Global Information Assurance Certification Paper

Case Study:
Applying CIA to Great Plains and MS SQL Server

For
GSEC Practical Submission
Practical Assignment 1.4b, Option 2

Submitted by
William G. Black
March 31, 2004

Mentored by David F. Severski
Seattle, WA
October 2003

Abstract

Securing Microsoft SQL Server is similar to most Microsoft (MS) products in that they are easy to initially install by following the wizard, yet prove difficult to properly and securely setup and administer. This practical is a combination case study and best practices document for applying the security concepts of CIA (Confidentiality, Integrity and Availability) and Access Controls when installing and configuring Microsoft SQL Server and Microsoft Great Plains eEnterprise Software on a Windows 2000 server. The examples given in this document are specific to MS SQL server and Great Plains, but my experience with other environments indicate the guidelines set forth in this document can be applied to other RDBMSs (Relational Database Management System).

It is beyond the scope of the document to address application programming security, SQL performance tuning or Windows 2000 security, except as they apply to this project. This document will cover the "gotchas" I experienced during an upgrade to a new SQL Server and I hope that others may benefit from my experience.

In the beginning

This project started in September 2000 as an upgrade from Great Plains Dynamics 5.5 (Pervasive SQL) to Great Plains eEnterprise 6.0 (SQL server 7.0). The upgrade also included modifying the Sales Order Processing module to accommodate a few table and screen changes for which programming was estimated at 50 hours. The primary cause for failure of the project was that the "few modifications" ballooned into an 18-month late project, which is already 4 times the originally budgeted cost. The result was a system so slow it was unusable and fraught with database corruption errors which the vendor was unable to resolve.

In May 2003, senior management and I met with a new consulting team that determined that both the software and hardware required improvements to reach our goal of a "conversation speed" system. The new consulting team was responsible for improving the software and I was tasked with upgrading the performance of the SQL server. Various options were analyzed to determine which features provided the highest availability with a deadline to have the new server in production on January 1, 2004.

Throughout the course of this project many security and setup mistakes were discovered that were made by the original vendor. The incomplete setup left the SQL Server and Great Plains application vulnerable to misuse. Applying the concepts of CIA and access controls I will discuss my findings, provide examples and explain the rationale behind the key decisions so others may apply these concepts to their circumstances.

CIA and Access Control Defined

Access control is defined as insuring that resources are only granted to those users entitled to them[1] (Cole et al. A-126). Access control includes:

- Separation of duty; the practice of splitting privileges among multiple individuals (Cole et al. 388)
- Least privilege; the principle requires that each subject is granted the most restrictive set of privileges needed for performance of authorized tasks[2] (Ruthberg 747).
- Need to know; ensures that only people that have a need to access certain information or resources will be authorized to do so (Cole et al. 1105).

Confidentiality is the need to ensure that information is disclosed only to those who are authorized to view it (Cole et al. A-128).

Integrity is the need to ensure that information is not been changed accidentally or deliberately and it is accurate and complete (Cole et al. A-134).

Availability is the need to ensure that the business purpose of the system can be met and is accessible to those who need to use it (Cole et al. A-126).

High availability is a term frequently used in marketing, but ironically, I struggled to find a consistent definition. For my purposes, it is the application of redundant and swappable hardware components in combination with software configuration to minimize unscheduled downtime or reduce the time required to recover from a system failure.

Recoverability as defined by me, is the process of sufficiently restoring your environment to allow the business to resume operations. The process of recovery may involve purchasing new equipment and/or recreating lost data and/or restoring tables/ files.

The Foundation (Availability)

Maintaining high-availability begins with redundant, hot-swappable architecture. Most mid-level servers can be configured with hot swap disk drives and power supplies and I suggest comparing the cost of these features to the cost of downtime. Note that my definition of recovery did not state resume normal operations. In a major disaster, it may be several hours or even days before all business processes are fully recovered. It is important to evaluate the cost of downtime for each business process and develop your hardware specifications accordingly. Critical business processes should be prioritized and recovered in the order specified in your Disaster Recovery plan.

During the investigation of the hardware upgrade, there were a couple of options we determined would improve performance and recoverability.

- 4-channel Architecture - To improve performance the recommendation is to put the Operating System, database (.MDB), transaction logs (.LOG) and temporary files (tempdb) on separate SCSI channels. "Separating the log files from data files on physical disks was found to give 10 percent performance gain over placing the logs and data files on the same (larger) volume. For recoverability, logs should never be placed on the same device as data"[3] (Xiong).

- Purchase sufficient memory for the SQL Server to load the entire database into memory. The reasoning for this recommendation is to reduce how frequently SQL accesses the disk subsystem. Disk activity, paging file and memory utilization can be monitored in Performance Monitor to measure effectiveness. In order to achieve this recommendation it is necessary to purchase the Enterprise versions of Windows 2000 server and SQL Server.

