Sistemas Distribuídos

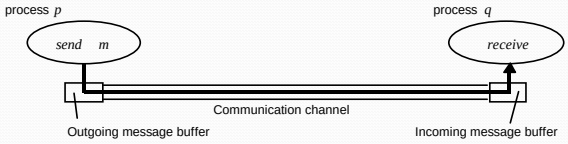
5 Modelo de Falhas e Segurança

- **Modelo de Falhas**
- **Modelo de Segurança**

Profª Ana Cristina B. Kochem Vendramin
DAINF / UTFPR

Modelo de Falhas

- Falhas de processo:
  - Crash/Fail Stop;
  - Erro ao executar as primitivas send e receive.
- Falhas de canais de comunicação:
  - Não transporta, corrompe ou duplica mensagens.



Modelo de Falhas

Class of failure	*Affects*	*Description*
Fail-stop	**Process**	Process halts and remains halted. Other processes may detect this state.
Crash	**Process**	Process halts and remains halted. Other processes may not be able to detect this state.
Omission	**Channel**	A message inserted in an outgoing message buffer never arrives at the other end's incoming message buffer.
Send-omission	**Process**	A process completes a *send*, but the message is not put in its outgoing message buffer.
Receive-omission	**Process**	A message is put in a process's incoming message buffer, but that process does not receive it.
Arbitrary	**Process** or **channel**	Process/channel exhibits arbitrary behaviour: it may send/transmit arbitrary messages at arbitrary times, commit omissions; a process may stop or take an incorrect step.

[CDK 01]

Falhas de Tempo

Class of Failure	*Affects*	*Description*
Clock	**Process**	Process's local clock exceeds the bounds on its rate of drift from real time.
Performance	**Process**	Process exceeds the bounds on the interval between two steps.
Performance	**Channel**	A message's transmission takes longer than the stated bound.

[CDK 01]

Mascarando Falhas

- O serviço mascara uma falha, escondendo-a ou convertendo-a em uma falha aceitável.
- Comunicação confiável definida em termos de:
  - **Confiabilidade**
  - **Validade**
    - Números de seqüência detectam mensagens ausentes.
    - Retransmissões de mensagens não entregues.
  - **Integridade**
    - *Checksums* mascaram mensagens corrompidas.
    - Números de seqüência detectam mensagens duplicadas.

Mascarando Falhas

- Redundância
  - Redundância de informação: bits extras são adicionados para recuperar bits errados.
  - Redundância de tempo: uma ação é executada e, se necessário, é executada novamente.
  - Redundância física: equipamento extra é adicionado para que o sistema como um todo tolere a falha de um ou outro componente.

Mascarando Falhas

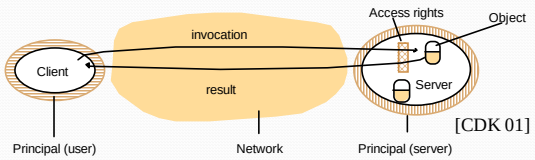
- Redundância ativa
  - Todos os processadores são usados o tempo todo como servidores (em paralelo) a fim de ocultar falhas completamente.
- *Primary backup*
  - Apenas um processador (*primary*) está ativo a cada instante. Se este falhar, um outro processador (*backup*) é ativado para operar em seu lugar.

Modelo de Segurança

- Segurança em processos, canais e objetos encapsulados pelos processos.
  - Direitos de acesso:
    - Protegem contra acesso não autorizado;
    - Especificam quem tem permissão de operar um objeto ou recurso;
    - Cada invocação ou resposta é associada à autoridade (chamada Principal) que a emitiu.

Modelo de Segurança – O Principal

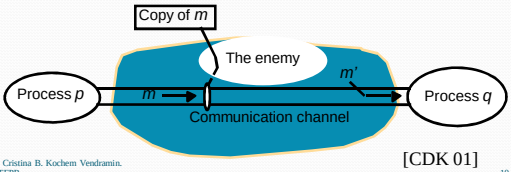
- O principal pode ser um processo cliente ou servidor.
- Cliente e servidor precisam checam a identidade do principal.
- Servidor deve verificar os direitos de acesso dos clientes antes de executarem operações em objetos.



