Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança Informática

2024/2025

Trabalho

Época especial

1 Objectivos

A componente de avaliação contínua da disciplina de Segurança Informática pretende familiarizar os alunos com alguns dos problemas envolvidos na programação de aplicações distribuídas seguras, nomeadamente a gestão de chaves criptográficas, a geração de sínteses seguras, cifras e assinaturas digitais, e a utilização de canais seguros à base do protocolo TLS. O trabalho será realizado utilizando a linguagem de programação Java e a API de segurança do Java.

O trabalho tem como objetivo fundamental a construção de uma aplicação distribuída. O trabalho consiste na concretização de um sistema **simplificado** de armazenamento de ficheiros, designado por **myCloud**, onde o utilizador usa um servidor central para armazenar os **seus ficheiros**.

2 Arquitectura do Sistema

O trabalho consiste no desenvolvimento de três programas:

- O servidor *myCloudServer*,
- A aplicação cliente *myCloud* que acede ao servidor via *sockets* TCP, e
- criarUser

A aplicação é distribuída de forma que o servidor fica numa máquina e o utilizador pode usar clientes em máquinas diferentes na Internet.

3 Funcionalidades

O sistema tem os seguintes requisitos:

1. O servidor recebe na linha de comandos a seguinte informação:
   - Porto (TCP) para aceitar ligações de clientes.

2. Cifrar/decifrar ficheiros

   myCloud -u *username* -p *password* -c *nome_de_ficheiro -t username_do_destinatario*

   myCloud -u *username* -p *password* -d *nome_de_ficheiro*

   em que:

   - o *username* e a *password* são utilizados para identificar o utilizador e a *password* de acesso à *keystore* do utilizador
   - assume-se que a *keystore* do utilizador já existe e é designada por keystore.*username*. Por exemplo, o utilizador maria tem uma *keystore* designada por *keystore.maria.* O *username* coincide com o *alias* na *keystore*. A *keystore* de cada utilizador deve ter os certificados necessários para a execução dos comandos. Caso os certificados necessários não existam, o comado deve dar erro. A criação das *keystores* e a importação dos certificados devem ser

realizadas manualmente com o comando *keytool*.

- a opção -c cifra o ficheiro *nome_de_ficheiro* utilizando um mecanismo de cifra híbrida. As chaves assimétricas dos utilizadores estão guardadas na *keystore*. Devem ser utilizadas chaves e algoritmos considerados seguros. A chave cifrada deve ser colocada num ficheiro com nome *nome_de_ficheiro.chave.username_do_destinatario*. O ficheiro cifrado deve ter o nome *nome_de_ficheiro.cifrado*

  Exemplo de utilização:

  myCloud -u maria -p passMaria -c bb.pdf -t silva

  *E*ste comando cifra o ficheiro bb.pdf de modo que apenas o utilizador silva possa decifrá-lo. Este comando cria os seguintes ficheiros: bb.pdf.cifrado e bb.pdf.chave.silva

- a opção -d decifra o ficheiro *nome_de_ficheiro* utilizando a chave guardada no ficheiro *nome_de_ficheiro.chave.username_do_destinatario*
- estas opções do cliente não contactam com o servidor

3. Assinar/validar a assinatura de ficheiros

   myCloud -u *username* -p *password* -a *nome_de_ficheiro*

   myCloud -u *username* -p *password* -v *nome_de_ficheiro*

   em que:

   - a opção -a assina o ficheiro *nome_de_ficheiro*. As chaves assimétricas do utilizador estão guardadas na *keystore*. Devem ser utilizadas chaves e algoritmos considerados seguros. A assinatura do ficheiro deve ser colocada num ficheiro com nome *nome_de_ficheiro.assinatura.username*
   - a opção -v verifica a assinatura do ficheiro *nome_de_ficheiro* utilizando a assinatura guardada no ficheiro *nome_de_ficheiro.assinatura.username.* O nome do ficheiro onde é guardada a assinatura tem a informação de quem assinou o ficheiro.
   - estas opções do cliente não contactam com o servidor

4. Enviar e receber ficheiros do cliente para o servidor e vice-versa

   myCloud -s endereço:porto_do_servidor -u username -p password -e nomes_de_ficheiros

   myCloud -s endereço:porto_do_servidor -u username -p password -r nomes_de_ficheiros

   em que:

   - a opção -s é usada para indicar o endereço IP e o porto do servidor
   - a opção -e é usada **apenas** para enviar ficheiros para o servidor. Os utilizadores partilham todos os ficheiros no servidor. Todos os ficheiros são enviados para a mesma diretoria do servidor. Caso já exista, no servidor, algum ficheiro com o mesmo nome, não deve ser enviado nenhum dos ficheiros.

     Caso se pretenda enviar um ficheiro cifrado e a respetiva chave, o comando deve ser usado da seguinte forma:

     myCloud -s 10.101.149.5:23456 -u maria -p passMaria -e aa.pdf.cifrado aa.pdf.chave.silva

   - a opção -r é usada **apenas** para receber ficheiros do servidor. Na linha de comandos, devem ser indicados todos os ficheiros a receber explicitamente. Por exemplo, para receber os ficheiros

enviados no exemplo anterior será executado o comando:

myCloud -s 10.101.149.5:23456 -u silva -p passSilva -r aa.pdf.cifrado aa.pdf.chave.silva

- a opção -p define a *password* do utilizador para se autenticar no servidor (ver explicação sobre autenticação de utilizadores). Nas opções -s e -e não é necessário aceder a keystores de utilizadores.

