Control methods of Database Security

Last Updated : 15 Dec, 2021

Database Security means keeping sensitive information safe and prevent the loss of data. Security of data base is controlled by Database Administrator (DBA).

The following are the main control measures are used to provide security of data in databases:

1. Authentication
2. Access control
3. Inference control
4. Flow control
5. Database Security applying Statistical Method
6. Encryption

These are explained as following below.

1. **Authentication :**
   Authentication is the process of confirmation that whether the user log in only according to the rights provided to him to perform the activities of data base. A particular user can login only up to his privilege but he can't access the other sensitive data. The privilege of accessing sensitive data is restricted by using Authentication.
   By using these authentication tools for biometrics such as retina and figure prints can prevent the data base from unauthorized/malicious users.
2. **Access Control :**
   The security mechanism of DBMS must include some provisions for restricting access to the data base by unauthorized users. Access control is done by creating user accounts and to control login process by the DBMS. So, that database access of sensitive data is possible only to those people (or users) who are allowed to

access such data and to restrict access to unauthorized persons. The database system must also keep the track of all operations performed by certain user throughout the entire login time.

3. **Inference Control :**
   This method is known as the countermeasures to statistical database security problem. It is used to prevent the user from completing any inference channel. This method protect sensitive information from indirect disclosure.
   Inferences are of two types, identity disclosure or attribute disclosure.
4. **Flow Control :**
   This prevents information from flowing in a way that it reaches unauthorized users. Channels are the pathways for information to flow implicitly in ways that violate the privacy policy of a company are called convert channels.
5. **Database Security applying Statistical Method :**
   Statistical database security focuses on the protection of confidential individual values stored in and used for statistical purposes and used to retrieve the summaries of values based on categories. They do not permit to retrieve the individual information. This allows to access the database to get statistical information about the number of employees in the company but not to access the detailed confidential/personal information about the specific individual employee.
6. **Encryption :**
   This method is mainly used to protect sensitive data (such as credit card numbers, OTP numbers) and other sensitive numbers. The data is encoded using some encoding algorithms.
   An unauthorized user who tries to access this encoded data will face difficulty in decoding it, but authorized users are given decoding keys to decode data.

Statistical Database Security

Last Updated : 27 Sep, 2022

Prerequisite – Control Methods of Database Security Certain databases may contain confidential or secret data of individuals of country like (Aadhaar numbers, PAN card numbers) and this database should not be accessed by attackers. So, therefore it should be protected from user access. The database which contains details of huge population is called Statistical databases and it is used mainly to produce statistics on various populations. But Users are allowed to retrieve certain statistical information of population like averages of population of particular state/district etc and their sum, count, maximum, minimum, and standard deviations, etc. It is the responsibility of ethical hackers to monitor Statistical Database security statistical users are not permitted to access individual data, such as income of specific person, phone number, Debit card numbers of specified person in database because Statistical database security techniques prohibit retrieval of individual data. It is also responsibility of DBMS to provide confidentiality of data about individuals. **Statistical Queries :** The queries which allow only aggregate functions such as COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION are called statistical queries. Statistical queries are mainly used for knowing population statistics and in companies/industries to maintain their employees' database etc. **Example –** Consider the following examples of statistical queries where EMP_SALARY is confidential database that contains the income of each employee of company.

Query-1:

```
SELECT COUNT(*)
FROM EMP_SALARY
WHERE Emp-department = '3';
```

Query-2:

```
SELECT AVG(income)
FROM EMP_SALARY
WHERE Emp-id = '2';
```

Here, the **"Where"** condition can be manipulated by attacker and there is chance to access income of individual employees or confidential data of employee if he knows id/name of particular employee. The possibility of accessing individual information from statistical queries is reduced by using the following measures –

1. **Partitioning of Database –** This means the records of database must be not be stored as bulk in single record. It must be divided into groups of some minimum size according to confidentiality of records. The advantage of Partitioning of database is queries can refer to any complete group or set of groups, but queries cannot access the subsets of records within a group. So, attacker can access at most one or two groups which are less private.
2. If no statistical queries are permitted whenever number of tuples in population specified by selection condition falls below some threshold.
3. Prohibit sequences of queries that refer repeatedly to same population of tuples.