[CDK 01]

Modelo de Segurança – O Principal

- Utiliza um computador conectado à rede para executar um programa que lê ou copia mensagens endereçadas a outros computadores;
- Executa um programa que gera mensagens com pedidos falsos fazendo-se passar por um usuário autorizado.



[CDK 01]

Mecanismos de Segurança

- Alguns mecanismos típicos de segurança incluem:
  - **Autenticação**: identifica quem (ou qual serviço) está solicitando recursos da rede.
  - **Autorização**: define o que pode ser feito depois de ter acesso à rede.
  - **Contabilidade (auditoria)**: procedimento de coleta de dados de atividade da rede. Grava acessos ou tentativas.
  - **Criptografia de dados**

Criptografia de Dados

- Processo para "embaralhar" os dados antes de colocá-los na rede de modo a protegê-los contra a leitura de qualquer pessoa que não seja o receptor pretendido.
- Consiste de duas partes:
  - Algoritmo de criptografia e chave de criptografia.
- As chaves consistem de dois tipos bem conhecidos:
  - Chave simétrica – chave secreta compartilhada
    - Ex.: DES (*Data Encryption Standard*).
  - Chave assimétrica
    - Ex.: Chave pública/privada.

Criptografia de Dados

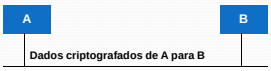
- Na chave simétrica, o remetente e o destinatário possuem a mesma chave secreta.
- Na chave pública/privada: tudo que for codificado pela senha pública só é decodificado com a senha privada e vice-versa.
- Dois recursos são possíveis:
  - Preservação do caráter confidencial dos dados.
  - Mecanismo de autenticação (assinatura digital).

Chave pública-privada

Assegurar caráter confidencial dos dados (só "B" poderá decodificar):

A codifica os dados usando a senha pública de B

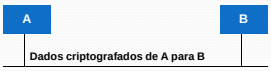
B decodifica os dados usando a sua senha privada.



Assegurar autenticação - assinatura digital (só "A" pode ter enviado os dados):

A codifica os dados usando a sua senha privada

B decodifica os dados usando a senha pública de A. (Certeza da origem)



Ameaças e Formas de Ataque

- Ameaças de segurança recaem em três classes:
  - ***Leakage* (Vazamento)**
    - Aquisição de informações por recipientes não autorizados;
  - ***Tampering***
    - Alteração não autorizada da informação;
  - ***Vandalism* (Vandalismo)**
    - Interferência na operação do sistema sem ganho para o infrator.

Ameaças e Formas de Ataque

- ***Eavesdropping* (escutar às escondidas)**
  - Ameaça à privacidade das informações → obter cópias de mensagens sem autorização.
- ***Masquerading***
  - Enviar ou receber mensagens usando a identidade de outro sem a autorização do mesmo.

Ameaças e Formas de Ataque

- ***Replaying***
  - Ameaça à integridade das informações que trafegam na rede.
  - Salva cópias de mensagens para enviá-las depois.
  - Não precisa possuir a chave secreta para causar esse tipo de ataque.
- ***Denial of service (DoS)***
  - Invocações excessivas nos serviços ou grandes transmissões de mensagens na rede resultando em recursos sobrecarregados.

Ameaças e Formas de Ataque

- ***Message tampering***
  - Intercepta mensagens e altera seus conteúdos antes de enviá-las ao destino.
  - Ataque ***man-in-the-middle***:
    - Atacante intercepta a primeira mensagem da troca de chaves criptografadas.
    - Substitui as chaves por outras que permitem que ele decodifique as mensagens subsequentes antes de codificá-las novamente com as chaves corretas e enviá-las.

Ameaças de Códigos Móveis

- Várias linguagens têm sido desenvolvidas para permitir que programas oriundos de servidores remotos sejam baixados e executados localmente (exemplo: java).
- Cada aplicação Java tem seu próprio ambiente de execução.
- Cada ambiente tem um gerente de segurança que determina quais recursos estão disponíveis para a aplicação.
- Ex.: o gerente de segurança pode não permitir que a aplicação leia ou escreva em arquivos ou oferecer a ela um acesso limitado a conexões de rede.