5. Autenticação dos utilizadores

O servidor mantém um ficheiro (designado por *users*) com os utilizadores do sistema e respetivas informações. Este ficheiro deve **ser um ficheiro de texto**. Cada linha tem um *username* e a respetiva password (com o *salt*), separador por **:**

Cada linha tem um *username*, o *salt* e a síntese da concatenação do *salt* com a *password*. Os *salts* são gerados de forma aleatória e são diferentes para cada utilizador. As sínteses são calculadas com a classe **MessageDigest**.

```
maria:w9CfDqX9Li5krpdJZgg/Qh:A46KPmM+bClnR5D8URnVAzG9heNbvxop5eQq1leAcuk=
alice:dbqPTW49yNLmOJK4RC:MAOgRGmbTqpwNdI5yIjZJICRG7CvKlRNOozCKx0QsyY=
```

Para adicionar utilizadores ao sistema, deve ser disponibilizado **um comando adicional** com a seguinte sintaxe:

criarUser *nome_de_user password_de_user*

o qual só poder ser usado na máquina do servidor e acede diretamente ao ficheiro dos utilizadores.

6. Integridade do ficheiro das passwords

O servidor deve proteger a **integridade do ficheiro das passwords**. Para tal, o ficheiro deve ser protegido com um MAC. O cálculo deste MAC utiliza uma chave simétrica calculada a partir da *password* do MAC.

No início da sua execução, o servidor pede a password do MAC para verificar a integridade do ficheiro das passwords. Se o MAC estiver errado, o servidor deve imprimir um aviso e terminar imediatamente a sua execução.

O MAC deve **ser verificado em todos os acessos ao ficheiro de passwords e atualizado caso o ficheiro seja alterado (no comando criarUser)**. O comando criarUser também terá de pedir a password do MAC ao utilizador. Para calcular o MAc deve ser usada a classe HMAC.

O MAC deve ser guardado num ficheiro de texto utilizado apenas para este efeito, designado por users.mac.

A chave para ser usada na geração do MAC deve ser obtida a partir da password do MAC do seguinte modo (neste exemplo a password do MAC é maria12:

```
byte [] pass = "maria12".getBytes();
SecretKey key = new SecretKeySpec(pass, "HmacSHA256");
```

O array de bytes deve ser convertido para string do seguinte modo:

```
String s = Base64.getEncoder().encodeToString(bytearray);
```

7. Na comunicação entre o cliente e o servidor pretende-se garantir a autenticidade do servidor (um atacante não deve ser capaz de fingir ser o servidor e assim obter a password de um utilizador) e a confidencialidade da comunicação entre cliente e servidor (um atacante não deve ser capaz de escutar a comunicação). Para este efeito, devem ser usados **canais seguros** (protocolo TLS/SSL). Este protocolo permite verificar a identidade do servidor utilizando chaves assimétricas.

4 Critérios de avaliação

A secção 3 descreve as funcionalidades cuja avaliação corresponderá aos trabalhos 1 e 2 da época normal.

Funcionalidade	Valorização	Validação
Cifrar/decifrar Funciona para ficheiros de qualquer dimensão Erros: file não existe // file existe // keystore Caso não seja possível comprovar o correto funcionamento das operações será classificado com 0	3	
Assinar/validar a assinatura de ficheiros Funciona para ficheiros de qualquer dimensão Erros: file não existe // file existe // keystore // certificado Caso não seja possível comprovar o correto funcionamento das operações será classificado com 0	3	
Enviar e receber ficheiros do cliente para o servidor e vice-versa Funciona para ficheiros de qualquer dimensão Erros: file não existe // file existe // no cliente // no servidor Caso não seja possível comprovar o correto funcionamento das operações será classificado com 0	3	
Autenticação dos utilizadores <u>Formato com user:salt:síntese(salt||password)</u> Sem síntese=0 // usa síntese incorreta/=0 Sem salt = -50% Sem MessageDigest = -50% <u>Autentica</u> Valida password e dá erro ao utilizador Algoritmo termina quando encontra Funciona para ficheiro com qualquer número de utilizadores <u>Criação de user</u> Adiciona user Verifica se já existe + msg de erro Ciclo termina qdo encontra alg funciona c/ qq nº de users	4.5	
<u>Integridade do ficheiro das passwords</u> Calcula o MAC e valida-o corretamente Calcula um novo MAC em todas as situações previstas (início caso necessário, e sempre que há alterações) Valida o MAC em todas as situações previstas (início e em todos os acessos) Apresenta mensagens de erro (quando MAC não existe, MAC incorreto, …) Funciona qualquer que seja a dimensão do ficheiro das passwords	3	
F. TLS/SSL	1.5	
Usam algoritmos não seguros	**Penalização** de 1 valor por cada caso	
Relatório (e entrega dentro do prazo)	1	
Preparação da apresentação para a avaliação e	1	

cumprimento do tempo		

5 Iptables/snort

Fazer a parteI-iptables **ou** a parteII-snort do trabalho 3 da época normal.

6 Entrega

Código e relatórios:

Até dia **17 de Julho**, às 23:55 horas. O código do trabalho deve ser entregue na página da disciplina. Os alunos deverão submeter o código do trabalho num ficheiro zip e um readme (txt) sobre como executar o trabalho.

As avaliações dos trabalhos serão realizadas no dia 18 de Julho.

No relatório da secção 3 devem ser apresentados e discutidos os seguintes aspetos:

- Os objetivos concretizados com êxito
- A tabela de autoavaliação de acordo com os critérios definidos
- Os problemas encontrados.
- A segurança da aplicação criada, identificando possíveis **fraquezas e melhorias** a incluir em versões futuras.

Este relatório deve ter no máximo 5 páginas.

Não serão aceites trabalhos por email nem por qualquer outro meio não definido nesta secção. Se não se verificar algum destes requisitos o trabalho é considerado não entregue.