Next Article

Recovery With Concurrent Transactions

Similar Reads

What is Cloud Database Security in DBMS?

Cloud Database Security in DBMS is a system located on a cloud computing platform. It consists of organized data set controlled and...

Intrusion Detection System (IDS)

Last Updated : 05 Feb, 2025

Intrusion is when an attacker gets unauthorized access to a device, network, or system. Cyber criminals use advanced techniques to sneak into organizations without being detected.

Intrusion Detection System (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations. Each illegal activity or violation is often recorded either centrally using an SIEM system or notified to an administration. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between 'bad connections' (intrusion/attacks) and 'good (normal) connections'.

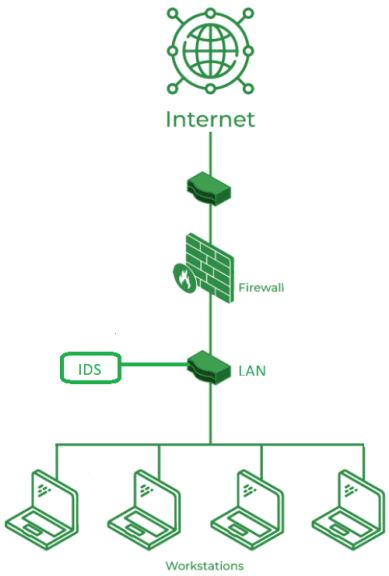
Common Methods of Intrusion

- **Address Spoofing:** Hiding the source of an attack by using fake or unsecured proxy servers making it hard to identify the attacker.
- **Fragmentation**: Sending data in small pieces to slip past detection systems.
- **Pattern Evasion:** Changing attack methods to avoid detection by IDS systems that look for specific patterns.
- **Coordinated Attack:** Using multiple attackers or ports to scan a network, confusing the IDS and making it hard to see what is happening.

Working of Intrusion Detection System(IDS)

- An IDS (Intrusion Detection Syst) monitors the traffic on a computer network to detect suspicious activity.

- It analyzes the data flowing through the network to look for patterns and signs of abnormal behavior.
- The IDS compares the network activity to a set of predefined rules and patterns to identify any activity that might indicate an attack or intrusion.
- If the IDS detects something that matches one of these rules or patterns, it sends an alert to the system administrator.
- The system administrator can then investigate the alert and take action to prevent any damage or further intrusion.



Intrusion Detection System (IDS)

Classification of Intrusion Detection System(IDS)

Intrusion Detection System are classified into 5 types:

- **Network Intrusion Detection System (NIDS):** Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the collection of known attacks. Once an attack is identified or abnormal behavior is observed, the alert can be sent to the administrator. An example of a NIDS is installing it on the subnet where **firewalls** are located in order to see if someone is trying to crack the **firewall**.

- **Host Intrusion Detection System (HIDS):** Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission-critical machines, which are not expected to change their layout.

- **Hybrid Intrusion Detection System:** Hybrid intrusion detection system is made by the combination of two or more approaches to the intrusion detection system. In the hybrid intrusion detection system, the host agent or system data is combined with network information to develop a complete view of the network system. The hybrid intrusion detection system is more effective in comparison to the other intrusion detection system. Prelude is an example of Hybrid IDS.

- **Application Protocol-Based Intrusion Detection System (APIDS):** An application Protocol-based Intrusion Detection System (APIDS) is a system or agent that generally resides within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application-specific protocols. For example, this would monitor the SQL protocol explicitly to the middleware as it transacts with the database in the web server.

