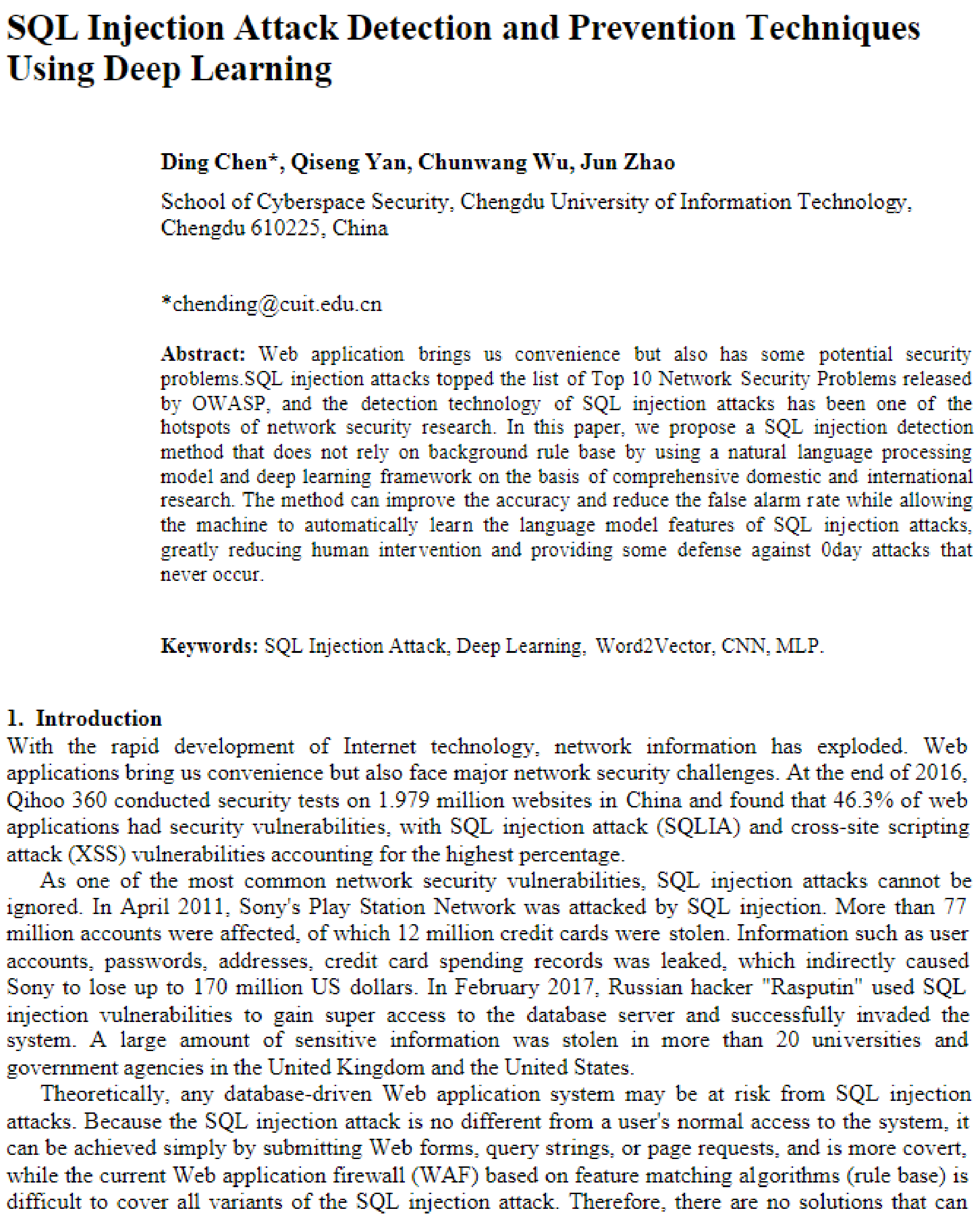
Threat Intelligence. (2022). *SQL Injection - What is it and How to Prevent Attacks?* Retrieved from threatintelligence.com: https://www.threatintelligence.com/sql-injection

Wimukthi, Y. (2022, October). *A comprehensive review of methods for SQL injection attack detection and prevention*. Retrieved from researchgate.net: https://www.researchgate.net/publication/364935556_A_comprehensive_review_of_methods_for_SQL_injection_attack_detection_and_prevention

8. Appendices

8.1 Appendix 1: SQLi Detection & Prevention

Below an article that discusses using deep learning to detect and prevent SQL injection attacks.



