Universidade Federal
de Campina Grande

Banco de Dados I
Unidade 13: Segurança

Prof. Cláudio de Souza Baptista, Ph.D.
Laboratório de Sistemas de Informação – LSI
UFCG

Introdução

- Uma das maiores preocupações em computação tem sido segurança da informação;
- Nos dias atuais, com o uso da Internet, os sistemas tornam-se onipresentes, entretanto também vulneráveis a ataques maliciosos;
- Portanto, os SGBDs trazem uma camada de segurança que visa compor toda o arsenal de segurança da informação numa corporação

Introdução

- **Definição:**
  - Segurança em Banco de dados diz respeito à proteção do banco de dados contra ataques intencionais ou não intencionais, utilizando-se ou não de meios computacionais .
- **Áreas envolvidas:**
  - roubo e fraude
  - perda de confidencialidade
  - perda de privacidade
  - perda de integridade
  - perda de disponibilidade

Introdução

- O subsistema de segurança é responsável por proteger o BD contra o acesso não autorizado.
- Formas de acesso não autorizado:
  - leitura não autorizada;
  - modificação não autorizada;
  - destruição não autorizada
- O ABD tem plenos poderes para dar e revogar privilégios a usuários.

Introdução

- **Motivação:** Exemplo Locadora
  - Apenas alguns empregados podem modificar preços dos DVDs;
  - Clientes usando o sistema de consulta, não devem ter acesso a outras funcionalidades (vendas, contabilidade, folha de pagamento, etc);
  - Apenas o pessoal da gerência deve ter acesso às informações dos empregados (por exemplo: empregados-a-demitir);
  - Clientes não devem ver o preço de compra de um produto

Introdução

- Controles de segurança computacionais
  - Adiciona-se uma camada à segurança provida pelo SO;
  - Autorização e autenticação;
  - Views;
  - Backup e recovery;
  - Integridade;
  - Stored procedures;
  - Criptografia;
  - Auditoria;
  - Procedimentos associados
    - e.g. upgrading, virus checking, proxy, firewall, kerberos, certificados digitais, SSL, SHTTP, etc.

Introdução

- Controles de segurança não computacionais
  - Política de segurança e plano de contingência;
  - Posicionamento seguro de equipamentos;
  - Controle de acesso físico;
  - Manutenção;

Introdução

- Duas abordagens para segurança de dados:
  - **Controle de acesso discreto:**
    - Um dado usuário tem direitos de acessos diferentes (privilégios) em objetos diferentes
    - Flexível, mas limitado a quais direitos usuários podem ter em um objeto
  - **Controle de acesso mandatório:**
    - cada dado é rotulado com um certo nível de classificação
    - A cada usuário é dado um certo nível de acesso
    - rígido, hierárquico

Introdução

- Em SQL:1999 temos:

Proteção	Privilégio	Aplica-se a
Ver	SELECT	Tabelas, colunas, métodos invocados
Criar	INSERT	Tabelas, colunas
Modificar	UPDATE	Tabelas, colunas
Remover	DELETE	Tabelas
Referenciar	REFERENCES	Tabelas, colunas
Usar	USAGE	UDT
Ativar	TRIGGER	Tabelas
Executar	EXECUTE	Stored Procedures

Introdução

- O que se espera do SGBD é o mesmo tratamento dada à tentativa de acesso a uma tabela inexistente (“no such table”);
- Portanto, se um usuário tentar acessar uma tabela que ele não tem privilégios para tal o erro será:

  “Either no such table or you have no privilege on the table”
- Razão: Segurança

Introdução

- O usuário tem um auth_ID que o identifica;
- Existe PUBLIC que representa todos usuários;
- Privilégios são atribuídos/revogados:
  - Usuários
  - Papéis (Roles)
- O criador de um objeto é o dono do objeto e assim tem todos os privilégios sobre o objeto, podendo autorizar a outros usuários alguns (ou todos) destes privilégios.
- A opção **with grant option**, permite ao usuário que recebeu um privilégio repassar para quem quiser.

