Segurança de Sistemas Computacionais

Trabalho Prático nº 2 (Versão inicial v0.9)

Data de Entrega:

Data limite 2/Junho/2019, 23h59

Notas:
À semelhança do Trabalho Prático nº 1 (TP1) este trabalho será entregue através do formulário que será disponibilizado e anunciado através de mensagem do CLIP a partir de 22/Maio/2019, podendo ser submetido a partir de 29/Junho 0h01 e até 2/Junho 23h59. Também à semelhança do TP1, após a avaliação inicial do trabalho, os alunos podem ser chamados a realizar demonstração e discussão em laboratório, em horários que serão anunciadas até 5/Junho. As slots de discussão para alunos escalonados terão lugar no período entre 3 e 12 de Junho de 2019.
Os alunos devem utilizar este enunciado, a partir do qual devem completar as especificações de acordo com as clarificações que entendam solicitar e discutir com o docente, tendo em vista as suas implementações específicas que serão depois reportadas em relatório com template fornecido. Na submissão do trabalho os alunos deverão ainda preencher uma ficha de referencia da implementação, cujo template será igualmente fornecido.

Resumo

O objetivo do trabalho é implementar **um sistema seguro para acesso e gestão remota de ficheiros, com segurança de comunicação baseada em TLS e com suporte acrescido de privacidade**. O sistema envolve a concepção e implementação de diversos módulos da solução, nomeadamente: um módulo de controlo de despacho de operações disponibilizadas ao cliente (através de uma API que pode concretizar um *endpoint* de serviço REST ou REQUEST/REPLY suportados em sockets TLS), um módulo de autenticação de utilizadores, um módulo de controlo de acessos e um módulo de gestão do armazenamento. Todas as comunicações entre todos os módulos do sistema serão protegidas por TLS, sendo o ambiente TLS parametrizável para utilização em modo de autenticação mútua, diferentes *ciphersuites* (com referencia base às parametrizações consideradas seguras e que possam ser suportadas e habilitadas em TLS v1.2). O suporte acrescido de privacidade permitirá que o repositório gerido pelo módulo de armazenamento de ficheiros possa ser concebido de forma a poder ser futuramente outsourced num sistema replicado de armazenamento CLOUD (não obstante esta concretização não ter lugar nos requisitos do trabalho), que emulará o ambiente através de um repositório em sistema de ficheiros usado por aquele módulo.

1. introdução

A arquitetura do sistema e os vários módulos da solução, de acordo com a funcionalidade a suportar, serão desenvolvidos em duas fases, sendo a primeira fase obrigatória para efeitos de avaliação (sendo avaliada de 0 a 14 pontos em escala de 20) e a segunda fase opcional, mas valorizando o trabalho até 5 pontos de acordo com os requisitos a cobrir nessa fase. Esta valorização será adicionada à implementação completa e correta dos objetivos realizados na fase 1. Cada fase está assim associada à implementação de objetivos específicos, como se descrevem a seguir.

2. Entrega do trabalho

A entrega do trabalho envolve:

a) entrega da implementação (código fonte do projeto de implementação, com todos os componentes necessários de modo a ser avaliado, testado e ensaiado em comprovação experimental)
b) Ficha-resumo da implementação (com base em *template* a disponibilizar para efeitos de submissão e entrega do trabalho) e que será submetido na entrega pelos alunos em formato PDF, como componente do formulário da entrega
c) Relatório sobre a implementação, com base em *template* de referência a disponibilizar para efeitos de submissão e entrega do trabalho) e que será submetido na entrega pelos alunos em formato PDF, como componente do formulário da entrega

Para efeitos de metodologia de desenvolvimento do trabalho, as sucessivas fases, suas funcionalidades e os serviços de segurança associados, podem ser desenvolvidas progressivamente, em diferentes pacotes autónomos de implementação dos objetivos de cada fase. Tal permitirá ir atendendo os requisitos de forma incremental, podendo cada fase constituir uma implementação de entrada para extensão na fase seguinte, até à entrega do trabalho para efeitos de avaliação.

