Segurança

- dados são um dos recursos mais valiosos e devem ser estritamente controlados e geridos.
- questões de segurança não incluem somente o sistema de getão de bases de dados e os serviços por ele prestados mas tambem o ambiente em que estao integrados.
- proteção desta relativamente a acessos intencionais ou não contra ameaças via meios de controlos baseados ou não em computadores.
- podem existir quebras na segurança de outros componentes do sistema que por sua vez podem afetar direta ou indiretamente o próprio sistema da bases de dados
- compreende sistemas de hardware, software, recursos humanos e dados.
- a segurança no passado era negligenciada -> mudança de mentalidade deve-se em parte ao aumento significativo de dados armazenados em sistemas de computação e ao reconhecimento e aceitaçao de que perda ou não disponibilidade desses dados pode causar perdas irreparáveis.
- sistemas de bases de dados devem ser seguros atraves de mecanismos e controlos de segurança adequados -> pretende-se minimizar de forma efetiva as perdas provocadas por eventos não esperados sem constranger os utilizadores.

    - situaçoes de quebra de segurança:
        - roubo e fraude
        - perda de confidencialidade
        - perda de privacidade
        - perda de integridade
        - perda de disponibilidade

Ameaças

- pode ser provocada por uma situaçao ou evento envolvendo uma pessoa, acção, ou circunstância que potencialmente, ou que na realidade, provoque algum tipo de dano na organização.
- danos podem ser tangiveis, tais como a perda de hardware, software ou dados, ou intangiveis, como a perda de credibilidade e confiança perante funcionários, organizaçao ou clientes.
- ameaças intencionais envolvem pessoas e podem ser efetuados por utilizadores autorizados ou não autorizados, os quais pode ser externos à organização.

Table 19.1 Examples of threats.

Threat	Theft and fraud	Loss of confidentiality	Loss of privacy	Loss of integrity	Loss of availability
Using another person's means of access	✓	✓	✓		
Unauthorized amendment or copying of data	✓			✓	
Program alteration	✓			✓	✓
Inadequate policies and procedures that allow a mix of confidential and normal output	✓	✓	✓		
Wire tapping	✓	✓	✓		
Illegal entry by hacker	✓	✓	✓		
Blackmail	✓	✓	✓		
Creating 'trapdoor' into system	✓	✓	✓		
Theft of data, programs, and equipment	✓	✓	✓		✓
Failure of security mechanisms, giving greater access than normal		✓	✓	✓	
Staff shortages or strikes				✓	✓
Inadequate staff training		✓	✓	✓	✓
Viewing and disclosing unauthorized data	✓	✓	✓		
Electronic interference and radiation				✓	✓
Data corruption owing to power loss or surge				✓	✓
Fire (electrical fault, lightning strike, arson), flood, bomb				✓	✓
Physical damage to equipment				✓	✓
Breaking cables or disconnection of cables				✓	✓
Introduction of viruses				✓	✓

Identificaçao de ameaças

- processo dificil que consome tempo, esforço e dinheiro. No caso de não haver a possibilidade de realizar tal tarefa de forma exaustiva, pelo menos, a indentificacão das principais ameaças deve ser feita.

- os danos provocados pela eventual concretização de uma ameaça dependem de um variado número de fatores tais como a existência de contra-medidas e de planos de contingência. (p.e: no caso da ocorrência de uma falha de hardware ao nivel dos sistemas de armazenamento secundário de informaçao toda a ativdade de processamento deve ser imediatamente terminada até o problema estar resolvido)

exemplos:

- hacking
- quebra ou desconexão de dados
- danos fisicos no equipamento
- falha de mecanismos de seguranca
- formaçao e treino inadequados
- alteração de programas

Periodos de Inatividade

- duraçao depende de vários fatores:
    - possibilidade de utilizar ou não hardware e software alternativo
    - momento em que foram realizadas as cópias de segurança
    - tempo necessário para recuperar o sistema
    - possibilidade de os dados perdidos serem ou não recuperados e recapturados

Contextos de ameaças

- base de dados
    - utlizadores
    - programadores/operadoes
    - administradores
- SGBD e aplicaçoes
- hardware

