Database Security

HNEU

Dept. IS

Assoc. prof. Fedko V.V.

Literature

1. Ramez Elmasri, Shamkant B. Navathe. Fundamentals of database systems: Sevens edition. - Pearson Education, 2016. – 1273 p.

2. Database Security. https://www.ibm.com/cloud/learn/database-security

3. SQL Server Security. https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/sql-server-security

4. Ethical Hacking. https://www.guru99.com/what-is-hacking-an-introduction.html

Test questions

en:

1. Describe threats to databases.

2. Describe purpose of inference control

3. What is SQL injection?

ru:

1. Опишите угрозы для баз данных.

2. Опишите цель контроля вывода

3. Что такое SQL-инъекция?

Contents

1. Main Concepts
2. Control Measures
3. Common threats and challenges
4. Database Security Priority Areas

1. Main Concepts

1. Types of Security
2. Threats to Databases
3. Database Security - Part of a Common System

1. Types of Security

Database Security Issues

Legal and ethical issues on the right to access information - for example, some information may be considered **confidential** and may not be legally accessible to outside organizations or persons.

Policy issues at the governmental, institutional or corporate level regarding which types of information should not be publicly available, such as **credit ratings** and **personal medical** records.

Systemic problems, such as system **levels**, at which various security functions should be performed, for example, whether the security function should be handled at the physical hardware level, at the operating system level, or at the DBMS level.

The need for some organizations **to identify multiple levels of security** and classify data and users based on these classifications - for example, **top secret, secret, confidential and unclassified**. An organization's security policy that allows access to various data classifications should be mandatory.

Threats to Databases

Loss of integrity. Database integrity refers to the requirement to **protect** information from **incorrect changes**. Integrity is lost if unauthorized changes are made to the data as a result of deliberate or accidental actions. If the loss of system or data integrity is not resolved, further use of the infected system or corrupted data may lead to **inaccuracies, fraud or erroneous decisions**.

Loss of availability. Database availability means the accessibility of objects to a user or program that has a **legal right** to these data objects.

Loss of confidentiality. Database confidentiality refers to the protection of data from unauthorized disclosure. Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.

Database Security - Part of a Common System

Database security is a complex and challenging endeavor that involves all aspects of information security technologies and practices. It’s also naturally at odds with database usability.

The security of the entire system is as strong as its **weakest link**, the database can be compromised, even if it was essentially completely secure.

The more accessible and usable the database, the more vulnerable it is to security threats; the more invulnerable the database is to threats, the more difficult it is to access and use. (This paradox is sometimes referred to as **Anderson’s Rule**.)

Database security must **address** and protect the following:

- The **data** in the database
- The database management system (**DBMS**)
- Any associated **applications**
- The physical database **server** and/or the virtual database server and the underlying hardware
- The computing and/or **network** infrastructure used to access the database

2. Control Measures

1. Types of control measures
2. Access control
3. Inference control
4. Flow control
5. Data encryption

1. Types of control measures

To protect databases from threats, there are general types of control measures:

1. access control,
2. inference control,
3. flow control,
4. encryption.

2. Access control

Database access control is a method of allowing access to company’s sensitive data only to those people (database users) who are allowed to access such data and to restrict access to unauthorized persons.

It includes two main components: authentication and authorization.

Authentication is a method of verifying the identity of a person who is accessing your database.

Authorization determines whether a user should be allowed to access the data or make the transaction he’s attempting.

Access control systems come in three variations:

1. Discretionary Access Control (DAC),
2. Mandatory Access Control (MAC),
3. Role Based Access Control (RBAC).

1. Discretionary Access Control (DAC)

Discretionary Access Control is a type of access control system that holds the business owner responsible for deciding **which people** are **allowed** in a specific location, physically or digitally.
DAC is the least restrictive compared to the other systems, as it essentially allows an individual complete control over any objects they own, as well as the programs associated with those objects.
The drawback to Discretionary Access Control is the fact that it gives the end user complete control to set security level settings for other users and the permissions given to the end user are inherited into other programs they use which could potentially lead to malware being executed without the end user being aware of it.