SQL Injection Attack Detection and Prevention Techniques Using Deep Learning

Ding Chen*, Qiseng Yan, Chunwang Wu, Jun Zhao

School of Cyberspace Security, Chengdu University of Information Technology, Chengdu 610225, China

*chending@cuit.edu.cn

Abstract: Web application brings us convenience but also has some potential security problems.SQL injection attacks topped the list of Top 10 Network Security Problems released by OWASP, and the detection technology of SQL injection attacks has been one of the hotspots of network security research. In this paper, we propose a SQL injection detection method that does not rely on background rule base by using a natural language processing model and deep learning framework on the basis of comprehensive domestic and international research. The method can improve the accuracy and reduce the false alarm rate while allowing the machine to automatically learn the language model features of SQL injection attacks, greatly reducing human intervention and providing some defense against 0day attacks that never occur.

Keywords: SQL Injection Attack, Deep Learning, Word2Vector, CNN, MLP.

1. Introduction

With the rapid development of Internet technology, network information has exploded. Web applications bring us convenience but also face major network security challenges. At the end of 2016, Qihoo 360 conducted security tests on 1.979 million websites in China and found that 46.3% of web applications had security vulnerabilities, with SQL injection attack (SQLIA) and cross-site scripting attack (XSS) vulnerabilities accounting for the highest percentage.

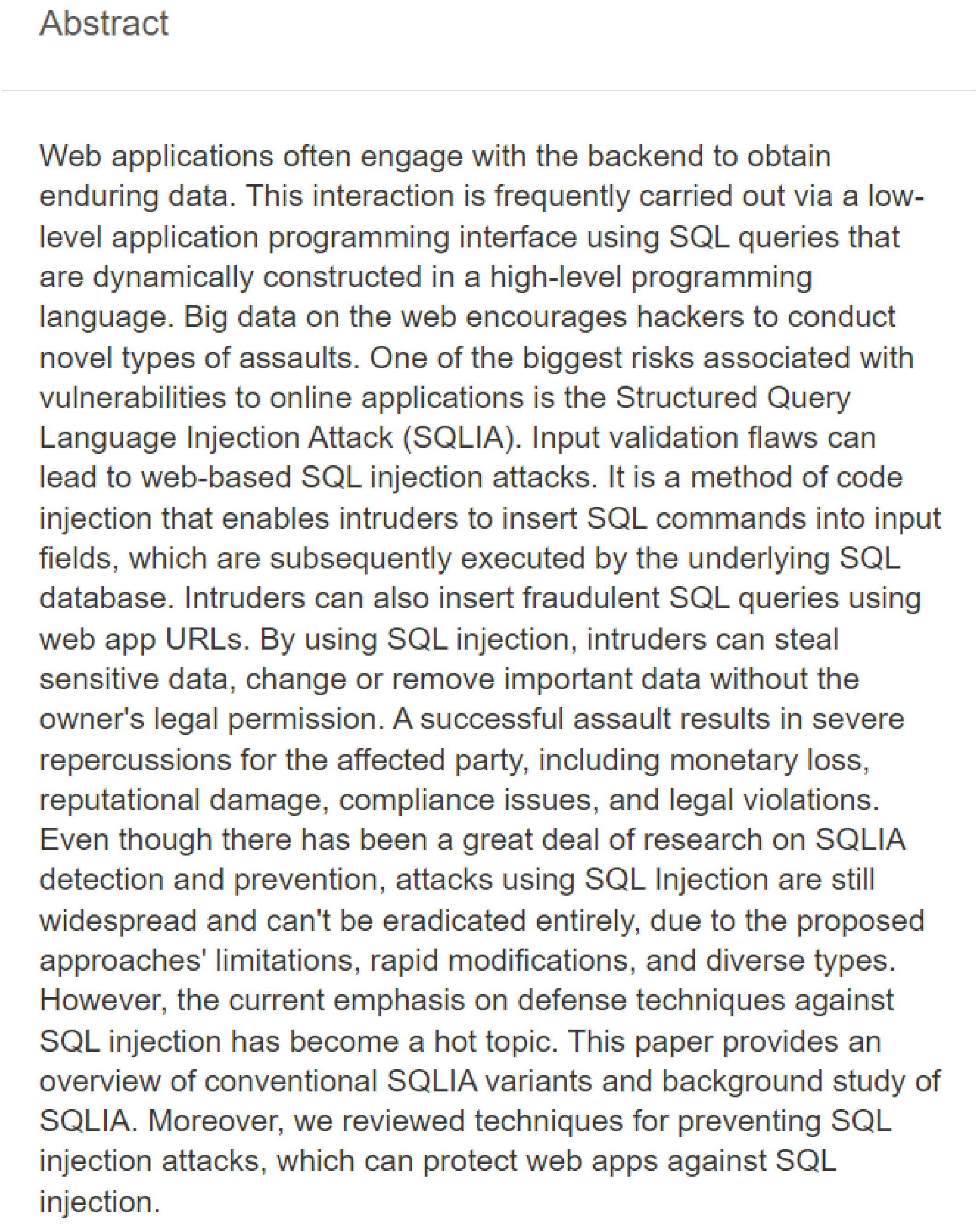
As one of the most common network security vulnerabilities, SQL injection attacks cannot be ignored. In April 2011, Sony's Play Station Network was attacked by SQL injection. More than 77 million accounts were affected, of which 12 million credit cards were stolen. Information such as user accounts, passwords, addresses, credit card spending records was leaked, which indirectly caused Sony to lose up to 170 million US dollars. In February 2017, Russian hacker "Rasputin" used SQL injection vulnerabilities to gain super access to the database server and successfully invaded the system. A large amount of sensitive information was stolen in more than 20 universities and government agencies in the United Kingdom and the United States.

