INFORMATIQUE
TECHNIQUE

Expert

SQL Server 2014

Optimisez l'exploitation de vos bases de données et tirez parti des dernières nouveautés

Téléchargement
www.editions-eni.fr

Olivier MAITRE

eni Editions

Table des matières

Les éléments à télécharger sont disponibles à l'adresse suivante :
http://www.editions-eni.fr
Saisissez la référence de l'ouvrage **EI14SQLOP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1
Introduction

Chapitre 2
Les différents composants des moteurs

Chapitre 3
Le In-Memory

Chapitre 4
Les outils clients

Chapitre 5
Les bases de données

Chapitre 6
La sécurité

Chapitre 7
L'audit et les changements de données

Chapitre 8
La maintenance des bases de données

Chapitre 9
Les very large databases (VLDB)

Chapitre 10
Les performances

Chapitre 6
La sécurité

1. Introduction

La sécurité des données est un enjeu capital. Une base de données peut être amenée à stocker des données très sensibles, confidentielles.

L'implémentation de la sécurité des données au sein d'une base se décline à deux niveaux. Tout d'abord le premier niveau consiste à sécuriser les accès en définissant quels utilisateurs accèdent à quelles données, tout en contrôlant le type d'accès. C'est la mise en place des comptes utilisateurs et des autorisations.

Puis le second niveau consiste à sécuriser les données elles-mêmes afin de se prémunir contre toutes intrusions : c'est le cryptage.

Ce chapitre ne reprend pas les fondamentaux de l'implémentation des différents aspects de la sécurité sous SQL Server mais propose plutôt un focus sur certains enjeux critiques de la sécurité et sur les nouveautés introduites dans ce domaine par la version 2014 du moteur.

2. La sécurité des accès

La sécurité des accès consiste à définir d'une part la population disposant de l'autorisation de se connecter à l'instance et aux bases de données, et d'autre part les droits et donc les actions permises pour chaque utilisateur connecté. La première étape concerne donc également l'authentification. Une personne qui se connecte à une instance doit être authentifiée par l'instance. Cette section commence par faire un rappel des méthodes d'authentification proposées par le moteur avant de mettre en avant les nouvelles autorisations apportées par la version 2014.

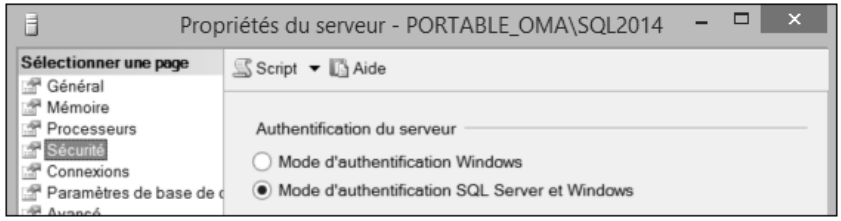
La sécurité des bases de données à relation contenant contenu et enfin la configuration de la surface d'exposition sont ensuite traitées.

2.1 Les modes d'authentification

Le moteur SQL Server propose deux modes d'authentification : le mode mixte, qui autorise les comptes SQL Server et les comptes Windows à se connecter à l'instance, et le mode Windows seulement qui comme son nom l'indique ne permet les connexions que par le biais de comptes Windows.

Le choix du mode d'authentification se fait durant l'installation du moteur et peut être changé par la suite.

- Pour changer le mode d'authentification d'une instance, depuis Management Studio, faites un clic droit sur l'instance et sélectionnez **Propriétés** puis choisissez le nœud **Sécurité** dans la fenêtre **Propriétés du serveur**. Choisissez le mode d'authentification souhaité puis cliquez sur **OK**. Un message vous avertit que l'instance doit être redémarrée pour que le changement soit effectif.



Changement du mode d'authentification d'une instance

2.1.1 L'authentification Windows

L'authentification Windows est une authentification intégrée qui récupère les informations du compte Windows de l'utilisateur pour se connecter, en s'appuyant sur l'architecture de domaine Active Directory. Il est impossible dans ce mode d'authentification de spécifier explicitement un compte et un mot de passe Windows. C'est forcément le compte de l'utilisateur courant qui est utilisé. C'est pour cela qu'on lui donne également le nom d'authentification intégrée.

