

### Comply with:

- PCI DSS, PCI Guidelines for Virtualization
- HIPAA, HITEC Act
- FISMA (FedRAMP)
- FDA

### Protect:

- MySQL Data
- PostgreSQL Data
- Configuration Files
- Backup files, exports, archives
- Transaction Logs
- Password Files

### Supported Platforms:

- Ubuntu, Debian
- Centos, Fedora, Redhat
- Oracle Enterprise Linux
- SuSE, Scientific Linux
- CloudLinux, Slackwear
- Rackspace Linux Xen Kernel

### System Requirements:

- Linux kernel 2.6.19 and higher\*
- Supported Linux Systems: CentOS, Red Hat, Ubuntu, Debian, Fedora & openSUSE.
- MySQL Server 4.x or 5.x
- Ecryptfs module (bundled with supported kernels).
- Keyutils
- Ecryptfs-utils
- \*Red Hat & CentOS can use 2.6.18-92 or higher.

### Out-of-the-box Data Encryption for MySQL & PostgreSQL

ezNcrypt for Databases is an out-of-the box application that will enable you to protect, encrypt and provide key management for your MySQL and PostgreSQL databases. It leverages transparent data encryption (TDE) features, traditionally reserved for only expensive and difficult-to-implement commercial databases. ezNcrypt for Databases delivers enterprise-class data security.

### ezNcrypt for Databases:

- Is an “out-of-the-box” method for protecting data – requiring **no expensive and complex changes to databases and applications**
- Enables you to encrypt data by using **AES, Twofish, and Blowfish encryption algorithms**
- Protects data at **the operating system level**
- Allows sensitive MySQL and PostgreSQL data to be **encrypted within the data files** to prevent access from the operating system
- Comes with **very low overhead** (typically less than 1%)
- Ensures little to **no performance impact** to databases and applications
- Includes **index support**
- Is built around **centralized key management**
- Provides separation of duties, making it **compliance ready**
- **Installs quickly** and is completely transparent to applications
- **Protects log files, configuration files, backups** and other database files and output
- **Realize ROI in months** – not years

### Features:

- **Transparent, Rapid Implementation** – Encrypt databases and files “in place” and avoids the need to re-architect databases, files, or storage networks. By running above the file system as a logical volume, ezNcrypt is transparent to users, applications, databases, and storage subsystems.
  - No coding
  - No modifications to applications
  - No database schema changes
- **Data Encryption** – Transparently encrypt, decrypt and access MySQL and PostgreSQL data in real-time (including data at rest).
- **High Performance** – Experience little-to-no database performance impact.
- **Centralized Management** – Manage with little administrative overhead. ezNcrypt combined with KSS key manager gives administrators a secure, easy-to-administer method for deploying and managing encryption keys.
- **Auditing** – Audit and report on access requests to protected files. The audit logs created by ezNcrypt are easy to integrate with various security and event management systems, providing added regulatory compliance for the protection of confidential data.

For additional information regarding ezNcrypt for Databases, please contact Gazzang at 281.523.5601 or email [sales@gazzang.com](mailto:sales@gazzang.com).

### Features (continued):

- **Scalability** – Scale ezNcrypt in large enterprise or complex cloud environments across large numbers of servers and files.
- **Transparent Data Encryption** – Perform real-time, page-level I/O encryption and decryption of data, configuration, transaction, log and other sensitive files.
- **Key Management** – The encryption key is NEVER stored on the same server with the data that is to be encrypted. It is securely stored in memory in Gazzang’s key storage system (KSS) for high availability.
- **Symmetric Key** – the key used is a symmetric key, secured by certificates, fingerprints, one-time passwords and other advanced technics – ensuring secure and controlled access.

### Key Management Workflow

#### Is the Linux exe trusted?

- Verify name
- Verify owner
- Verify location
- Process identifiers/fingerprints

#### If “OK,” then:

- Provide key
- Get transparent R/W access

#### Where is this Linux exe allowed or denied access to files/dir?

- Limited files or directories

#### If “OK,” then:

- Transparently uses key for R/W, etc.

#### If “Not OK,” then:

- Access is denied