Introdução

- Com respeito a DDL:
  - Um usuário pode executar qualquer comando DDL no esquema que ele é dono.
  - Um usuário NÃO pode executar nenhuma operação DDL no esquema que ele não é o dono.

Usuários e Papéis

- Identificador de usuário
  - Alguns SGBDs permitem que o usuário use o mesmo login e senha do SO
- Papéis (Roles)
  - É um identificador ao qual pode-se atribuir privilégios que não existem a princípio. Então pode-se atribuir a um usuário este papel (conjunto de privilégios) com um único comando GRANT.
  - Pode-se inclusive ao criar um papel usar outros papéis já cadastrados.
  - **Ex.** PapelVendedor, PapelVendedorSapatos, PapelVendedoFrutas.

Usuários e Papéis

- Pilha de autorizações

AuthID	Role name	
-	-	
José	(null)	Stored procedure
(null)	Vendedor	SQL Embutido
Carlos	(null)	Login no SO

Papéis - Roles

- Sintaxe SQL:1999

```
CREATE ROLE nome-papel
[WITH ADMIN {CURRENT_USER |
CURRENT_ROLE}]
```

- Para remover um papel:

```
DROP ROLE nome-papel;
```

Papéis - Roles

- Existem papéis padrões na maioria dos SGBD:
  - **DBA:** permite desempenhar o papel de administrados do banco de dados;
  - **Resource:** permite criar seus próprios objetos;
  - **Connect:** permite apenas se conectar ao banco de dados, mas deve receber os privilégios de alguém para acessar objetos.

Regras de Autorização

- Expressam os mecanismos de autorização em relações/visões/ stored procedures;
- São compiladas e armazenadas no dicionário de dados;
- São expressas em linguagem de alto nível (Ex. SQL);
- Uma maneira do SGBD implementar estas regras é usar uma matriz de autorização, onde cada linha corresponde a um usuário a um usuário e cada coluna corresponde a um objeto;
- M[i,j] => conjunto de regras de autorização que se aplica ao usuário i com relação ao objeto j.

Regras de Autorização

Ex.:

	Empregado	Departamento	Projeto
João	SELECT	UPDATE, SELECT	SELECT, DELETE, UPDATE
Maria	None	None	SELECT
Pedro	None	None	None
Ana	All	All	All

- O ABD fornece/revoga as autorizações de leitura, inserção, atualização e remoção aos usuários nas diversas tabelas/visões, e estes podem repassá-los caso receba autorização para tal.

Regras de Autorização

- O comando **GRANT**

```
GRANT lista-privilégios
ON objeto
TO lista-usuários [WITH GRANT OPTION]
[GRANTED BY
{CURRENT_USER|CURRENT_ROLE}]
```

Regras de Autorização

- O comando **GRANT**
  - **Lista de privilégios:**

Privilégio1[, privilégio2 ...]
|ALL PRIVILEGES

  - **Privilégios**

SELECT [coluna,...]
|SELECT (método,...)
|DELETE
|INSERT [coluna,...]
|UPDATE [coluna ...]
|REFERENCES [(coluna ...)]
|USAGE
|TRIGGER
|EXECUTE

Regras de Autorização

- O comando **GRANT**
  - **Lista de usuários:**

authID, [authID ...]
| PUBLIC

OBS.: authID pode ser **login** ou **role**

- A opção **GRANTED BY** indica se os privilégios concedidos são autorizados pelo o usuário corrente ou pelo role.

