Segurança Informática e nas Organizações 2015-16

Projeto:
SecureChat: Troca segura de mensagens

23 de Setembro de 2016 V 1.0

1 Introdução

Há hoje em dia inúmeros serviços que potenciam a interação entre pessoas, mas os mesmos têm a mais das vezes acesso aos conteúdos em claro trocados entre as mesmas. Tal é uma contrapartida do serviço prestado, onde a troco do seu fornecimento gratuito são realizadas mais-valias através da exploração da natureza dos conteúdos trocados, violando desta forma a sua confidencialidade e a privacidade dos respetivos donos.

1.1 Objetivo

O objetivo deste trabalho é o de desenvolver um sistema que permita que pessoas comuniquem entre si de forma segura, i.e., sem que os conteúdos trocados entre as mesmas sejam, ou possam, ser, devassados por terceiros.

O sistema a desenvolver terá por base um servidor, que funcionará como elemento de definição de uma comunidade (de pessoas interatuantes). Este servidor servirá apenas para fazer a ponte entre as mensagens enviadas por uma pessoa e quem as recebe. A sua funcionalidade é reduzida e, em termos de segurança, deve-se admitir que um servidor (ou quem o gere) é honesto (no sentido que cumpre a sua função primordial de encaminhador de mensagens) mas curioso (pode ter interesse em observar mensagens que lhe são enviadas) e potencialmente malicioso (pode tentar alterar ou fabricar mensagens).

Os utentes do sistema usam uma aplicação cliente, que interage com outra similar através do servidor antes referido. Todos os mecanismos de segurança são concretizados exclusivamente ao nível das aplicações cliente.

No âmbito deste projeto os alunos terão apenas de conceber e concretizar a aplicação cliente. O servidor não precisa de ser concretizado, ele será facultado pelos docentes da UC. Porém, os alunos terão de se adaptar ao modelo base de comunicação assumido pelo servidor.

Na realização do projeto poderão ser seguidas diferentes estratégias para concretizar os clientes, o que naturalmente criará soluções incompatíveis, i.e., onde o cliente desenvolvido por um grupo não consegue interagir com o cliente de outro grupo. Tal não constitui um problema. Também porque o código fonte (em Java) do servidor irá ser fornecido, o mesmo pode ser ajustado às necessidades dos alunos, porém tal deverá ser a exceção e não a regra.

2 Servidor

O servidor mantém uma lista de clientes ligados. Cada cliente possui uma identidade única, num determinado instante. Isto é, dois utentes podem usar a mesma identidade, mas não simultaneamente num mesmo servidor.

A interação dos clientes com o servidor faz-se via TCP: uma ligação TCP por cliente. A interação cliente-servidor é realizada através de objetos JSON encadeadas sobre o circuito virtual TCP.

Uma objeto JSON é uma sequência de caracteres delimitada por chavetas e contendo pares `"nome":"valor"` separados por vírgulas e espaços opcionais (quaisquer caracteres que criem uma separação de texto, seja na mesma linha ou em linhas diferentes). Caso o nome ou o valor tenha aspas, estas deverão ser precedidas pelo caráter \.

Um vetor de objetos JSON é um conjunto de objetos, separados por vírgulas e espaços, delimitados externamente por parêntesis retos.

Os objetos JSON trocados através do servidor podem conter pares nome-valor arbitrários, mas há alguns que são obrigatórios e interpretados pelo servidor. Esses pares possuem os seguintes nomes:

"command" : Nome do ação requerida ao servidor.

"src" : Identidade do remetente da mensagem.

"dst" : Identidade do recetor da mensagem.

"id" : Identidade de um cliente.

O servidor reconhece os seguintes comandos:

"register" : Registo da identidade do cliente. Deverá ser o primeiro comando enviado pelo cliente após estabelecer uma ligação com o servidor. É obrigatório que possua um par nome-valor com o nome `"src"` e com um valor não nulo.

Apenas é possível executar um comando destes (válido) por ligação cliente-servidor.

Assim que terminar a ligação da ligação TCP do cliente ao servidor, este último elimina automaticamente o registo do cliente, caso exista.

"list" : Listagem dos clientes ligados ao servidor. Se este comando tiver um par nome-valor com o nome `"id"` e um valor não nulo apenas será listado o cliente com a identidade indicada. Caso não possua tal par nome-valor, serão listados todos os clientes.

A listagem é feita através de um objeto que contém como valor do par nome-valor com o nome `"result"` um vetor JSON com as identidades registadas pelos clientes.

"send" : Envio de mensagem para um cliente. Caso o comando não especifique um par nome-valor com o nome `"dst"`, o comando será enviado para todos os clientes registados (difusão). Caso contrário, apenas será enviado para o cliente que tiver como identidade o valor desse par.

Todos os comandos enviados ao servidor retornam uma mensagem, gerada pelo servidor, e que é também um objeto JSON, a qual pode possuir um resultado do comando e contém sempre um código de erro. Esta mensagem possui sempre os seguintes nomes em pares nome-valor:

"result" : Resultado do comando. O seu valor depende do comando invocado e pode ser nulo.

"error" : Indica a ocorrência de um erro. Este erro pode ter os seguintes valores:

"ok" : Operação concluída com sucesso.