- **Protocol-Based Intrusion Detection System (PIDS):** It comprises a system or agent that would consistently reside at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the HTTPS protocol stream and accepting the related HTTP protocol. As HTTPS is unencrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS

- **Signature-Based Detection:** Signature-based detection checks network packets for known patterns linked to specific threats. A signature-based IDS compares packets to a database of attack signatures and raises an alert if a match is found. Regular updates are needed to detect new threats, but unknown attacks without signatures can bypass this system

Intrusion Detection System Evasion Techniques

- **Fragmentation:** Dividing the packet into smaller packet called fragment and the process is known as fragmentation. This makes it impossible to identify an intrusion because there can't be a malware signature.
- **Packet Encoding:** Encoding packets using methods like Base64 or hexadecimal can hide malicious content from signature-based IDS.
- **Traffic Obfuscation:** By making message more complicated to interpret, obfuscation can be utilised to hide an attack and avoid detection.
- **Encryption:** Several security features such as data integrity, confidentiality, and data privacy, are provided by encryption. Unfortunately, security features are used by malware developers to hide attacks and avoid detection.

Detection Method of IDS

- **Signature-Based Method:** Signature-based IDS detects the attacks on the basis of the specific patterns such as the number of bytes or a number of 1s or the number of 0s in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures. Signature-based IDS can easily detect the attacks whose pattern (signature) already exists in the system but it is quite difficult to detect new malware attacks as their pattern (signature) is not known.

- **Anomaly-Based Method:** Anomaly-based IDS was introduced to detect unknown malware attacks as new malware is developed

rapidly. In anomaly-based IDS there is the use of machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in the model. The machine learning-based method has a better-generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations.

Comparison of IDS with Firewalls

IDS and firewall both are related to network security but an IDS differs from a firewall as a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls restrict access between networks to prevent intrusion and if an attack is from inside the network it doesn't signal. An IDS describes a suspected intrusion once it has happened and then signals an alarm.

Why Are Intrusion Detection Systems (IDS) Important?

An Intrusion Detection System (IDS) adds extra protection to your cybersecurity setup, making it very important. It works with your other security tools to catch threats that get past your main defenses. So, if your main system misses something, the IDS will alert you to the threat.

Placement of IDS

- The most optimal and common position for an IDS to be placed is behind the firewall. The **'behind-the-firewall'** placement allows the IDS with high visibility of incoming network traffic and will not receive traffic between users and network.
- In cases, where the IDS is positioned beyond a network's firewall, it would be to defend against noise from internet or defend against attacks such as port scans and network mapper. An IDS in this position would monitor layers 4 through 7 of the OSI model and would use Signature-based detection method. Showing the number of attemepted breacheds instead of actual breaches that made it through the firewall is better as it reduces the amount of false positives. It also takes less time to discover successful attacks against network.

- An advanced IDS incorporated with a firewall can be used to intercept complex attacks entering the network. Features of and bridging mode. All of this in turn potentially reduces cost and operational complexity.
- Another choice for IDS placement is within the network. This choice reveals attacks or suspicious activity within the network. Not acknowledging security inside a network is detrimental as it may allow users to bring about security risk, or allow an attacker who has broken into the system to roam around freely.

Benefits of IDS

- **Detects Malicious Activity:** IDS can detect any suspicious activities and alert the system administrator before any significant damage is done.
- **Improves Network Performance:** IDS can identify any performance issues on the network, which can be addressed to improve network performance.
- **Compliance Requirements:** IDS can help in meeting compliance requirements by monitoring network activity and generating reports.
- **Provides Insights:** IDS generates valuable insights into network traffic, which can be used to identify any weaknesses and improve network security.

Disadvantages of IDS

- **False Alarms**: IDS can generate false positives, alerting on harmless activities and causing unnecessary concern.
- **Resource Intensive**: It can use a lot of system resources, potentially slowing down network performance.
- **Requires Maintenance**: Regular updates and tuning are needed to keep the IDS effective, which can be time-consuming.
- **Doesn't Prevent Attacks**: IDS detects and alerts but doesn't stop attacks, so additional measures are still needed.
- **Complex to Manage**: Setting up and managing an IDS can be complex and may require specialized knowledge.

