Procedures for Establishing User Access Controls to Electronic Protected Health Information

HIPAA Security Rule, Technical Safeguards, §164.312(A)(1)

GIAC HIPAA Security Certificate (GHSC) Practical Assignment (V1.0)

Barbara Filkins
August 10, 2004

Abstract

In any distributed system design, there are at least three levels of security protection that must be considered: network (including servers and workstations), applications that access and manage the sensitive data, and the electronic data itself. Security at the data level must be considered as part of an entity's HIPAA Security implementation.

Our emphasis for this paper is on the technical implications of database user access controls. This paper presents a practical case where three separate agencies, each representing a different aspect of health care, intend to share electronic protected health information (ePHI) with the goal of developing better outcome measures and improved access to care for their beneficiary population.

The main entity, GIAC Health, must establish technical safeguards for role-based user access to this database. Access must be compliant with HIPAA regulations for privacy and security, supportable with policy, and able to be managed and monitored with the tools currently available in the technical environment.

Table of Contents

Table of Figures

Table of Tables

Assignment One: Define the Environment

GIAC Health is a regional healthcare corporation that serves a diverse patient population of approximately 80,000 beneficiaries in the southwestern United States. The corporation is strongly aligned with a federal research agency. This agency bases their longitudinal studies for chronic disease on a subset of the corporation's patient population and, in return, makes available to GIAC Health extremely modern healthcare techniques and technology. The state's public health agency also accesses the beneficiary data to improve their understanding of this chronic disease and support their related health surveillance activities.

The operational arm of GIAC Health maintains both an electronic health record and a paper chart on each of their patients. The clinical studies branch maintains a parallel electronic record on each patient enrolled in a specific study. This information is shared with the federal agency and its researchers. The public health agency extracts information from the paper charts supplied by GIAC Health and imports the information into an independent Access database. The three entities recently entered into a cooperative project to develop a Reconciliation Database (RDB) with the goals of improving the overall quality of the clinical information being collected, developing better outcome measures, and improving the access to care for the patient population.

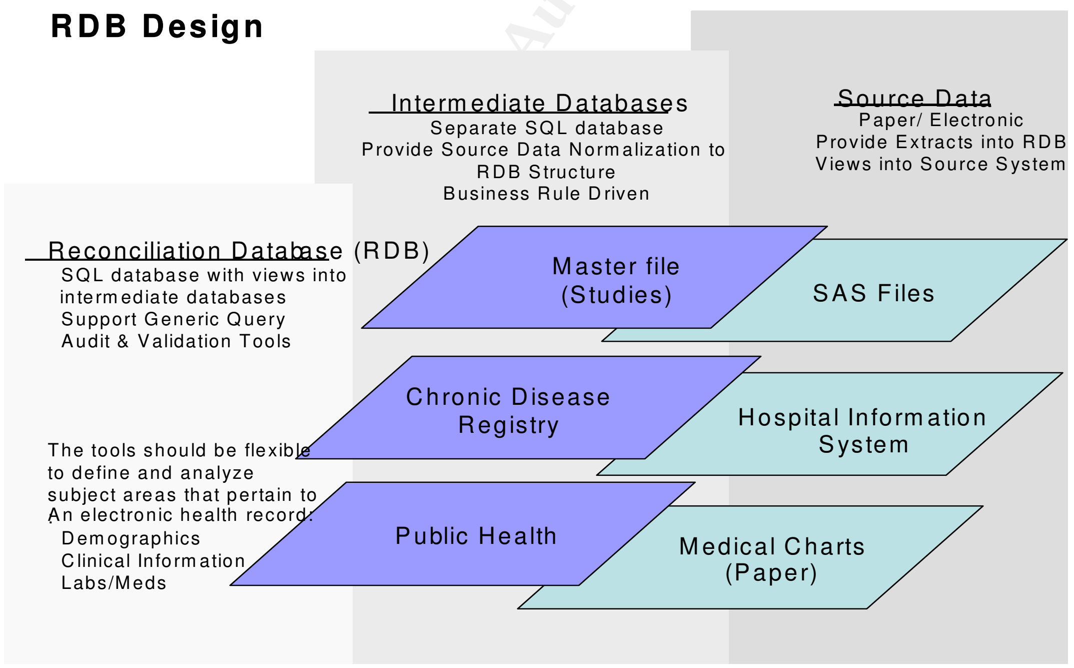


Figure 1: RDB Design

The RDB design approach is shown in Figure 1. Pertinent data from each entity's source system, whether electronic or paper based, is extracted into a corresponding intermediate database. The resulting data elements are then mapped against the RDB schema, transformed, and loaded into the RDB. The RDB can then be used as the basis for all analysis and decision making across the three entities.

Physically, the RDB is developed in Microsoft's SQL Server and located on a dedicated Windows 2000 platform at GIAC Health. Network access to the server and the RDB application is controlled by GIAC Health Network Operations. Users of the RDB have been granted network access, including Virtual Private Network (VPN), in accordance with the policies of GIAC Health and any relevant agreements established with their business associates, such as the federal research and state public health agencies.

