SQL Server running as local system and not a domain or local user

Running SQL Server as a Local System account, rather than under a domain or local user account, can pose several risks and limitations:

1. **Security Risks**: The Local System account has extensive privileges on the local machine, which means if SQL Server is compromised, the attacker could potentially gain control of the entire system. This high level of access increases the risk of both internal and external security breaches.

2. **Limited Network Access**: The Local System account does not have credentials outside of the local machine. This limits SQL Server's ability to interact with other servers on the network, such as for linked servers, remote backups, or file sharing. This can be a significant drawback in environments that rely on distributed or networked databases.

3. **Auditing and Accountability Issues**: Using a Local System account does not provide the same level of auditing and accountability as a domain or specific local user account. It becomes challenging to track and audit activities specific to SQL Server since actions appear to come from the system itself rather than a distinct account.

4. **Service Interactions**: SQL Server running under the Local System account might have unnecessary access to other system services and resources, which can be a security concern. Ideally, each service should operate with the minimum privileges necessary to perform its tasks.

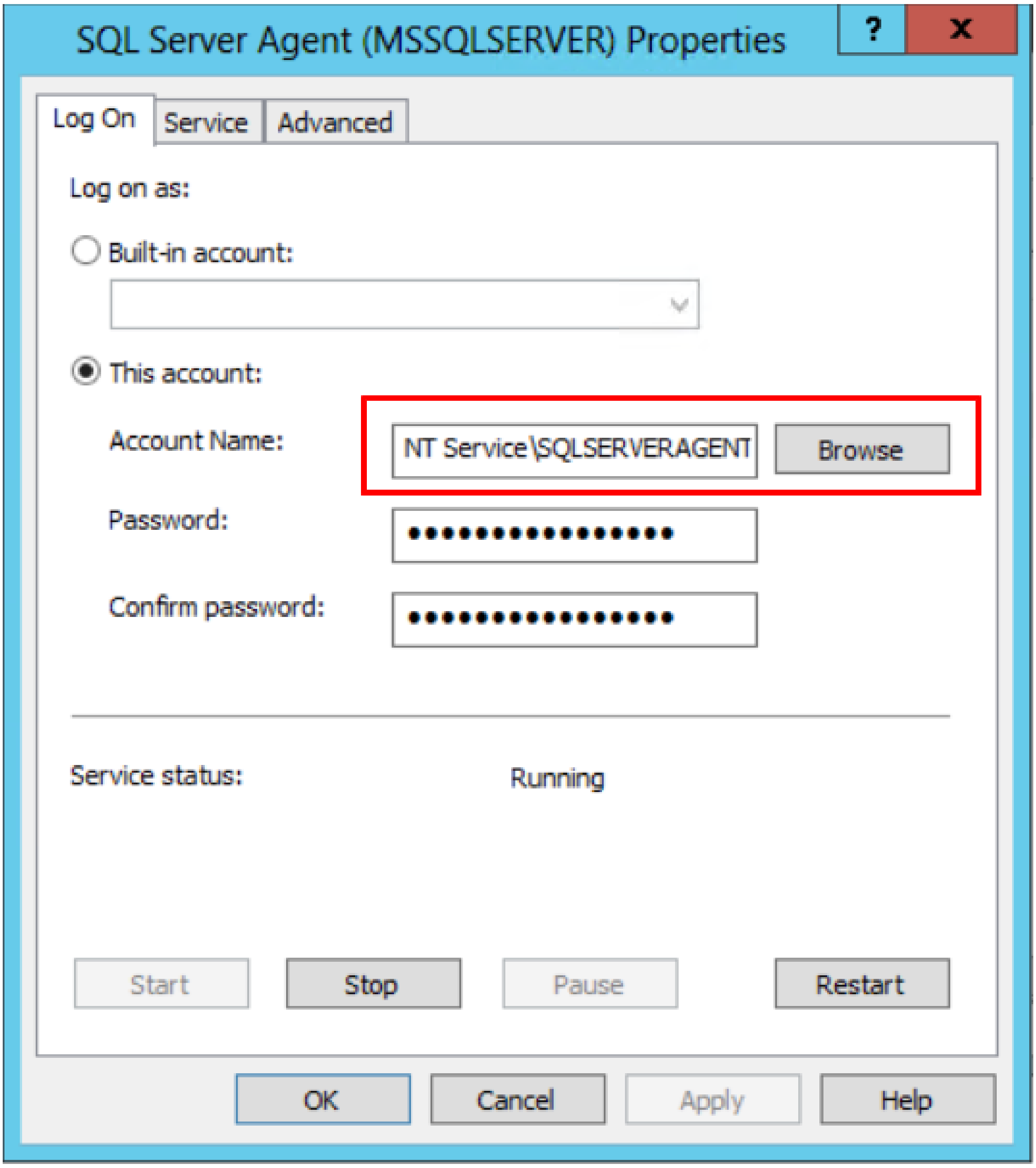
5. **Compliance and Best Practices**: Many compliance frameworks and security best practices advise against running services, especially those exposed to the network, under accounts with elevated privileges like Local System. Using a lower-privileged domain or local user account is often a compliance requirement.

6. **Configurational Limitations**: Certain SQL Server configurations, particularly those involving replication or certain integration services, may require a domain account due to their need for network access and permissions beyond the local machine.

While using the Local System account might be simpler in terms of configuration and management, it introduces significant security risks, reduces functionality, and can lead to compliance issues. It's generally recommended to use a domain or local user account with the specific permissions necessary for SQL Server's operation.

If your SQL Server needs to talk with other servers on your network a Domain User account should be used. Domain User Accounts also simplifies user management. Whichever account type is chosen it should be a minimally privileged account with as permissions as needed for the SQL Server to do its job.

The image below shows and example of the local system account, that could be changed to a local user or domain user.



SQL Server Agent (MSSQLSERVER) Properties

Log on as:

○ Built-in account:

● This account:

Account Name: NT Service\SQLSERVERAGENT

Password: ●●●●●●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●●●●●●

Service status: Running

Special considerations for SSRS

When changing the user that SQL Server Reporting Services runs as, you will want to back up the SSRS encryption key prior to making the change, then restore it after the change is made.

Changing the user account that SQL Server Reporting Services (SSRS) runs under may require you to restore the SSRS encryption key if the new account does not have the necessary permissions to access the encryption key.

The SSRS encryption key is used to encrypt sensitive data, such as connection strings and stored credentials, in the SSRS database. When you change the service account, the new account may not have access to the encryption keys that were created using the original account. In such cases, you would need to restore the encryption key to grant the new account access to the encrypted data.

To avoid any issues, follow these steps when changing the SSRS service account:

1. Backup the encryption key before making any changes to the service account. You can use the Reporting Services Configuration Manager to do this.

2. Change the service account in the Reporting Services Configuration Manager or by using SQL Server Configuration Manager.

3. If you encounter any issues related to the encryption key after changing the service account, restore the encryption key using the backup you created in step 1. This can also be done using the Reporting Services Configuration Manager.

By following these steps, you can ensure that the new service account has access to the encryption key and minimize the chances of issues related to encrypted data in SSRS.