



SUPPORTING HIPAA COMPLIANCE WITH MICROSOFT SQL SERVER 2008

RISK ADVISORY • TAX • FINANCE & ACCOUNTING

Dave Elliott,
CIPP/G/C, CISSP, CISA
Information Security
Center of Expertise

Chip Zodrow
Risk Advisory Services

Paul Rozek,
CGEIT
Information Security
Center of Expertise

Jefferson Wells delivers professional services in the areas of risk advisory, tax, and finance and accounting. We serve clients, including Fortune 500 and Global 1000 companies, through highly experienced professionals working from offices worldwide.

To learn more about our firm and our professional services, contact your local Business Development Manager or visit our Web site at www.jeffersonwells.com.

EXECUTIVE SUMMARY

Legislation relative to many types of industry compliance requirements continues to evolve, expanding in scope, impact and implications world-wide. Healthcare organizations in particular must be diligent sentinels to avoid compliance infractions, data loss, and security breaches that may lead to unnecessary costs, penalties and loss of consumer trust. The need for effective industry solutions to address growing business and information technology (IT) compliance risks has never been greater. Organizations must evaluate and implement appropriate and effective technologies to meet both current and future compliance and information security requirements.

The Health Information Portability and Accountability Act (HIPAA) of 1996 is recognized as one of the widest-reaching compliance regulations in the healthcare industry, governing health information privacy, security, organizational identifiers, and overall administrative practices. Its main goals include ensuring protected health information (PHI) is:

- Controlled and backed-up when stored and/or processed on computer systems and databases
- Encrypted while in transit on unsecured networks

TABLE OF CONTENTS

Executive Summary	1
HIPAA Background.	2
Microsoft SQL Server 2008.	2
HIPAA Technical Safeguards and SQL Server 2008	3
SQL Server 2008 Features.	4
SQL Server 2008	
Access Control	4
• Windows and SQL Security Principals and Logins	4
• Windows and SQL Authentication	5
• Windows and SQL Authorization.	5
• Emergency Access.	6
Data Integrity and Encryption.	7
• Transparent Data Encryption.	7
• Extensible Key Management.	8
Communications Security	8
Audit and Compliance	8
• SQL Server 2008 Audit	8
• Security and Storage of Audit Logs	10
• Policy-based Management.	11
Conclusion.	11

Continued on Page 2

- Only accessible by personnel with job responsibilities to access the PHI
- Monitored for both authorized and unauthorized access

Due to lack of enforcement in the “early years”, many healthcare organizations took a wait-and-see attitude when evaluating and remediating their HIPAA compliance problems. That attitude needs to be replaced by a proactive approach and the selection of appropriate technologies to achieve compliance. In recent months, the U.S. Department of Health and Human Services (HHS) issued an interim final rule to strengthen compliance enforcement for violations of the HIPAA rules and to encourage prompt corrective actions by healthcare providers. These changes are intended to increase consumer confidence that the privacy and security of their sensitive health information is being maintained.

In this white paper we provide guidance on specific Microsoft SQL Server 2008 features, and how they may be implemented to support the goals and technical safeguard requirements of HIPAA. The selection and implementation of this product’s features should be considered within the context of an organization’s current safeguards and overall compliance and risk mitigation strategies.

HIPAA BACKGROUND

HIPAA is a set of standards introduced by the U.S. Congress in 1996. The Act consists of rules governing PHI, including Security, Privacy, Identifiers, and Transactions and Code Sets. The purpose of the HIPAA Security Rule is to promote the protection and privacy of sensitive PHI used within the healthcare industry by organizations called “covered entities.” As a result of the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, both covered entities and business associates are now accountable to the HHS and individuals for appropriately safeguarding private patient information. For the purposes of this white paper, references to the term “covered entity” will also apply to business associates when discussing methods of, and requirements for, complying with the HIPAA Security Rule.

The HIPAA Security Rule itself has five major sections, created to identify relevant security safeguards that help achieve compliance: 1) Physical; 2) Administrative; 3) Technical; 4) Organizational Requirements; and 5) Policies, Procedures and Documentation Requirements.