2. Mandatory Access Control (MAC)

With **mandatory access control**, this security policy is centrally controlled by a security policy **administrator**; users do not have the ability to override the policy and, for example, grant access to files that would otherwise be restricted.

By contrast, discretionary access control (DAC), which also governs the ability of subjects to access objects, allows users the ability to make policy decisions and/or assign security attributes.

MAC-enabled systems allow policy administrators to implement organization-wide security policies. Under MAC (and unlike DAC), users cannot override or modify this policy, either accidentally or intentionally. This allows security administrators to define a central policy that is guaranteed (in principle) to be enforced for all users (ie. **military** institutions).

3. Role Based Access Control (RBAC)

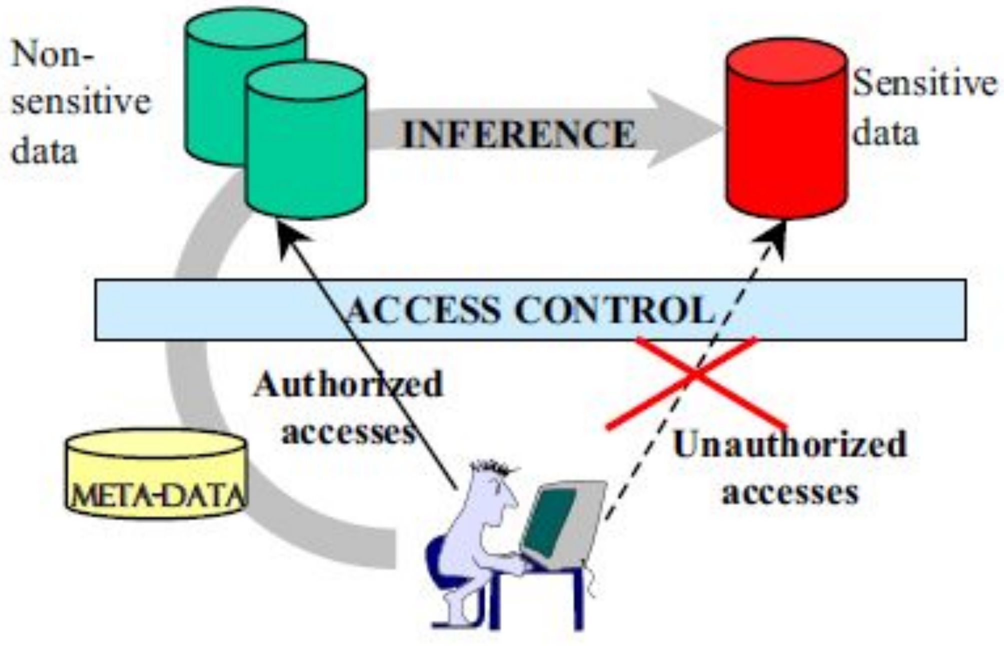
Role-based access control (RBAC) is a policy-neutral access-control mechanism defined around **roles** and **privileges**. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments.

RBAC addresses many needs of commercial and government organizations. RBAC can be used to facilitate administration of **security** in **large organizations** with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication.

3. Inference control

Inference is a way to infer or derive sensitive data from non sensitive data.

Statistical databases are used to provide statistical information or summaries of values based on various criteria. For example, a database for population statistics may provide statistics based on age groups, income levels, household size, education levels, and other criteria. Statistical database users such as government statisticians or market research firms are allowed to access the database to retrieve statistical information about a population **but not to access the detailed confidential information about specific individuals**. Security for statistical databases must ensure that information about individuals cannot be accessed.



Farkas, Csilla & Jajodia, Sushil. (2002). The Inference Problem: A Survey.. SIGKDD Explorations. 4. 6-11. 10.1145/772862.772864.

4. Flow control

Flow Control – Distributed systems encompass a lot of data flow from one site to another and also within a site. Flow control prevents data from being transferred in such a way that it can be accessed by unauthorized agents. A flow policy lists out the channels through which information can flow. It also defines security classes for data as well as transactions.

Prevents information from flowing in such a way that it reaches unauthorized users.