Table 1 summarizes the rules that govern access to electronic protected health information (ePHI) by the personnel from each organization. These rules were defined to ensure compliance with the HIPAA Privacy Rule. As the RDB contains electronic protected health information, the appropriate safeguards being implemented for RDB access are being addressed by GIAC Health in accordance with the HIPAA Security Rule.

Table 1: Business Requirements for User Roles

User Class	**Focus**	**Business Rule for Access to PHI (Minimum Necessary)**	
		Demographic	**Clinical**
GIAC Health -- Operations (Clinical and Administrative)	Covered entity. Treatment, payment, Healthcare operations	Full access to all demographic information contained in the RDB as defined by operational role	Full access to all clinical (medical and behavioral) information about a patient contained in the RDB as defined by operational role
GIAC Health -- Clinical Studies	Research for chronic disease	Access to all demographic information contained in the RDB under terms of current data sharing agreement with Operations	Full access to all medical information about a patient contained in the RDB under terms of current data sharing agreement with Operations
Public Health (External Agency)	Public health surveillance, research	Access to limited data set under terms of data sharing agreement with GIAC Health and consistent with § 164.514 (Federal Register, vol. 65, no. 250, 83818-82820)	Summary information only as defined in § 164.504 (Federal Register, vol. 65, no. 250, 83807-82810)

Assignment Two: Explanation

The HIPAA specification being addressed in this paper is §164.312(A)(1), Technical Safeguards, Standard, Access Control. The standard is to "implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in § 164.308(a)(4)" (Federal Register, vol. 68, no. 34, 8378). §164.308(a)(4) refers to the Administrative Safeguards for information access management. This standard is to "implement policies and procedures for authorizing access to electronic protected health information that are consistent with the applicable requirements of subpart E [i.e., the HIPAA Privacy Rule] of this part" (Federal Register, vol. 68, no. 34, 8377).

The business rules for user access to RDB data are outlined in Assignment One. Users have roles, determined both by the organization they belong to and the functions they must perform. Only a user with explicit access to specific information in the RDB is allowed to view, update, or otherwise manipulate that data. The RDB requires role-based access control and its security design must map these organizational and functional roles to system defined ones (Northcutt 209). GIAC Health's objectives are to first establish a policy for controlling user access to the RDB, then determine the appropriate levels of user access, and finally manage access to the database and its information.

The initial instantiation of the RDB is as a 'proof of concept', accessed by relatively few users. Some users will require remote access to the RDB from clients that are other than Windows 2000 or higher. Cost is also a factor. GIAC Health management has mandated that no third party tools other than tools native to SQL Server will be used to define and monitor the access to the RDB.

For problem clarification, an excellent first step is to translate these concepts and constraints into specific requirements that the policy, the database design, and the system implementation must address:

1) User access to the RDB shall be role based, controlled both by user unique identification and user assigned role(s) according to GIAC Health business rules related to the privacy and security of patient health information.

2) A user's access to any database object (i.e., table, view, stored procedure, constraint) shall be controlled by the permissions assigned to the user's role and the object.

3) The following user roles shall apply:

Table 2: RDB User Roles

Organization	Function		
	System Administrator	Analyst	Data Entry/Update
Operations	X	X	X
Clinical Studies		X	X
Public Health		X	

4) The system shall provide a means of authenticating user access independent of the user's network login / password. (In other words, the access credentials will be managed within SQL Server using mixed mode authentication.)

5) The system shall provide the ability to monitor all security actions related to user access and actions performed on database objects. The system shall capture, but not be limited to, the following information as part of the monitoring process:
   - User Name
   - Event or action (LOGIN/LOGOUT, SELECT, INSERT, UPDATE, DELET)
   - Event date/time
   - Database object affected or accessed
   - Event success or failure

6) The system shall provide all information gathered from the monitoring process for immediate review without directly impacting current database or server operations.

7) The system shall use tools native to SQL Server to define and monitor the access to the RDB.

Assignment Three: Reconciliation Database (RDB) User Access Policy

1.0 Overview

The Reconciliation Database (RDB) is being developed for improved chronic disease management. It represents an agreement between three separate organizations – GIAC Health Operations, GIAC Health Clinical Studies, and the State Public Health Agency – to share electronic protected healthcare information (ePHI) with the goal of developing better outcome measures and improved access to care in battling the incidence of chronic disease in the GIAC Health beneficiary population.

2.0 Purpose

The purpose of this policy is to establish procedures by which users access the RDB.

3.0 Scope

The scope of this policy includes all personnel who have access to the Reconciliation Database that resides at the main GIAC Heath facility. This policy only addresses access to the database itself. Individual access to the database server must be in compliance with all appropriate GIAC Health network, system, and/or remote user access policies. Individual access to specific RDB information is dependant upon which role(s) an individual user is assigned under the terms of the current data sharing agreement between GIAC Health and its business partners.

