SQL

There are two main ways to assess the security of SQL Server. The first is a **manual assessment**, which is essentially a trusted look at configuration settings while sitting down, logged onto the box. This would literally be walking through a hardening-type checklist, like this, to compare best practices on how it's actually set up. It's similar to an audit. Settings to look for in such an audit would include:

- Windows security settings involving file permissions, network access, etc.
- Default SQL Server stored procedures that aren't needed
- SQL Server services account privileges (i.e., regular user privileges)
- Type of authentication used (i.e., Windows versus mixed mode)
- Audit logging settings
- Physical and logical placement of the server relative to your Web and/or application servers (i.e., at least behind a firewall if not in an isolated DMZ)

In addition, Microsoft has a solid checklist for SQL Server 2000 security and resources for securing SQL Server 2005.

The second way to assess SQL Server is to **run various security assessment tools** against the underlying Windows OS and the actual database system itself. There are tools dedicated to both areas and each is essential. Running software on a secure OS is a key component of information security. This is especially true given how critical most SQL Server-based systems are in most organizations.

You can perform "automated" tests in two different ways. The first approach is to observe it as an unauthenticated outsider that should not have access to the system. This provides a true hacker's-eye view of areas that can be penetrated and exploited without authorized access. The other way to use automated assessment tools would be as an authenticated user, one with legitimate Windows and/or SQL Server accounts). This way you observe it with a trusted insider's view of exploitable security vulnerabilities. Both methods for running your security tests are equally beneficial, and most security tools support both options.

Oracle

Oracle Password Cracker in PL/SQL

This simple tool implements a Oracle database password cracker in PL/SQL. the main driving force behind this is to encourage DBA's to test the strength of passwords in their databases mopre easily by being able to use PL/SQL scripts simply from SQL*Plus without the need to download binaries or libraries and more. The main goal is simplicity and to encourage people to strengthen their database security.

Tools for Managing System Security

GlassFish Server provides the following tools for managing system security:

Administration Console

The Administration Console is a browser-based utility used to configure security for the entire server. Tasks include managing certificates, users, groups, and realms, and performing other

system-wide security tasks. For a general introduction to the Administration Console.

The asadmin utility

The asadmin command-line utility performs many of the same tasks as the Administration Console. You might be able to do some things with the asadmin utility that you cannot do with the Administration Console. For a general introduction to asadmin, asadmin Utility.

The keytool utility

The keytool Java Platform, Standard Edition (Java SE) command-line utility is used for managing digital certificates and key pairs. For more information.

The policytool utility

The policytool J2SE graphical utility is used for managing system-wide Java security policies. As an administrator.