3. Arquitetura de referência do sistema a desenvolver para Fase 1

O sistema deve ser concebido com base numa arquitetura distribuída de referência modelo cliente/servidor), representada na figura 1. Na arquitetura inicial (relativa à fase 1) existirão as seguintes entidades do modelo do sistema: **Client**e e **servidor de ficheiros (FServer)**. O **FServer** (que implementa o ponto de acesso ao serviço) funciona como um gateway que implementa o serviço de acesso remoto a ficheiros, sendo este constituído por mais três módulos principais: **um módulo de autenticação (FServerAuth), um módulo de controlo de acesso (FServerAccessControl)** e **um módulo de gestão de armazenamento de dados (FServerStorage).** Este último módulo implementará o sistema de armazenamento dos ficheiro no sistema de ficheiros local da máquina e que executar. Os anteriores módulos da solução constituem servidores independentes e autónomos, podendo ser distribuídos em diferentes máquinas (por exemplo, em diferentes instâncias de sistemas operativos virtualizados, com base em ambiente VirtualBox ou VMware. O deployment da solução deve ser flexível, de modo que os vários servidores possam ser colocados em operação na mesma ou e diferentes máquinas (ex., máquinas virtuais), mas podendo também executar na mesma máquina, para efeitos de desenvolvimento e comprovação experimental inicial.

Na arquitetura de sistema distribuído, os clientes e o **FServer** comunicam por TCP/IP, sendo as interações protegidas por TLS (ou SSL), com base em *sockets* SSL (usando o suporte Java JSSE). O suporte TLS constituirá uma camada de segurança transparente do protocolo do nível aplicação, propiciando as propriedades de segurança de transporte/sessão seguros com flexibilidade de parametrização, de acordo com as parametrizações de configuração permitidas pelo suporte Java JSSE, com todas as parametrizações que permitam que a interação se faça da seguinte forma:

- Todos os componentes devem ser suportados em interações TLS, com possibilidade de autenticação unilateral do lado servidor ou autenticação mútua, devendo conter as necessárias *keystores* ou *trusted stores*, de acordo com o *setup* para comprovação experimental.
- Os *endpoints* de configuração (do lado servidor) apenas permitirão estabelecer sessões a partir da configuração de um ficheiro que estipulará configurações da seguinte forma (exemplo):
-

TLS-PROT-ENF <tls protocol enforced>
TLS-AUTH: <auth-type>
CIPHERSUITES: <csuite options>

Em que <tls protocolo enforced> poderá ser: TLS-1.2 ou TLS-1.1
<auth-type> poderá ser: SERV ou MUTUAL
<csuite options> é uma lista de configurações expressadas na forma:
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

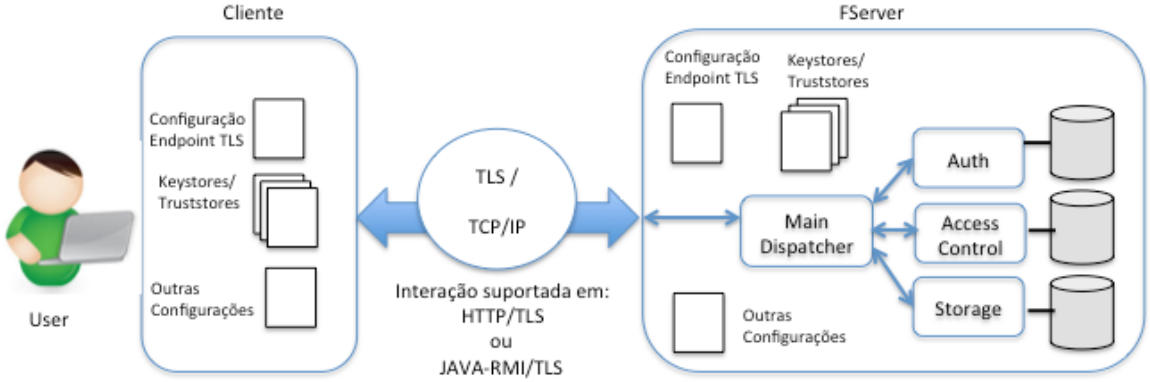

Fig. 1 Arquitetura e entidades do sistema (Fases 1 e 2)

A interação ao nível aplicação entre o cliente e o endpoint FServer (*Main Dispatcher*) e entre este e os restantes servidores, poderá ser suportada numa das seguintes opções:

- Opção REST/HTTP/TLS: corresponderá a desenvolver os servidores como *endpoints* de acesso REST suportados em TLS, com configuração sensível a ficheiros de configuração como indicados acima, de forma a impor as parametrizações de segurança que são requeridas na comunicação TLS.
- Opção Sockets TLS: corresponderá a desenvolver os servidores como *endpoints* de acesso REST suportados em TLS, com configuração sensível a ficheiros de configuração como indicados acima.
- Opção híbrida: que corresponde a uma possível solução híbrida entre as anteriores.

