**What Is Database Security?**

Database security is a set of practices and technologies used to protect database management systems from malicious cyberattacks and unauthorized use. Database security is a complex task that combines several information security disciplines—application security, [data security](https://satoricyber.com/data-security/what-is-data-security-threats-controls-and-solutions/), and endpoint security.

The goal of database security is to protect against misuse, data corruption, and intrusion, not only of the data in the database, but of the data management system itself and applications that access the database. Another aspect of database security is protecting and hardening the physical or virtual server hosting the database, and the surrounding computing and network environment.

**In this article we cover the following database security best practices:**

1. [Separate Database Servers and Web Servers](https://satoricyber.com/database-security/top-10-database-security-best-practices/#separate-database-servers-and-web-servers)
2. [Database Encryption At Rest And In Transit](https://satoricyber.com/database-security/top-10-database-security-best-practices/#database-encryption-at-rest-and-in-transit)
3. [Use Strong Authentication](https://satoricyber.com/database-security/top-10-database-security-best-practices/#use-strong-authentication)
4. [Continuously Discover Sensitive Data](https://satoricyber.com/database-security/top-10-database-security-best-practices/#continuously-discover-sensitive-data)
5. [Separate Tests From Production](https://satoricyber.com/database-security/top-10-database-security-best-practices/#separate-tests-from-production)
6. [Revoke Privileges Continuously](https://satoricyber.com/database-security/top-10-database-security-best-practices/#revoke-privileges-continuously)
7. [Deploy Physical Database Security](https://satoricyber.com/database-security/top-10-database-security-best-practices/#deploy-physical-database-security)
8. [Ensure Database User Accounts are Secure](https://satoricyber.com/database-security/top-10-database-security-best-practices/#ensure-database-user-accounts-are-secure)
9. [Monitor Database Activity](https://satoricyber.com/database-security/top-10-database-security-best-practices/#monitor-database-activity)
10. [Conduct Security Tests](https://satoricyber.com/database-security/top-10-database-security-best-practices/#conduct-security-tests)

Download The Data Security Guide (PDF)

[](https://satoricyber.com/linkout/44797)

**1. Separate Database Servers and Web Servers**

Separate your web server from your database server to improve security by maintaining isolation and preventing lateral movement. With separate servers, attackers cannot access the database even if they hack your admin account on the web server.

Keep any non-critical servers or programs separate from your database server. These servers might need to communicate for specific tasks, but they are not necessary for operating the database. When you enable communication, ensure you limit the permissions to the minimum required for successful operations. The principle of least privilege helps restrict an attacker’s ability to damage your database.

**2. Database Encryption At Rest And In Transit**

Strong encryption is a basic best practice for database security. Encrypt all database connections using the Transport Layer Security (TLS) protocol, protecting data in transit. You should also encrypt any disks containing a data store to prevent data loss or theft.

Leverage column-level encryption to ensure the confidentiality of sensitive data fields.

**3. Use Strong Authentication**

Database authentication is the process of confirming that users or service accounts attempting to connect to the database are who they say they are. A related process is authorization, which determines, based on the confirmed identity, what permissions the account should have on the database.

Because databases are almost always mission critical systems, all databases should have strong authentication enabled. If possible, use two-factor authentication, for example by combining a password or PIN with something the user owns, such as a security token or mobile phone.

**4. Continuously Discover Sensitive Data**

Many databases contain a mix of sensitive and non-sensitive data. If this is the case, you need to continuously audit your data and identify which tables or columns in your database are sensitive and require special protection. If you don’t know where sensitive data lives, you cannot adequately protect it, and this can result in compliance violations and data breaches.

Some regulations and compliance standards, especially in industries like healthcare, financial services, and telecom, have specific data discovery requirements. Make sure you follow the compliance standards that affect your organization in general and specific database workloads.

***Related content: Read our guide to***[***data security management***](https://satoricyber.com/data-security/the-6-pillars-of-data-security-management/)

**5. Separate Tests From Production**

A common cause of data breaches is that sensitive production data is stored on a database in a testing or staging environment, which is not as well protected as the production environment. Ensure that:

* Test environments are physically separate from production environments.
* Test environments have separate roles and permissions than production environments, and developers should not get access to production environments unless absolutely necessary.
* Test environments never contain real production data. Instead, you should create synthetic or anonymized datasets to enable testing on realistic data.
* There should be a strict, controlled process for promoting a database from testing to production, ensuring that the new version does not introduce bugs or security issues.