Theoretically, any database-driven Web application system may be at risk from SQL injection attacks. Because the SQL injection attack is no different from a user's normal access to the system, it can be achieved simply by submitting Web forms, query strings, or page requests, and is more covert, while the current Web application firewall (WAF) based on feature matching algorithms (rule base) is difficult to cover all variants of the SQL injection attack. Therefore, there are no solutions that can

Reference: Chen, D. (2021). *SQL Injection Attack Detection and Prevention Techniques Using DeepLearning*. Retrieved from researchgate.net: https://www.researchgate.net/publication/349022673_SQL_Injection_Attack_Detection_and_Prevention_Techniques_Using_Deep_Learning

8.2 Appendix 2: Comprehensive SQL injection detection and prevention methods

Below an article that discusses more comprehensive SQL injection detection and prevention methods.



Abstract

Web applications often engage with the backend to obtain enduring data. This interaction is frequently carried out via a low-level application programming interface using SQL queries that are dynamically constructed in a high-level programming language. Big data on the web encourages hackers to conduct novel types of assaults. One of the biggest risks associated with vulnerabilities to online applications is the Structured Query Language Injection Attack (SQLIA). Input validation flaws can lead to web-based SQL injection attacks. It is a method of code injection that enables intruders to insert SQL commands into input fields, which are subsequently executed by the underlying SQL database. Intruders can also insert fraudulent SQL queries using web app URLs. By using SQL injection, intruders can steal sensitive data, change or remove important data without the owner's legal permission. A successful assault results in severe repercussions for the affected party, including monetary loss, reputational damage, compliance issues, and legal violations. Even though there has been a great deal of research on SQLIA detection and prevention, attacks using SQL Injection are still widespread and can't be eradicated entirely, due to the proposed approaches' limitations, rapid modifications, and diverse types. However, the current emphasis on defense techniques against SQL injection has become a hot topic. This paper provides an overview of conventional SQLIA variants and background study of SQLIA. Moreover, we reviewed techniques for preventing SQL injection attacks, which can protect web apps against SQL injection.

Reference: Wimukthi, Y. (2022, October). *A comprehensive review of methods for SQL injection attack detection and prevention*. Retrieved from researchgate.net: https://www.researchgate.net/publication/364935556_A_comprehensive_review_of_methods_for_SQL_injection_attack_detection_and_prevention