# Global Information Assurance Certification Paper

# *Exploring Database Backups*

**GIAC Security Essentials Certification Practical**

*Diane Gregory*

*Version 1.4*

*Submitted April 4, 2003*

## Introduction

Database servers often include sensitive information from business partners and customers. Databases are not usually subjected to the same level of security as operating systems and networks. Databases are vulnerable to security attacks because of complexity, insecure password usage, improper configurations, and unrecognized system backdoors. However, database security is important because it is used to protect the organization's sensitive information and digital assets. Most of the organization digital assets are stored in off-the-shelf relational database products; it is necessary to protect access to the sensitive information and digital assets. As an organization begin to move its data online in the form of database storage, awareness of database security becomes pivotal and the need for it will continue to grow. This paper will discuss database security vulnerabilities, various methods to secure a database, various backup methods, how to manage backups, using passwords to protect backups, various backup media, securing backups, testing backups, and disaster recovery options.

## Database security vulnerabilities

Traditional database security only focus on user accounts, roles, and operating permission on specified database objects. These objects include access to tables and store procedures. There are several security vulnerabilities associated with database: risk associated with vendor-supplied software, risk associated with administration, and risk associated with user activity. Bugs, missing operating systems patches, vulnerable services and insecure choices for default implementations and configurations are risk associated with vendor-supplied software. To prevent risk associated with vendor-supplied software, an organization must stay current with the patch levels and avoid using default configurations. An organization must use the security options available correctly to avoid misuse of default settings and inappropriate granting of privileges to user who are not authorized to change system configurations. Passwords should be hardened to prevent inappropriate access to critical data. The above vulnerabilities need to be taken into consideration when securing database servers. Other vulnerabilities that may exist in critical database servers include lack of security feature maturity, database password management, operating systems back doors, and auditing. Built-in database security features need to handle issues such as account lockout, password expiration, and login hour restriction. An account should be lockout after three failed attempts. Password should expire after 90 days; users should receive

a message alerting them to change their password. The alert message should be displayed prior to the expiration date; letting the user know when the password will expire. Login restriction can be set up to prevent users from logging into the system after normal business hours. Managing database passwords can be a hassle because there is no mechanism to ensure that individual users are choosing strong passwords. Implementing a policy requiring users to choose strong password and supply them with examples of strong passwords is a way to ensure that users are using hard to guess passwords. Most database systems have powerful features that are convenient for Database Administrators (DBA); these powerful features can provide potential backdoors into a database server's host operating system. If an intruder compromise a Sybase or SQL Server, the intruder can gain system rights to the underlying operating system by the use of extended store procedures. The extended store procedure *xp_cmdshell* is an operating system command that allows a user the ability to run a command prompt at the server console. Auditing can be used to provide early warning signs that intruders are compromising the security of specific database servers and will provide a valuable clue for detecting and repairing any damage. However, that auditing system feature can only provide valuable security and alerting information if it is being properly used or configured. Database should be monitored regularly in order to detect and respond to security vulnerabilities for database severs. As critical business systems are deployed, security professionals, auditors, and DBA's need to be aware of database security.

**Securing Backups**

There are several methods used to secure database backups: encryption, physical security, and validation. An organization should use a backup system that encrypts the data in the event that the backups fall into the wrong hands. This will make it difficult for the theft to decrypt the data. If for some reason the tape or disk is missing, you should destroy the encryption key and create a new key. This will prevent the individual from finding the key and decrypting your backup. When encrypting backups you need to be careful because there are two types of failures that can occur. First, you might assume that your backups are secure when in fact they aren't. Second, You must make sure that the people who know those keys know how to decrypt it. It is your responsibility to know why you are encrypting the backups and who is able to access them and where the keys are stored. This information will be needed to access the data stored on the backups. But keep in mind that backups exist to make data accessible, whereas, encryption exists to make data inaccessible. The physical security of database backups should to be considered when it comes to securing the database backups. Physical security means that the database backups are stored in a safe and secured place. Safe from physical damage (i.e. fire) and secure for theft, misuse or just plain carelessness. Control physical access to the backup drives to prevent them from being stolen or tampered with. Validation is a method used to secure backups. Validation is all about ensuring that the data you think is on the backup media exist and that you are able to actually load the data from the backup onto the machine. You will need to confirm that you can restore from the backup media. Securing backups should be part of the overall security plan at your organization. An off-site storage for your backups is

recommended. This will protect your database backups in the event a disaster occurs in the building where your database backups are housed. However, when deciding on where to store your database backups, your must first choose an offsite facility that guarantees security of your backups and is open 24 hours, seven days a week. This is important because the facility where you store your backups should be available in the event of a disaster or system failure and you need the backups immediately to recover the system.