4.0 Policy

The following has therefore been implemented:

- A unique username and password will be assigned to each individual accessing the RDB. To further help ensure that these accounts are protected to reasonable standards, the password management policy found at http://www.sans.org/resources/policies/Password_Policy.pdf will be followed.
- Each individual will be assigned a role that determines their access to the data contained in the RDB.
- Each individual will be subject to monitoring while accessing the RDB. As a minimum, the following will be tracked:
    - User logins (success/failure)
    - User access to database objects
    - User actions on database objects (success/failure)

For user access to be granted, the individual must agree to the following:

- An individual will never allow their RDB user account to be used by another individual after its creation. Evidence of such action will make both the user whose account is being used and the interloper subject to enforcement action.
- An individual agrees to stay within the user rights associated with their role in the system. Any attempt to violate these rights will be monitored and the user shall be subject to enforcement.
- An individual understands and agrees to the monitoring of their activities within the RDB by GIAC Health.

5.0 Enforcement

The responsibility for complying with this policy belongs to the personnel responsible for creating accounts. Any user found to have violated this policy shall have their access to the RDB revoked and may be subject to disciplinary action, up to and including termination of employment.

6.0 Definitions

Note: Any relevant definitions should be included here.

7.0 Revision History

Note: Revision history for this policy should be provided here.

8.0 References

Note: Other GIAC Health policies or documentation should be referenced here.

Assignment Four: Procedures

The procedures outlined in this section are to ensure RDB access in accordance with previous policy. This discussion covers four main steps and provides practical examples. It is intended as a basis from which more comprehensive procedures can be developed. This discussion assumes that the reader is familiar with the SQL Server 2000 tools: Enterprise Manager, Query Analyzer, and SQL Profiler.

Step One: Establish RDB User Access

SQL Server user access involves two main steps. First, users must log into SQL Server itself via a user login account. Second, the user account on the server must connect to a specific database so that the user login can access the contents of the tables and other objects within that specific database. Because non-Windows users will be accessing the RDB, the RDB will use mixed mode authentication, where SQL Server validates a user's login against credentials stored in the server.

SQL Server has two standard or 'fixed' categories for user roles. These should be reviewed as a starting point for any custom roles that may be required. Server roles govern administrative permissions on a server wide basis, i.e., across all databases within that server. Table 3 describes the fixed server roles.

Table 3: Fixed SQL Server Roles ("Roles")

Fixed server role	Description
Sysadmin	Can perform any activity in SQL Server.
Serveradmin	Can set server-wide configuration options, shut down the server.
Setupadmin	Can manage linked servers and startup procedures.
Securityadmin	Can manage logins and CREATE DATABASE permissions, also read error logs and change passwords.
Processadmin	Can manage processes running in SQL Server.
Dbcreator	Can create, alter, and drop databases.
Diskadmin	Can manage disk files.
Bulkadmin	Can execute BULK INSERT statements.

Each database within a SQL Server has a set of fixed roles that govern administrative actions only within that database. Roles with the same name can exist in each database, but the scope of each role is only within that specific database. For example, the **RDB** and **Operational Intermediate Database** both have user IDs named **Operations_SysAdmin**. Adding **Operations_SysAdmin** to the **db_owner** fixed database role for the **Operational Intermediate Database** has no effect on whether **Operations_SysAdmin** in **RDB** is a member of the **db_owner** role for that database.

Table 4: SQL Server Fixed Database Roles ("Roles")

Fixed database role	Description
db_owner	Has all permissions in the database.
db_accessadmin	Can add or remove user IDs.
db_securityadmin	Can manage all permissions, object ownerships, roles and role

Fixed database role	Description
	memberships.
db_ddladmin	Can issue ALL DDL, but cannot issue GRANT, REVOKE, or DENY statements.
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements.
db_datareader	Can select all data from any user table in the database.
db_datawriter	Can modify any data in any user table in the database.
db_denydatareader	Cannot select any data from any user table in the database.
db_denydatawriter	Cannot modify any data in any user table in the database.

Roles allow a system administrator to establish login accounts that essentially have database but not server access. The role to which the GIAC Health RDB Administrator is assigned may allow addition or removal of user access from the RDB but not addition or removal of user accounts from the SQL Server containing the RDB.

Any database security planning should start with the development of a security decision matrix (Poolet). This provides the roadmap for assigning permissions to each database object, including views, tables, and stored procedures. This is actually a Create, Read, Update, Delete (CRUD) matrix tailored to the SQL Server environment: Relevant permissions include:

SELECT: Permits a user to access a table or view with a SELECT statement.
INSERT: Permits a user to add a new row of data in a table or view.
UPDATE: Permits a user to update values in a table or view.
DELETE: Permits a user to delete data from a table or view.
REFERENCES: Permits a user to make foreign key references to the primary key and unique columns of another table. Also permits a user to bind a database object, such as a view, to the underlying table schema, restricting changes to the view unless the underlying tables are changed or the reference removed.
EXECUTE: Permits a user to execute a stored procedure attached to a database object. For example, an audit report may use a stored procedure to gather the needed information. Since the report contains sensitive data, the system administrator may be the only one allowed to execute this stored procedure.

The Private Patient Info table contains those data elements that are considered sensitive to release, such as patient name and address. Access has been restricted for Studies and Public Health personnel per their data sharing agreement with Operations. Table 5 first shows a basic permission matrix that will be applied to all tables and views UNLESS otherwise indicated. The second matrix indicates how the permissions on the Patient Private Info table differ from the standard permissions matrix.