Regras de Autorização

- Autorizando papéis

```
GRANT role-name [, role-name ...]
To lista-usuários
[WITH ADMIN OPTION]
[GRANTED BY
{CURRENT_USER|CURRENT_ROLE}
```

- Um **role-name** pode ter um número ilimitado de privilégios ou outros roles

Exemplos

```
/*Permite a quem tenha o papel Gerente_Loja apenas ver a tabela empregados*/
GRANT SELECT ON EMPREGADOS TO GERENTE_Loja


/*Privilégios de remoção com permissão de repassar o privilégio */
GRANT DELETE ON Empregados TO Carlos WITH GRANT OPTION
```

Exemplos

```
/* Update de uma coluna específica */
GRANT UPDATE (preço) ON Produtos
TO Gerente_Loja

/* Privilégios de inserção */
GRANT INSERT ON Produtos
TO Carla, Maria, Marta

/* Inserção só em algumas colunas */
GRANT INSERT (id, preco, descricao, tipo)
ON Produtos TO Assistente
```

Exemplos

```
/* Acesso público em views */
GRANT SELECT ON MinhaVisão
TO PUBLIC;

/* referências (foreign key) */
GRANT REFERENCES (titulo)
ON FILMES TO Pedro;
```

Exemplos

- Um privilégio **TRIGGER** numa tabela permite criar um trigger para aquela tabela;
- O privilégio **EXECUTE** permite um usuário ou role executar uma determinada stored procedure.

Exemplo:

```
GRANT EXECUTE ON AumentaSalario TO isabel
```

Exemplos

- **ALL PRIVILEGES**
  - Permite especificar uma lista de privilégios que inclui todos os privilégios de um objeto específico no qual o usuário executando o **GRANT** tem o privilégio para dar o **grant** (recebeu **WITH GRANT OPTION** ou é o dono)

Ex.:

```
GRANT ALL PRIVILEGES ON Filmes to Patricia
```

Revoke

- Revoga autorização de privilégios;
- Se o usuário **A** tiver concedido o privilégio **P** para o usuário **B**, então **A** poderá, posteriormente, revogar o privilégio **P** de **B**, através do comando **REVOKE**;
- **Sintaxe:**

```
REVOKE <privilégios> ON <relação/visão> FROM <usuários>
```

Ex.:

```
REVOKE delete ON projeto FROM Marta, Ana
```

```
REVOKE update ON Empregado FROM Ana
```

```
REVOKE DBA FROM Bruno
```

BD Estatístico

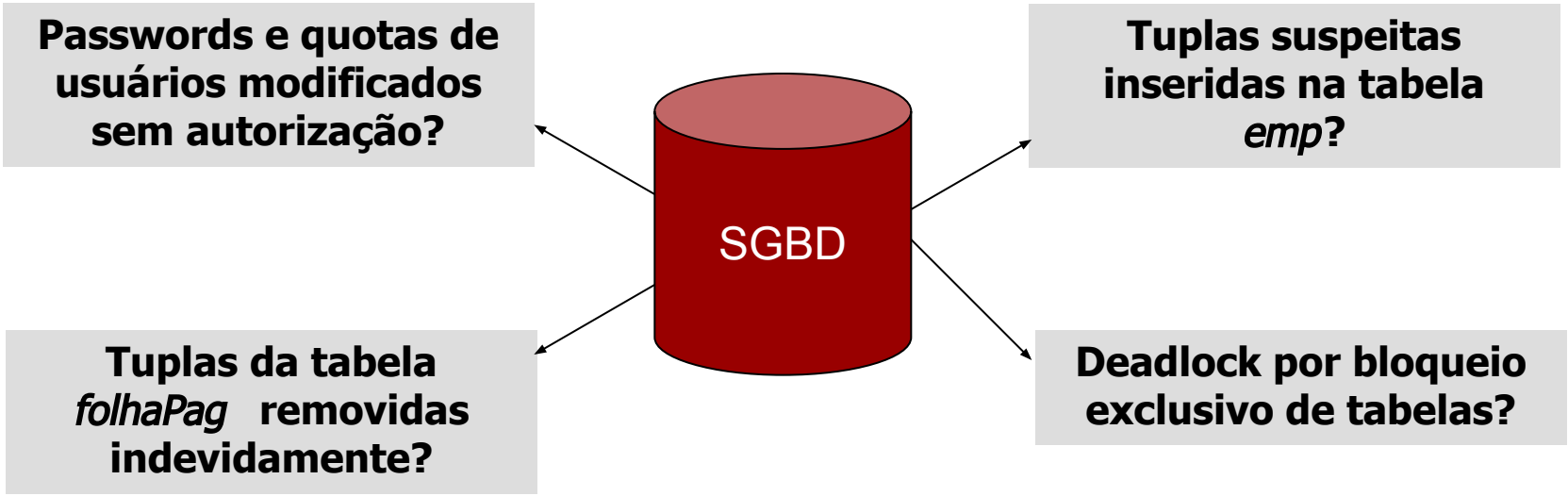
- Um banco de dados que permite queries que derivam informação agregadas (e.g. somas, médias)
  - Mas não queries que derivam informação individual
- Tracking
  - É possível fazer inferências de queries legais para deduzir respostas ilegais
  - Ex.:

```
WITH (STATS WHERE SEXO='M' AND FUNCAO = 'Programador') AS X : COUNT(X)
```

```
WITH (STATS WHERE SEXO = 'M' AND FUNCAO = 'Programador' AS X : SUM(X,SALARIO)
```