SQL Injection

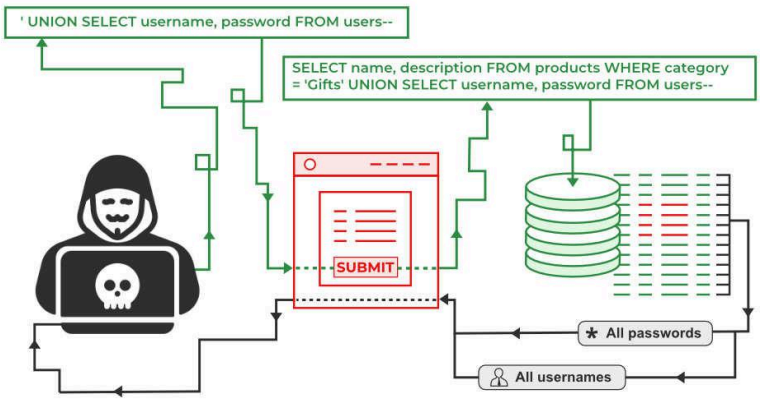
Last Updated : 13 Jan, 2025

SQL Injection is a security flaw in **web applications** where attackers insert harmful **SQL** code through user inputs. This can allow them to access sensitive data, change database contents or even take control of the system. It's important to know about SQL Injection to keep web applications secure.

In this article, We will learn about **SQL Injection** by understanding **How to detect SQL injection vulnerabilities, the** impact of a **successful SQL injection attack** and so on.

What is SQL Injection?

SQL Injection (SQLi) is a **security vulnerability** that occurs when an attacker is able to **manipulate** a web application's database queries by inserting **malicious SQL code** into user input fields. These injected queries can manipulate the underlying database to **retrieve**, **modify**, or **delete sensitive data**. In some cases, attackers can even escalate privileges, gaining full control over the database or **server**.



How Does SQL Injection Work?

SQL Injection typically works when a web application improperly validates user input, allowing an attacker to **inject malicious** SQL code. For example, if a web application uses user input (e.g., a username or

password) and directly inserts it into an SQL query without **proper sanitization**, an attacker can manipulate the query to perform unintended actions.

Example

Suppose we have an application based on student records. Any student can view only his or her records by entering a unique and private student ID.

```
SELECT * FROM STUDENT WHERE
STUDENT-ID == 12222345 or 1 = 1
```

SQL Injection based on 1=1 is always true. As we can see in the above example, **1=1** will return all records for which this holds true. So basically, all the student data is compromised. Now the malicious user can also similarly use other SQL queries.

Query 1:

```
SELECT * FROM USER WHERE
USERNAME = “” AND PASSWORD=””
```

Now the malicious attacker can use the ‘=’ operator to retrieve **private** and **secure** user information. So following query when executed retrieves protected data, not intended to be shown to users.

Query 2:

```
SELECT* FROM User WHERE
(Username = “” OR 1=1) AND
(Password=”” OR 1=1).
```

Since `'1'='1'` is always true, the attacker could gain unauthorized access to the application.

Types of SQL Injection

There are several types of SQL Injection attacks, each with different methods of exploiting the vulnerability. These include:

1. In-band SQL Injection

In-band SQL Injection is the most common type, where the attacker sends **malicious SQL queries** directly through the application interface. This method allows attackers to extract sensitive information or manipulate the database.

Example:

```
SELECT * FROM users WHERE id = 1; -- OR 1=1 --
```

This query would retrieve all users in the database because 1=1 is always true.

2. Error-based SQL Injection

This type of SQL injection exploits error messages generated by the database. Attackers can use the information provided in error messages to learn about the database structure and craft more sophisticated attacks.

Example:

```
SELECT * FROM users WHERE id = 1' -- ;
```

An error message could reveal details about the database schema, allowing the attacker to refine their attack.

3. Blind SQL Injection

In blind SQL injection, the attacker does not receive error messages but can infer information about the database by observing the behavior of the application. The attacker uses boolean conditions to test various aspects of the database.

Example:

```
SELECT * FROM users WHERE id = 1 AND 1=1;
```

If the response is different when `1=1` is changed to `1=0`, the attacker can infer information about the database.

4. Out-of-band SQL Injection

Out-of-band SQL injection relies on the attacker using a different communication channel to exfiltrate data from the database. This type of attack is less common but can be very effective.