There are three key differences between SQL standard and SQL Enterprise worth noting[4] (SQL Server Architecture - Maximum Capacity Specifications):

- There is a 1.8 GB memory limit in standard SQL Server versus a 32 GB limit in SQL Enterprise. There is also a memory limitation for the O/S which is 4 GB for Windows 2000 Server and 8 GB for Windows 2000 Advance Server. We have since migrated to Windows 2003 Advanced server that scales to 32 GB and supports 8 processors. Boot.ini switch for forcing the O/S to use
- Log shipping is another high availability feature which allows your transaction logs to be automatically transferred to a backup database server. Log shipping increases a SQL Server database's availability by automatically copying and restoring the database's transaction logs to another database on a standby server[5] (Talmage).
- SQL Server 2000 supports up to 4 processors, while SQL Enterprise supports up to 8. Again, keep in mind that Windows 2000 Advanced Server is required to support 8 processors.

Carefully perform cost/benefit analysis when evaluating any of these options and use SQL Profiler and Windows Performance Monitor to evaluate the effectiveness of your decisions. Redundancy improves availability by reducing the risk that a single hardware failure would result in downtime. The ability to recover quickly from a system failure begins with a solid hardware foundation and ends with a tested recovery process.

SQL Server Setup

Now that we have a solid hardware foundation, the next step is to install the operating system. As previously indicated the scope of this document does not cover Windows 2000 security other than this reminder to apply all patches before proceeding. Only then will you be ready to install SQL 2000 Server. The information contained in this section was synthesized from various sources including Microsoft training courses and

TechNet articles and credit has been given when quoted specifically. These is not an all-inclusive best practices document and for more complete information refer to the references in the Bibliographies section or use your favorite Internet search engine. This is my list of best practices and all of these were "gotchas" which the original vendor failed to address.

1) Remove the server from the network to prevent the Slammer worm from infecting the server before you have a chance to apply the latest patches (Rapoza 50)[6].

2) Disable the default SQL ports (TCP 1433 and UDP 1434) on your firewall prior to putting the server on the network. Don't forget to re-enable the ports if your SQL Server is accessible from the Internet.

3) Assign a strong password to the SA account[7] (Microsoft SQL Server: 10 Steps to Help Secure SQL Server 2000). It may seem silly to remind people of something so basic, but I have actually had a programmer ask me to remove the SA password because his software wouldn't work with a password (yes, true story and no I didn't).

4) Create a separate Database owner (DBO) for each database, especially those databases that are supported by a consultant or employee that should not have SQL Server administrator privileges. Additionally the DBO account allows the individual access to only the database necessary to perform their work. This is especially important in a SQL environment with multiple databases.

5) Run the KILLPWD utility. Microsoft recommends that administrators delete or remove the installation files, or run the KillPwd utility[8] (Microsoft Security Bulletin MS02-035) to eliminate any traces of passwords used during installation. This problem has been corrected in SP3 for SQL 2000, but it is still a good practice to run the utility to remove any left-over setup files.

6) Use the Full Recovery Model – Best option when recovery from damaged media is the highest priority [9] (Chapter 11 - Database Backup and Recovery) or you need a point in time recovery to a test environment for assisting in problem resolution. Don't forget to include the SQL dump files in your backups.

In my environment we have decided that two hours is an acceptable risk/loss period. My largest database is over 6 GB and the backup to another server takes approximately 25 minutes. On an order entry system that handles 3-5 thousand transactions per hour, log backups consume less than 300 MB of disk space and take less than 60 seconds to complete. My point here is disk space has become so inexpensive that I don't know of any reason not to implement the full recovery model.

Great Plains Access Control

With SQL server securely installed on a solid foundation it is time to install Great Plains eEnterprise. After the lengthy process of installing Great Plains is complete (approx. 14 hours in my environment), then it's time to address access control. Great Plains employs a 2-tier system with a controlling database (Dynamics) granting permissions to multiple company databases. During installation Great Plains creates a SQL role named DYNGRP for each company database. User Logins are created in Great Plains and then granted permission to access the specified company. When a user is granted access to a company the logon name is placed in the DYNGRP for the corresponding company database.

Great Plains uses a granular access control system for granting permission to the various windows, forms etc. based upon either classes or User ID. Great Plains then uses DYNGRP to access the tables on behalf of the user. The information in the Dynamics security structure is used in determining which windows, forms etc. the user is permitted to access.