This white paper focuses exclusively on the *technical* safeguard requirements. Organizations must implement reasonable and appropriate solutions and management policies and procedures to comply with HIPAA technical standards and implementation specifications. It is critical to perform a formal security risk assessment for each of the safeguards in the HIPAA Security Rule. Management’s decisions related to risk aversion and tolerance must be documented in the security risk assessment to identify potential compliance gaps. For many organizations, it is difficult to determine how the Rule applies and what type of supporting technologies and processes are needed to achieve compliance. As technology solutions are not specifically defined or named in the Rule, organizations have to choose from a myriad of vendors and solutions, as appropriate, for their environments. Compliance efforts must include formal evaluations of vendors and their products to see how well they support the administrative, operational and management controls – including technical controls in network servers, databases and applications.

As part of the American Recovery and Reinvestment Act signed into law on February 17, 2009, the HITECH Act increased penalties for HIPAA violations based on a tiered structure, depending on the severity of each violation. Patients and other parties must be notified of security breaches to PHI. This means covered entities must be able to effectively monitor and retain security event logs on systems containing or using PHI. Policies and procedures should be in place to regularly monitor and review security event logs to ensure activities affecting PHI are authorized and appropriate. Such activities may include, but are not limited to, account logons and logoffs, file accesses, changes and potential security incidents.

Microsoft SQL Server 2008 provides robust technical security auditing features and functionality that can be implemented to address HIPAA technical safeguard requirements and facilitate effective and timely monitoring of security event logs. SQL Server 2008 features are most effective when used in conjunction with multiple layers of security defenses. Selection and implementation of a product such as SQL Server 2008 should be in accordance with an organization’s formal security risk assessment to ensure optimal compliance value.

MICROSOFT SQL SERVER 2008

Organizations concerned with HIPAA compliance can take advantage of features introduced in SQL Server 2008 to meet these compliance requirements. Features of particular interest to companies storing, processing,

or transmitting electronic protected health information (ePHI) include:

- Database auditing capabilities with SQL Server Audit
- Transparent Database Encryption (TDE)
- Extensible Key Management (EKM)
- Policy-based Management
- Reporting Services
- Performance data collection
- Fewer services and potential vulnerabilities enabled at installation
- Improved control over privileged access and separation of duties
- Granular access control

HIPAA TECHNICAL SAFEGUARDS AND SQL SERVER 2008

The HIPAA Security Rule identifies technical safeguards as the policies, procedures and technologies used to protect and control access to ePHI. The Rule does not define specific technologies or vendor solutions for the implementation of these safeguards. Covered entities have flexibility regarding safeguard selection and implementation, described as a “flexibility of approach.” The appropriateness of specific technical safeguards for a given environment must be determined through a formal risk analysis and management process, as defined in Section 164.308(a) (1) of the Rule, which should determine how and when specific technical safeguards are implemented. Vendors that understand the compliance landscape and real-world problems their customers face, can facilitate and accelerate product evaluation and selection by proactively addressing

TABLE 1: HIPAA TECHNICAL SAFEGUARDS

The following table associates available SQL Server 2008 security features to the defined technical safeguard standards and implementation specifications.

Technical Safeguard HIPAA 45 CFR §164.312	MS SQL Server 2008 Feature
Access Management	
Access Control Standard - §164.312(a)(1)	Windows/Active Directory & SQL Server Authentication and Authorization
Unique User Identification Specification - Required §164.312(a)(2)(i)	Windows/Active Directory & SQL Server Authentication
Emergency Access Procedure Specification - Required §164.312(a)(2)(ii)	Emergency Access Policy, Procedures and Pre-Staged Accounts; SQL Server Audit
Automatic Logoff Specification - Addressable § 164.312(a)(2)(iii)	Windows/Active Directory Group Policy Enforcement
Encryption and Decryption Specification - Addressable § 164.312(a)(2)(iv)	TDE); EKM; Cell-level Encryption
Person or Entity Authentication Standard - § 164.312(d)	Windows and SQL Server Authentication; Domain Password Policy Enforcement; Transport Layer Security (TLS\SSL)
Audit and Compliance	
Audit Controls Standard - § 164.312(b)	SQL Server Audit; Policy-Based Management
Data Integrity	
Integrity Standard - § 164.312(c)(1)	Database and Application Development Standards and Guidelines; Constraints, Triggers and Referential Integrity; Database Performance Collection; Database Backups
Mechanism to authenticate ePHI Specification - Addressable § 164.312(c)(2)	Transport Layer Security (TLS\SSL)
Secure Communications	
Transmission Security Standard - § 164.312(e)(1)	Transport Layer Security (TLS\SSL)
Integrity Controls Specification - Addressable § 164.312(e)(2)(i)	
Encryption Specification - Addressable § 164.312(e)(2)(ii)	

these requirements and clearly communicating relevant product features and capabilities.