A complete RDB security design includes a matrix for each database object, covering tables, views, and stored procedures. Role names should be descriptive, not only of the organization but also the defined set of tasks that discrete groups of people perform. This matrix can help determine whether the fixed database roles will work for a given class of users. The same format can be used to define any server roles that may be needed for database administration staff.

Table 5: Decision Support Matrix

Security Decision Matrix

Roles		Permissions				
Standard Permissions -- All Tables and Views						
		Select	Insert	Update	Delete	References
Operations	Sys Admin	X	X	X	X	X
	Analyst	X				
	Data Entry	X	X	X		
Studies	Analyst	X				
	Data Entry	X	X	X		
Public Health	Analyst	X				
Patient Private Info Table Permissions						
		Select	Insert	Update	Delete	References
Operations	Sys Admin	X	X	X	X	X
	Analyst	X				
	Data Entry	X	X	X		
Studies	Analyst	X				
	Data Entry		Access not needed nor allowed			
Public Health	Analyst		Access not needed nor allowed			

The following steps are involved in establishing user access to the RDB:

1. *Define SQL Server database roles that describe the data use requirements for the RDB activities.* Both organizational and functional requirements need to be considered.

2. *Map the role permissions to the RDB database objects.* This is where the security decision matrix is a handy tool as it can be used to present and confirm the security business rules to people who are not technology or programming oriented. Note: Every user in a database belongs to the **public** database role. Permissions assigned to the **public** role apply to all database users. If a user has not been specifically granted permissions on an object, they use the permissions assigned to **public**.

3. *Assign individual logins to these roles.* A user requests access and agrees to the terms of the policy in Assignment Three. A complete list of users who have access to the RDB is prepared. Each individual login account is assigned a unique user ID and password. Each individual account is also assigned to at least one role and possibly more than one. (Note: In SQL Server 2000, a user can belong to multiple roles.)

4. *Write and execute security scripts based on these mappings.* The security matrix is the guide to writing any SQL scripts to granting (or revoking) permissions. The GRANT command assigns permissions to groups and roles; the REVOKE command removes permissions already granted. This can also be used for

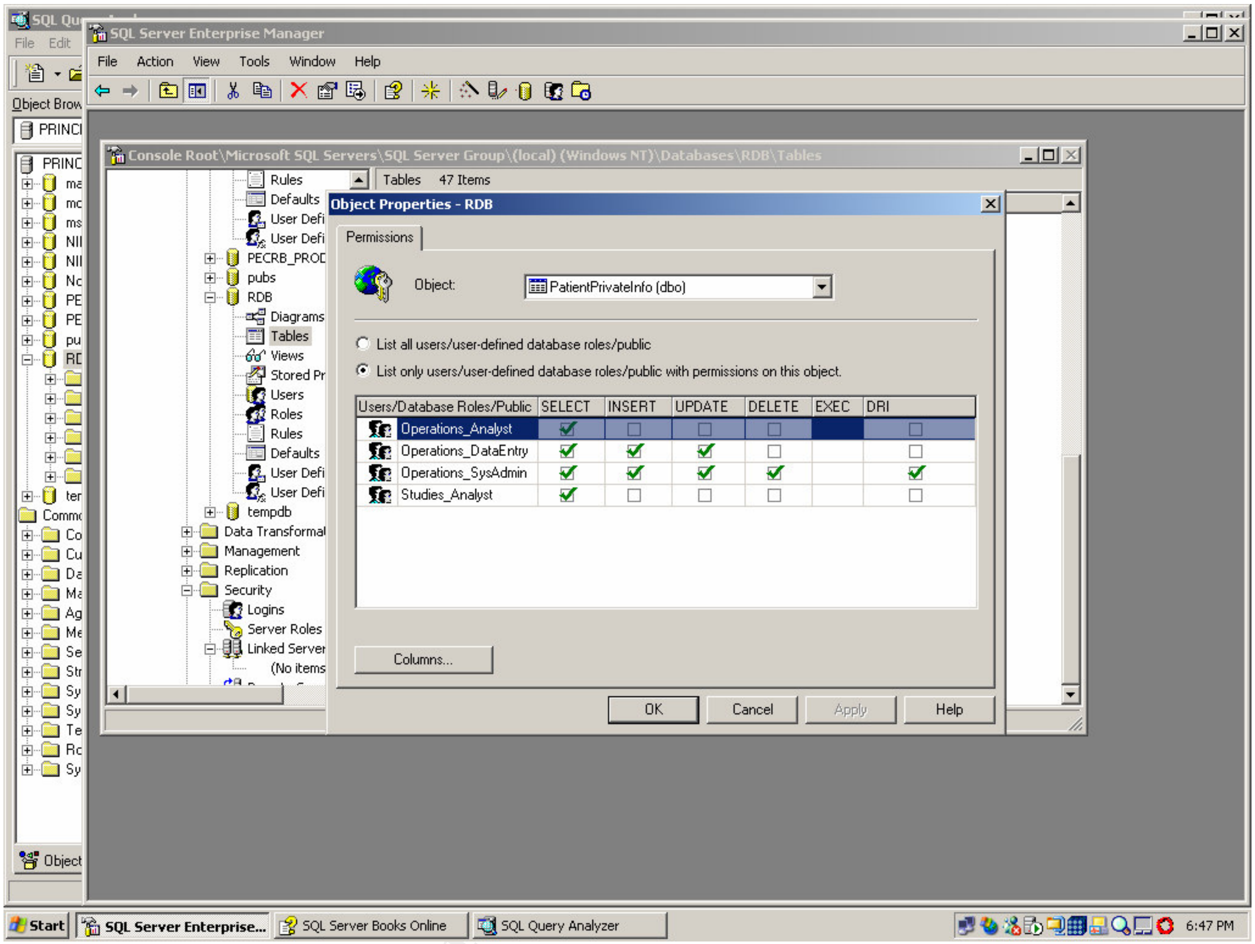
assigning permissions through Enterprise Manager. (Note: Rights assigned individually override those that are assigned via a role.)