Example:

```
SELECT * FROM users WHERE id = 1; -- ;
```

The attacker might direct the database to send a DNS request or HTTP request with the extracted data.

5. Time-based Blind SQL Injection

In this form of blind SQL injection, the attacker sends a query that causes a time delay (e.g., using `SLEEP`), allowing them to infer whether the query was true or false based on the response time.

Example:

```
SELECT * FROM users WHERE id = 1 AND 1=1 SLEEP(5);
```

If the query takes 5 seconds to execute, the attacker knows that the query is true.

Impact of SQL Injection Attacks

- **Unauthorized access to sensitive data**: Attackers can retrieve personal, financial, or confidential information stored in the database.
- **Data integrity issues**: Attackers can modify, delete, or corrupt critical data, impacting the application's functionality.
- **Privilege escalation**: Attackers can bypass authentication mechanisms and gain administrative privileges.

- **Service downtime**: SQL injection can overload the server, causing performance degradation or system crashes.
- **Reputation damage**: A successful attack can severely harm the reputation of an organization, leading to a loss of customer trust.

Detecting SQL Injection Vulnerabilities

To detect SQL injection vulnerabilities, consider the following:

- **Input validation testing**: Test inputs by inserting special characters like `--`, `;`, `'`, or `"` to see if they cause errors or unintended behavior.
- **Automated tools**: Use tools like SQLMap, **Burp Suite**, or **OWASP ZAP** to scan for vulnerabilities.
- **Review source code**: Inspect source code for insecure coding practices such as concatenating user inputs directly into SQL queries.
- **Monitor error messages**: Unexpected or detailed error messages can indicate that the application is vulnerable.
- **Penetration testing**: Regularly perform penetration testing to identify security gaps.

Preventing SQL Injection Attacks

There are several best practices to prevent SQL injection attacks:

1. Use Prepared Statements and Parameterized Queries

Prepared statements and parameterized queries ensure that user inputs are treated as data rather than part of the SQL query. This approach eliminates the risk of SQL injection.

Example in PHP (using MySQLi):

```
$stmt = $conn->prepare("SELECT * FROM users WHERE username
= ? AND password = ?");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
```

Stored procedures are predefined SQL queries stored in the database. These procedures can help prevent SQL injection because they don't dynamically construct SQL queries.

Example:

```
CREATE PROCEDURE GetUserByUsername (IN username
VARCHAR(50))
BEGIN
   SELECT * FROM users WHERE username = username;
END;
```

3. Whitelist Input Validation

Ensure that user inputs are validated before being used in SQL queries. Only allow certain characters and patterns, such as alphanumeric input, for fields like usernames or email addresses.

4. Use ORM Frameworks

Object-Relational Mapping (ORM) frameworks like **Hibernate** or **Entity Framework** can help prevent SQL injection by automatically handling query generation, preventing dynamic query construction.

5. Restrict Database Privileges

Grant the minimum required database permissions to users. Ensure that applications can only perform necessary actions (e.g., SELECT, INSERT), and restrict permissions like DROP TABLE or ALTER.

6. Error Handling

Configure the database and application to not display detailed error messages to the user. Instead, log errors internally and display generic error messages to end users.

Conclusion

SQL injection remains one of the most dangerous **security vulnerabilities** in web applications. By understanding how SQL injection attacks work and following best practices for prevention, developers can protect their applications from **unauthorized data access**, **data corruption**, and other security breaches. Ensuring secure **input validation**, using **parameterized queries**, and regularly testing for vulnerabilities are essential to maintaining a secure web application.

Next Article

SQL Performance Tuning

Similar Reads

SQL Performance Tuning

SQL performance tuning is an essential aspect of database management that helps improve the efficiency of SQL queries and ensures that...

15+ min read

SQL Stored Procedures

SQL Stored Procedures are a powerful feature in database management systems (DBMS) that allow developers to encapsulate SQL code and...

15+ min read

Dynamic SQL in SQL Server

In SQL Server, at times the SQL Queries need to be dynamic and not static, meaning the complete SQL query may be built dynamically at run...

15+ min read

SQL | Subquery