Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança Informática

2023/2024

Trabalho 1

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.

A primeira fase do 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 **mySNS**, onde o utilizador usa um servidor central para armazenar os **seus ficheiros** referentes a exames e prescrições médicas. Nesta fase vão ser asseguradas apenas as funcionalidades do médico.

Na segunda fase do trabalho serão adicionadas outras funcionalidades que permitirão que vários utilizadores usem o mesmo servidor e partilhem ficheiros entre si. Nesta fase será assegurada a autenticação dos utilizadores e o controlo de acesso.

Por fim, na terceira fase do trabalho serão configurados mecanismos de segurança ao nível do servidor. Será configurada a *firewall* do servidor e utilizado o snort para a detecção de intrusões.

2 Arquitectura do Sistema

O trabalho consiste no desenvolvimento de dois programas:

- O servidor *mySNSServer*, e
- A aplicação cliente *mySNS* que acede ao servidor via *sockets* TCP.

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. O cliente pode ser utilizado com as seguintes opções na linha de comandos:

   mySNS -a <serverAddress> **-m <username do médico> -u <username do utente> -sc {<filenames>}+**

   mySNS -a <serverAddress> **-m <username do médico>-u <username do utente> -sa {<filenames>}+**

   mySNS -a <serverAddress> **-m <username do médico> -u <username do utente> -se {<filenames>}+**

   mySNS -a <serverAddress> **-u <username do utente> -g {<filenames>}+**

   Em que:

   - -a <serverAddress>

   Identifica o servidor (*hostname* ou endereço IP e porto; por exemplo 127.0.0.1:23456).

   - -u <username do utente>

Identifica o nome do utente para o qual se destinam os ficheiros. Nesta fase do trabalho é usado, por exemplo, para identificar o *alias* do utilizador na *keystore*.

- -m <username do médico>

Identifica o nome do médico que, por exemplo, assina os ficheiros.

- -sc {<filenames>}+

O cliente cifra um ou mais ficheiros e envia-os para o servidor. Caso algum dos ficheiros já exista no servidor ou não exista localmente, apresenta uma mensagem de erro ao utilizador e continua para os ficheiros seguintes. O cliente usa **cifras híbridas**. Assim, a chave usada para cifrar cada ficheiro deve ser cifrada no cliente e enviada para o servidor.

Cada uma destas chaves pode ser guardada num ficheiro cujo nome deve ser o nome do ficheiro original com extensão chave_secreta.username_do_utente

Os ficheiros cifrados são guardados no servidor com extensão cifrado.

Por exemplo:

mySNS -a 127.0.0.1:23456 -m silva -u maria -sc exame1.png relatorio1.pdf

envia para o servidor os ficheiros cifrados e armazena-os no servidor, com os nomes exame1.png.cifrado e relatorio1.pdf.cifrado, na diretoria do utilizador maria; e

envia para o servidor as chaves cifradas e armazena-as nos ficheiros *exame1.png.chave_secreta.maria* e *relatorio1.pdf.chave_secreta.maria* na diretoria do utilizador maria.

O médico deve ter na sua *keystore* o certificado da maria. Estes certificados devem ser colocados previamente na *keystore* dos utilizadores, manualmente. As *keystores* dos utilizadores devem ter um nome com o formato *silva.keystore*, por exemplo para o caso do utilizador silva.

- -sa {<filenames>}+

o cliente assina um ou mais ficheiros e envia-os para o servidor. Caso algum dos ficheiros já exista no servidor ou não exista localmente, apresenta uma mensagem de erro ao utilizador e continua para os seguintes. As assinaturas devem ser guardadas separadamente em ficheiros com extensão *assinatura.username_do_medico*. Os ficheiros assinados são guardados no servidor com extensão assinado (este ficheiro é exatamente igual ao original).

Por exemplo:

mySNS -a 127.0.0.1:23456 -m silva -u maria -sa exame2.png relatorio2.pdf

envia para o servidor os ficheiros exame2.png e relatorio2.pdf, e armazena-os no servidor, com os nomesexame2.png.assinado e relatorio2.pdf.assinado na diretoria do utilizador maria, e

e envia para o servidor as assinaturas, armazenando-as nos ficheiros exame2.png.assinatura.silva e relatorio2.pdf.assinatura.silva na diretoria do utilizador maria.

- -se {<filenames>}+

o cliente assina e cifra um ou mais ficheiros e envia-os para o servidor. Caso algum dos ficheiros já exista no servidor ou não exista localmente, apresenta uma mensagem de erro ao utilizador e continua para os seguintes. O cliente usa **envelopes seguros (para simplificar o trabalho, os alunos não precisam de cifrar a assinatura)**. Os ficheiros são guardados no servidor com extensão seguro.

Por exemplo:

mySNS -a 127.0.0.1:23456 -m silva -u maria -sa exame3.png relatorio3.pdf

envia para o servidor os ficheiros exame3.png e relatorio3.pdf cifrados e armazena-os no servidor, em ficheiros com os nomes exame3.png.seguro e relatorio3.pdf.seguro e

envia as assinaturas – idêntico à opção -sa

envia as chaves – idêntico à opção -sc

- -g {<filenames>}+

  o cliente recebe um ou mais ficheiros. Caso algum dos ficheiros já exista localmente ou não exista no servidor, apresenta uma mensagem de erro ao utilizador e continua para os seguintes.

  O cliente decifra os ficheiros que tenham sido cifrados.

  O cliente verifica a assinatura dos ficheiros que tenham sido assinados.