Auditoria

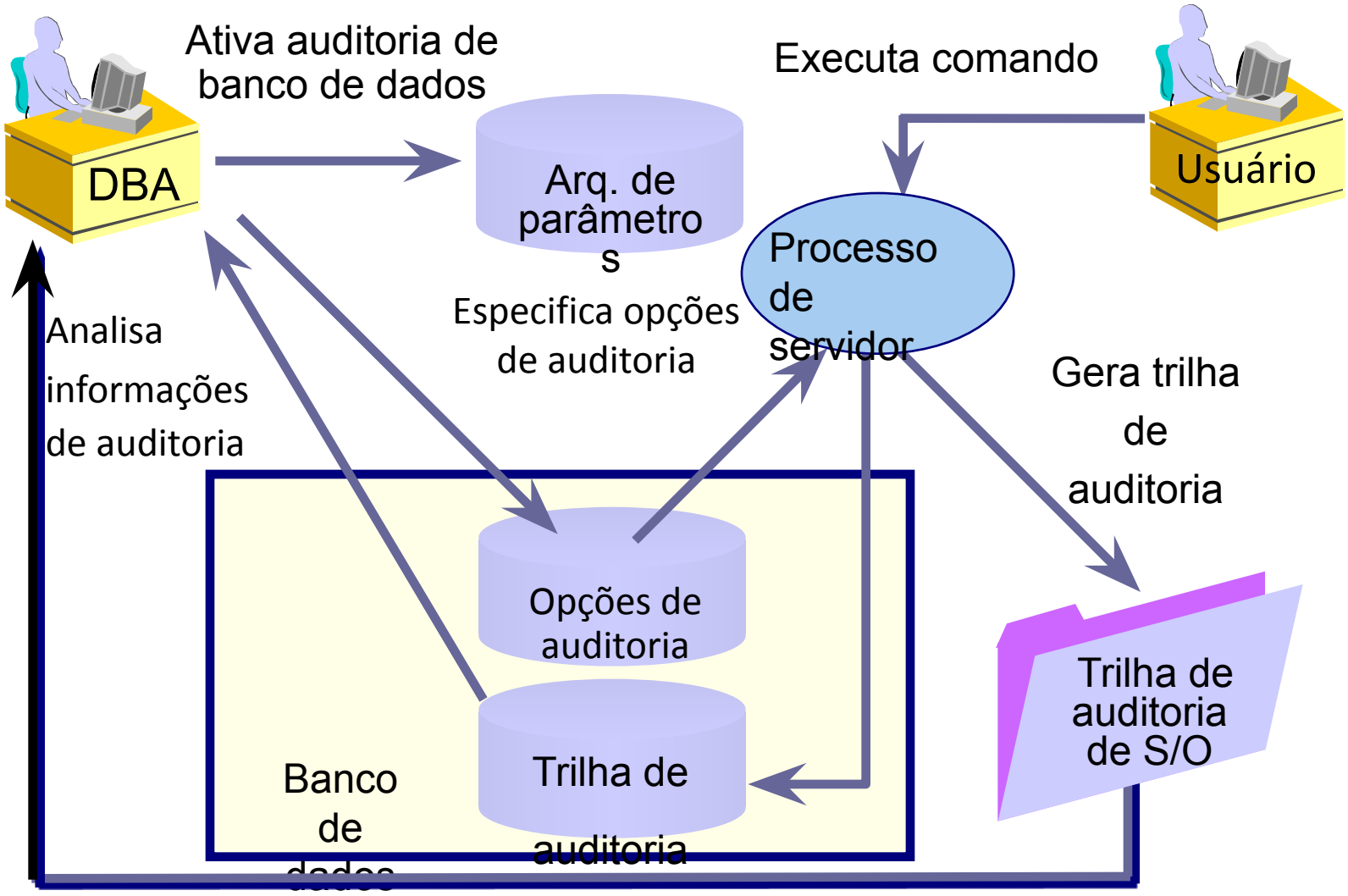
Auditoria



Solução: Auditar, Investigar

??? Quem fez o quê e quando ???

Auditoria


Ativa auditoria de banco de dados
Executa comando
DBA
Usuário
Arq. de parâmetros
Processo de servidor
Especifica opções de auditoria
Analisa informações de auditoria
Gera trilha de auditoria
Opções de auditoria
Trilha de auditoria de S/O
Banco de dados
Trilha de auditoria

Diretrizes da Auditoria

- Defina as finalidades da auditoria
  - Atividade de banco de dados suspeita
  - Reúna informações históricas
- Defina o que você deseja auditar
  - Auditar usuários, instruções ou objetos
  - Por sessão
  - Com ou sem sucesso
- Gerencie a trilha de auditoria
  - Monitorar o crescimento da trilha de auditoria
  - Proteger a trilha de auditoria de acesso não-autorizado

Diretrizes da Auditoria

- Avaliar o propósito de auditoria, evitando auditoria desnecessária.
  - Que tipo de atividade do BD você suspeita?
  - Quem são os suspeitos?
- Auditar, inicialmente, de forma genérica e ir especializando.
- Apesar do custo baixo deve-se limitar o nº de eventos auditados o máximo possível para minimizar:
  - O impacto de performance na execução de comandos auditados
