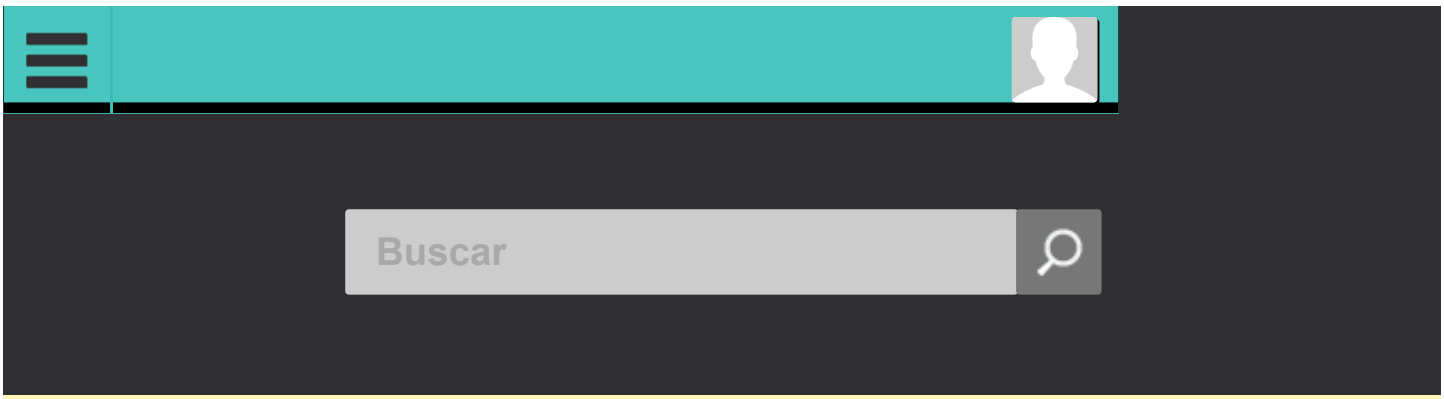

SQL Magazine 134 - Índice

Segurança de banco de dados no SQL Server e Oracle

Este artigo apresenta uma série de para aprimorar a qualidade de dados no SQL Server e para o Oracle.

Fique por dentro

Este artigo descreve técnicas importantes que podem ajudar a melhorar a

segurança dos bancos de dados SQL Server e Oracle. Estes exemplos serão úteis àqueles que pretendem administrar ou dar suporte em bancos de dados atuando e mantendo um ambiente de produção dentro dos padrões de uma segurança eficiente. Serão discutidos itens como políticas de acessos, sistema operacional, permissões, conexões, roles e acesso multicamadas.

Nos ambientes corporativos, o conceito de segurança vai muito além de implementar uma senha forte. Em empresas de grande porte, esta área tem a função de definir regras em todas as esferas da organização, incluindo conceitos de segurança física que devem ser seguidos à risca.

Quando o assunto é segurança, uma palavra é fundamental: confiança. Quando um processo de segurança é fragilizado, a confiança pode ser perdida, e isso pode causar prejuízos à organização. Políticas de trocas de senha devem ser pensadas e adequadas a cada ambiente. Ferramentas podem ajudar neste processo. Firewalls, anti-spam, políticas de acessos, baselines devem ser criados para que no final o resultado seja um ambiente confiável.

No contexto de banco de dados, os SGBDs SQL Server e Oracle não deixam a desejar neste aspecto. Os dois são excelentes ferramentas que se forem bem configuradas podem deixar a informação bem protegida.

O SQL Server tem como grande aliado o seu sistema operacional Windows Server e as ferramentas da Microsoft que, juntas, podem dificultar bastante os programas mal-intencionados. Já o Oracle possui parâmetros que auxiliam nas configurações voltadas para uma boa segurança.

Na prática, implementar estas medidas não é difícil, porém cuidados devem ser tomados e regras devem ser criadas. Uma boa segurança sempre deve estar

associada a uma disciplina dentro da empresa. De nada adianta ter todo um processo de segurança bem implementado e não seguir este processo.

O processo de segurança deve ser iniciado com uma análise do ambiente de banco de dados da empresa. Em ambientes grandes, existem várias equipes de desenvolvimento, analistas e pessoas envolvidas em resolver problemas, suporte e outros.