Lorsque l'on choisit le mode d'authentification Windows, la connexion à l'instance via un compte SQL Server est impossible. Pendant l'installation, le compte "sa" (login SQL Server disposant de tous les droits, automatiquement créé lors de l'installation d'une instance) est désactivé et un mot de passe lui est attribué par le programme d'installation afin de sécuriser le compte.

NTLM vs Kerberos

Deux protocoles d'authentification peuvent être utilisés pour se connecter à une instance SQL Server : NTLM et Kerberos. NTLM est le protocole historique, basé sur un compte utilisateur et un mot de passe et utilisant un mécanisme de stimulation/réponse (challenge-response). Kerberos est plus sécurisé mais nécessite néanmoins certains prérequis. Si l'un des prérequis suivants n'est pas respecté, l'authentification Kerberos échoue et c'est NTLM qui est utilisé :

- Le client doit choisir le mode d'authentification Windows.
- Le client et le serveur SQL Server doivent se trouver dans le même domaine ou dans des domaines approuvés.
- Le SPN du service doit être créé sur le compte de démarrage du service SQL Server (voir le chapitre Les différents composants des moteurs - Les différents services pour en savoir plus sur les SPN). Lorsque le client demande à se connecter à un serveur SQL Server, il s'adresse au contrôleur de domaine et demande un accès au service SQL en l'identifiant par son SPN. Le contrôleur interroge alors le catalogue Active Directory pour récupérer le compte qui contient ce SPN, c'est-à-dire le compte de démarrage du service SQL Server. Et s'il n'en trouve pas, il renvoie le compte machine du serveur. Le client se présente alors avec un ticket d'authentification au service SQL. Ce dernier ne peut pas déchiffrer lorsqu'il démarre avec un compte autre que le compte machine (compte système local). Et dans ce cas l'authentification Kerberos échoue.
- Le serveur SQL doit être correctement enregistré dans le DNS car c'est le nom complet du serveur (FQDN) qui est utilisé dans le SPN.

Remarque

Le choix du protocole se fait automatiquement. Si les prérequis nécessaires à Kerberos sont respectés, c'est ce protocole qui est utilisé, sinon c'est NTLM sous certaines conditions.

Les groupes Windows

Dans la pratique, plutôt que d'ajouter directement les comptes Windows des utilisateurs dans l'instance SQL Server afin de leur donner les accès, il est préférable d'ajouter des groupes Windows, apportant une souplesse à l'utilisation. Ainsi, les droits peuvent être attribués à des groupes plutôt qu'à des utilisateurs individuels. En outre, pour autoriser un nouvel utilisateur ou supprimer les droits d'un utilisateur existant, il suffit de l'ajouter dans le bon groupe Windows, sans avoir à intervenir au niveau de l'instance SQL Server.

Remarque

Et depuis SQL Server 2012, il est (enfin !) possible d'affecter un schéma par défaut à un groupe Windows.

2.1.2 L'authentification mixte

L'authentification mixte comprend l'authentification Windows, en permettant en plus l'authentification SQL Server, c'est-à-dire l'authentification par le biais de comptes et mots de passe SQL Server. Lorsque l'on choisit ce mode d'authentification, la première chose à faire est de définir un mot de passe fort pour le compte "sa", le compte administrateur par défaut.

Remarque

Le compte "sa" est le compte historique de l'administrateur par défaut de toutes les instances SQL Server, quelle que soit la version. Bien connu, il est recommandé de lui affecter un mot de passe fort et de changer ce mot de passe régulièrement pour sécuriser le compte au maximum, voire de le désactiver.

Lorsque l'on se connecte dans ce mode d'authentification, il faut préciser le compte SQL Server et son mot de passe dans la chaîne de connexion, ce qui peut poser problème dans des scripts ou batchs qui contiennent alors ces informations de connexion en clair. Par ailleurs, ces informations par défaut transitent en clair sur les réseaux, et nécessitent dans certains cas la mise en place du cryptage des communications (voir le chapitre Les différents composants des moteurs - Les différents services).

Moins sécurisée que l'authentification Windows et donc non recommandée, l'authentification SQL Server reste néanmoins très utilisée car elle apporte les avantages suivants :

– La connexion Windows entre domaines non approuvés est impossible. La connexion SQL Server peut être utilisée dans ce cas.
– L'authentification Windows permet également la connexion à une instance SQL Server à partir de clients installés sur des plateformes non Windows, pour lesquelles l'utilisation de compte Windows est impossible (client Linux par exemple).
– Elle permet enfin la prise en charge d'applications ne supportant pas l'authentification Windows.