Impacto das ameaças

- na avaliaçao de ameaças e na análise dos seus possiveis impactos, varias questores podem ser levantadas. No caso da falha de hardware com corrupção de sistemas de armazenamento secundário, ja referida anteriormente:
    - existira hardware alternativo que pode ser utilizado?
    - esta alternativa sera segura?
    - quando é que as ultimas copias de segurança de base de dados(dados e logs) foram realizadas
    - as copias de segurança estao armazenadas em local seguro?
    - se a base de dados atual necessitar de ser novamente criada, quanto tempo levará estre processo a ser realizado
    - qual foi o volume de processamento que foi perdido
    - Será que somos capazes de recuperar os dados?
    - Poderáo as atividade da organização ser realizadas sem o suporte do sistema que está inativo? Por quanto tempo?

Por exemplo, se ocorrer uma falha de hardware corrompendo o armazenamento secundário, toda a atividade de processamento deve parar até o problema estar resolvido.
Uma organização necessita de identificar os tipos de ameaças às quais pode estar sujeita e iniciar planos e contra-medidas adequadas. Obviamente, pode não ser "cost-effective" gastar tempo, esforço e dinheiro em ameaças que apenas resultem numa menor inconveniência. Algumas das ameaças podem ser raras, mas devem ter-se em conta se o possivel dano for elevado.

Contra-medidas (podem ir desde a definicao de controlos fisicos ate a aplicaçao de procedimentos administrativos)
- autorização e credenciais de acesso
- vistas de dados
- cópias de segurança e recuperação
- controlo e garantia de integridade
- cifragem de dados
- tecnologia RAID

Computer-Based Control

Autorização - dar acesso legítimo ao sistema. Apenas deverá ser feito a pessoas qualificadas. Isto não implica que tenha acesso a toda a base de dados. A autorizacao pode ser dada apenas a uma parte da mesma.

Autenticação - garantir que um utilizador é quem diz ser, através de uma password pessoal e um código. Isto permite o uso autorizado de um sistema computacional, mas não garante acesso ao DBMS ou aplicaçoẽs associadas. Um administrado de sistema é normalmente responsável por dar acesso aos utilizadores (através das contas individuais que possuem a password e código).

Controlo de Acesso

Grant ou revoke de privilégios
Isto permite que um utilizador possa ou não manipular um certo objeto e devem ser dados apenas os necessários para os utilizadores fazerem as suas tarefas.

Discretionary Access Control (DAC)

A maior parte dos DBMS usa isto através do GRANT e REVOKE e dá e retira privilégios.

Apesar de eficaz, isto tem algumas fraquezas. Em particular, um utilizador não autorizado pode enganar um user autorizado para lhe fornecer dados sensíveis (aos quais o primeiro não tem acesso). O exemplo apresentado no livro envolve o Assistente e o Gestor. O 1º não tem acesso aos dados do cliente, no entanto pode criar uma nova relação à qual tem acesso e pode dar privilégios de escrita ao Gestor. Em seguida, pode alterar o DBMS para copiar dados a que apenas o Gestor tem acesso para esta nova relação, acabando finalmente por apagar as instruçoẽs, cobrindo assim os seus passos. Desta forma, nunca violou os privilégios e tem acesso aos dados.

Para dar a volta a estes problemas, é necessária uma nova abordagem. Assim, nasceu o MAC - Mandatory Access Control. Apesar de quase todos os DBMS terem DAC, apenas alguns têm MAC.

Mandatory Access Control (MAC)

Baseado em políticas de sistema que não podem ser alteradas por utilizadores individuais. Nesta política, a cada objeto de dados é fornecida uma classe e a cada utilizador um nível de segurança. Em seguida, são impostas regras que conferem o estatuto a cada classe.

Um modelo popular é o de Bell-LaPadula que envolve objetos, sujeitos, classes de segurança e clearance. Cada objeto da base de dados tem uma classe de segurança, cada sujeito tem uma clearance. As classes de segurança são ordenadas: Top Secret (TS) > Secret (S) > Confidential (C) > Unclassified (U). O mesmo se aplica às clearances. Baseia-se em duas regras:

- Simple Security Property: Um sujeito S só pode ler um objeto O se a classe de S for >= classe de O. Por exemplo, um user com clearance TS pode ler um objeto de classe C, mas um user de clearance C nao pode ler um objeto de classe C.
- *_ Property: Um sujeito S so pode escrever um objeto O se a clearance de S for <= classe de O. Ou seja, um sujeito com clearance S só pode escrever objetos de clearance S e TS.