Os alunos deverão adoptar desde o início uma das opções acima para concepção e desenvolvimento do trabalho. Preferencialmente a opção escolhida deverá depois ser usada como referencia para toda a implementação do trabalho.

Na arquitetura indicada na figura 1, o ficheiro de configuração de ciphersuites TLS nos endpoints do lado servidor devem designar-se: **servertls.conf**. Todos os restantes ficheiros de configuração podem ter nomes escolhidos pelos alunos, sendo depois estes explicados no relatório e submissão do trabalho, sendo a flexibilidade e transparência das configurações para os diferentes modos de operação TLS um factor qualitativo de avaliação do trabalho.

2. Operação do sistema pra a Fase 1

Clientes

Os clientes permitem que os utilizadores possam ter acesso ao serviço de ficheiros remoto, permitindo autenticar os utilizadores e verificar o controlo de acesso e depois guardar, gerir e aceder a ficheiros no repositório de ficheiros remoto implementado no *filesystem* local do sistema onde executa o **FServer**. Os ficheiros serão organizados numa arquitetura hierárquica (de diretórios) a partir de uma raíz no sistema de ficheiros local do **FServer.** Nessa hierarquia, cada utilizador, corresponderá a uma subárvore, cuja raiz corresponde ao *username* de cada utilizador suportado.

Deste modo, do lado do **FServer**, qualquer ficheiro pode sempre ser designado remotamente na forma

/username/dir1/dir2/dir3/file

Em que *username* corresponde a um diretório *home* do utilizador "*username*", na raiz da hierarquia do sistema de ficheiros remoto gerido pelo **FServer**, através do servidor **Storage**.

Autenticação do cliente. Ao iniciar um cliente, cada utilizador deve autenticar-se com um par <"*username*","*password*"> ou <"email" "password"> sendo a autenticação dos clientes suportada pelo módulo de autenticação do **FServer**. O protocolo de autenticação entre cliente e **AServer** poderá ser implementado em duas opções possíveis (sempre protegido por TLS ou SSL):

- Uma opção corresponde a implementar um protocolo da fase de autenticação do sistema *Kerberos* (bastando que seja baseado no protocolo de autenticação base *Kerberos V5*). Notar que esta interação estará protegida por sua vez pelos serviços de segurança da camada TLS (SSL).
- Outra opção corresponde a usar um protocolo com base num acordo *Diffie Hellman*, como a seguir se indica:

  Cliente > FServer : username
  FServer > Cliente:
  SecureRandom_1 || assinatura de Yaserver
  Cliente > AServer:
  { H (PWD || SecureRandom_1+1 } Ks || SecureRandom_2 || Assinatura de Ycliente
  AServer > Cliente: {Assinatura (A || Ktoken1024 || TTL) || SecureRandom_2+1}Ks

Em que:
Yaserver e Ycliente são parâmetros públicos de um acordo *Diffie-Hellman*, assinados (autenticados) pelo cliente e pelo **AServer**
Ks: uma chave gerada com base no acordo
Ktoken: Chave *token* para acesso ao **FServer** assinado por **AServer**. Notar que este *token* será válido durante um TTL (from: data; to: data) como credencial autenticada de acesso por A, ao **FServer**.

Após autenticação bem sucedida, será controlado o acesso do cliente pelo módulo *Access Control*, para todas as operações que o cliente pretende realizar.

O cliente disponibilizará ao utilizador a seguinte funcionalidade:

- Login username password (obrigatório fase 1)
- ls username (obrigartório na fase 1)
  // mostra ficheiros ou diretórios do utilizador *username* na sua home-root
- ls username path (opcional na fase 1)

// mostra ficheiros ou diretórios do utilizador *username* na path indicada, sendo esta especificada na forma /a/b/c, a partir do directório home do utilizador.