The following demonstrates a simple security script. First, logins are added to the database server and each user's default database is established as the RDB. Then, these users are granted specific access to the RDB database and roles are established. Finally, various users are assigned to the roles and the restrictions on the Patient Private Info table established for each role according to the security decision matrix:

```
USE master
EXEC sp_addlogin 'John'
EXEC sp_defaultdb 'John', 'RDB'
EXEC sp_addlogin 'Sarah'
EXEC sp_defaultdb 'Sarah', 'RDB'
EXEC sp_addlogin 'Betty'
EXEC sp_defaultdb 'Betty', 'RDB'
EXEC sp_addlogin 'Ralph'
EXEC sp_defaultdb 'Ralph', 'RDB'
EXEC sp_addlogin 'Diane'
EXEC sp_defaultdb 'Diane', 'RDB'
USE RDB
EXEC sp_grantdbaccess 'John'
EXEC sp_grantdbaccess 'Sarah'
EXEC sp_grantdbaccess 'Betty'
EXEC sp_grantdbaccess 'Ralph'
EXEC sp_grantdbaccess 'Diane'
EXEC sp_addrole 'PublicHealth_Analyst'
EXEC sp_addrole 'Studies_Analyst'
EXEC sp_addrole 'Studies_DataEntry'
EXEC sp_addrole 'Operations_SysAdmin'
EXEC sp_addrole 'Operations_Analyst'
EXEC sp_addrole 'Operations_DataEntry'
EXEC sp_addrolemember 'PublicHealth_Analyst', 'John'
EXEC sp_addrolemember 'PublicHealth_Analyst', 'Sarah'
EXEC sp_addrolemember 'Studies_DataEntry', 'Diane'
EXEC sp_addrolemember 'Operations_DataEntry', 'Diane'
EXEC sp_addrolemember 'Operations_SysAdmin', 'Betty'
EXEC sp_addrolemember 'Operations_Analyst', 'Ralph'
GRANT SELECT ON PatientPrivateInfo TO Studies_Analyst, Operations_Analyst
GRANT SELECT, INSERT, UPDATE ON PatientPrivateInfo TO Operations_DataEntry
GRANT ALL on PatientPrivateInfo to Operations_SysAdmin
```

This script gives the Diane, the data entry clerk for both Operations and Studies, the ability to add new demographic information or update existing data regarding a patient's residence (contained in the Patient Private Information table), while analyst Ralph can only select the information. John, as an analyst for Public Health, is denied access to this table even to select the data. Betty, on the other hand, has complete access to this table as system administrator. Not allowing Diane to delete information enforces an audit process that requires a more stringent review before patient information is completely deleted from the system.

Reviewing the permissions assigned to this table in the Enterprise Management results in the following.



Object Properties - RDB

Permissions

Object: PatientPrivateInfo (dbo)

○ List all users/user-defined database roles/public

● List only users/user-defined database roles/public with permissions on this object.

Users/Database Roles/Public	SELECT	INSERT	UPDATE	DELETE	EXEC	DRI
Operations_Analyst	☑	□	□	□		□
Operations_DataEntry	☑	☑	☑	□		□
Operations_SysAdmin	☑	☑	☑	☑		☑
Studies_Analyst	☑	□	□	□		□

Figure 2: Database Object Assignment in Enterprise Manager

Step Two: User Access Management

All user accounts and logins, roles, and account and access permissions within the server and the database should be reviewed periodically. The following is a list of commends that are helpful for this review. From Query Analyzer, type EXEC followed by the one of the following commands:

Command	Result
sp_helprole	Lists all roles in the database, both fixed and user defined
Sp_helprolemember	Lists database roles and the users assigned to each role
Sp_helprotect	List permissions associated with role
Sp_helpuser	Lists all user associated with a database. This command allows you to discover the user assigned to a role. For example, the command: EXEC sp_helpuser @name_in_db = 'Operations_DataEntry' will return all users assigned to the role of Operations Data Entry

Enterprise Manager provides a GUI front-end for reviewing user access and roles. Refer to Microsoft's product documentation for further information. SQL Server 2000 Books on Line (BOL) can either be downloaded from

http://www.microsoft.com/sql/techinfo/productdoc/2000/books.asp

or accessed on-line at

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/startsql/getstart_4fht.asp.