Under the HIPAA Security Rule, implementation of standards is required, and implementation specifications are categorized as either “required” or “addressable.” For required specifications, covered entities must implement the specifications as defined in the Security Rule. For addressable specifications, a covered entity must assess whether the implementation of the specification is reasonable and appropriate for its environment and the extent to which it is appropriate to protect ePHI. Following that assessment, the covered entity must either implement the addressable specification, or document why it would not be reasonable and appropriate to implement and identify alternative and/or compensating safeguards as reasonable and appropriate.

SQL SERVER 2008 FEATURES

As indicated in Table 1 on Page 3, SQL Server 2008 provides a set of robust features that can be leveraged to support compliance with the HIPAA Security Rule Technical Safeguard requirements.

The following sections provide an overview of these features, the associated technical safeguards and specific implementation guidance.

SQL SERVER 2008 ACCESS CONTROL

SQL Server 2008 provides a strong and integrated set of features to effectively manage user and entity access to ePHI. These features include integration with Microsoft Windows Active Directory domain authentication and authorization services, expanded Kerberos protocol support, pre-defined database roles, encryption of authentication credentials over the network, enforcement of domain password policy (on Windows 2003 servers and later), and more granular access rules governing security principals, actions and objects.

A key aspect of any access control strategy is the principle of least privilege, whereby individuals are granted only the access required to fulfill their job responsibilities. This principle should be enforced for all users and entities, including support personnel, system and service accounts and others that access ePHI maintained by a covered entity or business associate. Access rules and privileges should be based on the protection of ePHI confidentiality (read), integrity (modify/delete) and availability (delete/service

management), and recognize that information and underlying services can be compromised through the actions of both authorized individuals and unauthorized individuals with malicious intent.

Securing and limiting access to ePHI is a key objective of the HIPAA Security Rule. In general, access control requires covered entities to develop and implement policies, procedures and technical solutions for systems that maintain ePHI to restrict access to authorized persons or entities. To meet this objective, covered entities must implement appropriate safeguards to ensure only authorized individuals can read, modify or delete sensitive patient health information. Regardless of the technology or information system in use, access should be managed in accordance with rules developed and implemented as part of the Information Access Management Standard [Administrative Safeguards (§ 164.308(a)(4))], which should address core access principles, including least privilege, separation of duties and accountability. Specific access control technologies are not defined in the Rule, and compliance can be achieved through a combination of methods and technologies depending on the organization’s risk assessment and mitigation strategies.

Windows and SQL Security Principals and Logins

Access to SQL Server 2008 resources is granted to security principals (users, groups, roles and logins). By configuring and managing SQL Server to rely primarily on Windows/AD authenticated security principals for user access management, organizations can more easily and consistently enforce unique user account naming standards at the enterprise level and meet the required specification for *unique user identifiers*. Regardless of the type of security principal in use, formal policies and procedures should be followed in naming and assigning security principals to ensure user accountability and to support the audit of actions performed against ePHI.

The requirement for unique user identifiers should also be enforced for all SQL authenticated security principals. Since the unique user identifier requirement is aimed at ensuring accountability for actions performed against ePHI, the creation and use of shared accounts should be minimized, with formal shared account password checkout/change procedures, activity logging, audit and review when they are used. This is particularly important for shared administrative accounts that may have broad access to ePHI.