**Backup Methods**

There are different types of backup methods: full, differential, incremental, and disk imaging. The various backup methods are very useful. However, an organization will have to weigh the cost and benefit of the various methods when scheduling backups of the organization's data. The various database backup mechanisms have advantages and disadvantages, but it is up to the organization to determine which methods best suits the overall organizations need.

*Full backups*
A full backup is a copy of the entire database (e.g. store procedure, tables, views, permissions, etc.). Full backups are performed without having to take the database offline. Full backups utilize a large amount of system resources and may impact the database response time. Full backups and incremental backups will need to be stored together in the event that you have to use both of them. If a tape is lost or becomes corrupted, you will be unable to perform a full restore of the database. Full backups include all of the files stored on your drive. When you perform a full backup, you will not have to search through several tapes to find the files you need because a full backup will contain all the files you need to restore your database in the event of a system failure or attack. A great amount of time is involved in performing a full backup. But, the overall benefit is rewarding when you need to restore your organization's critical data. A full backup is used in the event of a catastrophic failure to restore a system to operational status. A Database Administrator performs a full backup of the database. A full backup of your databases should be performed at least on a monthly or weekly basis.

*Differential Backups*
Differential backups are used to limit the amount of time required to perform a full backup. A differential backup is a copy of the data that has changed since you performed the last full backup. Differential backups utilize fewer resources and do significantly impact database performance. Early file versions are replaced when doing a differential backup. Differential backups capture cumulative changes; therefore, you will need at least two tapes to perform a full system restore: The last full backup and the last differential backup tape. This will include the last full backup and the most recent differential backup. Differential backups provide more efficient restores in the event of a system failure or disaster. There are several disadvantages associated with differential backups. Data backed up following the last full backup will get larger and larger each day. Example, if the Full backup was done on Friday then Wednesday's Differential

backup would have the data that was backed up on the Monday tape and on Tuesday's tape plus whatever was changed or created on Wednesday. Redundant backups. Each day's backup would store much of the same information plus the latest information added or created since the last Full Backup.[1] Differential backups can speed up the time it takes to recovery or restore your database and transaction log backups.

*Incremental Backups*

Incremental backup is a method used to backup only the files that have changed or been added to the system since doing the last backup. Increment backups run quicker because it only captures the data that has changed. Incremental backups should be performed each of the remaining days of the week that is followed by a full backup. In order to successfully restore a system from full loss, you have to restore from the full backup tape, then restore from every incremental backup that has been made in the order they were created. Incremental backups require less data store space and is faster since you are only backing up the file that has been modified since you perform the last backup. Incremental backups require multiple tapes in order to restore. The files may be spread over all of the tapes used since your last full backup. It may be possible that you will have to search several tapes in order to find the file you need to restore. Increment backups should be performed each of the remaining days of the week following a full backup. To properly restore a system from full loss, you must first restore from the full backup tape, then restore from every incremental backup that was made in the order in which they were made.[2] Your full backup and increment backup tapes should be stored together in the same case just in case you need them to restore a system. A full restore cannot be performed if tapes are missing or are corrupted.

*Disk image backups*

A disk image backup is a copy of the drive or partition that hold Windows (usually the C drive). This type of backup is a copy of the drive itself, including the boot sector. A backup can be stored on CD and/or another partition. If Windows becomes corrupted, a disk image backup can be used to restore the entire reconfiguration; you would not have to reinstall programs, drivers, etc. For example, you can delete "C" drive and replace it with the disk image backup. Drive Imaging and Norton Ghost are the two possible programs that can make drive imaging backups. Drive Image is relatively easy to use and has intuitive interface and numerous options for making image files from any Windows partition. A disk image can be compressed, be password protected, and can span multiple files if required. Norton Ghost is not so easy to use and has less intuitive interface. Ghost is mainly designed for large-scale workstation cloning. Ghost is cheaper if it is purchased as part of systems works.

## ***Managing backups***

[1] http://www.seagate.com/support/kb/tape/4062.html

[2] SANS Institutes, "SANS Security Essential V: Windows Security

Backups should be managed carefully in order to ensure that it could be used to restore your system when needed. A backup will contain the descriptive text that you provided at the time the backup was created. It will also include the expiration information. This information is used to identify a backup, determine when the backup can be safely overwritten, and identify all the backups on a backup medium to determine which backup needs to be restored. When working with backups, they should be kept in a secure location at an offsite facility. Old backups should be stored for a designated amount of time in the event that the most recent backup is damaged, destroyed, or lost. However, it is up to your organization to determine how long to keep these older backups. A system should be established to overwrite backups by reusing the oldest backups first. Expiration date should be used on backups to prevent premature overwriting. Backup media should be properly labeled in order to prevent overwriting of critical backups. This allows you to identify the data stored on the backup media or specific backup set easily.

**Using Password to protect your backups**

Microsoft® SQL Server™ 2000 supports password protection for backup media and backup sets. Passwords are not required to perform backup operations, but they provide an added level of security. You can use them in addition to using SQL Server security roles.[3] Using password protection helps safeguard your backups against unauthorized restoration of databases, unauthorized appends to the media, and unintentional overwriting of the media. You can use passwords to protect both your media and backups. You can set a password on a media; once a password is set, all data stored on the media is protected. You set the media password when the media header is written; Once protected it cannot be altered. You will need to supply the password when using append or restore operations. The media can only be used for SQL Server backup and restore operations. A Microsoft Windows NT® 4.0 or Windows® 2000 backup will not be able to share the media. Setting a password on a backup will only protect the particular backup set. Different backup set passwords can be used for each backup set on the media. A backup password is set when the backup is written to the media. Once a password has been defined for the backup set, you will have to supply it before you can perform any restore of that backup set.

**Backup Media**

There are several types of backup media: floppy, zip disk, CD, tapes, and hard disk. There are both advantages and disadvantages of using the different backup medias. Floppies and zip disk are easy to use, highly portable, inexpensive, and is a random access media. The disadvantage is limited capacity and reliability. CD is highly portable, durable, reliable, inexpensive, and is a random access media. CDs are more difficult to use and have limited capacity. There are two types of CDS: CDR/CDRW. CDRWs have to be blanked before reusing them and they are time consuming. Tapes are cheap and have high capacity. There are two types of tapes: DAT/DLT. The drives

[3] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_bkprst_3ulq.asp

that support the tapes are expensive. Tapes have sequential access media and the write and recovery time is longer with tapes. Hard disk are inexpensive, fast, is a random access media, and easy. The disadvantage of hard disk is that it is more expensive and you have to determine where to back the disk up.

**Testing Backups**

Testing backups ensures your ability to recover your data in the event of a failure or disaster. Your backups should be very reliable, in the event that you accidentally erase everything on your hard drive. If your backups are reliable, you will be able to restore your system. There are several discrepancies that can occur in a tape backup. Check to make sure that your tape is not worn-out. If your tape is worn-out, data cannot be recorded properly. A tape can be damage when trying to load data. So be careful when you have to load the data from the tape to the computer. You will not know if your backup is successful unless you test. There is backup software that allows you to compare the data stored on the backup with the original data. Automated test should be run after every backup to ensure the validity of the storage media. Using manual testing can be used to test your backups. Manual testing involves restoring the backup onto a test or spare computer. Try to open the files or execute the application to confirm that the data has not been corrupted. This type of test will ensure your ability to restore your important files as needed. There are several reasons to test your backups: backup failure, backup software incompatibility, compression, and incomplete data. An organization can run the backups for months without realizing that the backups don't work properly. The only way to found out is if a system goes down and the backups are needed to recover the system. In the process of recovery, you noticed that the files are missing or corrupted. To avoid the above scenario, an organization must test the backups to ensure they are working properly. Often an organization will need to upgrade the system, in doing so previous backups may become obsolete; therefore, you need to test the backups after you have applied changes to the system to ensure that the previous backups work properly. You do not want to wait until you need to recover the system to find out that your backups are obsolete. Backup software often uses compression features in order to store more data on a smaller storage device. The compression features may compromise the integrity of the data stored – thus making it incompatible. Make sure that data stored on backups are complete. Make sure that all directories are included on the backups. You don't want to wait until a system failure occurs to realize that your backups do not include all of the directories.