Step Three: Establish Monitoring on User Access

Microsoft has implemented the ability to enable C2 auditing for a given SQL Server 2000 database. The disadvantage with this approach is that it is an "all-or-nothing" solution, resulting in potentially large volumes of data. Additionally, C2 requires the database service be stopped and restarted for this auditing to take effect. In the early stages of the RDB project, GIAC Health had very specific requirements on which event categories need to be monitored for user access, including specific tables containing sensitive patient demographic information. GIAC Health elected to use Server Side Traces, allowing flexibility in determining what data to capture and providing finer control over when traces can be started and stopped, without the inconvenience of having to restar the database service.

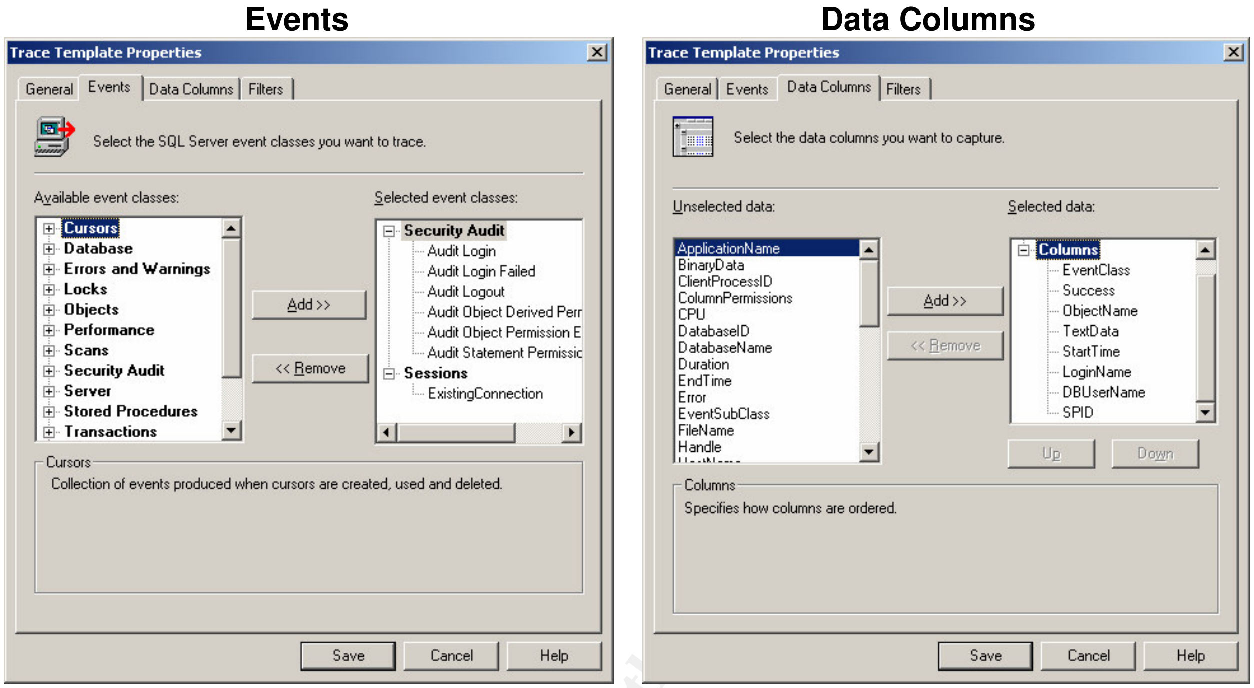
The area of immediate interest for auditing user events include logins (Failed, Successful) and logouts and the attempted violation by a user of the permissions on a database object (i.e., table) according to their role. A record is needed of whether a user accessed data using a SELECT, INSERT, UPDATE, or DELETE statement.

For the initial implementation of the RDB, GIAC Health chose to look at SQL Server's auditing capabilities accessible through the Windows GUI based tool -- SQL Server 2000 Profiler. SQL Server Profiler was used to define a trace that captures this basic information. The Profiler can also be used to review the output from the trace process.

1) *Develop an appropriate trace template.* Templates contain certain combinations of events, data columns, and filters. Events are the activities generated within the SQL Server engine, such as a user login, the execution of a stored procedure, the completion of a SQL statement, or a security permission check on an object. Data columns are attributes of events, like start and stop times. Filters are just that – they provide the ability to filter on the values of the event attributes to help minimize server resources and better analyze the results ("SQL Profiler").

   A basic pre-filter (i.e., placing the filter when the trace is defined) was applied to capture activities related only to the RDB. This was done to conserve server side resources. Several initial runs were made with all events and all data columns selected. Multiple database operations were performed (such as adding and revoking user accounts, permissions, and updating tables) to determine what combination of events and columns generates the best results.

The following figure show the events and data columns within the Profiler that were selected to generate the desired output:



Events

Trace Template Properties

General | Events | Data Columns | Filters

Select the SQL Server event classes you want to trace.

Available event classes:
- Cursors
- Database
- Errors and Warnings
- Locks
- Objects
- Performance
- Scans
- Security Audit
- Server
- Stored Procedures
- Transactions

Selected event classes:
- Security Audit
  - Audit Login
  - Audit Login Failed
  - Audit Logout
  - Audit Object Derived Perr
  - Audit Object Permission E
  - Audit Statement Permissic
- Sessions
  - ExistingConnection

Cursors
Collection of events produced when cursors are created, used and deleted.

