Segurança e Auditoria de Sistemas

- Políticas,
- Modelos e mecanismos de controle de acesso;
- Políticas discricionárias.

Controle de acesso

- Consiste em:
  - mediar cada solicitação de acesso de um usuário autenticado a:
    - um recurso ou
    - dado mantido pelo sistema,
  - determinar se a solicitação deve ser autorizada ou negada

Controle de acesso

- Praticamente todos os recursos de um sistema operacional típico estão submetidos a um **controle de acesso**.
  - arquivos,
  - áreas de memória,
  - semáforos,
  - portas de rede,
  - dispositivos de entrada/saída, etc.

Controle de acesso

- Sujeitos:
  - entidades que exercem um papel ativo no sistema:
    - como processos, threads ou transações.
  - Normalmente um sujeito opera em nome de um usuário, que pode ser um ser humano ou outro sistema computacional externo.
- Objetos:
  - entidades passivas utilizadas pelos sujeitos:
    - como arquivos, áreas de memória ou registros em um banco de dados.
  - Um sujeito pode ser visto como objeto por outro sujeito (quando um sujeito deve enviar uma mensagem a outro sujeito).
  - Tanto sujeitos quanto objetos podem ser organizados em grupos e hierarquias, para facilitar a gerência da segurança..

Políticas, modelos e mecanismos de controle de acesso

- Uma política de controle de acesso é uma visão abstrata das **possibilidades** de acesso a recursos (objetos) pelos usuários (sujeitos) de um sistema.
  - Essa política consiste basicamente de um conjunto de regras
    - Definindo os acessos possíveis aos recursos do sistema e eventuais condições necessárias para permitir cada acesso.
    - Por exemplo, as regras a seguir poderiam constituir parte da política de segurança de um sistema de informações médicas:
      - Médicos podem consultar os prontuários de seus pacientes;
      - O supervisor geral pode consultar os prontuários de todos os pacientes;

Políticas, modelos e mecanismos de controle de acesso

- As regras ou definições individuais de uma política são denominadas **autorizações**.
- Uma política de controle de acesso pode ter autorizações baseadas em:
  - identidades
    - como sujeitos e objetos
  - atributos
    - como idade, sexo, tipo, preço, etc.;
  - as autorizações podem ser:
    - individuais (a sujeitos) ou
    - coletivas (a grupos);
  - também podem existir autorizações positivas (permitindo o acesso) ou negativas (negando o acesso);
  - dependentes de condições externas (como o tempo ou a carga do sistema).
- Política administrativa,
  - define quem pode **modificar/gerenciar** as políticas vigentes no sistema.

Políticas, modelos e mecanismos de controle de acesso

- O conjunto de autorizações de uma política deve ser:
  - completo,
    - cobrindo todas as possibilidades de acesso que vierem a ocorrer no sistema, e
  - consistente,
    - sem regras conflitantes entre si (por exemplo, uma regra que permita um acesso e outra que negue esse mesmo acesso).
- A política deve respeitar o princípio do privilégio mínimo,
  - um usuário nunca deve receber mais autorizações que aquelas que necessita para cumprir sua tarefa.

Políticas, modelos e mecanismos de controle de acesso

- Existem muitas formas de se definir uma política, que podem ser classificadas em quatro grandes classes:
  - políticas discricionárias,
  - políticas obrigatórias,
  - políticas baseadas em domínios e
  - políticas baseadas em papéis

Políticas, modelos e mecanismos de controle de acesso

- Geralmente a descrição de uma política de controle de acesso é muito abstrata e informal.
- Para sua implementação em um sistema real, ela precisa ser descrita de uma forma precisa, através de um modelo de controle de acesso.
  - Um modelo de controle de acesso é uma representação lógica ou matemática da política, de forma a facilitar sua implementação e permitir a análise de eventuais erros.
  - Em um modelo de controle de acesso, as autorizações de uma política são definidas como relações lógicas entre atributos do sujeito (como seus identificadores de usuário e grupo) atributos do objeto (como seu caminho de acesso ou seu proprietário) e eventuais condições externas (como o horário ou a carga do sistema).

Políticas, modelos e mecanismos de controle de acesso

- os mecanismos de controle de acesso são as estruturas necessárias à implementação de um determinado modelo em um sistema real.
- É de fundamental importância a separação entre políticas e mecanismos, para
  - permitir a substituição ou modificação de políticas de controle de acesso de um sistema sem incorrer em custos de modificação de sua implementação.
- O mecanismo de controle de acesso ideal deveria ser capaz de suportar qualquer política de controle de acesso.

Políticas discriminárias

- As políticas discricionárias (DAC - Discretionary Access Control) se baseiam na atribuição de permissões de forma individualizada.
  - pode-se claramente conceder (ou negar) a um sujeito específico ***s*** a permissão de executar a ação ***a*** sobre um objeto específico ***o***.
  - regras de uma política discricionária têm a forma ⟨**s, o, +a**⟩ ou ⟨**s, o, −a**⟩,
    - para respectivamente autorizar ou negar a ação ***a*** do sujeito ***s*** sobre o objeto **o**
    - também podem ser definidas regras para grupos de usuários e/ou de objetos devidamente identificados.
    - Por exemplo:
      - O usuário Beto pode ler e escrever arquivos em /home/beto
      - Usuários do grupo admin podem ler os arquivos em /suporte
  - O responsável pela administração das permissões de acesso a um objeto pode ser o seu proprietário ou um administrador central.
  - A definição de quem estabelece as regras da política de controle de acesso é inerente a uma política administrativa, independente da política de controle de acesso em si.