- mkdir username path (opcional na fase 1)
  // cria um directório na path indicada
- put username path/file (obrigatório fase 1 com um único diretório home do utilizador)
  // coloca um ficheiro file na path indicada
- get username path/file (obrigatório fase 1 com um único diretório home do utilizador)
- // obtem ficheiro file na path indicada
- cp username path1/file1 path2/file2 (
  (obrigatório fase 1 com um único diretório home do utilizador)
  // copia ficheiro file 1 na path 1 para file 2 na path 2
- rm username path/file (obrigatório fase 1 com um único diretório home do utilizador)
  // apaga ficheiro file na path indicada
- rmdir username path (opcional na fase 1)
  // apaga ficheiro file na path indicada
  // remove a path indicada se não houverem ficheiros, senão devolve erro
- file username path/file (opcional na fase 1)
  // Mostra atributos do ficheiro file na path indicada, devolvendo o nome, se é um diretório ou se é um ficheiro, o tipo de ficheiro, data da criação, data da última modificação

FServerAuth

Este módulo gere uma tabela de autenticação (de estrutura inspirada no ficheiro /etc/passwd de um sistema UNIX, Linux ou Mac OSX). Cada entrada está associada a um utilizador, como se ilustra a seguir:

hj:hj@fct.unl.pt:Henrique Domingos:**********:TRUE

Os campos (neste caso separados por ":") possuem respetivamente a seguinte informação:

- Username:Email:Nome:**********:BOOLEAN
- O campo ********* possui a password (ou uma sua transformação, por exemplo uma síntese da mesma) cifrada por uma chave simétrica. Trata-se de um segredo partilhado com o cliente. No cliente, pode usar-se um esquema do tipo PBEEncryption para implementar o protocolo de autenticação.
- O valor booleano indica que:
  - TRUE: o utilizador pode ser autenticado
  - FALSE: o utilizador está bloqueado e não pode ser autenticado (não podendo assim usar o sistema)

FServerAccessControl

Este componente deve implementar uma política de controlo de acesso. Para tal usará um ficheiros de configuração que implementa uma tabela de controlo de acesso do tipo discricionário.

- Especificação do ficheiro access.conf:
  username1: deny // Utilizador não autorizado a ler ou escrever ficheiros
  username3: allow read // só pode ler ficheiros
  username4: allow read write // pode ler e escrever

- Qualquer *username* que não conste do ficheiro *access.conf*, não poderá ter acesso ao serviço. Neste ficheiro is usernames podem ser expressos pelos identificadores ou endereços Email,

conforme estão configurados na tabela de autenticação gerida pelo módulo FServerAuth.

- Notar que o módulo **FServerAccessControl** implementará o conceito associado ao princípio de mediação completa, com a respetiva gestão de privilégios. Por outro lado, nenhum acesso a ficheiros por parte de utilizadores pode ser feito sem o necessário escrutínio da correta autenticação (via **FServerAuth**) e das reras de controlo de acesso (via FServerAccessControl), o que deverá ser coordenado pelo módulo **FServer**.

FServerStorage
Este módulo implementa o serviço de armazenamento de ficheiros.

- É o módulo que gere o filesystem remoto como pretendido. A sua função é criar - escrever ou ler ficheiros (ou criar diretórios), suportando as operações solicitadas pelos utilizadores remotos (operações solicitadas via o endpoint FServer).
- Tem um ficheiro local de configuração com tabelas que definem a raiz do *file-system* remoto (sob a qual estão os diretórios de ficheiros dos utilizadores numa estrutura hierárquica), tendo por base uma diretoria no filesystem.

3. Arquitetura de referência do sistema a desenvolver para Fase 2

A fase 1 envolve extensões à arquitetura inicialmente desenvolvida na fase 1. Na fase 2, para além da extensão da funcionalidade dos anteriores módulos na passagem da fase 1 para a fase 2, a arquitetura irá incorporar alguns módulos adicionais: um módulo de indexação no cliente que permitirºa que todos os ficheiros sejam armazenados no módulo de Storage de forma a que os ficheiros serão armazenados pelo módulo de storage com as seguintes garantias a fornecer aos clientes:

- Confidencialidde: os ficheiros geridos e armazenados, são mantidos e operados sempre cifrados;
- Integridade: os ficheiros estão armazenados de forma a ficar preservada e detectável a sua integridade;
- Autenticidade: os ficheiros serão armazenados com provas de assinatura digital do dono desses ficheiros

Por outro lado, o módulo de armazenamento será estendido por um módulo proxy (**FServerProxy**) que permitirá ao componente **FServerStorage** usar repositórios de dados outsourced - *Cloud* (ex., *Dropbox, AmazonEC3, GoogleDrive,* ou mesmo um ou mais serviçso de Cloud Storage) em vez do sistema de ficheiros local usado pelo módulo **FServerStorage**. Para esse efeito, o módulo tratará de replicar e manter os ficheiros em diferente repositórios, o que aumentará as garantias de disponibilidade e tolerância a falhas do armazenamento. Neste contexto apenas se prevê tolerar falhas por paragem (*fail-stop only model*) em alguma réplica do sistema de armazenamento
Os requisitos da Fase 2 serão inicialmente apresentados e discutidos na aula (semana 6 a 9 de Maio/2019), sendo incluídas posteriormente linhas de orientação da implementação na versão 1.0 deste enunciado.