Data Columns

Trace Template Properties

General | Events | Data Columns | Filters

Select the data columns you want to capture.

Unselected data:
ApplicationName
BinaryData
ClientProcessID
ColumnPermissions
CPU
DatabaseID
DatabaseName
Duration
EndTime
Error
EventSubClass
FileName
Handle

Selected data:
- Columns
  - EventClass
  - Success
  - ObjectName
  - TextData
  - StartTime
  - LoginName
  - DBUserName
  - SPID

Columns
Specifies how columns are ordered.

Figure 3: Events and Data Columns Assignments in Profiler

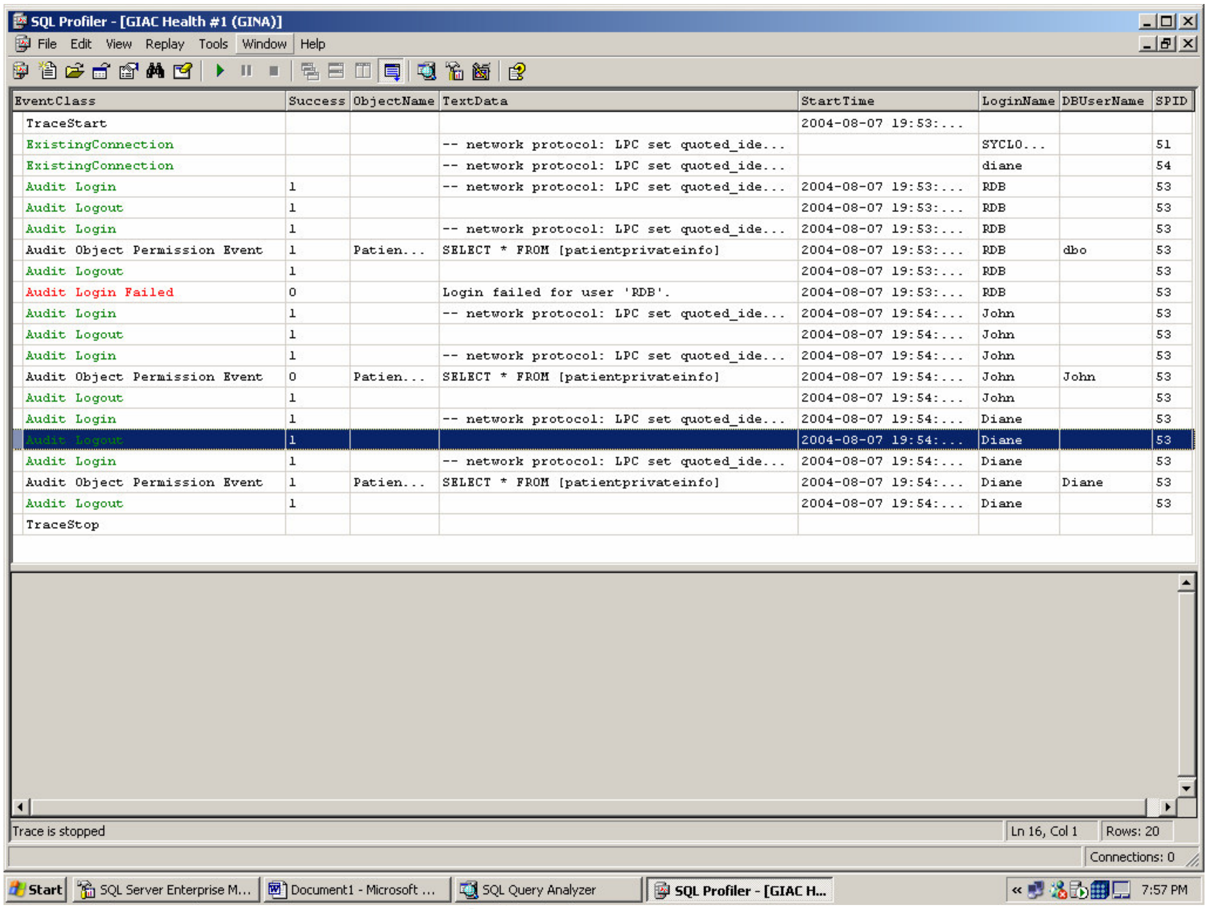
2) *Save the trace template.* GIAC Health saved the template under the name RDB User Access. Only one log is generated for monitoring all events. GIAC Health deemed this procedure more manageable since the procedures were being developed to support the RDB access policy.

3) *Run the resulting trace.* The trace template is first selected and an output file is named. The figure shows the results of a short test scenario:

User RDB connects to the RDB with SQLQuery, logging with the proper password
User RDB runs a select query to return all rows on PatientPrivateInfo that completes successfully
User RDB disconnects
User RDB attempts reconnection without the proper password
Reconnection fails
User John connects to the RDB with SQLQuery, logging in with his proper password
User John attempts to run the same select query on PatientPrivateInfo, but, as a member of PublicHealth_Analyst role, he is denied access.
User John disconnects
User Diane connects to the RDB with SQLQuery, logging in with his proper password
User Diane attempts to run the same select query on PatientPrivateInfo and, as a member of Studies_Analyst role, she is successful.
User Diane logs out.

