International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V68I1P108

Volume 68 Issue 1, 35-37, January 2020

Original Article

Information Security Least Privilege Requirement Analysis for SQL Database Backups

Chirag Goel

Sr. Database Administrator, Illinois Institute of Technology, Chicago, USA

Received Date: 15 November 2019
Revised Date: 10 January 2020
Accepted Date: 16 January 2020

***Abstract* —** *Security is getting more vigilant and granular in information technology. To protect enterprise data proper implementation of security and vigilance in access if necessary. We are going to analyze the least privilege needed to perform backup and restore in SQL Server. We will start with Microsoft documented server and database roles and then proceed with the analysis and outcome of each permission we provision foran account. We will discussthe best techniques to perform the backup operation and without exposing data with excessive permissions.*

***Keywords* —** *Backup, privilege, restore, sysadmin, database management system, server roles, database roles.*

I. INTRODUCTION

Information securityrun by the principle of least privileges. We ran into a scenario where we need to give the least possible rights to a service account to perform a backup of the database from Production and restore it in lower environments.

II. PROBLEM STATEMENT AND DIAGNOSIS

We are working on the scenario where a backup of the Production database required to be restored on the Non-Prod server. Traditionally database administrators used a service account running SQLServer service to perform this operation. We will call the service account running SQL server service as **SA** from now on this for this analysis. SA has sysadmin permissions on SQL Server and can perform any operation. This increasesthe surface area of penetration by hackers as in scenario when you use a third-party system to automate the backups in your environment account used to run Backup will be shared among many teams that expose risk for getting account compromised.
Let's discuss some of the best practices to perform this operation.

1. Create a service account for backup operations in one environment. Let's call it Prod_svc
2. Create a Shared storage location to store the backups. This location should be specifically used for backups and should not be used with other use. Let's call this location as \\sharedLocation\.
3. Create a service account for non-Prod or other servers and let's call it NonProd_svc.

We started with below permissions

Prod_svc

- SQLAgentUserRole
- BackupOperator Role
- Read, Write and delete permissions on\\sharedLocation\

NonProd_svc

- DB_creator
- SQLAgentUser Role
- Read, Write and delete permissions on \\sharedLocation\

Here are details of each of the above role:

1. *SQLAgentUserRole:*
SQLAgentUserRole is the least privileged of the SQL Server Agent fixed database roles. It has permissions on only operators, local jobs, and job schedules. Members of SQLAgentUserRole have permissions on only local jobs and job schedules that they own.[1]

2. BackupOperator Role
Members of the db_backupoperator fixed database role can back up the database.[2]

3. DBcreator
Members of the dbcreator fixed server role can create, alter, drop, and restore any database.[3]

The above permissions are the minimum required access set needed for backup and restore operation to happen.

Below is the setup we have for this to happen with automation. We used the SQL server agent to automate these.

1. Create a job on Prod server:
   a. Backup of Prod Database
   b. Send notification
2. Job on Non-Prod Server
   a. Restore Database
   b. Delete old backup file

c. Rename current backup file to _old(This is needed so that next day backup can run successfully)

The job ran with the below error.

III. ERROR STATEMENT I

The SELECT permission was denied on the object 'sysjobs', database 'msdb', schema 'dbo'.
The error indicatesselect permission is needed on dbo.sysobj in msdb database. So, we requested below permissions to resolve the above error. Recommend way to assign permission is through groups so that database administrators don't have to deal with removing individually from servers and access revocation can be done through Active Directory.

IV. Table I
Permissions provisioned to resolve error statement I

Account	Permissions
Prod_svc	Prod-Msdb-Read
NonProd_svc	NonProd-msdb-read

Once we added supposed to be needed permission, we tried running the job again which gave below error.

V. ERROR STATEMENT II

Error 14393, Level 16, State 1, Procedure sp_start_job, Line 42, Message: Only owner of a job or members of role sysadmin or SQLAgentOperatorRole can start and stop the job.

Explanation**:**The above error indicatesthat members of SQLAgentOperator role can start and stop the job. So, we requested additional permissions.

VI. Table II
Permissions provisioned to resolve error statement II

Account	Permissions
Prod_svc	Prod- instance-SQLAgentOperatorRole
NonProd_svc	NonProd- instance-SQLAgentOperatorRole

Once the above permissions have been added we tried running the job again. And here is the error we got.

VII. ERROR STATEMENT III

Now we got **access denied** on the Shared location. We did add permissions to the service account running the backup for the shared location but it failed with access denied to file location. Even though the service account running job has access but service account running SQL Server service doesn't have access to the shared location. Logically reviewing when we are running a job with some particular service account then permissions of that account should be sufficient enough to run the job but in SQL Server this is not the case. SQL Server expects service account running its services should have the complete access on the file system to which it interacts which includesthe location where data, log and tempdbfiles are stored. So, we provisioned below permissions for the SQL Server service account.