É muito fácil um analista solicitar um acesso full ao banco de dados, desta forma ele não se preocupa em ter que acessar uma ou outra tabela (acessando o banco de dados por completo, ele pode verificar o ambiente como um todo). Estas solicitações de acesso ficam ainda mais complicadas quando o analista conhece profundamente o sistema ou até quando ele já trabalha há muito tempo na empresa. Neste ponto existe uma certa vaidade em “ter que acessar tudo” e dessa forma, se o gerente permite este tipo de acesso, começam a existir “pessoas autorizadas” a fazer tudo e acessar tudo dentro de um database.

Neste artigo iremos demonstrar alguns detalhes que podem melhorar muito a proteção do SQL Server e do Oracle tais como políticas de acesso e uso de roles.

Políticas de acessos

As políticas de acesso nos databases podem ser tratadas da mesma forma, independente do banco de dados usado. Tais políticas devem ser criadas pela gestão de dados e devem ser seguidas para todo ambiente de databases (bancos transacionais e data warehouses).

Definir políticas implica diretamente em atividades relacionadas a permissões de acesso e como os usuários terão acesso ao database.

Nos databases SQL Server e Oracle existem as roles, que devem ser usadas para manter uma forma eficiente de proteger os dados. Além disso, o uso de roles pode ajudar muito na administração do ambiente. As roles facilitam as manutenções de acesso e criação de novos acessos. Roles são conjuntos de permissões que são concedidas levando em conta perfis diferentes de acesso.

Uma forma de definir políticas de roles seria a criação de um perfil para cada tipo de módulo do sistema. Outra estratégia seria definir tipos de acessos, usuários básicos que pudessem somente ter leitura no database e usuários avançados que pudessem executar procedures e escrever no database. Assim, quando um perfil é previamente determinado, fica muito mais fácil manter todos os acessos e administrar os acessos às tabelas, procedures e views.

Um ambiente seguro de banco de dados contempla ambientes distintos de desenvolvimento, homologação e produção. Na maioria das empresas, o ambiente de homologação e até mesmo o de desenvolvimento são ambientes controlados e mesmo o desenvolvedor mais experiente não tem acesso a eles inteiramente.

A grande necessidade de ter ambientes distintos, porém iguais em relação à estrutura, é evidenciar testes e conceder um ambiente para as equipes de desenvolvimento com uma liberdade maior para desenvolver melhor os processos nas fases de desenvolvimento e fazerem a manutenção e implementação dos sistemas já existentes.

Hoje, as empresas possuem estratégias para mascarar os dados que são considerados críticos de forma que no ambiente de desenvolvimento estes dados fiquem seguros. Um bom exemplo seria uma tabela de clientes onde existem CPFs. Estes campos devem ser mascarados de alguma forma. Além disso, manter um ambiente de homologação atualizado reflete também características em relação aos

testes de performance. Afinal, o ambiente produtivo pode ter um impacto diferente da homologação se uma mesma consulta SQL for executada nos dois ambientes (podemos ter uma tabela da produção bem maior que a mesma tabela na homologação).

Um aspecto que deve ter a atenção de DBAs e gestores é a questão das senhas de usuários de aplicação e as senhas de administradores dos bancos de dados. A senha de admin das instâncias SQL Server e Oracle devem ser definidas e trocadas de tempos em tempos. Existem programas que auxiliam no armazenamento destas senhas e podem ser usadas pelos DBAs, afinal, quando o ambiente começa a crescer e existem vários usuários de aplicação nestes ambientes, um armazenamento de senhas é extremamente importante para o gerenciamento destes acessos. Um bom exemplo deste tipo de aplicativo seria o KeePass.

Uma baseline de cada ferramenta de banco de dados deve ser escrita para que as políticas de segurança sejam seguidas corretamente e adotadas por outras equipes que venham a ocupar os cargos de DBA posteriormente.

Considerações sobre o sistema operacional

Um dos grandes diferenciais do Oracle é sua grande abrangência em relação ao tipo de sistema operacional que ele pode ser instalado. Entretanto, devemos destacar que esse pode ser também um ponto de atenção em relação à segurança, pois, um sistema operacional bem protegido em uma rede protegida faz toda a diferença. Sendo assim, seja Unix, Linux ou Windows, proteger o servidor onde será hospedado o Oracle é de extrema importância. Afinal, o sistema operacional e uma rede pode ser a porta de entrada para conexões mal-intencionadas no banco de dados.

