Database Credentials Coding Policy

Last Update Status: *Updated October 2022*

1. Overview

Database authentication credentials are a necessary part of authorizing application to connect to internal databases. However, incorrect use, storage and transmission of such credentials could lead to compromise of very sensitive assets and be a springboard to wider compromise within the organization.

2. Purpose

This policy states the requirements for securely storing and retrieving database usernames and passwords (i.e., database credentials) for use by a program that will access a database running on one of <Company Name>'s networks.

Software applications running on <Company Name>'s networks may require access to one of the many internal database servers. In order to access these databases, a program must authenticate to the database by presenting acceptable credentials. If the credentials are improperly stored, the credentials may be compromised leading to a compromise of the database.

3. Scope

This policy is directed at all system implementer and/or software engineers who may be coding applications that will access a production database server on the <Company Name> Network. This policy applies to all software (programs, modules, libraries or APIS that will access a <Company Name>, multi-user production database. It is recommended that similar requirements be in place for non-production servers and lap environments since they don't always use sanitized information.

4. Policy

4.1 General

4.1.1 In order to maintain the security of <Company Name>'s internal databases, access by software programs must be granted only after authentication with credentials. The credentials used for this authentication must not reside in the main, executing body of the program's source code in clear text or easily reversible encryption. Database credentials must not be stored in a location that can be accessed through a web server. Algorithms in use must meet the

standards defined for use in NIST publication FIPS 140-2 or any superseding document, according to date of implementation. The use of the RSA and Elliptic Curve Cryptography (ECC) algorithms is strongly recommended for asymmetric encryption.

4.2 Specific Requirements

4.2.1 Storage of Data Base Usernames and Passwords

- Database usernames and passwords may be stored in a file separate from the executing body of the program's code. This file must not be world readable or writeable.
- Database credentials may reside on the database server. In this case, a hash function number identifying the credentials may be stored in the executing body of the program's code.
- Database credentials may be stored as part of an authentication server (i.e., an entitlement directory), such as an LDAP server used for user authentication. Database authentication may occur on behalf of a program as part of the user authentication process at the authentication server. In this case, there is no need for programmatic use of database credentials.
- Database credentials may not reside in the documents tree of a web server.
- Passwords or pass phrases used to access a database must adhere to the Password Policy.

4.3 Retrieval of Database User Names and Passwords

4.3.1 If stored in a file that is not source code, then database user names and passwords must be read from the file immediately prior to use. Immediately following database authentication, the memory containing the user name and password must be released or cleared.

4.3.2 The scope into which you may store database credentials must be physically separated from the other areas of your code, e.g., the credentials must be in a separate source file. The file that contains the credentials must contain no other code but the credentials (i.e., the user name and password) and any functions, routines, or methods that will be used to access the credentials.

4.3.3 For languages that execute from source code, the credentials' source file must not reside in the same browsable or executable file directory tree in which the executing body of code resides.

4.4 Access to Database Usernames and Passwords

4.4.1 Every program or every collection of programs implementing a single business function must have unique database credentials. Sharing of credentials between programs is not allowed.
4.4.2 Database passwords used by programs are system-level passwords as defined by the Password Policy.
4.4.3 Developer groups must have a process in place to ensure that database passwords are controlled and changed in accordance with the Password Policy. This process must include a method for restricting knowledge of database passwords to a need-to-know basis.
4.4.4 Users and/or software accessing sensitive data must be subjected to proper access control and should not be able to perform privileged operations that are out of scope of said user and/or software.

4.5 Coding Techniques for Implementing this Policy

[Add references to your site-specific guidelines for the different coding languages such as Perl, JAVA, C and/or Cpro.]

5. Policy Compliance

5.1 Compliance Measurement

The Infosec team will verify compliance to this policy through various methods, including but not limited to, business tool reports, internal and external audits, and feedback to the policy owner.

5.2 Exceptions

Any exception to the policy must be approved by the Infosec team in advance.

5.3 Non-Compliance

An employee found to have violated this policy may be subject to disciplinary action, up to and including termination of employment.

A violation of this policy by a temporary worker, contractor or vendor may result in the termination of their contract or assignment with <Company Name>.

Any program code or application that is found to violate this policy must be remediated within a 90-day period

6. Related Standards, Policies and Processes

- Password Policy

7. Definitions and Terms

- Credentials
- Executing Body
- Hash Function
- LDAP
- Module

8. Revision History

Date of Change	Responsible	Summary of Change
June 2014	SANS Policy Team	Formatted into new template and made minor wording changes.
October 2022	SANS Policy Team	Converted to new format.