  Sugere-se que esta opção seja realizada incrementalmente, à medida que sejam realizadas as outras opções. Ou seja, quando a opção -c for realizada, sugere-se que seja realizada a parte da opção -g que trata os ficheiros cifrados.

Toda criptografia assimétrica no trabalho deve usar RSA com chaves de 2048 bits. A criptografia simétrica deve ser efetuada com AES e chaves de 128 bits.

Os utilizadores devem ter um par de chaves na sua keystore. As keystores devem ser criadas previamente através do comando keytool.

A unicidade dos nomes dos ficheiros deve ser assegurada qualquer que seja a opção com que foram enviados, para o mesmo destinatário (utente), ou seja, caso um ficheiro seja enviado com a opção sc para a maria, caso volte a ser enviado para a maria com a opção sc ou outra qualquer, o sistema deve dar erro.

4 Avaliação

A avaliação dos projetos será feita segundo uma abordagem funcional, onde cada funcionalidade descrita no enunciado deve ser apresentada pelos alunos. **Não serão consideradas funcionalidades incompletas ou valorizada qualquer implementação não funcional.** É obrigatório apresentarem os projetos em duas máquinas distintas do laboratório (servidor e cliente em máquinas separadas). De preferência, devem utilizar o sistema operativo Linux.

Cada uma das opções/funcionalidades apresentadas será valorizada de acordo com a seguinte tabela.

Funcionalidade	**Valorização**	**Validação**
Opção sc		
Cifra um ficheiro e envia (podem mostrar que funciona com o decifra da aula ou com a opção -g). Para terem a certeza que funciona devem usar duas máquinas distintas para o cliente e para o servidor. Verifica erros locais e avisa o utilizador (ficheiros não existem, keystore, etc) Verifica erros remotos (servidor) e avisa o utilizador (ficheiros já existem, etc)	1 0.25 0.25	
Cifra vários ficheiros e envia (podem mostrar que funciona com o decifra da aula ou com a opção -g). Verificação de erros locais e remotos para vários ficheiros	1 0.25+0.25	
Cifra e envia ficheiros de qualquer dimensão.	1	
Não usam chaves e algoritmos seguros	**Penalização** de até 2	
Preparação da apresentação para a avaliação e cumprimento do tempo	0.5	
Opção ss		
Assina um ficheiro e envia (podem mostrar que funciona com o exercício da aula ou com a opção -g). Para terem a certeza que funciona devem usar duas máquinas distintas para o cliente e para o servidor. Verifica erros locais e avisa o utilizador (ficheiros não existem, keystore, etc) Verifica erros remotos (servidor) e avisa o utilizador (ficheiros já existem, etc)	1 0.25 0.25	
Assina vários ficheiros e envia (podem mostrar que funciona com o	1	

exercício da aula ou com a opção -g). Verificação de erros locais e remotos para vários ficheiros	 0.25+0.25	
Assina e envia ficheiros de qualquer dimensão.	1	
Não usam chaves e algoritmos seguros	**Penalização** de até 2	
Preparação da apresentação para a avaliação e cumprimento do tempo	0.5	
Opção se		
Cifra, assina o ficheiro e envia (podem mostrar que funciona com o exercício da aula ou com a opção -g). Para terem a certeza que funciona devem usar duas máquinas distintas para o cliente e para o servidor. Verifica erros locais e avisa o utilizador (ficheiros não existem, keystore, etc) Verifica erros remotos (servidor) e avisa o utilizador (ficheiros já existem, etc)	1 0.25 0.25	
Cifra e assina vários ficheiros e envia (e funciona – podem mostrar que funciona com o exercício da aula ou com a opção -g). Verificação de erros locais e remotos para vários ficheiros	1 0.25+0.25	
Cifra e assina e envia ficheiros de qualquer dimensão.	1	
Não usam chaves e algoritmos seguros	**Penalização** de até 2 valores	
Preparação da apresentação para a avaliação e cumprimento do tempo	0.5	
Opção g		
Recebe um ficheiro e decifra e/ou valida assinatura. Para ficheiros cifrados Para ficheiros assinados Para ficheiros cifrados e assinados Verifica erros locais e avisa o utilizador (ficheiros já existem, keystore, etc) Verifica erros remotos (servidor) e avisa o utilizador (ficheiros não existem, etc)	 1 0.5 0.5 0.25 0.25	
Recebe vários ficheiros e decifra e/ou valida assinatura (cifrados, assinados, cifrados e assinados). Verificação de erros locais e remotos para vários ficheiros	0.5+0.25+0.25 0.25+0.25	
Recebe vários ficheiros de qualquer dimensão (cifrados, assinados, cifrados e assinados).	0.5+0.25+0.25	
Preparação da apresentação para a avaliação e cumprimento do tempo	0.5	
Relatório: Identificação do Grupo (apenas os alunos que participaram na realização do trabalho), esta tabela preenchida, opções consideradas relevantes na concretização do trabalho.	1	

5 Relatório

O relatório será entregue no moodle**.**

6 Entrega

Código:

Dia **7 de Abril**, até às 20:00 horas. O código do trabalho deve ser entregue da seguinte forma:

- Os grupos devem inscrever-se atempadamente de acordo com as regras afixadas para o efeito, na página da disciplina.
- Na página da disciplina submeter o código do trabalho num ficheiro zip e um readme (txt) sobre como executar o trabalho.

Relatório:

Dia **7 de Abril**, até as 23:55 horas, no moodle.

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.