A proteção do sistema operacional seja em databases SQL Server ou Oracle deve ser

bem-feita e a definição de acesso a disco e outros componentes de hardware devem ser bem controladas. Esta definição pode ser bem orientada pela equipe de sistemas operacionais juntamente com os DBAs. Algumas atenções devem ser aplicadas em relação aos acessos e à segurança do SO. A seguir, demonstramos alguns pontos que devemos ter atenção:

· No banco de dados Oracle:

o Remover os usuários desnecessários dos grupos ORADBA e ORAOPER. Os usuários que pertencem aos grupos do sistema operacional ORADBA e ORAOPER conectam-se ao banco de dados sem a necessidade de fornecer senhas. Estes privilégios permitem acesso administrativo ao banco de dados;

o Defina corretamente as permissões no *ORACLE_HOME* permitindo acesso somente para o Oracle e administradores do banco de dados. Todos os arquivos no diretório *$ORACLE_HOME* devem ser configurados para uso apenas por usuários proprietários de processos Oracle e pelo grupo de administradores do banco de dados;

o Defina corretamente as permissões da chave do Registry do Oracle. O acesso indevido à chave de Registry do Oracle permite que usuários maliciosos manipulem informações de configuração do banco de dados, podendo comprometê-lo.

· No banco de dados SQL Server:

o Configure corretamente o sistema de arquivos. O NTFS é o sistema de arquivos mais estável, ele habilita opções de segurança como listas de controle de acesso (ACLs) a arquivos e diretórios e criptografia de arquivos do sistema de arquivos com criptografia (EFS);

o Isolamento de serviços. Execute serviços separados do SQL Server sob contas separadas do Windows. Sempre que possível, use contas de usuário Local ou do

Windows separadas e com poucos direitos para cada serviço do SQL Server;

o Protocolos NetBIOS e SMB. Todos os servidores na rede devem ter os protocolos desnecessários desabilitados, incluindo NetBIOS e SMB. Os servidores Web e DNS (Sistema de Nome de Domínio) não requerem NetBIOS ou SMB. Nesses servidores, desabilite os dois protocolos para reduzir a ameaça de acessos indevidos;

o Não instale o SQL Server em uma máquina que é controladora de domínio;

o Restrinja arquivos do SQL Server. As pastas que contêm os datafiles devem ter os seus acessos restritos;

o Restrinja o acesso às chaves de registro do SQL Server. Nestas chaves, apenas a permissão do grupo de DBAs deve ser concedida.

É muito importante a verificação dos itens de segurança no sistema operacional antes e após a instalação do gerenciador de banco de dados. Um servidor bem protegido proporciona ao banco de dados uma boa segurança. O uso de firewall deve ser analisado e implementado juntamente com a configuração adequada das portas TCP/IP.

Da mesma forma que a performance do servidor que hospeda um banco de dados deve ser analisada, a segurança segue o mesmo critério e deve ser analisada antes mesmo de uma instalação. Neste aspecto, a troca de conhecimentos entre as áreas de banco de dados e administração de servidores deve ser bem alinhada.

Considerações sobre os padrões do SGBD

Quando uma ferramenta de banco de dados é instalada em um servidor, seja ela SQL Server, Oracle, DB2 ou Teradata, é importante lembrar que muitos dos recursos são disponibilizados como um padrão que foi definido pelo fornecedor da ferramenta,

porém nem sempre estes padrões são os melhores para o mundo corporativo e, sendo assim, é importante atentarmos para alguns pontos que, se forem modificados, podem melhorar muito a segurança dos bancos de dados:

· No Oracle:

o Aplique os patches de segurança necessários para corrigir todas as falhas conhecidas do produto. O banco de dados padrão precisa que estas patches sejam instaladas e atualizadas após sua instalação;

o Desabilite o parâmetro de inicialização *REMOTE_OS_AUTHENT*. Ele permite que usuários conectem-se ao banco remotamente sem fornecer senha;

o Altere o parâmetro *UTL_FILE_DIR* no arquivo de inicialização do banco adicionando os caminhos dos diretórios que são permitidos acessar;

o Restrinja o acesso às packages *UTL_SMTP*, *UTL_TCP*, *UTL_HTTP* e *UTL_FILE*. Estas packages permitem acesso externo ao banco de dados;

o Habilite a auditoria do Oracle para conexões. Desta forma, qualquer conexão mal-intencionada será registrada.

· No SQL Server:

o Instale o último service pack e hotfixes disponíveis. Este procedimento tem a função de corrigir possíveis falhas de segurança que foram descobertas após o lançamento da versão;

o Aumente o número de arquivos ErrorLogs. Esta opção deixa o troubleshooting mais fácil em uma situação de incidentes nos databases;

o Habilite a opção *Enforce Password Policy*. Desta forma, as senhas do SQL Server

obedecerão ao padrão Windows de senhas;

o Use a opção de modo misto para autenticação do SQL Server *SQL Server and Windows Authentication mode*;

o Habilite a auditoria de login no SQL Server. Desta forma, toda falha de conexão poderá ser observada.

Considerações sobre permissões

Iremos falar neste tópico sobre as principais permissões do banco de dados e como podemos melhorar a segurança em relação a elas. Em alguns ambientes, muitas vezes as permissões SYS/SYSTEM (Oracle) e SA (SQL Serve) ficam de posse de outras áreas que não são a de banco de dados. Este é um assunto polêmico que muitas vezes gera discussões.

O importante é ter em mente que as principais permissões de acesso citadas habilitam o usuário a fazer qualquer coisa no banco de dados. Embora apenas o DBA precise ter um acesso FULL, outros setores da empresa muitas vezes têm este acesso, e isso de certa forma é um item que gera desconforto em relação aos DBAs. Afinal, como manter a estabilidade do ambiente se os administradores de rede possuem acesso full? E como responsabilizar um DBA pelos dados perdidos, se o outro analista de sistemas pode ter a permissão também de apagar os dados? É necessário controlar as senhas principais de cada instância de banco de dados e saber a quem conceder estas permissões. Neste tópico iremos citar como melhorar a segurança do banco de dados em relação às permissões e privilégios:

· No Oracle:

o Remova os privilégios de sistema *ANY* dos usuários. Este privilégio concede acesso

a qualquer objeto de qualquer schema no banco de dados;

o Remova privilégios de sistemas ao *Public.* Tais privilégios devem ser concedidos cuidadosamente e apenas a usuários autorizados;

o Remova permissões de DBA a usuários desnecessários. Esta permissão permite acesso à role DBA que é uma das mais poderosas em relação a acessos no Oracle;

o Remova permissões SYSDBA de usuários desnecessários. Esta permissão permite que usuários executem qualquer ação no banco de dados;

o Desabilite os usuários SYS e SYSTEM no Oracle.

· No SQL Server:

o Remova usuários da role *System Administrator*. Usuários dessa role podem ter acesso full ao banco de dados SQL Server;

o Remova o grupo *BUILT-IN/Administrators* do SQL Server, desta forma usuários do SO não terão acesso ao SQL Server. Ao realizar esta remoção, um outro grupo de DBAs deve ser inserido no SQL Server;

o Remova o privilégio *Server Permissions* dos usuários desnecessários;

o Remova usuários desnecessários do grupo MSSQLServer;

o Remova permissões SYSADMIN de usuários desnecessários no banco de dados. Esta permissão permite que o usuário faça qualquer coisa no database. Além desta permissão, outras devem ter atenção dedicada a elas: *ServerAdmin*, *ProcessAdmin*, *SecurityAdmin*, *SetupAdmin*, *BulkAdmin*;

o Desabilite o usuário SA de uso no SQL Server.

Considerações sobre conexões

Quando o assunto é conexão com o banco de dados, todo o cuidado é pouco. É através das conexões que os acessos mal-intencionados podem chegar ao banco de dados. Vejamos alguns itens que podem melhorar as conexões:

· No banco de dados Oracle:

o Altere a opção para que as configurações não sejam feitas pelo aplicativo “lsnrctl”, mas apenas pelo arquivo no *listner.ora*;

o Desabilite a senha do listener para que sejam feitas administrações remotas.

· No banco de dados SQL Server:

o Altere a porta TCP/IP para uma porta diferente da porta padrão do SQL Server (1433);

o Desabilite a conexão DAC (dedicada) no SQL Server. O SQL Server fornece uma conexão especial para administradores quando conexões padrão com o servidor não são possíveis.

Considerações sobre roles

Como já foi dito anteriormente, a melhor prática que as empresas adotam em relação a permissões e acessos ao banco de dados é o uso de roles. As roles facilitam muito toda a administração e ajudam o DBA a manter as políticas de segurança nos objetos (tabelas, views e procedures). Quando um sistema é criado, a definição de como os acessos serão concedidos é função do DBA. Tendo em mãos todos os tipos de acesso, incluindo DBLinks (Oracle) e Linked Server (SQL Server), o DBA deve criar um login de aplicação e a ele devem ser concedidas as permissões necessárias para

acesso às tabelas e outros objetos. Este usuário será usado pelo aplicativo que se conecta ao banco de dados e todos os acessos serão originários dele. Dito isso, a seguir demonstraremos como criar um login, criar uma role, associar o login a uma role e em seguida conceder o acesso a uma tabela para uma role.

No banco de dados Oracle, o login, role, associação entre logins e roles, e concessão de acesso a uma tabela para esta role podem ser definidos conforme a **Listagem 1**. Os mesmos passos referentes ao SQL Server podem ser observados na **Listagem 2**.

Listagem 1. Definição de roles no Oracle.

```
  -- criando o login
  CREATE USER user_sqlmagazine IDENTIFIED BY ‘S4Q8L9MAG’
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;
  -- Criando uma role no banco de dados
  CREATE ROLE aplconsulta;
  -- Associando a role ao login no banco de dados.
  GRANT aplconsulta TO user_sqlmagazine;
  -- Concedendo acesso em uma tabela para a role
  GRANT SELECT, UPDATE, DELETE, INSERT ON hr.tb_cadastro TO aplconsulta
```

Listagem 2. Definição de roles no SQL Server.

```
  -- Criando um login no SQL Server.
  CREATE LOGIN user_sqlmagazine WITH PASSWORD = ‘S4Q8L9MAG’
  GO
  -- Criando um usuário no banco de dados e associando este usuário ao login do SQL

USE DB_APL
  GO
  CREATE USER user_sqlmagazine FOR LOGIN user_sqlmagazine;
  GO
-- Criando uma role no banco de dados.

USE DB_APL
  GO
  CREATE ROLE aplconsulta
```

```
  GO
-- Associando o usuário do banco de dados à role criada

USE DB_APL
  GO
  EXEC SP_ADDROLEMEMBER ‘aplconsulta’,’user_sqlmagazine’;
-- Concedendo acesso em uma tabela para a role
  GRANT SELECT, UPDATE, DELETE, INSERT ON tb_cadastro TO aplconsulta
```

No exemplo apresentado, podemos notar que a role faz a função de ligação entre o objeto (tabela) e o usuário. Neste aspecto, o usuário nunca acessa diretamente uma tabela. Se por acaso esta tabela não pode ser manipulada pela role, o usuário nunca poderá fazer nada com ela. Em outras estratégias o usuário poderia acessar diretamente uma tabela, porém a administração destes usuários se tornaria muito mais complexa com o passar do tempo. Em uma estratégia com roles, a única ação a fazer é adicionar um novo usuário em uma role, caso contrário, este usuário teria que ter as mesmas permissões do usuário anterior (direto a tabela). Quando um ambiente tem muitos usuários, esta carga de acessos fica mais complicada ainda. Desta forma, com o uso de roles, a manutenção dos privilégios se transforma em algo fácil e bem dinâmico.

Acesso multicamadas

Existe ainda uma estratégia que pode ser adotada em muitos ambientes e que de certa forma ajuda muito na segurança dos databases. O conceito de mais uma camada durante a conexão faz com que o usuário tenha acesso a um tipo de máscara que oculta o nome da tabela. Este conceito é implementado com o uso de synonym. O synonym é como se fosse um apelido que é criado para uma tabela. Assim, a segurança do objeto tabela e o seu nome verdadeiro fica resguardado dentro do banco de dados. Quando um synonym é criado, o acesso da role precisa ser concedido para

o synonym e não para a tabela em específico. Conhecidos os conceitos, a **Listagem 3** apresenta como criar um synonym e expõe como conceder permissões para uma role de um synonym no Oracle. Para bases de dados SQL Server, estes mesmos comandos são apresentados na **Listagem 4**.

