12 STEPS TO PROTECTING SENSITIVE DATA IN STRUCTURED DATABASES GUIDEBOOK

Protecting sensitive data in structured databases is crucial for maintaining the confidentiality, integrity, and availability of information. Here are some of the best ways to safeguard sensitive data in structured databases:

1. **Data Encryption:** Implement encryption mechanisms such as Transparent Data Encryption (TDE) or column-level encryption to protect data at rest. This protection ensures that even if the database is compromised, the data remains encrypted and unreadable without the appropriate keys.
    - SQL Server provides Transparent Data Encryption (TDE) for encrypting the entire database, as well as options for encrypting specific columns using functions like ENCRYPTBYPASSPHRASE or ALWAYS ENCRYPTED feature.

2. **Access Control:** Utilize role-based access control (RBAC) and strong authentication mechanisms to limit access to sensitive data. Only authorized users should have access to specific data based on their roles and responsibilities.
    - Utilize Windows authentication or Active Directory integration for authentication and grant permissions to users and roles based on the principle of least privilege.

3. **Database Auditing and Logging:** Enable auditing and logging features to monitor database activity. Keep detailed logs of all database transactions, including who accessed the data, what changes were made, and when they occurred. This helps in identifying and investigating security incidents.
    - Enable SQL Server Audit to track and log database activities. Configure auditing for sensitive operations such as access to sensitive data, changes to permissions, and failed login attempts. Review audit logs regularly to detect and investigate suspicious activities.

4. **Data Masking and Anonymization:** Mask or anonymize sensitive data in non-production environments to reduce the risk of unauthorized access during development, testing, or training. Replace actual data with characters such as asterisks or randomized data values.
    - Leverage SQL Server's Dynamic data masking or tools such as Redgate's Data Masker and SQL Data Generator

5. **Parameterized Queries and Prepared Statements:** Use parameterized queries and prepared statements to prevent SQL injection attacks, which can be used to extract sensitive data from databases. This tactic helps mitigate the risk of unauthorized access through malicious inputs.
    - Use stored procedures to encapsulate database logic.
    - Validate user input against an allow-list (whitelist) of acceptable values. Reject any input that doesn't match the expected format or falls outside the allowed range.

6. **Database Firewalls:** Deploy database firewalls to monitor and filter incoming and outgoing database traffic. This helps detect and prevent unauthorized access, SQL injection attacks, and other malicious activities targeting the database.

Contact ODGA's Data Protection and Governance Team for assistance.

- Ideally, install your application on a server separate from your database.
- Configure Windows Firewall (or 3rd party vendor) to only allow SQL Server traffic (TCP 1433, 4022, 135, 1434, UDP 1434). Other ports may be necessary for other SQL Services.

7. **Regular Patching and Updates:** Keep the database management system (DBMS) and associated software up to date with the latest security patches and updates. This process helps address known vulnerabilities and reduce the risk of exploitation by attackers.
    - Work with your ISO to review the Tenable scan results to ensure patches are being applied appropriately.

8. **Data Loss Prevention (DLP):** Implement DLP solutions to monitor and control the movement of sensitive data within the database and prevent unauthorized data exfiltration. DLP tools can help detect and block attempts to access or transmit sensitive data.
    - Contact VITA if you're interested in a data loss prevention solution.

9. **Database Activity Monitoring (DAM):** Deploy DAM solutions to monitor real-time database activity and detect anomalous behavior indicative of security threats or policy violations. These tools enable proactive responses to potential security incidents.
    - Vendors you may want to consider are SolarWinds, Dynatrace, Datadog or Redgate.

10. **Regular Security Assessments:** Conduct regular security assessments and penetration testing of the database environment to identify vulnerabilities and weaknesses. Address any findings promptly to enhance the overall security posture of the database.
    - In addition to Tenable and Acunetix scans, you should run Microsoft Defender for SQL to identify security issues on your SQL Server. 3rd parties such as Mandiant or Verizon can conduct penetration tests.

11. **Data Backup and Recovery:** Implement regular data backups and ensure that backups are securely stored and encrypted. This measure helps mitigate the impact of data breaches or disasters by enabling data recovery from a known good state.
    - Make sure your backup schedule is current with VITA. Conduct a quarterly audit with VITA to ensure the appropriate databases on your SQL servers are being backed up and conduct periodic restore tests.

12. **Employee Training and Awareness:** Provide comprehensive training to database administrators and users on security best practices, data handling policies, and procedures for securely accessing and managing sensitive data.
    - Ensure your database administrations complete the Sec 527 role-based training for data custodians and "Secure Database Administration"

By implementing these best practices, organizations can strengthen the security of structured databases and better protect sensitive data from unauthorized access, data breaches, and other security threats.

Contact ODGA's Data Protection and Governance Team for assistance.