The resulting trace is shown in the figure below. Note that all event classes that complete successfully return a success value of 1. The failed login and attempted

SELECT on the Patient Private Info table by John return a value of 0 that indicates failure.


SQL Profiler - [GIAC Health #1 (GINA)]
File
Edit
View
Replay
Tools
Window
Help
Trace is stopped
Ln 16, Col 1
Rows: 20
Connections: 0
Start
SQL Server Enterprise M...
Document1 - Microsoft ...
SQL Query Analyzer
SQL Profiler - [GIAC H...
7:57 PM

EventClass	Success	ObjectName	TextData	StartTime	LoginName	DBUserName	SPID
TraceStart				2004-08-07 19:53:...			
ExistingConnection			-- network protocol: LPC set quoted_ide...		SYCLO...		51
ExistingConnection			-- network protocol: LPC set quoted_ide...		diane		54
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:53:...	RDB		53
Audit Logout	1			2004-08-07 19:53:...	RDB		53
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:53:...	RDB		53
Audit Object Permission Event	1	Patien...	SELECT * FROM [patientprivateinfo]	2004-08-07 19:53:...	RDB	dbo	53
Audit Logout	1			2004-08-07 19:53:...	RDB		53
Audit Login Failed	0		Login failed for user 'RDB'.	2004-08-07 19:53:...	RDB		53
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:54:...	John		53
Audit Logout	1			2004-08-07 19:54:...	John		53
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:54:...	John		53
Audit Object Permission Event	0	Patien...	SELECT * FROM [patientprivateinfo]	2004-08-07 19:54:...	John	John	53
Audit Logout	1			2004-08-07 19:54:...	John		53
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:54:...	Diane		53
Audit Logout	1			2004-08-07 19:54:...	Diane		53
Audit Login	1		-- network protocol: LPC set quoted_ide...	2004-08-07 19:54:...	Diane		53
Audit Object Permission Event	1	Patien...	SELECT * FROM [patientprivateinfo]	2004-08-07 19:54:...	Diane	Diane	53
Audit Logout	1			2004-08-07 19:54:...	Diane		53
TraceStop							

Figure 4: SQL Profiler Resulting Trace

Step Four: Define Routine Procedures to Monitor User Access

The final step is to define routine procedures for monitoring RDB user access. These include:

- *Review the security decision matrix at regular intervals and when events occur that should trigger a review.* Trigger events include: changes to HIPAA related regulations that affect the business rules for ePHI access; changes to data sharing agreements between the entities, such as when Clinical Studies initiates a new research project; any occurrence of a privacy or security incident that involves the RDB.
- *Compare the security decision matrix against the current user accounts, roles, and logins for both the server and the RDB database to ensure that the system meets the requirements.* This should occur both periodically and when the events described above occur.

- *Review the Profiler traces for completeness and compliance with the policy.* In addition to the regular review of the audit logs, certain events such as the addition or deletion of a new user or new database role, policy updates or changes, should trigger a mini 'audit' to ensure that the RDB configuration matches the current security profile.

Summary Checklist

The four procedural steps to ensure secure user access to the RDB at GIAC Health are summarized below. The same steps can be followed when establishing any database that contains ePHI.

1. **Establish Database User Access.** Define the business rules for access and develop organizational and functional roles. Develop a security decision matrix that embodies the requirements in the user access policy for a given database. Use the appropriate SQL Server tools to translate the requirements in this matrix into database and server roles and the assignment of database object permissions. Ensure that all users have unique IDs for access to the database and assign each individual user to the correct role(s).
2. **User Access Management.** Develop the scripts or other procedures/tools needed to periodically review the assignment of user rights against database objects in the database.
3. **Establish Monitoring for User Access.** Either use the native SQL Server tools, such as Profiler, or develop a more robust capability using triggers and additional tables that support an audit capability.
4. **Define Routine Procedures to Monitor User Access.** Check the database usage history and review events and patterns of activity that may correlate with spurious events. These can be done as part of periodic reviews/audits or when specific events occur that should trigger the action.

References

1. “45 CFR Parts 160. 162, and 164. Health Insurance Reform: Security Standards: Final Rule”. *Federal Register*, vol. 68. no. 34. (February 20, 2003), 8334-8381.

2. “45 CFR Parts 160 and 164. Standards for the Privacy of Individual Identifiable Health Information: Final Rule”. *Federal Register*, vol. 65. no. 250. (December 28, 2000), 82462-82829.

3. Northcutt, Stephen, ed. HIPAA Security Implementation, Version 1.0. SANS Press, 2004.

4. Poolet, Michelle A. “Solutions by Design: The Security Matrix.” SQL Server Magazine. March 2002 (2002): 69.

5. “Roles; Logins, Users, Roles, and Groups; Logical Database Components, Database Architecture”. Microsoft SQL Server 2000. MSDN Library. Microsoft. URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_da_7v1h.asp (July 8, 2004)

6. “SQL Profiler Terminology”. Microsoft SQL Server 2000. MSDN Library. Microsoft. URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_mon_perf_7oah.asp (July 9, 2004)