Listagem 3. Criando um synonym no Oracle.

```
CREATE SYNONYM TabeladeCadastro FOR hr.tb_cadastro

-- Concedendo acesso a role do synonym.
GRANT SELECT, UPDATE, DELETE, INSERT ON TabeladeCadastro TO aplconsulta
```

Listagem 4. Criando um synonym no SQL Server.

```
CREATE SYNONYM TabeladeCadastro FOR tb_cadastro

-- Concedendo acesso a role do synonym.
GRANT SELECT, UPDATE, DELETE, INSERT ON TabeladeCadastro TO aplconsulta
```

Informações sobre permissões

Neste último tópico demonstraremos algumas tabelas de sistema que são importantíssimas para o monitoramento de novas permissões criadas em um banco de dados e para manutenções de permissões existentes. O DBA tem, acima de tudo, que estar sempre monitorando o ambiente e revendo as permissões criadas. É importante ter em mente que se um usuário já teve em algum momento uma permissão que lhe dava acesso para criar usuários e para conceder novas permissões, pode ser que este usuário tenha acesso à base de dados de várias maneiras. O monitoramento é indispensável neste aspecto. A seguir expomos as principais tabelas de sistema que auxiliam o DBA a monitorar o banco de dados em relação à segurança:

· No banco de dados Oracle:

o **DBA_USERS:** Verifica usuários do sistema;

o **DBA_TAB_PRIVS:** Verifica atributos do usuário;

o **DBA_SYS_PRIVS:** verifica privilégios do usuário;

o **DBA_ROLE_PRIVS:** Verifica privilégios e roles;

o **DBA_ROLES:** Verifica roles de usuários.

· No banco de dados SQL Server:

o **sys.database_permissions:** Verifica permissões;

o **sys.database_role_members:** Verifica permissões de roles;

o **sys.schemas:** Verifica os schemas de um banco de dados;

o **sys.syslogins:** Verifica cada login no SQL Server;

o **sys.server_role_members:** Verifica as server roles.

Segurança é um assunto que de certa forma atinge a todas as esferas de uma organização. Saber lidar com esse tipo de situação em uma organização é essencial para a aceitação das políticas de uma forma geral. Alguns fatores podem depender ainda de aspectos culturais da empresa.

Acima disso tudo, fazer uma segurança requer tempo e análises variadas. Nos ambientes organizacionais, muitas vezes a segurança vira uma prioridade e quando os diretores percebem que existe uma falha e que isso pode de certa forma influenciar no negócio como um todo, a ordem de estabelecer a segurança aparece sem medir

escalas e faz com que os analistas e DBAs tenham que tratar este assunto como prioridade.

A segurança é algo que precisa ser definida e mantida. Sabemos que da mesma forma que estamos tentando fazer a segurança funcionar, existem pessoas mal-intencionadas trabalhando no sentido contrário.

Links

SQL Server Books Online

http://technet.microsoft.com/en-us/library/ms130214(v=sql.105).aspx

Oracle Database Documentation Library

http://docs.oracle.com/cd/B19306_01/index.htm

Flavio Climaco

Atua na ramo de Tecnologia da Informação há mais de 10 anos. É formado na Área de Tecnologia da Informação pela UNIPAC/JF, Pós Graduado em Administração de Banco de Dados pelo CES/JF e MBA em Administração de Banco de Dados pela F [...]

Publicado em 2015

+ Mais conteúdo sobre SQL

Não há comentários

Meus comentarios

Postar dúvida / Comentário

Mais posts

Video aula

Entendendo os tipos de dados para caracteres no MySQL - Curso Completo MySQL - Aula 34

Video aula

Entendendo o funcionamento dos campos de ponto flutuante

- Curso Completo MySQL - Aula 33

Video aula

Aprendendo a trabalhar com campos decimais exatos - Curso Completo MySQL - Aula 32

Video aula

Entendendo os limites dos campos inteiros - Curso Completo MySQL - Aula 31

Artigo

Recuperação de bases de dados Oracle

Artigo

Particionamento no Oracle

Listar mais conteúdo