Unique user identification provides a way to monitor individual user activities, and a mechanism to hold those individuals accountable for actions and activities that impact ePHI confidentiality, integrity and availability. Consider the following when administering SQL Server security principals for individuals and entities:

- Access management policies and procedures should ensure consistent account and security principal naming conventions across systems
- Use Windows domain authenticated logins and groups for authorization and access management
- Minimize and strictly control the use of shared administration and privileged system accounts. Where shared accounts are required, additional safeguards should be considered to monitor activities performed, including activity logging and formal password management, change, and checkout procedures for all shared accounts

Windows and SQL Authentication

To ensure individuals and entities are appropriately and securely authenticated when accessing or transferring ePHI, the *Person and Entity Authentication Standard* requires covered entities to validate a person or entity requesting access to ePHI is who they claim to be. The protection of ePHI confidentiality and integrity requires secure authentication whenever ePHI is accessed or transferred. SQL Server 2008 supports both Windows and SQL authentication; however, it is recommended that Windows authentication be used wherever possible to support unique and auditable user identifiers and strong authentication policies throughout the enterprise. The use of Windows authentication simplifies administration and allows Kerberos domain authentication to be leveraged. When authenticating entities that do not reside within the same trusted network, enabling Transport Layer Security (TLS\SSL) transmission is essential to protect session security.

For local authentication, SQL Server 2008 can be configured to support domain password policy enforcement for SQL logins; password complexity is always enforced for application roles and encryption key passwords. Although certificates used to encrypt the SQL authentication channel may be self-signed, it is recommended that certificates generated by a mutually trusted certificate authority be used to establish more secure SQL authentication channels.

While the management of Windows domain accounts is outside the scope of this discussion, there are certain

factors that should be considered when implementing and configuring SQL Server 2008:

- Access management policies and procedures should ensure consistent account and security principal naming conventions across systems, and user security principal names should not be reused
- Servers should be configured to use Windows Authentication mode unless there is a specific requirement (e.g., to support non-Windows client authentication)
- Access management processes for both individuals and entities should be established and consistently performed to centrally provision and periodically review all access to servers and databases hosting ePHI
- The use of shared administrative, system and service accounts should be minimized and strictly controlled. Where shared accounts are required, additional safeguards should be considered to monitor activities performed, including detailed activity logging and formal shared account management procedures
- Strong password policies should be enforced through domain group policy, including password length, complexity, history, expiry and account lockout settings
- Password protected screensaver settings should be enforced through domain group policy to enforce authentication confirmation when workstations are left unattended to support *Automatic Logoff* requirements

If using SQL Server Authentication:

- Disable or rename the system administrator or “sa” account
- Configure SQL servers to enforce domain password policies for all logins
- Use certificates generated by a mutually trusted certificate authority rather than self-signed certificates for TLS\SSL, especially when communicating over unprotected networks

Windows and SQL Authorization

Access to SQL Server 2008 can be accomplished using both Windows and SQL Server security principals. To effectively manage user access to SQL Server resources and to support the least-privilege objectives of the *Access Control Standard*, organizations should grant access using domain group membership and/or user-defined roles

wherever possible. In addressing Access Control, covered entities must develop access management policies, procedures and rules appropriate to the risks in their environments. In all cases, companies should:

- Ensure all access for administrators, operations personnel and database users is granted based on least-privilege and directly related to the users' job responsibilities
- Assign least-privilege rights consistently at both the server and database level

When managing access using domain and local groups:

- Use domain groups rather than local groups for rights management, wherever possible
- Create and assign unique, local Windows logins for each SQL Server service. Verify the login has the lowest possible rights assigned
- Confirm the "guest" account has been disabled by revoking the CONNECT privilege for all databases except the MASTER, msdb and tempdb databases

When administering server and database roles:

Fixed server roles

- Limit the assignment of the system administrator role (i.e., sysadmin), and explicitly assign server roles to accounts only as required by job responsibilities and account purposes
- Do not assign sysadmin to local Windows administrators, domain administrators, or database owners
- Verify separation of duties exists between database sysadmin users and server administrators
- Closely provision and manage "fixed server roles" that can impact ePHI confidentiality, integrity and server operations. Such roles include sysadmin, serveradmin, securityadmin, processadmin, setupadmin, bulkadmin, diskadmin, dbcreator and public roles

Database roles

- Do not use fixed database roles
- For users performing specific database operations, define and assign user-defined database roles, taking into account the potential effect on ePHI confidentiality (read), integrity (write/modify) and availability (system/service, access and deletion)
- Grant access using password protected application roles with consideration of the potential impact on ePHI confidentiality (read), integrity (write/modify) and availability (system/service, access and deletion)

- Assign users to custom defined roles that restrict access based on least-privilege and established access control rules