The original vendor insisted on assigning security to individuals rather than classes. Classes in Great Plains function in similar manner as Windows groups and the argument for using is the same for both. My suggestion is to assign every user to a class even if the class includes a single user. As employees change roles within the company or new employees are hired, classes make it easier to manage and administer security.

My "gotcha" in this area was pointed out by an external audit. Like most accounting systems Great Plains has several functions for closing the month and moving transactions to the historical tables. Unfortunately Great Plains also includes the ability to remove individual historical transactions. Work with your accounting manager to determine who, if anyone should have permission to delete historical transactions. For example the accounting supervisor should not be able to delete general ledger history. This access control can prevent the accounts payable clerk from creating a check transaction and working with the accounting supervisor to remove the transaction from history.

My other "gotcha" was failing to change the password to the system setup menu. This is one of the few security items the original vendor did implement. Over time many people learned the password so they could remove users from the activity file. Initially this wasn't a major concern as you can only modify user security when logged into Great Plains using the SA account. But, when Great Plains is improperly shut down the user remains in the activity file which occurred regularly as the vendor struggled to complete the customized module. The customized module also included a similar file for tracking invoices that are "open" and so invoices also remained after an improper shutdown. Users began reporting that they were "kicked out" of Great Plains. I determined people were attempting to clear a user from Invoice Activity and were incorrectly deleting the user from User Activity. So don't forget to require a password for accessing the System Setup menu and change if it becomes compromised.

The granular security module followed by Great Plains provides adequate access controls for separation of duties and least privilege. Unfortunately it is difficult to administer, requiring patience to learn the hierarchy and terminology used in applying permissions. Another disadvantage is the lack of read-only option and the controlling program model applies the same permissions for each company.

Integrity

Integrity is a less tangible concept then confidentiality and is more difficult to define and enforce. It is a vague, over-arching concept which applies to every thing we do from using computers to driving our cars and is a founding principle for information security. The threat of employee abuse of computer and Internet resources are so common that an entire industry is devoted to monitoring and restricting employee computer activity. I have collected the following narrowed definitions of integrity and will provide examples of how I have implemented them in my organization.

Computer system Integrity – I see this covering the traditional area of security where the primary focus is to prevent the bad guys from penetrating and compromising your systems. Examples in my organization include a strong password policy, proper firewall configuration, prioritized user maintenance for removal of fired employees and physical security of the equipment.

Information Integrity – This is the responsibility of programmers, consultants and other parties responsible for administering access to data. Primary focus is on maintaining access controls, regular backups and disaster recovery. I believe the threat is more likely to be an authorized user deliberately manipulating data with the intent to deceive or misuse the information. Examples include developing an approval policy for changing file or application permissions, limiting access to CD burners and establishing a baseline for monitoring system logs and network activity.

Personal Integrity – Ethics is defined by Webster's dictionary as the discipline dealing with what is good and bad and with moral duty and obligation. Ethics is a primary component of integrity and touches all levels of an organization from the boardroom to the mailroom. I believe that finding and retaining qualified, competent, honest employees is one of the most difficult challenges facing companies today. Information System employees should be members of professional organizations who abide by a published code of conduct. Background checks should be mandatory for highly sensitive positions. Examples implemented in my organization include independent audit of computer system access controls, including ethics question during the interview process and ethics awareness training to staff employees. None of these guarantee an ethical employee, but they do educate employees about the importance of ethics in the organization. Ultimately, you must simply trust your employees.

Business Integrity – Is defined by Microsoft as "the vendor of a product behaves in a responsive and responsible manner" [10] (Mundie) and is listed as one of the four goals of Microsoft's Trustworthy Computing Initiative. I would expect this to be the goal of any company wishing to remain in business and causes one to question the sincerity of Microsoft's initiative[11] (Anderson). I know that this is part of my organizations mission statement. Being a predominantly Microsoft shop, I am interested in where this initiative goes.

As technology continues to spread through the organization I often wonder what role computer professionals will play in the work place of the future. Are we handed responsibility for cellular phones, electronic access controls and alarm systems because trust is already an inherent part of our job responsibility or just because they are electronic devices?

Confidentiality

If your company has HR/payroll data or is publicly traded, confidentiality has recently become a primary concern. HIPAA and Sarbaines-Oxley contain many requirements for protecting confidentially and integrity of the data stored on your networks. All employees must be aware of these regulations even if not in an industry specifically governed by them. There is a chance they may send or receive information from an industry that is governed by these and other regulations.

In my organization there are users who routinely write their own reports or extract data to Excel or Access. I have created two SQL server logons that are granted read-only access and are used for report writing; one includes access to the payroll and HR tables and the other does not. Restricting the accounts to read-only prevents users from accidentally writing out to the tables. Limiting knowledge of the password for accessing the HR tables to the HR manager restricts who can view this sensitive information.