VIII. Table III
Permissions provisioned to resolve error statement III

Account	Permission
Prod-SQL Server Service Account	SharedLocation-Read
Prod-SQL Server Service Account	SharedLocation-Write
Prod-SQL Server Service Account	SharedLocation-Delete
NonProd-SQL Server Service Account	SharedLocation-Read
NonProd-SQL Server Service Account	SharedLocation-Write
NonProd-SQl Server Service Account	SharedLocation-Delete

IX. ERROR STATEMENT IV

Error 229, Level 14, State 5, Procedure xp_sqlagent_enum_jobs, Line 1, Message: The EXECUTE permission was denied on the object 'xp_sqlagent_enum_jobs', database 'mssqlsystemresource', schema 'sys'.
Explanation: Errorindicate execute permission is denied on the 'xp_sqlagent_enum_jobs' object which is in system database ResourceDB. As resourcedb is hidden in SQL Server, we need to find a way to reach the true error. Here we provide read and execute in master which give access to account to read objects in master database and execute the stored procedure mentioned in the error above. To resolve this error, we provisioned the below access.

X. Table IV
Permissions provisioned to resolve error statement IV

Account	Permission
Prod_svc	Prod- Master-Read
Prod_svc	Prod Master-Execute
NonProd_svc	NonProd- Master-Read
NonProd_svc	NonProd- Master-Execute

Finally, after adding these sets of permissions both the jobs ran fine.

XI. CONCLUSION

As we are designing new principles in IT so we need to design a new principle in Security. Security is the base of sound and secure data. By the above approach, we reduced the exposure of database server and database itself by creating distinct accounts for different servers and by providing the least privilege. This research can be implemented with SQL Servers in enterprise environments.This will benefit the enterprise information

security approach and provide the procedure to minimize the use of sysadmin privileges. Below is the final set of minimum permissions needed to perform backup on the server and restore it on a different server along with automation.

XII. Table V
Concluded set of Permissions

Account	Permission
Prod_svc	SQLAgentUserRole
Prod_svc	BackupOperator Role
Prod_svc	Read, Write and delete permissions on \\sharedLocation\
Prod_svc	Prod-Msdb-Read
Prod_svc	Prod- instance-SQLAgentOperatorRole
Prod_svc	Prod- Master-Read
Prod_svc	Prod Master-Execute
NonProd_svc	DB_creator
NonProd_svc	SQLAgentUser Role
NonProd_svc	Read, Write and delete permissions on \\sharedLocation\
NonProd_svc	NonProd-msdb-read
NonProd_svc	NonProd- instance-SQLAgentOperatorRole
NonProd_svc	NonProd- Master-Read
NonProd_svc	NonProd- Master-Execute
Prod-SQL Server Service Account	Read, Write and delete permissions on \\sharedLocation\
NonProd-SQL Server Service Account	Read, Write and delete permissions on \\sharedLocation\

REFERENCES

[1] SQL server agent fixed database roles, https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent-fixed-database-roles?view=sql-server-ver15

[2] Database level roles, https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-ver15

[3] Server level roles, https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-ver15

[4] Li, C., Wang, S. A Data Model for Supporting On-Line Analytical Processing, Proc. of the 5th International Conference on Information and Knowledge Management, 1996, pp. 81-88.

[5] Pedersen, D., Riis, K., Pedersen, T.B. (2002), A Powerful and SQL-Compatible Data Model and Query Language For OLAP, Proc. of the 13th Australasian Database Conference (ADC2002), Melbourne, Australia

[6] Wang, H., Zaniolo, C, Using SQL to Build New Aggregates and Extenders for Object Relational Systems, Proc. of the 26th VLDB Conference, Cairo, Egypt,2000.

[7] Plattner, H., A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database, SIGMOD'09, Providence, Rhode Island, USA,2009.

[8] Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D., Zdonik, S. , The Object-Oriented Database System Manifesto, Proc. Of the First International Conference on Deductive and Object-Oriented Databases, Kyoto, Japan, pp.223-240,1989.

[9] Zaniolo, C., Intelligent Databases: Old Challenges and New Opportunities, Journal of Intelligent Information Systems, 1, pp.271-292,1992.

[10] Connolly T, Begg C. Database system a practical approach to design, implementation, and management. 5th ed. Boston: Addison-Wesley; 2009.

[11] Chan, M.Y. and Cheung, S.C. Applying white box testing to database applications. CSTR, Hong Kong University of Science and Technology, HKUST-CS99-01. 1999.

[12] Chays D., Deng, Y., Frankl, P.G., Dan S., Vokolos, F.I. and Weyuker, E.J. An AGENDA for testing relational database applications. Software Testing, Verification, and Reliability. 14 17--44. 2004.