Other authorization considerations:

- Users and sysadmins should not be granted direct access to data, backup, or log files
- Users should not be given OS-level access to servers; this includes restricting both local Windows and Remote Desktop Protocol access
- User and application logins should not be assigned grantable privileges

Emergency Access

Commonly referred to as "break the glass" or "firecall" solutions, *Emergency Access Procedures* should be designed to provide timely access to ePHI in emergency situations where individuals may have a need to access ePHI. Covered entities are required to make provisions to grant ePHI access to appropriate personnel in the case of an emergency. SQL Server 2008 can support these scenarios through pre-staged account profiles and logging of emergency account activities. Emergency access procedures may consist of a set of instructions and practices to be followed in emergency situations and pre-defined emergency profiles to be used. Access to, and use of, the procedures should be restricted to actual emergency situations, and should include appropriate logging and audit of user actions when such procedures are activated. It is the responsibility of each organization to determine likely emergency scenarios and the appropriate access procedures for each scenario. Key to this process is determining who is likely to require access to ePHI in an emergency, the scope of access and what procedures should be developed for enabling, communicating and disabling emergency access. When establishing these procedures, covered entities should:

- Pre-stage emergency accounts using consistent domain and local server principal naming conventions to facilitate administration, activity logging and discovery
- Configure granular auditing on emergency accounts to capture ePHI activity logs
- Formally review the purpose of every use of emergency accounts. The procedures should be used only in the case of a validated emergency. Frequent use of these procedures may indicate broader issues with the access management processes
- Ensure "Emergency Access Procedures" are part of the HIPAA training programs and are appropriately documented and approved

DATA INTEGRITY AND ENCRYPTION

Maintaining the *integrity* of ePHI is a key focus of the Security Rule, as corrupt or inaccurate health information can directly affect patient life and health. Covered entities must develop and implement policies and procedures to protect ePHI from unauthorized alteration or destruction. There are many ways data can become corrupt or inaccurate, including errors during data entry, failure of storage devices, programming and communication errors and failure of storage devices. Malicious activities and malware can also have a negative effect on data integrity.

As with previous versions of SQL Server, the following should be considered to ensure the integrity of ePHI:

- Verify application databases are designed and built to enforce referential integrity. This includes the use of primary or foreign keys, unique indexes, check constraints and triggers
- Verify applications are written with appropriate and adequate application controls, including input and output validation
- Verify the use of secure development methodologies within all application and database logic
- Verify database configuration and performance monitoring are enabled and examined
- Monitor the database encryption flags

While server and operating system security is outside the scope of this discussion, servers should be built to implement minimum secure configurations, including but not limited to:

- Only minimum required services are enabled
- Strong access controls exist and enforce the concept of least privilege
- Patches and hot fixes are applied in a timely manner
- Backups of the server and data are regularly performed and tested
- Database backups are encrypted

Transparent Data Encryption

With the introduction of TDE in SQL Server 2008, a full range of data encryption solutions is available for data at rest. This feature can provide an effective solution to support increasing requirements to protect ePHI through encryption of data at rest. In addition to the previously available drive, file

and granular cell-level encryption, data can now be encrypted at the database level. TDE does not replace other encryption functions, and in some instances, a combination of encryption methods may be appropriate.

TDE encrypts both data and logs as the records are written to SQL database files (*.mdf) in real-time, including backups, snapshots and transaction logs. It does not protect data in use, such as when the operating system pages data to memory, or when paged memory is stored to disk as part of memory management. When data is in use, pages are read and decrypted and exist in clear text within buffer pool memory. To protect data in transit, TLS\SSL should be enabled to protect communications between the database server and any clients.

When implementing TDE, consider the following:

- Because of the write-once design of transaction logs, TDE does not encrypt transaction entries that have already been written to disk. Due to this design, not all data written to logs may be encrypted even after TDE has been enabled. Data written before a change in an encryption key will remain encrypted with the previous key
- Back-up all past and present certificates used to protect the Database Encryption Key (DEK), which must be present in the Masterdb for encrypted files to be restored or reloaded
- Leverage EKM to protect DEKs using Hardware Security Modules (HSM)

The use of *encryption* and *decryption* to protect ePHI at rest is an addressable implementation specification, and covered entities have flexibility with respect to when and how to implement encryption solutions. Each covered entity must assess when data encryption is reasonable and appropriate for its environment through risk assessment. When considering the type of encryption to implement, consider database design, database operations and application constraints.