Be aware of what data is taken off-site by your consultants. Unless the consultant has a need to work with your HR or payroll tables I suggest removing these tables from the SQL backup provided to them. To simplify the process I suggest creating a SQL maintenance job that only includes the tables necessary for the consultants to complete their assignments.

My "gotcha" in this area turned out to be Great Plains Explorer. Explorer is a report writing tool which I discovered has it's own set of permissions for restricting access to the tables. It took some digging to find that Explorer permissions were located under Alternative eEnterprise Windows in the security module. To maintain confidentiality I suggest restricting access to the HR and Payroll tables from the classes that are not privileged to view them.

Conclusion

In the end, the upgrade project was a success as there was a dramatic improvement in the response time of the application. As you will recall the baseline to open PBK was initially 40 seconds. Changes to the software reduced the response time to 5 and the hardware upgrade ultimately reduced the time to less than 1 second. A major re-design of one section of the custom module eliminated the data integrity errors.

The project was also a success for implanting proper security and access controls. During the upgrade I realized that in addition to a lack of project management skills, the Great Plains vendor also had a minimal understanding of proper security procedures for Great Plains and SQL Server. The new SQL server is correctly configured based on best practices; access controls are properly applied to Great Plains and confidence in the reliability of the upgraded system has greatly improved for myself, management and the end-users.

Many of the lessons described here were learned using the timeless process of trial and error. I learned the importance of using Access Controls for preventing users from viewing or modifying information they are not privilege to. In applying security to my systems I learned there is a trade-off between cost and convenience. Security controls must be convenient enough for employees to use without being burdensome and cost effective enough for senior management to fund. There will always be a gap between what a computer system can afford to check, both in processing time and money, and what we would like to enforce.

An additional benefit of this project was that I developed the disaster recovery and backup procedures for SQL server. Two final thoughts that I mentioned earlier but are worth mentioning again – First; document the recovery steps with sufficient detail that anyone, not just the author, can perform the recovery and second; test your recovery strategy prior to a failure. Time is of the essence during recovery and testing reduces stress and recovery time. It's not a question of if; but when you will be required to recover your database.

My hope is that following the suggestions in this document allows a system administrator to sleep soundly at night knowing that they have a secure, documented system that isn't being comprised. Also remember that each bullet point in any best practices document must be evaluated for cost versus benefit as well as support your organization's mission statement, policies and goals.

Biblography

[1] Eric Cole. SANS Security Essentials with CISSP CBK Volume 1 and 2. 2.1 ed. : The SANS Institute, 2003.

[2] Ruthberg, Zella G., and Harold F. Tipton, ed. Handbook of Information Security Management. Boston: Auerbach Publications, 1993.

[3] Xiong, Man. "Microsoft SQL Server 2000 Scalability Project – Server Consolidation." March 2002. Microsoft Corporation and Dell Corporation. 5 Mar 2004. <http://www.dell.com/downloads/global/solutions/ServerConsolidation1.doc>.

[4] "SQL Server Architecture - Maximum Capacity Specifications." January 2002. Microsoft. 27 February 2004. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_ts_8dbn.asp>.

[5] Talmage, Ron. "Log Shipping in SQL Server 2000, Part 1 ." December 2001. SQL Server Magazine. 27 Feb 2004. <http://www.winnetmag.com/SQLServer/Article/ArticleID/23056/23056.html>.

[6] Rapoza, Jim. "SQL Server 2000 systems still unsecure." eWeek January 12 2004: 50.

[7] "Microsoft SQL Server: 10 Steps to Help Secure SQL Server 2000." 28 June 2003. Microsoft Corporation. 2 January 2004. <http://www.microsoft.com/sql/techinfo/administration/2000/security/securingsqlserver.asp>.

[8] "Microsoft Security Bulletin MS02-035." 10 July 2002. Microsoft Security Bulletin. 5 May 2003. <http://www.microsoft.com/technet/security/bulletin/ms02-035.mspx>.

[9] "Chapter 11 - Database Backup and Recovery." 19 July 2001. Microsoft Corporation. 28 April 2003. http://www.microsoft.com/technet/prodtechnol/sql/2000/books/c11ppcsq.mspx>.

[10] Mundie, Craig. "Trustworthy Computing." October 2002. Microsoft Corporation. 29 March 2004. <http://download.microsoft.com/download/a/f/2/af22fd56-7f19-47aa-8167-4b1d73cd3c57/twc_mundie.doc>.

[11] Anderson, Ross. "`Trusted Computing' Frequently Asked Questions." August 2003. University of Cambridge Computer Laboratory. 27 Mar 2004. < http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html>.

Last Updated: July 27th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS