ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 29: Segurança (parte 3)

Prof. Renan Alves

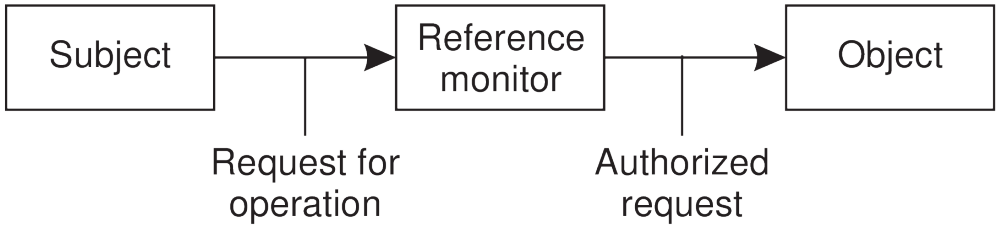
Escola de Artes, Ciências e Humanidades — EACH — USP

24/06/2024

Controle de acesso: Modelo geral

Autorização

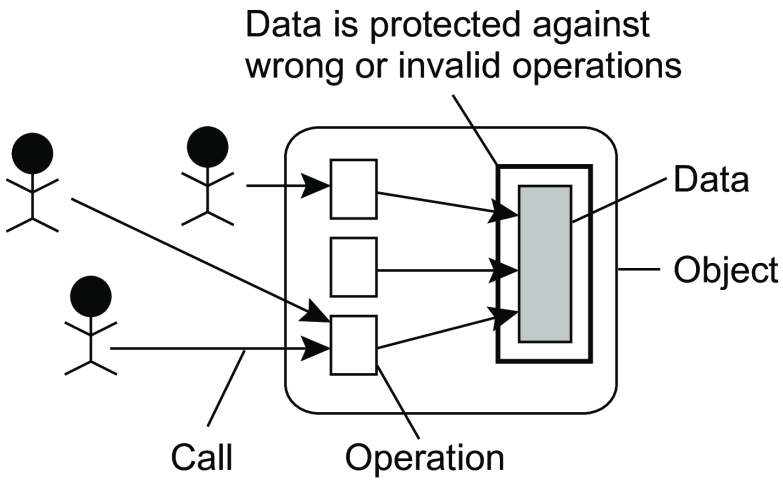
Garantir que entidades autenticadas tenham acesso apenas a recursos específicos.



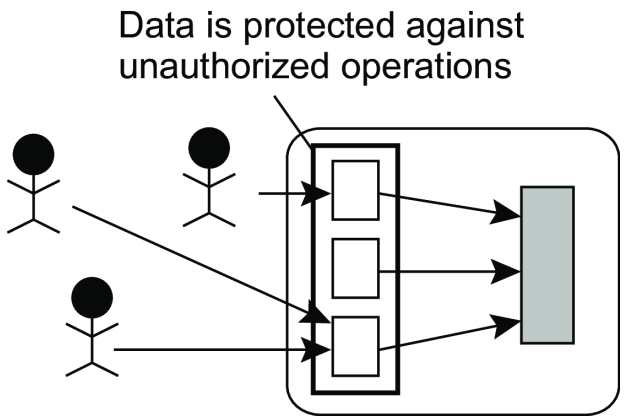
Observação

O monitor de referência precisa ser inviolável: geralmente é implementado totalmente a nível de sistema operacional, ou em um servidor remoto seguro.

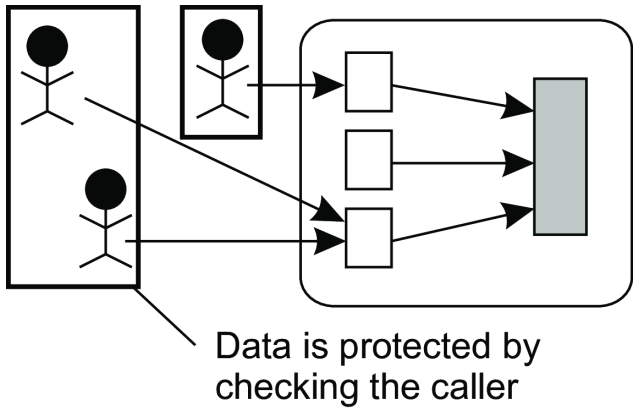
Proteção



...contra operações inválidas



...contra acesso não autorizado



...contra invocadores não autorizados

Políticas de controle de acesso

Como definir quem tem acesso a que?

1. Controle de acesso obrigatório: um administrador central define quem tem acesso a quê.
2. Controle de acesso arbitrário: o proprietário de um objeto pode alterar os direitos de acesso do objeto.
3. Controle de acesso baseado em cargo: os usuários não são autorizados com base em sua identidade, mas com base no cargo que possuem dentro de uma organização.
4. Controle de acesso baseado em atributos: atributos dos usuários **e** dos objetos que desejam acessar são considerados para decidir sobre uma regra de acesso específica.

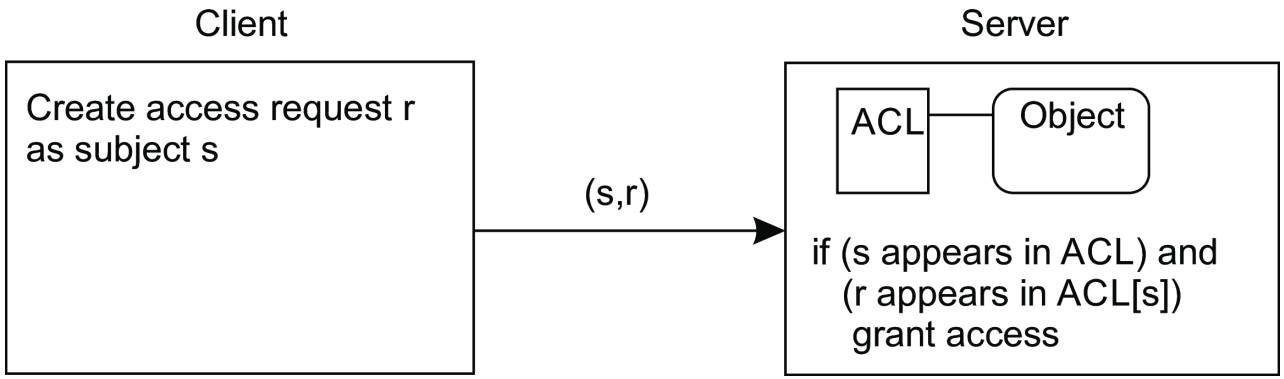
Observação

As políticas podem ser combinadas.

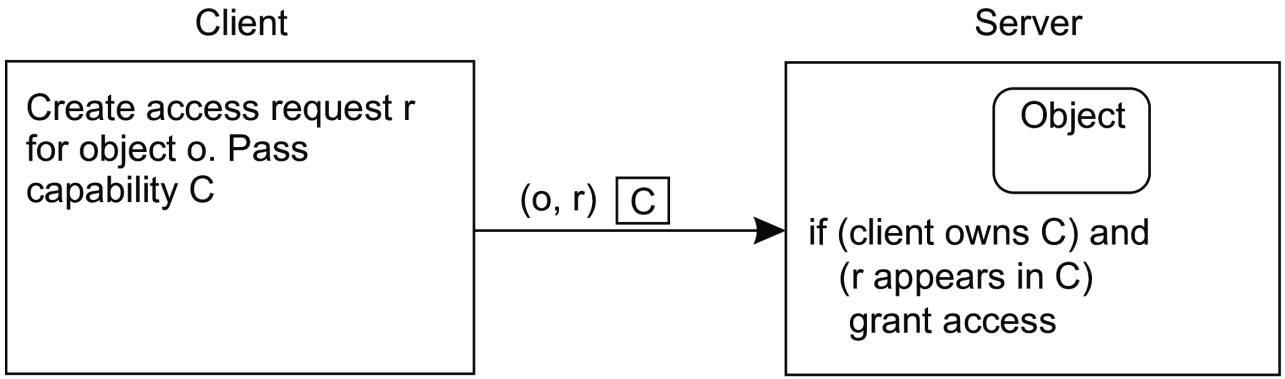
Matriz de controle de acesso

Funcionamento

Ideia original: construção de uma matriz na qual $M[s, o]$ descreve os direitos de acesso que o sujeito s tem em relação ao objeto o. Porém é impraticável, então simplificar para listas de controle de acesso (ACL) ou capacidades.



Lista de controle de acesso



Capacidades

Exemplo: controle de acesso baseado em atributos

Distinguir diferentes classes de atributos (exemplo de uma universidade):

- Atributos de usuário: nome, data de nascimento, funções atuais, endereço residencial, departamento, qualificações obtidas, status de contrato, etc. Atributos podem depender da função (por exemplo, professor ou aluno).
- Atributos do objeto: qualquer coisa – criador, última modificação, número de versão, tipo de arquivo, tamanho do arquivo, mas também informações relacionadas ao seu conteúdo.
- Atributos ambientais: descrevem o estado atual do sistema, por exemplo, data e hora, carga de trabalho atual, status de manutenção, propriedades de armazenamento, serviços disponíveis, etc.
- Atributos de conexão: fornecem informações sobre a sessão atual, por exemplo, endereço IP, duração da sessão, estimativas de largura de banda e latência disponíveis, tipo e força da segurança utilizada.
- Atributos administrativos: refletem políticas globais, por exemplo, configurações mínimas de segurança, regulamentos gerais de acesso e durações máximas de sessão.

Delegação

Qual é o problema?

Alice usa um provedor de serviços de e-mail que armazena sua caixa de entrada. Ela precisa fazer login no provedor para acessar seu e-mail. Alice quer usar seu próprio cliente de e-mail local. Como permitir que esse cliente de e-mail aja em nome de Alice? Como delegar os direitos de acesso de Alice ao seu cliente de e-mail?

Delegação

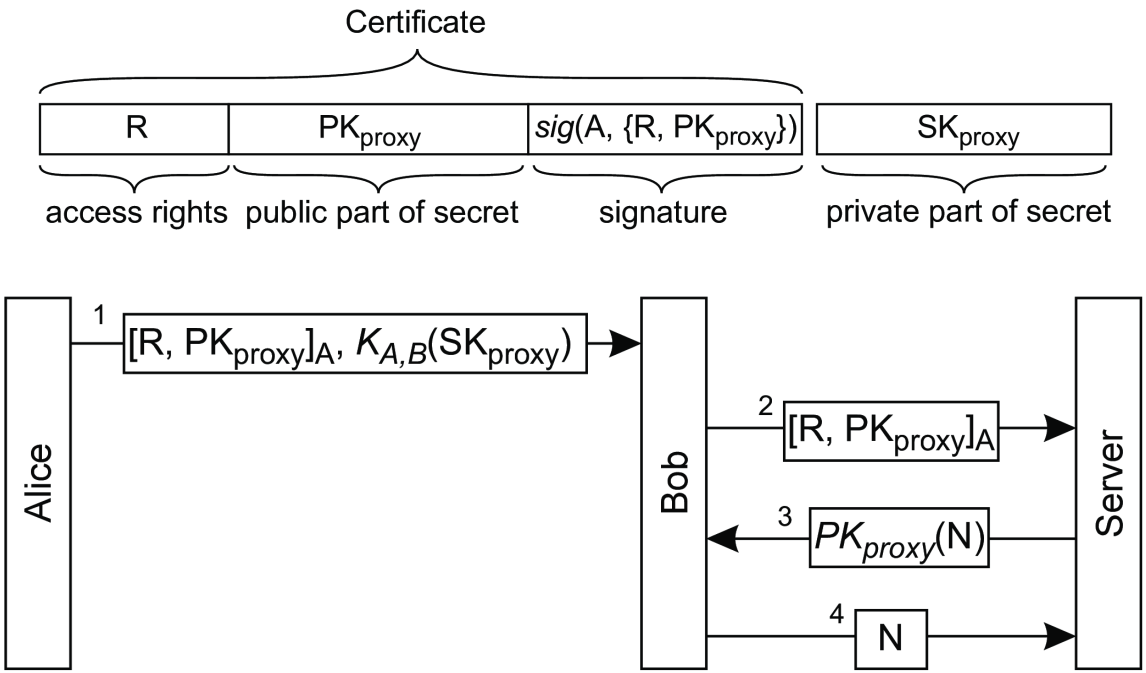
Qual é o problema?

Alice usa um provedor de serviços de e-mail que armazena sua caixa de entrada. Ela precisa fazer login no provedor para acessar seu e-mail. Alice quer usar seu próprio cliente de e-mail local. Como permitir que esse cliente de e-mail aja em nome de Alice? Como delegar os direitos de acesso de Alice ao seu cliente de e-mail?

Observação

Não é uma boa ideia entregar todas as credenciais de usuário para uma aplicação: por que a aplicação ou a máquina seriam confiáveis? $\Rightarrow$ uso de um proxy de segurança.

Proxy de segurança



Como funciona

1. Alice passa alguns direitos *R* para Bob, juntamente com uma chave secreta SK_{proxy}
2. Quando Bob quer exercer seus direitos, ele passa o certificado
3. O servidor quer que Bob prove que conhece a chave secreta
4. Bob prova que conhece, e assim que Alice delegou *R*.

Exemplo: Open Authorization (OAuth)

Quatro diferentes papéis

- Proprietário do recurso: tipicamente um usuário final.
- Cliente: uma aplicação que desejamos que aja em nome do proprietário do recurso.
- Servidor de recursos: Uma interface através da qual uma pessoa normalmente acessa o recurso.
- Servidor de autorização: uma entidade que emite certificados para um cliente em nome de um proprietário de recurso.

Passos iniciais

1. A aplicação cliente se registra no servidor de autorização e recebe seu próprio identificador, *cid*.
2. Alice quer delegar uma lista R de direitos $\Rightarrow$

$$\text{Cliente: } envia\ [cid, R, H(S)]$$

com o hash de um segredo temporário S

Completando o processo

Passos finais

3. Alice é solicitada a fazer login no servidor de autorização e confirmar a delegação R para o cliente.
4. O servidor envia um código de autorização temporário AC para o cliente.
5. O cliente solicita um token de acesso final:

$$\text{Cliente: } envia\ [cid, AC, S].$$

   O envio de S para o servidor de autorização permite que ele verifique a identidade do cliente (calculando $H(S)$).