Cell or column-level encryption was introduced with SQL Server 2005. Although this solution provides a more granular level of encryption than TDE and may be appropriate in some circumstances, the following issues should be considered:

- Cell-level encryption introduces the potential for significant performance issues and limits indexing and searching capabilities
- Applications need to be modified to support cell-level encryption

Extensible Key Management

EKM is a feature introduced with SQL Server 2008. EKM allows the integration of external cryptographic providers and external management of portions of the encryption key hierarchy. This feature can support the external management of keys used to protect TDE database encryption and cell-level encryption keys, including keys used to directly encrypt cell-level data. When enabled, EKM can provide a common interface to third-party key management and HSM to encrypt the keys used for data encryption and to directly encrypt the data itself. Once registered with EKM, these modules can be used by SQL Server to leverage the extended functionality provided by the HSM. These solutions work seamlessly with SQL Server 2008 databases and support enterprisewide, dedicated key management. This allows the key management function to be performed by a dedicated key management system instead of SQL Server. When implementing EKM, remember to:

- Store all keys separately from the data (SQL Server 2008 supports the use of HSMs to provide the physical separation of keys from data)
- Configure SQL Server Audit to monitor access to encryption keys

COMMUNICATIONS SECURITY

As with previous releases, SQL Server 2008 supports TLS\SSL, which provides server validation to clients and entities requesting session encryption and secures authentication sessions. This helps protect against server identity spoofing that could affect ePHI confidentiality. Through mutual authentication and session encryption, these safeguards can help validate the authenticity and integrity of transmitted ePHI.

To ensure the confidentiality and integrity of ePHI in transit, covered entities must determine the risks associated with methods used to establish communications between systems hosting ePHI. These risks and requirements should be assessed for all client, vendor and trading partner communications. Where there is a significant risk of unauthorized ePHI disclosure or unauthorized modification, appropriate session encryption and integrity safeguards should be implemented. As encryption and integrity safeguards are addressable implementation specifications, covered entities have flexibility in choosing when to implement communication encryption, the type(s) of encryption to use and other appropriate session parameters.

Various methods can be used to achieve these objectives, including the implementation of appropriate network protocols and data and message authentication codes.

When implementing TLS\SSL to provide session confidentiality and integrity, the following should be considered:

- To ensure the confidentiality of both local authentication and ePHI in transit, SQL Server should be configured to refuse connections for clients that do not support session encryption. To accomplish this, ensure the Database engine is configured to “Force Encryption” in the SQL Server Network Configuration
- Use certificates generated by a mutually trusted certificate authority rather than server-generated certificates, especially when communicating over unprotected networks

AUDIT AND COMPLIANCE

SQL Server 2008 introduces additional functionality to support audit and compliance objectives, including new SQL Server Audit and Policy-based Management solutions. Appropriately managed, SQL Server Audit can help support HIPAA requirements for Audit Controls through granular event configuration and improved management, security and centralization of audit records. Policy-based Management can assist in defining, monitoring and enforcing organizational security policies through a cohesive management interface, allowing organizations to configure, monitor, and in many cases, enforce policy compliance across multiple server instances.

The specific events and activities to be monitored will vary by organization and environment, and should be determined through the assessment of risk to ePHI confidentiality, integrity and availability.

SQL Server 2008 Audit

SQL Server Audit provides a mechanism to configure and manage database audit functions, and can be used to replace the audit functionality previously provided by SQL Trace. This audit solution facilitates ease of management, integration with System Center Operations Manager and collections services, and more detailed configuration and logging of events to support investigation and discovery. With SQL Server Audit, audit data can be written to specified files, Windows Application and/or Windows Security logs, providing support for various audit management approaches. The ability to write to the Windows Security log provides added protection of audit record integrity and secure integration with System Center Operations Manager through its Audit Collection Services. The audit function can be configured to provide extremely granular event logging by action, database object and/or security principal. This level of granularity allows covered

entities to apply audit rules focusing on specific security principals, actions and objects and combinations thereof, based on perceived risks to ePHI confidentiality, integrity and availability. Changes to the audit configuration itself are also written to the audit log, supporting policy compliance monitoring and enforcement.