Suitable for database over multiuser system or network

Flow control checks that information contained in some data objects does not flow

(explicitly or implicitly) into less protected objects.

- A clearance for a security class can be assigned to each application program.
- Like a DB user, each application program is

subjected to the same read/write restrictions.

5. Data encryption

Suppose we communicate data, but our data falls into the hands of a nonlegitimate user. In this situation, by using encryption we can disguise the message so that even if the transmission is diverted, the message will not be revealed.

Database encryption is the process of converting **data**, within a **database**, in plain text format into a meaningless cipher text by means of a suitable algorithm.

Database decryption is converting the meaningless cipher text into the original information using keys generated by the **encryption** algorithms.

It enhances security and privacy when access controls are bypassed, because in cases of data loss or theft, encrypted data cannot be easily understood by unauthorized persons.

3. Common threats and challenges

1. Insider threats
2. Human error
3. Exploitation of database software vulnerabilities
4. SQL/NoSQL injection attacks
5. Buffer overflow exploitations
6. Denial of service (DoS/DDoS) attacks
7. Malware
8. Attacks on backups

1. Insider threats

An internal threat is a security risk from any of three sources with privileged access to the database:

An evil insider who intends to do harm

A negligent insider who makes mistakes that make the database vulnerable to attack

An infiltrator is an outsider who somehow obtains credentials using a scheme such as phishing, or by gaining access to the credential database itself.

Insider threats are often the result of giving **too many employees access privileges** for privileged users.

2. Human error

- Accidents,
- weak passwords,
- password sharing,

and other unwise or uninformed user behaviours continue to be the cause of nearly half (49%) of all reported data breaches.

3. Exploitation of database software vulnerabilities

Hackers make a living by detecting vulnerabilities in all types of software, including database management software, and targeting them vulnerabilities.

All major vendors of commercial database software and the open source database management platform issue regular **security patches** to address these vulnerabilities, but not applying these patches on time can increase your vulnerability.

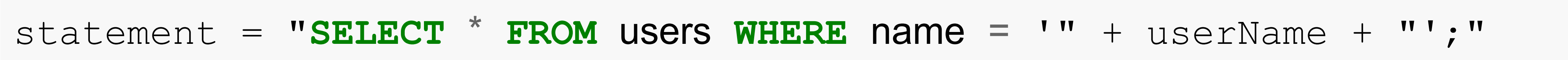
4. SQL/NoSQL injection attacks

A database-specific threat involves inserting arbitrary SQL or non-SQL attack lines into database queries served by web applications or HTTP headers.

Implication:

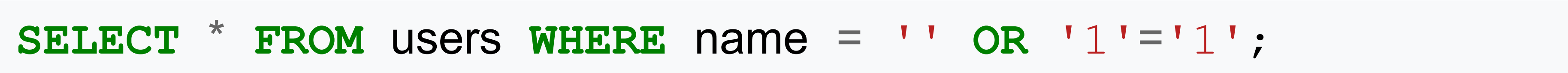
- An attacker can inject malicious content into the vulnerable fields.
- Sensitive data like User Names, Passwords, etc. can be read from the database.
- Database data can be modified (Insert/Update/ Delete).
- Administration Operations can be executed on the database

Example. **Incorrectly filtered escape characters**



```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

setting the "userName" variable as: ' OR '1'='‘1