O servidor de autorização agora (1) verificou que Alice quer delegar direitos de acesso para o cliente, e (2) verificou a identidade do cliente $\Rightarrow$ retorna um token de acesso para o cliente (por um canal seguro).

Autorização descentralizada com WAVE

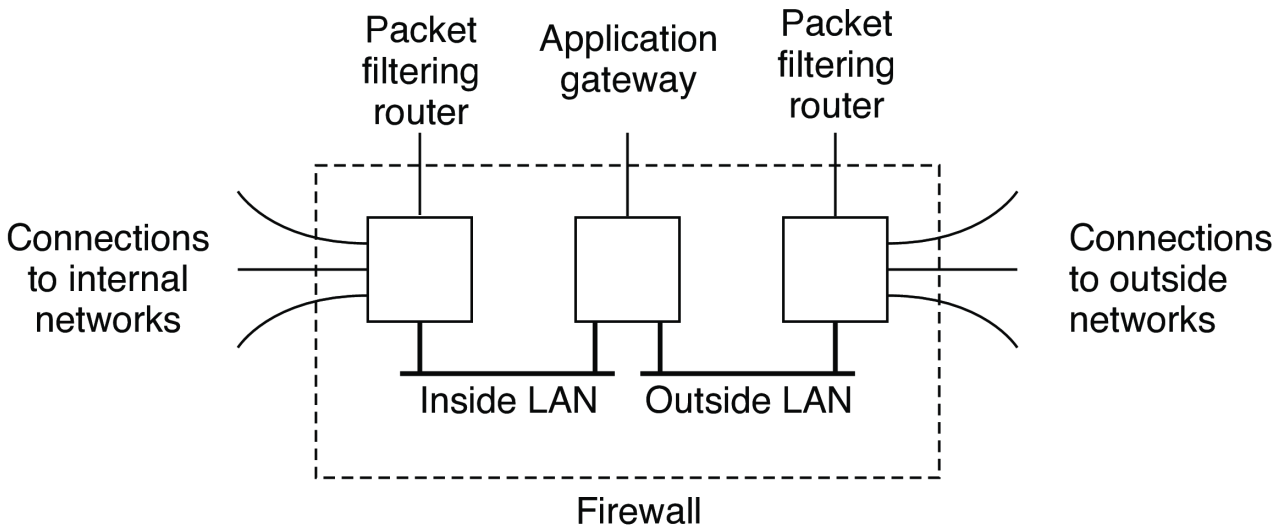
Simplificadamente:

- **W**AVE is an **A**uthorization **V**erification **E**ngine
- Baseado em um grafo direcional para representar autorizações
- Se Bob tem permissão para acessar um recurso de Alice: existe um caminho no grafo de A até B
- Grafo precisa ser armazenado de forma distribuída e confiável (semelhante a um blockchain)
- Delegação é feita utilizando um esquema baseado em criptografia de chave pública de forma que, se Alice der direitos a Bob, e Bob der (alguns destes) direitos para Chuck, a permissão pode ser verificada sem que Alice ou Bob estejam online

Firewalls

Essência

Impedir que coisas indesejadas entrem e prevenir tráfego indesejado de saída.



Diferentes tipos de firewalls

- Gateway de filtragem de pacotes: opera como um roteador e filtra pacotes com base no endereço de origem e destino.
- Gateway de nível de aplicação: inspeciona o conteúdo de uma mensagem de entrada ou saída (e.g., gateways que filtram spam).

Sistemas de detecção de intrusão

Duas abordagens

- Baseado em assinatura: corresponde a padrões de intrusões conhecidas no nível da rede. Problemático quando séries de pacotes precisam ser correspondidas, ou quando novos ataques ocorrem.
- Baseado em anomalia: assume que podemos modelar ou extrair comportamentos típicos para, posteriormente, detectar comportamentos atípicos ou anômalos. Depende fortemente de tecnologias modernas de inteligência artificial.

Detecção colaborativa de intrusão

Essência

- Diversos sensores medindo dados que podem ajudar na detecção de intrusão.
- Sensores são agrupados em comunidades, onde há um líder (é possível participar de mais de uma comunidade)
- Os líderes das comunidades colaboram entre si

Melhorando detecção

Gerenciar as comunidades para maximizar precisão e acurácia:

$$\text{Acurácia}: \quad \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precisão}: \quad \frac{TP}{TP + FP}$$

FP/TP = falsos e verdadeiros positivos; FN/TN = falsos e verdadeiros negativos