At a minimum, covered entities should consider logging and monitoring the following:

System access

- Failed and successful logins
- Access and permission management activities
- Creation, modification and deletion of security principals

System and configuration activities

- System configuration and policy changes
- Database configuration and administration activity
- Service management (shutdowns and restarts)
- Audit trail access and changes to audit specifications

Access to ePHI

- Activities performed using emergency access accounts and procedures
- Select access to ePHI
- Activities that modify ePHI and other sensitive data

Activity monitoring

- Capture and monitor sysadmin and db_owner activities
- Enable and use the SQL Server Performance Data Collector service to capture and report on database and server health
- Verify audit reports are periodically reviewed, and issues are addressed

Event logging

To effectively audit database actions relevant to HIPAA, the audit groups in Table 2 below should be defined for all users accessing, and all servers hosting, ePHI. The audit of specific database-level actions should be

TABLE 2: SQL SERVER AUDIT GROUPS

Area	Audit group
Authorization Changes	SERVER_ROLE_MEMBER_CHANGE_GROUP
Authentication and Credential Changes	SUCCESSFUL_LOGIN_GROUP FAILED_LOGIN_GROUP LOGOUT_GROUP LOGIN_CHANGE_PASSWORD_GROUP
Server Operations	SERVER_OPERATION_GROUP SERVER_STATE_CHANGE_GROUP BACKUP_RESTORE_GROUP
Security Events	AUDIT_CHANGE_GROUP
Server Access/Changes	SERVER_OBJECT_CHANGE_GROUP SERVER_PERMISSION_CHANGE_GROUP SERVER_PRINCIPAL_CHANGE_GROUP SERVER_PRINCIPAL_IMPERSONATION_GROUP SERVER_OBJECT_PERMISSION_CHANGE_GROUP SERVER_OBJECT_OWNERSHIP_CHANGE_GROUP DBCC_GROUP SERVER_STATE_CHANGE_GROUP
Database Access/Changes	APPLICATION_ROLE_CHANGE_PASSWORD_GROUP DATABASE_CHANGE_GROUP DATABASE_OWNERSHIP_CHANGE_GROUP DATABASE_PERMISSION_CHANGE_GROUP DATABASE_ROLE_MEMBER_CHANGE_GROUP DATABASE_PRINCIPAL_CHANGE_GROUP DATABASE_PRINCIPAL_IMPERSONATION_GROUP DATABASE_OBJECT_CHANGE_GROUP DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP DATABASE_OBJECT_ACCESS_GROUP

configured by applying the SELECT, UPDATE, INSERT AND DELETE audit groups to any tables containing ePHI in accordance with risks to data confidentiality, integrity and availability identified during risk assessment.

Storage and security of audit logs

To provide increased security, the SQL Server Audit object should be configured to store logs in a remote file share. A remote and centralized audit collection service

TABLE 3: POLICY-BASED MANAGEMENT BASELINE VALUES

Safeguard	Facet	Condition	Operator/Value	Target Object(s)	Note
Access Control	Database Security	@ISOwnerSysadmin	=False	ePHI databases	
	Login	@LoginType	=Windows User	Logins that have access to ePHI	Applies to Windows domain logins
	Server Security	@CrossDBOwnershipChainingEnabled	=False	Server	
	Surface Area Configuration	@AdHocRemoteQueriesEnabled	=False	Server	
Authentication	Login	@PasswordExpirationEnabled	=True	SQL Logins	For SQL server authentication
	Login	@PasswordPolicyEnforced	=True	SQL Logins	For SQL server authentication
	Server Security	@LoginMode	=Integrated	Server	For Windows domain authentication
Encryption	Certificate	@Keylength	[=1024]	DEK	
	Certificate	@PrivateKeyEncryptionType	=MasterKey	TDE Certificate (Master DB)	When EKM is not in use
	Certificate	@ExpirationDate, @StartDate	=<DateTime value>	TDE Certificate (Master DB)	When EKM is not in use
	Database	@EncryptionEnabled	=True	ePHI databases	
	Symmetric Key	@EncryptionAlgorithm	=TripleDES or AES256	DEK	
Secure Communications	Server	@TcpEnabled	=True	Server	
	Server	@NamedPipesEnabled	=False	Server	
System Security	Surface Area Configuration	@ClrIntegrationEnabled	=False	Server	
	Surface Area Configuration	@DatabaseMailEnabled	=False	Server	
	Surface Area Configuration	@OleAutomationEnabled	=False	Server	
	Surface Area Configuration	@ServiceBrokerEndpointActive	=False	Server	
	Surface Area Configuration	@SoapEndpointsEnabled	=False	Server	
	Surface Area Configuration	@SqlMailEnabled	=False	Server	
	Surface Area Configuration	@WebAssistantEnabled	=False	Server	
	Surface Area Configuration	@XPcmdShellEnabled	=False	Server	
Audit	Audit	@Enabled	=True	Audits (all)	
	Server	@AuditLevel	=All	Server	
	Server Audit Specification	@Enabled	=True	Server Audit Specifications (all)	
	Database Audit Specification	@Enabled	=True	Database Audit Specifications (all)	