```
SELECT * FROM users WHERE name = '' OR '1'='1';
```

https://www.guru99.com/learn-sql-injection-with-practical-example.html

5. Buffer overflow exploitations

Buffer overflow occurs when a process attempts to write **more data** to a **fixed-length block** of memory than it is allowed to hold.

Buffer overflow vulnerabilities typically **occur** in code that:

- Relies on external data to control its behavior
- Depends upon properties of the data that are enforced outside of the immediate scope of the code
- Is so complex that a programmer cannot accurately predict its behavior

In this case, a buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. **Writing outside** the bounds of a block of allocated memory **can**

- corrupt data,
- crash the program,
- cause the execution of malicious code.

Attackers may use the excess data, stored in adjacent memory addresses, as a foundation from which to launch attacks.

Buffer Overflow Solutions:

- to use automatic protection at the language level
- bounds-checking enforced at run-time, which prevents buffer overrun by automatically checking that data written to a buffer is within acceptable boundaries.

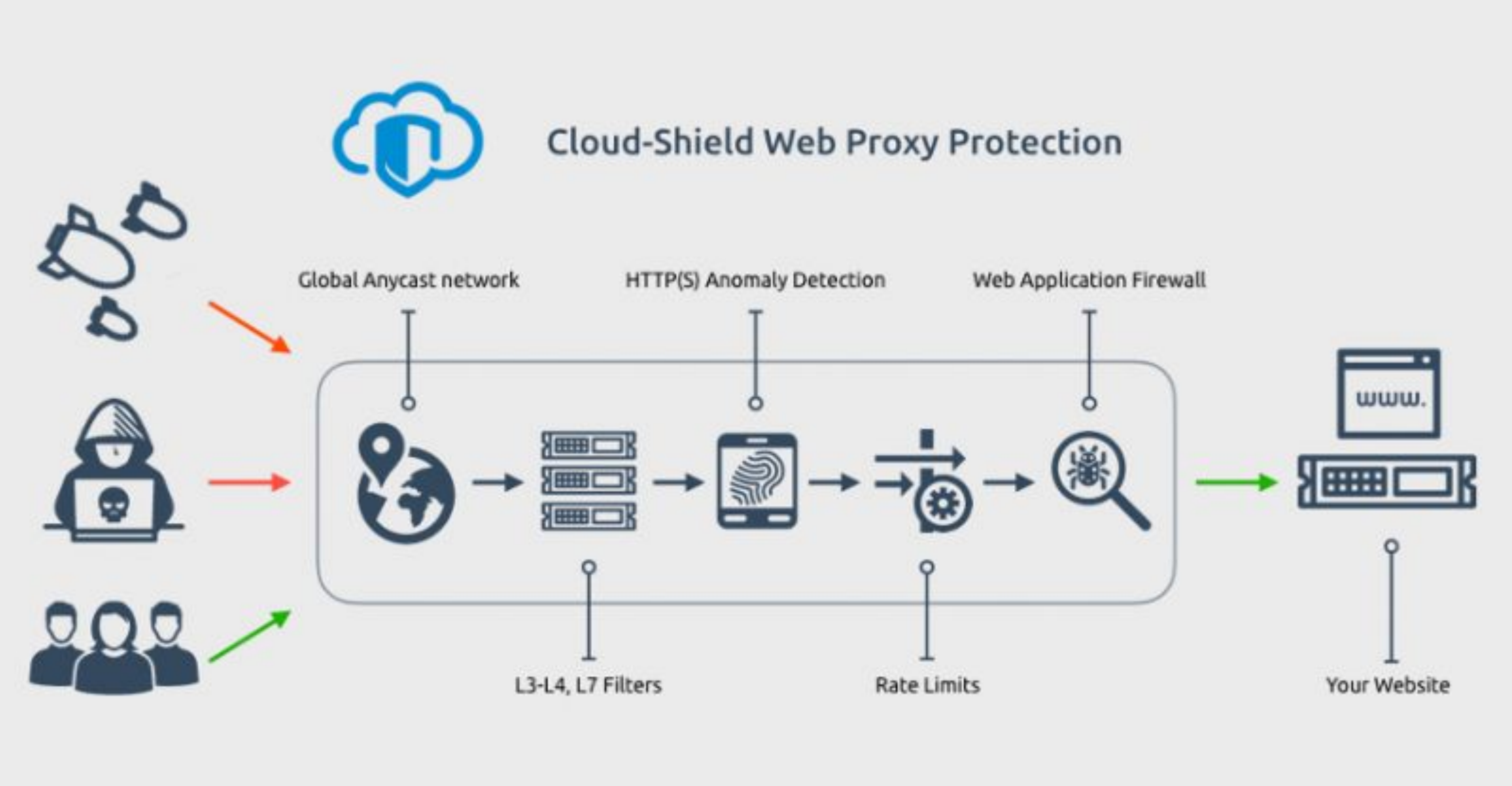
6. Denial of service (DoS/DDoS) attacks

In a **denial of service** (DoS) attack, the attacker floods the target server — in this case, the database server — with so many queries that the server can no longer perform legitimate queries from real users, and in many cases the server becomes unstable or crashesor making it extremely slow.

In a **distributed denial of service** (DDoS) attack, a stream arrives from multiple servers, making it difficult to stop the attack.

DoS/DDoS Attacks Solutions:

- security patches for operating systems,
- router configuration,
- firewalls
- intrusion detection systems.



7. Malware

Malware is software written specifically to exploit vulnerabilities or otherwise cause damage to the database. Malware may arrive via any endpoint device connecting to the database’s network.

The objective of targeting an organization would be to steal sensitive data, disrupt business operations or physically damage computer controlled equipment. Trojans, viruses, and worms can be used to achieve the above-stated objectives.

Malware Solutions:
A range of antivirus software, firewalls and other strategies are used to help protect against the introduction of malware, to help detect it if it is already present, and to recover from malware-associated malicious activity and attacks.

8. Attacks on backups

Threats are compounded by the following:

Growing data volumes: Data capture, storage, and processing continues to grow exponentially across nearly all organizations. Any data security tools or practices need to be highly scalable to meet near and distant future needs.

Infrastructure sprawl: Network environments are becoming increasingly complex, particularly as businesses move workloads to multicloud or hybrid cloud architectures, making the choice, deployment, and management of security solutions ever more challenging.

Increasingly stringent regulatory requirements: The worldwide regulatory compliance landscape continues to grow in complexity, making adhering to all mandates more difficult.