  - O tamanho do audit trail

Audit Trail

- Audit trail: componente de todo SGBD que armazena histórico de informações de auditoria
  - Oracle: tabela SYS.AUD$
  - DB2: log DB2AUDIT.LOG

- O SO também pode ter um audit trail. Podendo ser usado em conjunto com o do BD.

Audit Trail

- Algumas informações do audit trail:
  - Nome do login do usuário no SO;
  - Nome do usuário no BD;
  - Identificador de sessão;Identificador do terminal;
  - Nome do objeto do esquema acessado;
  - Operação executada ou tentada;
  - Código de conclusão da operação;
  - Data e hora.

SQL Injection

- É uma forma de alterar um código SQL e assim ter acesso a dados não autorizados.
- Ex.: Suponha uma Stored Procedure get_salario(nomedep), que, no seu corpo, executa o seguinte SQL:

```
SELECT e.mat, e.nome, e.salario
FROM Empregado e JOIN Departamento  d
        ON e.depto = d.coddep
WHERE e.nomedepto = nomedep
```

SQL Injection

- Um usuário não malicioso poderia invocar get_salario(‘Vendas’), obtendo os salários do departamento de vendas.
- Um usuário malicioso pode invocar get_salario(‘Vendas’’ OR ‘’1’’= ‘’1’)

- Neste caso, veria os salários de todos os empregados