



redhat.

Red Hat Database

Open Source Database, Open Source Security

January 2002

Abstract

Database security is a critical component of an information technology (IT) security strategy. Red Hat Database, the commercial open-source database management system offering from Red Hat, has security features that enable it to perform as a secure database management system (DBMS). This paper discusses how Red Hat Database builds on open-standards based foundation technologies (such as Kerberos, SSL, and SSH) to provide this functionality.

Overview

Security of computer systems is a critical aspect of any information technology strategy. The security of database management systems is an important component therein. Beyond ensuring the security of the host operating system, a database administrator needs to ensure that the DBMS software itself is secure, and that it provides the full complement of features to enable this security.

Red Hat Database, the open-source database for Red Hat Linux, is powered by PostgreSQL. The full source code to the product is freely available, unlike with proprietary solutions.

Many enterprises are recognizing that open source technology offers clear benefits over similar proprietary software, especially when addressing security issues:

- **Fast Fixes**

Open source solutions feature fast bug fixes because programmers around the world have access to the program source code.

- **Freely Auditable Source Code**

Because the source code to open source software is freely available, any experienced programmer can audit it for security holes and problems. This almost guarantees the application is free of secret backdoors.

- **Ability to Disable / Enhance Features**

Access to the program source code means that an organization can easily produce custom product versions. So, should an organization determine that a particular feature presents a security risk, that feature can be easily disabled by removing it, hence guaranteeing it cannot be used. Also, custom enhancements can be made to the program source for interoperability with legacy operating systems and security systems.

Red Hat Database provides numerous security features that are built on such open standards as Kerberos, SSL, and SSH:

- **Authentication security**

Ensuring that only authorized users can connect to the database system.

- **Transmission security**

Ensuring that data is not vulnerable to snooping while in transit over the network.

- **User-level and group-level access control**

Ensuring that users and groups have access only to information to which they are entitled.

Used together, these features ensure that people can access only data for which they are authorized, and only authorized people can access the data.



Authentication Security

The first line of defense in ensuring a secure database is authentication security; that is, making sure that only people who should be able to connect to a database server can connect. Red Hat Database has support for several mechanisms for ensuring this requirement is enforced. These methods fall into two broad categories:

- **Token-based authentication**

Under token-based authentication systems, a user proves his or her identity by sending a token (such as a password) that the server associates with that user.

- **Connection-origin based authentication**

These are weaker authentication methods that automatically trust connections that originate from a particular user, as determined by some simple connection-origin based test (IP address, ident protocol, etc.). These methods are useful in a trusted environment, such as during database application development, but are not suitable for an enterprise production environment. They are not discussed further in this document.

Red Hat Database uses token-based methods to authenticate database users. These features are based around two core technologies: password authentication, using the typical UNIX password handling methods; and Kerberos, the industry-standard secure authentication system.

Password-based authentication is a familiar mechanism for most users of computer systems. The password-authentication method of Red Hat Database involves the client prompting the user interactively for a password, which is then, after being encrypted, transmitted to the server. This authentication scheme is one of the easiest to implement. Because Red Hat Database can read password files in the standard UNIX `/etc/passwd` format, administrators can choose to use the same password file across multiple systems—database or otherwise. Also, coupled with one of the transmission security technologies discussed later (in order to prevent password hash replay attacks), password authentication is an effective mechanism for end-to-end access control for a Red Hat Database system.

Red Hat Database can also be configured as a Kerberos service. This allows Red Hat Database to be integrated into an overall, open-source, enterprise security system that is built on technologies such as Linux PAM (Pluggable Authentication Modules) and Kerberos. As many proprietary solutions, such as Microsoft Active Directory, are also built with Kerberos, Kerberos-enabled solutions such as Red Hat Database can also serve well in mixed system environments. The Further Reading section has more information about Kerberos.

Transmission Security

Information in transit across data networks can be vulnerable to snooping. Transmission security techniques, usually involving end-to-end encryption, ensure that data is not vulnerable to interception while in transit. Red Hat Database can easily be deployed to use two common transmission security techniques: Secure Sockets Layer (SSL) and Secure Shell (SSH) tunnels.

Red Hat Database has native support for connections over SSL. This is the same open protocol that is used to protect World Wide Web transactions, such as on most e-commerce sites. Not only does it provide end-to-end encryption, SSL also enables clients to verify a server's identity, by means of a security certificate purchased from one of several commercial certificate authorities. This proof of identity ensures that a malicious party cannot create a rogue server that impersonates a database server. SSL provides a reliable, accepted mechanism whereby transmissions can be protected from snooping in transit.

Red Hat Database connections can also be tunneled over SSH using SSH software, such as the freely available OpenSSH implementation. This is helpful if you already have deployed SSH in your organization, as it does not require any additional set-up. SSH provides end-to-end encryption, as well as a server identification, albeit in a weaker form than that of SSL. However, SSH does provide excellent security against data being snooped in transmission. Additionally, it does not require the purchase of a security certificate, and thus is a potentially more cost-effective solution for securing network transmissions to Red Hat Database servers.

Using the transmission security mechanism provided for Red Hat Database, you can prevent the interception of important data while in transit, thus ensuring the security of data stored on a Red Hat Database server.



User-level and Group-level Access Controls

Connection security and transmission security serve to ensure that only authorized users may connect to a database server. However, often finer grained security is required. For example, a manager from engineering might be authorized to view information about his department's budget, but should not be able to access information about the salaries of other managers. This is accomplished through user-level and group-level access-control lists.

Red Hat Database provides user-level and group-level access control lists (ACLs). These ACLs allow access to tables and databases to be granted or revoked from users. In order to facilitate administration, users may also be made members of one or more groups. For example, all employees working in accounting may be added to a group accounting. Then, the accounting group can be granted access to accounting-related tables and databases, and all members of the group will have that access.

In addition, by exploiting the SQL Views feature present in Red Hat Database, access to selected fields of a database table can be restricted or granted. For example, consider the case of a retail salesperson accessing a product database. Perhaps it is desirable that the salespeople know only the wholesale price of a product, but not the actual cost to the company. In this case, a view on the product list table could be constructed that includes only the information that the salespeople should be able to view (product number, description, retail price, etc.). Then, using the ACL feature, the salespeople group can be granted access to this view, rather than to the product list table. By combining Red Hat Database features as in this example, most access restriction scenarios can be accommodated.

Conclusions

Red Hat Database offers a strong set of features for enabling secure database systems. These features encompass the three key security areas: authentication security, transmission security, and access control. By combining these features with a good security strategy, and by making use of updates and information available via Red Hat Network, you can make your Red Hat Database data environment secure.



Further Reading

Red Hat Database

1. *Red Hat Database Getting Started/Installation Guide*
(available from redhat.com)
2. *Red Hat Database Administrator and User's Guide*
(available from redhat.com)
3. *Red Hat Database*
<http://www.redhat.com/software/database>
4. *Red Hat Database Project*
<http://sources.redhat.com/rhdb/>

PostgreSQL

PostgreSQL Global Development Group
<http://www.postgresql.org>

Kerberos

Kerberos FAQ
<http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html>

SSH/SSL

1. *OpenSSH Project*
<http://www.openssh.com>
2. *SSH: The Secure Shell: The Definitive Guide*
Daniel J. Barrett and Richard E. Silverman
O'Reilly, 2001