

Computação Distribuída [Engenharia Informática] - 2023

[Painel do utilizador](#) / [Minhas disciplinas](#) / [Computação Distribuída-7348 \[Engenharia Informática\]-23](#) / [Geral](#) / [Mini-teste #2 - turma de noite](#)

Iniciada quinta, 23 de novembro de 2023 às 21:31

Estado Terminada

Terminada quinta, 23 de novembro de 2023 às 22:45

Tempo gasto 1 hora 13 minutos

Pergunta **1**

Respondida

Nota: 10,00

Como é feito o controlo de acessos num sistema **MAC (Mandatory Access Control)** ? Ilustre com o seguinte exemplo: um cliente está a tentar ler um ficheiro remoto, tendo chamado o serviço: **read (ficheiro, utilizador)**

O controlo de acessos num sistema MAC para o exemplo dado é feito da seguinte forma:

1. O serviço `read()` irá verificar as classes de segurança do ficheiro e do utilizador.
Se o nível de segurança do utilizador for inferior ou igual à classe de segurança do ficheiro,
2. O serviço irá devolver o conteúdo do ficheiro ao utilizador.
3. Caso contrario o serviço não ira permitir o acesso ao ficheiro.

Pergunta **2**

Respondida

Nota: 10,00

Como é feito o controlo de acessos num sistema **DAC (Discretionary Access Control)** ? Ilustre com o seguinte exemplo: um cliente está a tentar ler um ficheiro remoto, tendo chamado o serviço: **read (ficheiro, utilizador)**

O acesso seria feito da seguinte forma

1. Agente pede autorização ao monitor de segurança
2. O monitor verifica se o agente(Utilizador) está ou não autorizado através de uma matriz de direitos e acessos
3. Caso o agente tenha privilégios de leitura para o ficheiro o monitor permite o acesso caso contrario nega o acesso.

Pergunta **3**

Respondida

Nota: 10,00

É boa ideia a segurança de uma comunicação cifrada estar baseada no secretismo do algoritmo de cifra ? Porquê ? Qual é a solução correta ?

Utilizar uma solução de segurança com uma comunicação cifrada baseada no secretíssimo do algoritmo de cifra não é uma boa ideia.

A razão é porque caso o algoritmo seja descoberto pelo atacante este irá conseguir decifrar todas as comunicações efetuadas.

Para resolver esta situação uma solução com chaves privadas e publicas talvez fosse mais correcta.

Pergunta **4**

Respondida

Nota: 10,00

Considere 2 agentes, A e B, cada um com o seu par de chaves pública e privada, respetivamente:

A: KAPub, KAPriv B: KBPub, KBPriv

Considere que A quer enviar a B a mensagem M com **privacidade**. Represente a mensagem que A deve enviar a B e descreva porque é que a mensagem garante privacidade.

Nota: Não percebi muito bem como representar mas penso que o pedido é o seguinte:

A --> B: (M)KBPub

A Mensagem é cifrada por A com a chave publica de B, o que significa que a mesma só pudera ser decifrada com a chave privada de B pois só B conhece a mesma(chave).

Pergunta **5**

Respondida

Nota: 10,00

Considere o programa em SUN RPC abaixo. Diga onde está definida e descreva o funcionamento da função que o servidor exporta para o cliente.

```
/* date.x */  
  
program DATEPROG {  
    version DATEVERS {  
        long BINDATE(void) = 1;  
    } = 1;  
} = 0x3012225;  
  
/* dateproc.c - remote procedures; called by server stub */  
  
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <rpc/rpc.h>  
  
#include "date.h"  
  
long * bindate_1_svc(void* arg1, struct svc_req *arg2) {  
    static long timeval; /* must be static */  
    timeval = time((long *) 0);  
    return (&timeval);  
}
```

O seguinte programa RPC está definido no cliente

O dateproc gera uma data baseada na data atual e devolve essa data.

Pergunta **6**

Respondida

Nota: 10,00

```
public class Server implements Hello {  
    public Server() {}  
    public String sayHello() {  
        return "Hello, world!";  
    }  
    public static void main(String args[]) {  
        try {  
            Server obj = new Server();  
            Hello stub = (Hello)  
                UnicastRemoteObject.exportObject(obj, 0);  
            // Bind the remote object's stub in the registry  
            Registry registry = LocateRegistry.getRegistry();  
            registry.bind("Hello", stub);  
        }  
    }  
}
```

No exemplo de um servidor Java RMI acima, diga para que serve a linha **registry.bind("Hello", stub);**

A linha `registry.bind("Hello", stub);` serve para gravar o objeto remoto em registo, ou seja, a a partir desse momento o cliente pode usar esse objecto como se fosse um objecto local utilizado apenas a referencia obtida através do RMI.

Pergunta **7**

Respondida

Nota: 10,00

Descreva resumidamente a função do seguinte componente do WSDL: PortType

A função da componente WSDL: PortType serve para descrever a funções disponíveis desse service.

Pergunta **8**

Respondida

Nota: 10,00

Descreva 3 ameaças ou ataques à segurança em sistemas distribuídos

3 ameaças ou ataques à segurança de um sistema distribuído são

1. Brute forcing
2. Problemas de Permissões
3. Roubo de credenciais

Pergunta **9**

Respondida

Nota: 10,00

Considere 2 agentes, A e B, uma autoridade de certificação S, e as 4 mensagens do protocolo Needham - Schroeder de autenticação com chave secreta:

1. $A \rightarrow S: A, B, Na$
2. $S \rightarrow A: \{Na, B, Kab, \{Ts, A, Kab\}Kb\}Ka$
3. $A \rightarrow B: \{A, Ta\}Kab, \{Kab, A, Ts\}Kb$
4. $B \rightarrow A: \{Ta + 1\}Kab$

Na mensagem 2, como é que A obtém a chave Kab ? Como é que tem a certeza que a chave é válida ?

A obtém a chave através S utilizando o sistema de ticketing comunica com a autoridade de Certificação(S) e esta valida a sua comunicação, caso seja permitido S entrega uma chave Kab a A

Pergunta **10**

Respondida

Nota: 10,00

Quais são os problemas que se pretendem resolver com os Web Services e porque razão o Sun RPC e o Java RMI não são adequados para resolver esses problemas ?

Os problemas que se pretende resolver com Web Services são:

1. Compatibilidade: ao contrario de Sun RPC e Java RMI, Web Services permitem que os mesmos sejam programados em qualquer linguagem, inclusive que Servidor e Cliente tenham linguagens de programação diferentes. O que torna os Web Services muito mais flexíveis e compatíveis com diversos sistemas de computação distribuídos.
2. Dependência da localização Geográfica: Os Web Services ao contrario de Sun RPC e Java RMI não tem qualquer tipo de dependência na localização geográfica e na timezone. Isto porque funciona de maneira a ser assíncrono, não quer que o cliente/servidor estejam com o mesmo fuso horário nem que usem o mesmo tipo de caracteres.
3. Fácil integração com outros serviços: Os WebServices como tem poucas dependências e são muito flexíveis são facilmente integrados com sistemas já existentes e em produção. Possibilitando assim cruzar informações entre sistemas distribuídos que outrora necessitariam ser substituídos por novos. Os webservices permitem assim reaproveitar sistemas que já estão a ser usados numa empresa.

[Previous Activity](#)[Ir para...](#)

Next Activity