Polyinstantiation

Com o uso de MAC surge um problema muito claro.
Por exemplo, uma tabela tem várias entradas de classe S, entre as quais uma que tem a chave primária 1. Também tem várias entradas de classe C. Um user de clearance C só vê as de classe C, enquanto que um user de clearance S vê todas. Desta forma, quando o user de clearance C tenta inserir uma entrada cuja chave primária tem valor 1, não pode uma vez que a restrição de integridade é violada. Assim, este user pode inferir que existe uma entrada cuja chave é 1 a um nível de segurança superior. Isto compromete a segurança dos dados, uma vez que um user de clearance inferior não pode saber nada sobre o que está acima dele. A solução é passar a clearance na chave. Desta forma, ficariam duas entradas na tabela: (1, C) e (1, S) e o user de clearance C nunca saberia nada sobre os níveis acima de si. A isto chama-se poli-instanciação.

Neste caso existiriam dois tuplos com a mesma chave para um utilizador de acesso superior. Assim, a situação pode ser resolvida assumindo que o tuple com maior clearence tem prioridade sobre os outros.
A grande desvantagens de MAC é que há demasiada rigidez nas suas classes, levando a que seja demasiado inflexível e difícil de lidar em certos casos.

Views

Uma view é uma relação que não existe na base de dados mas pode ser chamada como se fosse. Assim, é possível esconder diversas vistas da base de dados de um utilizador. Esconde também atributos e outras relações sem que o utilizador se aperceba que estes existem. Uma vista pode ser definida para várias relações e os privilégios podem ser dados ou retirados a esta vista sem afetar as relações que estão por trás. É mais restritivo que o uso de permissões (DAC) e menos restritivo que MAC.

Backups, Restauro e Logging

Os primeiros dois sao obvios.

Quanto a logging, deve manter registos das transaçoẽs e mudanças na base de dados para dar apoio em caso de falhanço. Com isto, em caso de falhanço pode ser recuperado para o seu último estado consistente **usando uma cópia de restauro e o log file.**
Se não há logging, só o restauro de um backup pode salvar os dados mas quaisquer mudanças feitas desde o último backup são perdidas.

Integridade

As restrições de integridade ajudam a evitar que os dados sejam corrompidos. Devem ser vistas também com um meio de segurança.

Encriptação

De acordo com a sensibilidade dos dados pode ser recomendado usar encriptação. Alguns DBMS permitem isto, no entanto há degradação de performance uma vez que e necessário codificar e descodificar os dados.

Há várias técnicas de encriptação de dados, sendo estas reversiveis ou irreversiveis. As irreversiveis não permitem que os dados originais sejam conhecidos, no entanto, podem ser usadas para obter informação estatistica.

Também permite comunicação segura. Para transmitir dados sobre redes inseguras é necessário:
- uma chave de encriptação para encriptar o texto
- um algoritmo de encriptaçao
- uma chave de desencriptação
- um algoritmo de desencriptação

Encriptação simétrica: Usa a mesma chave para a encriptação e desencriptação (A chave necessita de ser pelo menos tão longa como a mensagem).
Encriptação assimétrica: Diferentes chaves para a encriptação e desencriptação.

Os algoritmos simétricos são mais rapidos de executar, mas na prática ambos os algoritmos são usados em conjunto.

RAIDS (Redundant Array of Independent Disks)

Também ajuda a manter os dados seguros. Razões óbvias.

O hardware do DBMS deve ser tolerante a falhas, e então deve poder continuar a operar quando um dos componentes falha. Isto sugere a existência de componentes redundantes que pode facilmente ser integrados no sistema quando um dos componentes falha.

Assim, os RAIDS consistem em arrays de discos independentes que são organizados de forma a melhorar a leitura e aumentar a performance. A performance é aumentada através de "data striping", na qual os dados são segmentados entre partições de igual tamanho, que são distribuidas transparentemente por vários discos. Isto dá a ilusão de um único grande disco. "Data striping" também ajuda a balancear a carga dos discos.