Vazamento de Informações

- Se a transmissão de uma mensagem entre dois processos pode ser observada, algumas informações podem ser recolhidas pela simples existência da mensagem.
- Exemplo:
  - Uma inundação de mensagens para um acionista em um mercado de ações pode indicar um alto nível de negociações naquela bolsa.

Cenário 1 – Comunicação autenticada por um servidor

- Alice quer acessar arquivos mantidos por Bob em um servidor na rede local onde ela trabalha
- Sara é um servidor de autenticação que é gerenciado de forma segura
- Sara cria usuários com senhas e mantém chaves secretas para todos os principais no sistema que ela serve
- Por exemplo:
  - Sara conhece a chave secreta de Alice KA e Bob KB

Comunicação autenticada por um servidor

- Alice envia uma mensagem não criptografada para Sara, identificando-se e requisitando um ticket para acessar Bob.
- Sara envia uma resposta para Alice criptografada em KA consistindo de um ticket criptografado em KB (para ser enviado a Bob em cada requisição de acesso a arquivo) e uma nova chave secreta KAB para uso enquanto estiver na mesma seção de comunicação com Bob.
  - Então, a resposta que Alice recebe aparece assim:
    - {{Ticket}KB, KAB}KA

Comunicação autenticada por um servidor

- Alice decodifica a resposta usando seu KA.
  - Se o recipiente não for Alice, então ele não saberá a senha de Alice e não será capaz de decifrar a mensagem.
  - Alice não pode decifrar ou modificar o ticket porque ele está criptografado com KB.
- Alice envia o ticket para Bob juntamente com a sua identidade e uma requisição R para acessar um arquivo:
  - {Ticket}KB, Alice, R.

Comunicação autenticada por um servidor

- O ticket originalmente criado por Sara é:
  {KAB, Alice}KB .
  - Bob decifra o ticket usando sua chave KB.
  - Então, Bob consegue a identidade autêntica de Alice e uma nova chave compartilhada secreta KAB para uso quando for interagir com Alice (chamada de *session key* porque pode ser usada seguramente por Alice e Bob em uma seqüência de interações).

Cenário 2 – Uso de chave pública para distribuir chaves privadas compartilhadas

- Bob e Alice estabelecendo uma chave secreta KAB:
- Alice acessa um serviço de distribuição de chaves para obter o certificado de chave pública de Bob.
  - É chamado de certificado porque é assinado por uma autoridade de confiança (*trusted authority*) - uma pessoa ou organização que é amplamente conhecida como confiável.
  - Após verificar a assinatura, Alice lê a chave pública de Bob KBpub do certificado.

Uso de chave pública para distribuir chaves privadas compartilhadas

- Alice cria uma nova chave compartilhada KAB e a codifica usando KBpub.
  - Ela manda a mensagem para Bob juntamente com um nome que identifica unicamente o par de chaves pública/privada (visto que Bob deva possuir várias chaves públicas).
  - Então, Alice envia para Bob:
    - keyname, {KAB}KBpub
- Bob seleciona a chave privada correspondente KBpriv para decifrar KAB.

Controle de Acesso aos Recursos

- Servidores recebem requisições na forma <op,principal,recurso>
- Onde:
  - op: operação requisitada;
  - principal: identidade do principal fazendo o pedido;
  - recurso: identifica o recurso na qual a operação será aplicada.
- Para cada requisição, o servidor autentica o principal e verifica se o mesmo possui direitos de acesso necessários para realizar a operação desejada no recurso especificado.

Referências Bibliográficas

- Coulouris, George; Dollimore, Jean; Kindberg, Tim. Distributed Systems: concepts and design. Third Edition. Addison-Wesley 2001.
- Coulouris, George; Dollimore, Jean; Kindberg, Tim; tradução João Tortello. Sistemas Distribuídos: conceitos e projeto. 4. ed. Bookman 2007.