Cybersecurity skills shortage: Experts predict there may be as many as 8 million unfilled cybersecurity positions by 2022..

4. Database Security Priority Areas

1. Database Security Extension
2. Physical security, access controls and accounts
3. Encryption, software and applications
4. Backup and Auditing

1. Database Security Extension

Because databases are nearly always network-accessible, any security threat to any component within or portion of the network infrastructure is also a threat to the database, and any attack impacting a user’s device or workstation can threaten the database. Thus, database security must extend far beyond the confines of the database alone.

When evaluating database security in your environment to decide on your team’s top priorities, consider each of the following areas:

1. Administrative and network access controls
2. End user account/device security
3. Encryption
4. Database software security
5. Application/web server security
6. Backup security
7. Auditing

2. Physical security, access controls and accounts

Physical security: Whether your database server is on-premise or in a cloud data center, it must be located within a secure, climate-controlled environment. (If your database server is in a cloud data center, your cloud provider will take care of this for you.)

Administrative and network access controls: The practical minimum number of users should have access to the database, and their permissions should be restricted to the minimum levels necessary for them to do their jobs. Likewise, network access should be limited to the minimum level of permissions necessary.

End user account/device security: Always be aware of who is accessing the database and when and how the data is being used. Data monitoring solutions can alert you if data activities are unusual or appear risky. All user devices connecting to the network housing the database should be physically secure (in the hands of the right user only) and subject to security controls at all times.

3. Encryption, software and applications

Encryption: ALL data—including data in the database, and credential data—should be protected with best-in-class encryption while at rest and in transit. All encryption keys should be handled in accordance with best-practice guidelines.

Database software security: Always use the latest version of your database management software, and apply all patches as soon as they are issued.

Application/web server security: Any application or web server that interacts with the database can be a channel for attack and should be subject to ongoing security testing and best practice management.

4. Backup and Auditing

Backup security: All backups, copies, or images of the database must be subject to the same (or equally stringent) security controls as the database itself.

Auditing: Record all logins to the database server and operating system, and log all operations performed on sensitive data as well. Database security standard audits should be performed regularly.

https://www.ibm.com/cloud/learn/database-security

Test questions

en:

1. Describe threats to databases.

2. Describe purpose of inference control

3. What is SQL injection?

ru:

1. Опишите угрозы для баз данных.

2. Опишите цель контроля вывода

3. Что такое SQL-инъекция?