**Learn more:**

* [**Blog: Data Security Projects Keep Data Teams Away From Their Core Responsibilities**](https://blog.satoricyber.com/data-security-projects-keep-data-teams-away-from-their-core-responsibilities/?l=l-middle&f=datasec-dbsec-bp)
* [**How Satori’s Data Security Platform Helps Scale Your Data Security Across Databases, Data Warehouses, and Data Lakes**](https://satoricyber.com/product/?l=l-middle&f=datasec-dbsec-bp)
* [**Set a demo meeting**](https://satoricyber.com/get-a-demo/?l=l-middle&f=datasec-dbsec-bp)

**6. Revoke Privileges Continuously**

Be vigilant about enforcing the principle of least privilege. Users should have access to a database only as long as they need it for their daily roles, and should only have the exact permissions they need to carry out their role. When a user no longer requires a permission, it must be revoked.

Privilege creep is a common problem in database systems, where additional privileges are granted as needed and are not revoked. A good way to manage excessive privileges is a privilege access management (PAM) system. These systems provide visibility of all permissions granted to sensitive systems, and can assign “just in time” privileges for individuals performing maintenance on a database, revoking them automatically when maintenance is complete.

**7. Deploy Physical Database Security**

Your data center or database server may be vulnerable to physical infiltration by threat actors (both outside and within your company). If infiltrators or malicious insiders can physically access your database server, they might exfiltrate or corrupt your data or install malware that grants them remote access.

Cyber-attacks that exploit physical security vulnerabilities may be difficult to prevent or detect with digital security controls alone. You should apply additional security measures to protect physical assets, including machines, storage facilities, or workspaces with access to sensitive data.

If you use a hosting service, ensure it has a good reputation and takes security seriously. Don’t use a free hosting service that may lack adequate security. If you host your servers, implement physical security measures and restrict physical access to essential personnel. Protect sensitive areas with locks, cameras, and security staff. Maintain a log of all access to restricted areas to enable the investigation and mitigation of a breach.

**8. Ensure Database User Accounts are Secure**

Restrict database access to the minimum number of users required. Only provide the administrative privileges required to complete a job, restricting access to the times your users need it. Comprehensive access management might not be practical for smaller organizations, but it is still important to manage permissions via roles or groups and not grant them directly to individual users.

If your organization is larger, consider using an automated access management solution. Access management software can generate temporary passwords with limited privileges, so authorized users must authenticate every time they access the database. This approach prevents password sharing and ensures that all sessions and activities are properly logged. Administrators might want to share passwords for the sake of convenience, but you should not allow this practice as it complicates accountability.

Secure user accounts using these standard procedures:

* Establish and enforce a strong password policy
* Encryption stored password hashes
* Enable automatic account locking after several login attempts
* Establish a policy for deactivating accounts when employees switch to different roles or leave the company.

**9. Monitor Database Activity**

Monitor all logins and login attempts to your database and operating system. Regularly review the logs to identify anomalous activity. You might set up an alerting system to notify relevant individuals or teams of suspicious activity.

Continuous monitoring allows you to identify compromised accounts quickly if an attacker breaches your databases or an employee performs a suspicious or negligent task. Monitoring also helps you identify the creation of unauthorized accounts (i.e., a hacker creating an account without your permission) or when users share accounts.

Use a [database activity monitoring](https://satoricyber.com/data-security/database-activity-monitoring-uses-features-how-to-choose/) (DAM) solution to provide independent monitoring and help keep track of admin activity. Keep database activity logs and conduct regular audits to provide records for investigation purposes.

**10. Conduct Security Tests**

Once you’ve implemented your security policy, you need to test its effectiveness regularly. Perform penetration testing to identify unsecured aspects of your database and conduct frequent vulnerability assessments. Security testing allows you to discover and fix issues quickly before they result in a breach.