2.2 Les rôles et les privilèges

Les droits sont donnés aux utilisateurs par le biais de rôles et de privilèges déclinés au niveau serveur (instance) ou base de données. Voyons les nouveautés dans ce domaine apportées par la version 2014 du moteur SQL Server.

2.2.1 Le niveau serveur

Trois nouvelles autorisations de niveau serveur apparaissent avec SQL Server 2014.

SELECT ALL USER SECURABLES

Cette autorisation permet au compte qui en bénéficie d'accéder à l'ensemble des données des bases de données qui lui sont accessibles. Jusqu'à présent, il fallait explicitement attribuer les droits de lecture à l'utilisateur (user) associé au compte (login) dans chacune des bases pour lui en donner l'accès. Il suffit dorénavant de donner le privilège **SELECT ALL USER SECURABLES** au login.

Mais ce nouveau privilège offre également une possibilité qui était attendue de longue date. Considéré non plus comme une autorisation mais comme une restriction, et associé à l'autorisation **CONTROL SERVER**, ce privilège peut donner la possibilité de créer des comptes d'administrateur de bases de données ayant tous les droits, sauf celui de visualiser les données elles-mêmes. Les données les plus confidentielles ou sensibles sont alors soumises à un meilleur contrôle : même les administrateurs de bases de données n'y ont pas accès.

▶ Pour créer un compte ayant tous les droits sauf celui de consulter les données, créez tout d'abord un login (Windows ou SQL Server) au niveau de l'instance en utilisant la commande suivante :

```
CREATE LOGIN test_dba WITH PASSWORD = 'T35t_db#';
```

Puis mettez en place les droits sur le login en vous appuyant sur les privilèges **CONTROL SERVER** et **SELECT ALL USER SECURABLES** :

```
GRANT CONTROL SERVER TO test_dba;
GO
DENY SELECT ALL USER SECURABLES TO test_dba;
GO
```

Ainsi, il est dorénavant possible de créer un rôle de serveur fixe disposant des droits mentionnés précédemment, puis de peupler ce rôle avec les comptes des administrateurs de bases de données. On a donc la possibilité de créer ainsi le rôle DBA, capable de toutes les actions d'administration mais ne pouvant pas accéder aux données elles-mêmes.

Attention néanmoins dans ce dernier cas : le compte d'administrateur dispose effectivement de tous les droits d'administration, y compris celui... de donner les droits de lecture des données à un autre compte ou rôle que lui-même ! Rien ne l'empêche en effet d'octroyer le droit **SELECT ALL USER SECURABLE** au rôle dont il fait partie, ce qu'il ne peut pas faire sur son propre compte.

Ce levier est donc un moyen d'éviter que l'administrateur de bases de données n'accède à des données sensibles de manière intempestive, mais ne l'interdit pas dans l'absolu. Un système d'audit des accès ou des changements de privilèges viendra donc compléter l'implémentation de ces droits dans ce contexte (voir le chapitre L'audit et les changements de données).

Créez deux comptes SQL Server dba1 et dba2 ainsi qu'un rôle de serveur fixe db_admins. Dans un cas, vous allez donner les autorisations directement sur le compte dba1, dans le deuxième cas, vous affecterez les autorisations au rôle db_admins et ajouterez le compte dba2 en tant que membre de ce rôle :

```
Use master
GO
--Création du premier utilisateur dba1 et mise en place des droits
CREATE LOGIN dba1 WITH PASSWORD = 'dba', CHECK_POLICY=OFF;
GO
GRANT CONTROL SERVER TO dba1;
GO
DENY SELECT ALL USER SECURABLES TO dba1;
GO

--Création d'un rôle de serveur fixe et affectation des droits
CREATE SERVER ROLE db_admins;
GO
GRANT CONTROL SERVER TO db_admins;
GO
DENY SELECT ALL USER SECURABLES TO db_admins;
GO

--Création du deuxième utilisateur et ajout dans le rôle
CREATE LOGIN dba2 WITH PASSWORD = 'dba', CHECK_POLICY=OFF;
GO
ALTER SERVER ROLE db_admins ADD MEMBER dba2
GO
```

```
00 %
Résultats
Commande(s) réussie(s).
```

Création d'un login et d'un rôle type administrateur de bases de données