Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança Informática

2021/2022

Trabalho 2

1 Objetivos

Esta fase do trabalho estende a anterior, possibilitando aos alunos experimentarem diversos mecanismos de segurança, tais como: cifras, assinaturas, comunicação com um protocolo seguro (TLS – Transport Layer Security) e gestão básica de certificados.

A envolvente do trabalho continua a ser a mesma, ou seja, a concretização de um sistema de assinatura digital, designado por myAutent. O trabalho será realizado utilizando a linguagem de programação Java e a API de segurança do Java, e ferramentas complementares. Neste trabalho, tal como no anterior, o cliente e o servidor devem ser executados dentro da sandbox.

De modo a **alargar o uso de assinaturas digitais, e equilibrando os requisitos de usabilidade e de segurança,** este sistema guarda no servidor as chaves assimétricas dos utilizadores.

O servidor gera e verifica as assinaturas digitais dos ficheiros.

2 Modelo de adversário

Iremos assumir no trabalho que existe um adversário que pretende comprometer o correto funcionamento do sistema. O adversário terá um conjunto de capacidades que poderão ser empregues na realização das suas ações maliciosas. Torna-se assim necessário dotar o sistema dos mecanismos de proteção que lhe possibilitem manter um funcionamento correto ainda que se encontre sob ataque.

Vamos assumir que o adversário tem as seguintes capacidades:

- Acesso à rede : tendo o adversário acesso à rede, poderá escutar os pacotes trocados entre o cliente e o servidor. Potencialmente, também poderá tentar corromper, alterar, introduzir, e reproduzir mensagens de forma a enganar quer o cliente quer o servidor.

- Controlar um ou mais utilizadores : o adversário controla uma (ou mais) conta(s) de utilizadores do sistema. Através desta(s) conta(s), ele poderia tentar aceder a ficheiros para os quais não tem permissões ou corromper ficheiros com informação de outros utilizadores.

- Acesso à máquina que executa o servidor em modo de leitura : o adversário tem acesso em modo de leitura aos ficheiros armazenados no servidor. Com esse acesso, ele pode potencialmente observar informação que eventualmente seria confidencial.

Em seguida indicam-se e discutem-se as proteções que devem ser adicionadas ao sistema.

3 Alterações a adicionar ao sistema

Nesta fase, os alunos devem usar a mesma arquitetura da 1ª fase. De seguida são descritas as alterações a adicionar ao sistema.

3.1. Opções dos clientes:

myAutentClient -u <userId> -a <serverAddress> [-p <password>] -c <userId> <nome> <password>

- A criação de um novo utilizador inclui adicionalmente a criação de um par de chaves RSA para este utilizador. As chaves são guardadas numa *keystore* do utilizador no **servidor**. A *password* de acesso à *keystore* e a *password* de acesso à chave privada deve coincidir com a *password* do utilizador.

myAutentClient -u <userId> -a <serverAddress> [-p <password>] -l

- Esta opção não sofre alterações.

myAutentClient -u <userId> -a <serverAddress> [-p <password>] -e {<filenames>}+

- envia para o servidor um ou mais ficheiros. Caso algum dos ficheiros já exista no servidor ou caso algum dos ficheiros não exista localmente, apresenta uma mensagem de erro ao utilizador.

  O servidor cria a assinatura digital de cada ficheiro com a chave do utilizador**, guarda-a** localmente **e envia-a** para o cliente.

  O cliente e o servidor guardam as assinaturas em ficheiros com extensão *signed.alias*, **em que *alias* corresponde ao *alias* do utilizador que assinou o ficheiro**. Adicionalmente, o servidor assegura a confidencialidade destes ficheiros – ver secção 3.2.

myAutentClient -u <userId> -a <serverAddress> [-p <password>] -s {<filenames>}+

- o cliente gera a síntese de cada ficheiro e envia para o servidor **apenas** as sínteses. O servidor cria as assinaturas digitais com a chave do utilizador e envia-as para o cliente. O cliente guarda cada assinatura num ficheiro com extensão *signed.alias*.

myAutentClient -u <userId> -a <serverAddress> [-p <password>] -d {<filenames>}+

- o cliente recebe do servidor um ou mais ficheiros e respetivas assinaturas digitais. Devem ser previstos tipos de erros idênticos aos da opção -e.

myAutentClient -a <serverAddress> -v {<filenames>}+

- para cada ficheiro, o cliente gera a sua síntese e envia para o servidor a síntese gerada e a assinatura digital.

  O servidor verifica as assinaturas e envia o resultado dessa verificação para o cliente.

3.2 . O servidor deve concretizar os seguintes mecanismos de segurança:

A. O servidor também deve proteger a **confidencialidade das passwords**. Com este objetivo, as passwords devem ser armazenadas utilizando mecanismos baseados em algoritmos de sínteses e *salts*. O ficheiro das passwords deve manter o mesmo formato do trabalho 1.

B. 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 admin que é pedida ao utilizador quando **inicia a execução do servidor**.

No início da sua execução, o servidor deve usar o MAC para verificar a integridade do ficheiro. Se o MAC estiver errado, o servidor deve imprimir um aviso e terminar imediatamente a sua execução. Se não há MAC a proteger o ficheiro, o servidor deve imprimir um aviso e perguntar ao utilizador se pretende calcular o MAC, adicionando-o ao sistema. O MAC deve **ser verificado em todos os restantes acessos ao ficheiro e atualizado caso o ficheiro seja alterado**. O MAC pode ser guardado num ficheiro utilizado apenas para este efeito.

Caso o ficheiro de passwords não exista, deve ser criado com o utilizador admin.

C. 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.

D. Os ficheiros dos utilizadores armazenados no servidor devem ser protegidos de eventuais ataques que possam ocorrer na máquina servidora, nomeadamente que tenham em vista **observar o seu conteúdo**. Para isso, deverão utilizar criptografia híbrida com as chaves assimétricas dos utilizadores.

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

4 Relatório e discussão

No relatório devem ser apresentados e discutidos os seguintes aspetos:

- Os objetivos concretizados com êxito
- Os problemas encontrados.
- A segurança da aplicação criada, identificando possíveis **fraquezas e melhorias** a incluir em versões futuras.

O relatório deve ter no máximo 5 páginas.

5 Entrega

Código. Dia **1 de Maio**, até as 23:55 horas. O código do trabalho deve ser entregue da seguinte forma:

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

Relatório. Dia **2 de Maio**, até as 12:00 horas.

1. Na página da disciplina submeter o relatório num ficheiro pdf.

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.

6 Avaliação dos Trabalhos

As melhorias ao trabalho 1 serão valorizadas da seguinte forma:
Nota final do trabalho 1 = 50% da nota da 1ª entrega + 50% da nota da versão melhorada entregue com o trabalho 2

Caso os alunos tenham efetuado melhorias ao trabalho 1 devem identificá-las e explicá-las numa secção do relatório do trabalho 2.

A avaliação dos trabalhos e dos alunos será efetuada na última semana de aulas. A avaliação dos alunos será individualizada. Os trabalhos serão avaliados nos computadores dos laboratórios do DI utilizando mais do que uma máquina de modo a assegurar que os sistemas de ficheiros dos clientes e do servidor são distintos.