Concórdia
Serviço Local de Troca de Mensagens

2024-04-18

1 Descrição geral do serviço

Pretende-se desenvolver em C um serviço de conversação entre utilizadores locais de um sistema Linux[1].

Do ponto de vista funcional, o serviço deverá suportar o envio de mensagens para um utilizador, bem como a sua leitura pelo respetivo destinatário.

O serviço deverá suportar comunicação assíncrona, como no caso do mail, mas também poderá querer suportar comunicação síncrona, como acontece como acontece nos serviços de conversação instantânea. Por uma questão de simplicidade, sugere-se que considere que o tamanho das mensagens nunca excede os 512 caracteres.

O serviço deverá também suportar a gestão de utilizadores e de grupos de utilizadores do serviço. Em concreto, deverá permitir a adição e remoção de utilizadores (sempre locais ao sistema).

O serviço deverá também suportar a noção de grupos privados de conversação, Nesse sentido, deverá oferecer mecanismos para criação de grupos, remoção de grupos e de gestão dos seus membros.

2 Aspetos de segurança do serviço

Tendo em conta que o foco desta unidade curricular é no aspeto de segurança de sistemas informáticos, as equipas deverão ter como preocupação central a proteção da confidencialidade e integridade das mensagens trocadas entre os utilizadores, bem como, obviamente, a disponibilidade do próprio serviço.

No sentido de exercitar os mecanismos de controlo de acesso do Linux, as equipas deverão suportar – pelo menos em parte – o seu modelo de segurança no modelo de segurança do próprio sistema operativo. Ou seja, deverão ter em conta: as noções de utilizador e grupo de utilizadores Linux; as permissões definidas para cada objeto do sistema de ficheiros necessário ao funcionamento do serviço; as noções de processo e a eventual execução de programas com os privilégios associados ao utilizador ou grupo donos do mesmo.

A este respeito, sugere-se o estudo da abordagem ao desenvolvimento seguro do `qmail`, um agente de transferência de mail (MTA), desenvolvido por Daniel J. Bernstein, entre 1995 e 1998. Nele poderão encontrar um exemplo de uma arquitura modular focada na segurança, em que componentes de software executam como processos com acesso restrito aos recursos do sistema.

[1] Tenha como referência uma distribuição que permita, se necessário, utilizar listas estendidas de controlo de acesso (por exemplo, distribuições baseadas em Debian ou Ubuntu)

3 Aspetos de valorização

No desenho deste serviço, cada equipa poderá querer suportar aspetos funcionais adicionais que considere relevantes, porventura em consequência de decisões relativas ao modelo e arquitetura de segurança adotados no seu desenho e desenvolvimento.

As equipas poderão, por exemplo, querer adicionar outros mecanismos de garantia das propriedades de segurança do serviço. Por exemplo, poderão querer adicionar mecanimos criptográficos que melhorem a resiliência a vulnerabilidades que possam vir a ser descobertas ao nível do serviço ou do sistema operativo e, não menos importante, vulnerabilidades resultantes da utilização do mesmo (p. ex: um utilizador pode deixar sem vigilância uma sessão autenticada).

No mesmo sentido, as equipas poderão querer integrar o serviço com a infraestrutura de registo de eventos e/ou a de gestão de serviços do sistema, ou ainda, poderão querer (re)confirmar a identidade de um utilizador, exigindo, porventura, a introdução de um código de autenticação OTP (por exemplo).

Note que a descrição dos aspetos funcionais e de segurança do serviço permite, intencionalmente, o desenho e a implementação de soluções muito diferentes. Nesse sentido, a equipa deverá ter bem em conta que, tão ou mais importante do que a implementação do serviço, é o seu modelo e arquitetura de segurança.

4 Critérios de avaliação e a sua ponderação

Relatório (40-50%): O relatório, deverá ser preparado em formato PDF, em A4, 11pt, espaçamento simples. Não deverá ter mais do que 10 páginas (sem capas e índices), podendo incluir outra documentação numa secção opcional de anexos. Deverá ser estruturado da seguinte forma:

- A secção de introdução descreve os contornos da solução desenhada, as suas preocupações e decisões arquiteturais mais relevantes;
- A secção de arquitetura funcional descreve os programas (por exemplo, a semântica de cada operação e dos seus argumentos), os processos envolvidos na operação do serviço, aspetos de interoperabilidade (por exemplo, formatos e APIs internas), e, por último, as componentes de sofware desenvolvidas e eventuais dependências de componentes. Deverá ilustrar estes aspetos arquiturais com base em diagramas simples mas rigorosos e completos de modo a facilitar a compreensão desta secção.
- Partindo da arquitetura funcional, uma nova secção deverá descrever e explicar agora as decisões tomadas no domínio da segurança do serviço. Por exemplo, quem são os donos e quais as permissões definidas em cada objeto do sistema de ficheiros, como como dos processos necessários à execução do serviço. Se necessário, os diagramas usados nesta secção poderão omitir alguns detalhes apresentados na arquitetura funcional (por exemplo, os formatos de dados) e/ou expandi-los com as anotações necessárias.
- Uma secção de reflexão deverá justificar detalhadamente os aspetos funcionais e, em particular, de segurança que considere mais relevante. Nomeadamente, a necessidade do conjunto particular de utilizadores e grupos de utilizador do sistema, permissões bem como outros mecanismos complementares que possam ter sido empregues. Deverá também explicar eventuais aspetos de modularidade e encapsulamento das componentes de software desenvolvidas, que ferramentas foram utilizadas para identificar problemas no código dos programas (por exemplo, Valgrind, flags de compilação, outras ferramentas).
- Uma secção de conclusão resume não só os pontos fundamentais das decisões tomadas, identifica eventuais limitações e formas de as superar.

Implementação (20-35%): Deverá ser entregue uma Makefile, com objetivos (*targets*) para a compilação, teste de componentes (se aplicável) e instalação do serviço. Deverá também ser fornecido um README com instruções relevantes, identificando, eventualmente, dependências de componentes externas que devem ser instaladas ou que terão de ser compiladas (se código-fonte incluído), bem como, exemplos de utilização dos vários programas do serviço. Deverá ser também disponibilizado todo o código-fonte e outros ficheiros necessários à instalação e utilização do serviço.

Discussão (30%): Em sessão presencial com a equipa docente a equipa apresentará a solução desenhada e desenvolvida, focando a atenção nos aspetos de segurança do serviço. Deverá ser capaz de demonstrar todo o processo de compilação, de teste (se aplicável), de instalação e de operação do serviço. Relembra-se que, sendo o trabalho desenvolvido em grupo, a avaliação é, no entanto, individual. Todos membros da equipa deverão ser capazes de participar na discussão do trabalho e justificar todas as decisões tomadas.

Valorizações: Os aspetos de valorização funcionarão como bonificação sobre a nota base. A fórmula concreta desta bonificação será anunciada nos próximos dias.