A figura 2 constitui uma referencia inicial da arquitetura para a concepção e implementação da fase 2 do trabalho.

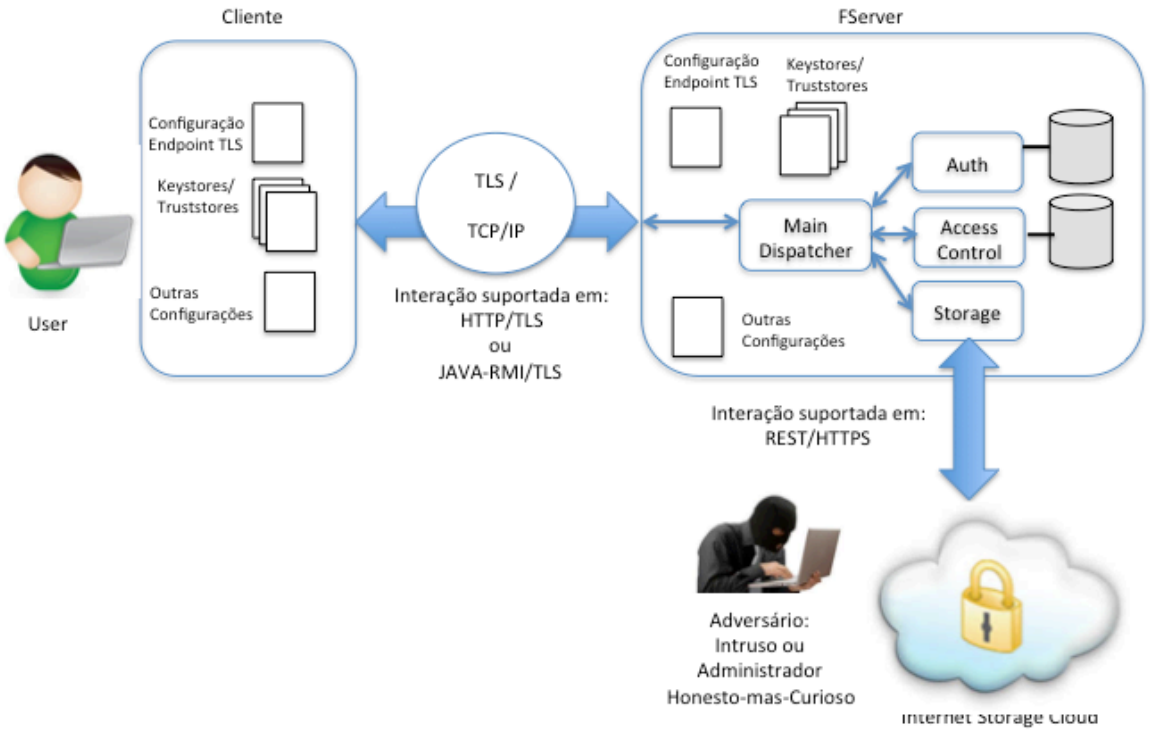

Figura 2. Arquitetura estendida para utilização de uma solução Internet Storage Cloud

4. Operação e demonstração do sistema com implementação na Fase 2

A operação e demonstração do sistema, conforme implementações por parte dos grupos será discutida em aula, consoante a concepção de cada grupo. Algumas linhas de orientação comuns ser incluídas na versão 1.0 do enunciado.

5. Critérios de avaliação

A referencia de critérios de avaliação é a seguinte:
FASE 1: de 0 a 14 valores (sendo o máximo obtido com a completude, correção e qualidade do trabalho, bem como a sua entrega nos prazos estabelecidos e assim sem penalizações).

FASE 2: de 0 a 6 valores (sendo o máximo obtido com a completude, correção e qualidade do trabalho, bem como a sua entrega nos prazos estabelecidos e assim sem penalizações).