provides the resources to remove access to logs from sysadmins and other privileged users and to consolidate logs from other servers. By implementing the enhanced auditing features of SQL Server 2008, locating audit logs in a central repository and using SQL Server Integration Services and SQL Server Reporting Services, a secure and comprehensive facility to generate actionable audit reports is available. Third-party tools are no longer required for audit logging and reporting. To ensure the confidentiality and integrity of audit logs:

- Users, including fixed database roles, should not be able to monitor or modify log files
- Changes to audit specifications should be logged and monitored

Policy-based Management

With Policy-based Management, SQL Server configuration options and security settings can be tested and monitored.

“Out of compliance” alerts can be distributed based on a central policy, and alerts should be enabled to enforce policy and schema across the database infrastructure.

Policy-based Management should not be considered a security enforcement tool, as privileged users may issue SQL statements or reconfigure settings that circumvent policy. This feature can, however, assist in defining, configuring and monitoring security settings across multiple instances of SQL Server. In SQL Server 2008, a number of nonessential features are disabled by default to minimize security vulnerabilities. Policy-based Management can be configured to selectively enable additional required features and to help ensure other features remain disabled.

When implementing Policy-based Management, consider the baseline values for testing as shown in Table 3 on Page 10.

CONCLUSION

Microsoft SQL Server 2008 provides a suite of technical security and auditing features that can be leveraged to effectively address the technical safeguard requirements within the HIPAA Security Rule. Policy-based Management features can help promote consistency of security within systems, databases and applications, and assist with the ongoing monitoring of, and compliance with, management policies and standards. In addition, SQL Server 2008 features such as TDE, support for TLS\SSL, strong authentication/password controls and role-based access controls, can be enabled to encrypt and safeguard health records at rest and in transit.

No single technology solution ensures end-to-end compliance. This white paper provides insight into Microsoft SQL Server 2008 security features and how these features can support an organization’s HIPAA compliance and risk mitigation strategies. Successful compliance with HIPAA, or other regulations, will continue to require due diligence and action. Compliance can be achieved most effectively through the integration of multiple layers of security defenses, educating management to obtain adequate funding, and employing appropriate experts in database security, auditing and internal controls.

Logging security events may facilitate notification for action. It is critical for management to require appropriately timed monitoring and review of security event logs. Formal independent audits of the implemented

security and auditing features should also be performed on a regular basis, as there is little benefit in discovering internal control problems days or weeks after they occurred. How a covered entity defines its monitoring and Internal Audit policies and procedures should be based on the outcome of its HIPAA risk assessment. If a security incident occurs, the failure to effectively address security risks serves as proof in an inquiry that a covered entity was capable of knowing what was occurring, but failed to exercise timely corrective action.

Management is encouraged to closely scrutinize its vendors and solutions to ensure they continue to evolve and keep pace with the regulatory environment. Organizations need to formally evaluate specific products’ technical capabilities when making purchasing decisions – even if they have chosen not to implement a given safeguard at present. They need to know appropriate product security features will be available if required in the future and that vendors have compliance in mind with respect to their ongoing product development and support.

For more information:

www.microsoft.com/technet
www.hhs.gov/ocr/privacy
www.microsoft.com/downloads
www.jeffersonwells.com/mssql2008hipaa