**Disaster Recovery Options**

Disaster Recovery is the ability to respond to an interruption in services. It includes a disaster recovery plan that will be used to restore an organization critical business functions. They're a two ways to handle recovery in an event of a disaster: install-and-go and turn key. Install-and-go is an option used by an organization that normally contracts with a vendor to deliver equipment within a specified time to a designated site. This method is used in a client server environment where a quick return to operation is not imperative. The turnkey option is used by an organization that encompasses the

contracting of firm that will provide a client for recovery. The backup media from the client company is used to restore the data and client staff will run the system from a remote location. This method is best suited for mainframe systems environment. There are many recovery models used to recovery data during a disaster. In Microsoft SQL Server 2000, there are three recovery models available: simple, full, and bulk-logged.

*Simple recovery model*
Simply recovery requires the least administration. In the simple recovery model, the database is recovered up to the most recent full database or differential backup. A database can't be restored to the point of failure or to a specific point in time. Since transaction log backups are not used, a minimum amount of transaction log spaced is used. Transaction logs are not backed up. Instead the transaction logs are frequently truncated. This is an automated process. Simple recovery model processes bulk operations the fastest.

*Full recovery model*
In the full recovery model, the database is recovered up to the point of failure or to a specific point in time. In order to guarantee this degree of recoverability, all operations, including bulk operations such as select into, create index, and bulk loading data, are fully logged. The full recovery model use database backups and transaction log backups to provide protection against media failure. The benefits of full recovery include: no work lost due to a lost or damaged data file and can recovery to any point in time. However, if the log is damaged, changes made since the most recent log backup must be redone. SQL Server logs all operation; rows inserted using bulk copy program (bcp) or bulk insert operation is written to the transaction log. Bulk insert operations will be much slower because every modification must be logged. The full recovery model will provide the better option in the event that a data file is corrupted. While the database is in full recovery mode, transaction will reside in the transaction log until it is backed up. The space from the older transactions will become free once the database is back online. New transaction will be logged on the space.

*Bulk-logged recovery*
Bulk-logged recovery allows bulk-logged operations. Bulk-logged recovery offer the following benefits: high-performance bulk copy operations and minimal log space is used by bulk operations. Recovery can be made to the end of a transaction log backup when the log backup contains bulk changes. Point-in-time recovery is not supported in bulk-logged recovery. If the log is damaged, or bulk operations occurred since the most recent log backup, changed made would have to be redone. Otherwise, no work will be lost. Bulk-logged recovery uses the least amount of log space for bulk operations such as bulk insert, bulk copy program, create index, etc. Log space can be smaller if you are using bulk operations. Bulk operations run faster because only the operation that has occurred needs to be recorded, not every modification to the database.

Full recovery and Bulk-logged recovery will provide the greatest protection of data. Both models rely on the transaction log to provide full recoverability and to prevent work

loss in the event of a system failure. But keep in mind, that you can switch between full and bulk-logged recovery model relatively easily.

**Conclusion**

While securing your database backups is very essential to everyday life. Databases are not usually subjected to the same level of security as operating systems and networks. Protecting your data by any means necessary means protecting the servers that backup that data. Database backups are very important to your organization. Therefore, it is pivotal that you protect them. Database backups are needed to restore your system in the event of a failure or disaster. As such, you need to take the necessary precautions to secure your backups. This can be achieved by encrypting your backups and or storing them off-site. Of the various backup method discuss, it is up to the organization to determine which one best suit's the overall business needs. Managing your backups becomes important because you will need them to restore your system. Keeping them in a secured location and storage of older backups will aid in your recovery process in the event of a system failure or disaster. Using a password to protect your backup is an added level of security. Testing your backups periodically is needed to ensure that you are able to recover successfully during a major IT outage or system failure. All of the above mentioned factors could be considered when you need to protect your database backups.

References:


1. http://www.hfc.org.uk/publications/98july/0797backup.htm
2. http://pages.friendlycity.net/~sdove/techtalk/storage.htm
3. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_bkprst_1jzn.asp
4. Protecting Your Database: http://www.oracle.com/oramag/oracle/00-May/index.html?o30sec.html
5. Sans Institute Track 1 – SANS Security Essentials 1.5 SANS Security Essentials V: Windows Security.
6. http://www.jsware.net/jsware/diski.html
7. http://www-personal.umd.umich.edu/~kordus/options.htm
8. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_bkprst_1jzn.asp
9. http://www.seagate.com/support/kb/tape/4062.html
10. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_bkprst_3ulq.asp
11. Delaney, Kalen. “Database Recovery Models: SQL Server 2000 recovery models give you backup-and-restore flexibility. June 2000. URL: http://www.sqlmag.com/Articles/Index.cfm?ArticleID=8551 (21 April 2003)