Matriz de controle de acesso

- O modelo matemático mais simples e conveniente para representar políticas discricionárias é a **Matriz de Controle de Acesso**.
  - Nesse modelo, as autorizações são dispostas em uma matriz
  - **linhas** correspondem aos **sujeitos** do sistema e
  - **colunas** correspondem aos **objetos**.
  - Formalisando:
    - um conjunto de sujeitos S = {s1,s2,...,sm},
    - um conjunto de objetos O = {o1,o2,...,on} e
    - um conjunto de ações possíveis sobre os objetos A = {a1, a2, . . . , ap},
    - cada elemento Mij da matriz de controle de acesso é um sub-conjunto (que pode ser vazio) do conjunto de ações possíveis, que define as ações que si ∈ S pode efetuar sobre oj ∈ O:

      $\forall si \in S, \forall oj \in O, Mij \subseteq A$

    - Por exemplo
      - um conjunto de sujeitos S = {Alice, Beto, Carol, Davi},
      - um conjunto de objetos O = { f ile1, f ile2, program1, socket1} e
      - um conjunto de ações A = {read, write, execute, remove}

Matriz de controle de acesso

- Por exemplo
  - um conjunto de sujeitos S = {Alice, Beto, Carol, Davi},
  - um conjunto de objetos O = { f ile1, f ile2, program1, socket1} e
  - um conjunto de ações A = {read, write, execute, remove}

	file1	file2	program1	soccket1
Alice	Read Write remove	Read write	execute	write
Beto	Read write	Read Write remove		Read write
Carol		read	execute	Read append
Davi	read	append	read	

Tabela de autorizações

- Na **Tabela de Autorizações**,
  - as entradas não-vazias da matriz são relacionadas em uma tabela com três colunas:
    - sujeitos,
    - objetos e
    - ações
  - cada tupla da tabela corresponde a **uma autorização**.
  - Esta abordagem é muito utilizada em sistemas gerenciadores de bancos de dados (DBMS - Database Management Systems)
    - facilidade de implementação e consulta nesse tipo de ambiente.

Tabela de autorizações

- Matriz de controle de acesso sob a forma de uma tabela de autorizações para Carol.

Sujeito	Objeto	Ação
Carol	file2	read
Carol	program1	execute
Carol	socket1	read
Carol	socket1	write

Lista de controle de acesso

- Outra abordagem usual é a **Lista de Controle de Acesso**.
  - Para cada objeto é definida uma lista de controle de acesso (**ACL** - Access Control List), que contém
    - a relação de sujeitos que podem acessá-lo,
    - suas respectivas permissões.
  - Cada lista de controle de acesso corresponde a uma coluna da matriz de controle de acesso.
  - É simples de implementar e
  - É bastante robusta.
  - É a mais usada em sistemas operacionais,
  - Em geral, somente o proprietário do arquivo pode modificar sua ACL, para incluir ou remover permissões de acesso.
  - Por exemplo,
    - o sistema de arquivos associa uma ACL a cada arquivo ou diretório, para indicar quem são os sujeitos autorizados a acessá-lo.

Lista de controle de acesso

- Exemplo, as listas de controle de acesso relativas à matriz de controle de acesso:

  ACL(file1) ={Alice : (read, write, remove, owner), Beto : (read, write),Davi : (read) }

  ACL(file2) ={Alice : (read, write), Beto : (read, write, remove, owner), Carol : (read), Davi : (write) }

  ACL(program1) = { Alice : (execute), Beto : (read, owner), Carol : (execute), Davi : (read) }

  ACL(socket1) = { Alice : (write), Carol : (read, write), Davi : (read, write, owner) }

Lista de capacidades

- Outra abordagem possível para a implementação da matriz de controle de acesso é a Lista de Capacidades (CL - **Capability List**),
  - uma lista de objetos que um dado sujeito pode acessar e suas respectivas permissões sobre os mesmos.
  - Cada lista de capacidades corresponde a uma linha da matriz de acesso.
  - Uma capacidade pode ser vista como uma ficha ou token:
    - sua posse dá ao proprietário o direito de acesso ao objeto em questão.
  - Capacidades são pouco usadas em sistemas operacionais, devido à sua
  - Dificuldade de implementação e possibilidade de fraude,
    - pois uma capacidade mal implementada pode
      - ser transferida deliberadamente a outros sujeitos, ou
      - modificada pelo próprio proprietário para adicionar mais permissões a ela.
    - Outra dificuldade inerente às listas de capacidades é a administração das autorizações:
      - por exemplo, quem deve ter permissão para modificar uma lista de capacidades, e
      - como retirar uma permissão concedida anteriormente a um sujeito?

Lista de capacidades

CL(Alice) ={f ile1 : (read, write, remove, owner), file2 :(read,write), program1 : (execute), socket1 : (write) }

CL(Beto) ={file1 :(read,write), file2 :(read,write,remove,owner), program1 : (read, owner) }

CL(Carol) ={file2 :(read), program1 : (execute), socket1 : (read, write) }

CL(Davi) ={file1 :(read), file2 :(write), program1 : (read), socket1 : (read,write,owner)}