"registered" : Cliente já registado.

"re-registered" : Tentativa de duplo registo do mesmo cliente.

"unknown" : Cliente desconhecido.

2.1 Funcionalidades a concretizar

O trabalho a realizar, e que incide sobre o funcionamento do cliente, deverá contemplar as seguintes funcionalidades:

1. Envio de uma mensagem para um destinatário, cifrada com uma chave derivada de uma senha.

2. Envio de uma mensagem para um destinatário, cifrada com a chave pública do mesmo e usando cifra híbrida.

3. Acordo de uma chave de sessão entre dois interlocutores, usando (i) uma senha, (ii) o algoritmo de Diffie-Hellman ou (iii) o envio de uma chave de sessão cifrada com a chave pública do interlocutor.

4. Envio de uma mensagem para um destinatário, cifrada com a chave de sessão comum.

5. Assinatura de uma mensagem com um par de chaves assimétricas criado pelo remetente.

6. Autenticação do cliente com o Cartão de Cidadão.

   Esta autenticação deverá ser possível em todas as funcionalidades relativas à autenticação do próprio perante terceiros quando (i) envia mensagens ou quando (ii) fornece dados para que terceiros lhe enviem mensagens de forma confidencial e íntegra.

Todo e qualquer envio de mensagens cifradas deverá ser acompanhado pelo envio de um valor de controlo de integridade ou, opcionalmente, de uma assinatura. O recetor de uma mensagem deverá indicar o grau de confiança que tem na identidade do remetente (elevada, caso tenha sido verificada, baixa, caso seja apenas retirada sem validação da mensagem recebida)

Idealmente as mensagens enviadas devem especificar os algoritmos usados na sua criação, os quais devem ser usados na sua validação. Tal permite uma evolução da aplicação ao longo do tempo. Porém, são aceitáveis, mas menos valorizados, projetos que usem um conjunto de algoritmos fixos (criptográficos ou outros, tais como de codificação).

3 Avaliação do Projeto

Os projetos devem ser realizados em grupos de 2 elementos. A nota final dependerá de 3 aspetos:

1. O grau de satisfação dos requisitos expostos neste enunciado. Isto é, quantas das funcionalidades pedidas foram implementadas.
2. O grau de complexidade da solução apresentada. São mais valorizadas soluções simples que conseguem o maior grau de integração de funcionalidades e que melhor satisfazem a experiência dos utentes. É também valorizada a identificação, discussão e proposta de solução de alguma eventual vulnerabilidade na aplicação proposta.
3. A participação individual de cada elemento do grupo. Esta será aferida em discussão oral e em casos extremos pela participação no repositório de código do grupo. **O desconhecimento de quaisquer partes relevantes do projeto apresentado será interpretado como não tendo participado na sua realização ou contribuído de forma relevante para a mesma**.

Relativamente à interface, a mesma será fundamentalmente avaliada tendo em conta a quantidade de informação, e a sua relevância, apresentada ao utente, mas não a sua qualidade gráfica.

3.1 Datas

No dia **28 de outubro de 2016** será necessário entregar um documento, com um limite de **2 (duas) páginas**, especificando:

- A linguagem escolhida.
- O formato completo dos comandos.

Será fornecido *feedback* sobre esta informação, devendo este ser utilizado para refinar o trabalho. Esta avaliação intermédia contribui com 1 valor para a nota final do trabalho em grupo.

No dia **31 de outubro de 2016** haverá uma avaliação intermédia com a duração máxima de 10 minutos por grupo e na presença de todos os elementos da turma. Nela, cada grupo deve fazer uma demonstração de um conjunto de funcionalidades mínimas que deverão existir nessa altura. Considerando uma distribuição do esforço mais ou menos homogénea ao longo do semestre,

espera-se que na altura desta avaliação intermédia os vários grupos já tenham protótipos funcionais com o registo de clientes no servidor, a listagem dos clientes registados e os dois primeiros itens das funcionalidades requeridas para os clientes (ver Secção 2.1). Esta avaliação intermédia contribui com 2 valores para a nota final do trabalho em grupo.

No final do semestre, para além da demonstração final do trabalho, os alunos deverão entregar um relatório da sua realização e fazer uma apresentação oral do trabalho. Esta apresentação será seguida de uma discussão individualizada onde o grupo fará a defesa do seu trabalho. O relatório deverá referir todas as decisões tomadas pelo grupo na realização do projeto e todos os requisitos não cumpridos. O relatório deverá ainda conter imagens devidamente comentadas que evidenciem a correção da solução implementada. As datas de entrega do relatório e de apresentação e discussão do trabalho serão oportunamente indicadas.

3.2 Bónus

Serão atribuídos pontos de bónus a funcionalidades que aumentem de forma interessante e razoável a segurança do sistema. Não se procuram sistemas complexos mas sim eficientes. Um exemplo é a extensão para vários destinatários (todos ou apenas uma lista), dos mecanismos de segurança idealizados para um destinatário. Outro é a possibilidade de criar, ao nível do cliente ou do servidor, mecanismos de controlo de acesso (e.g. só autorizar diálogos com alguns clientes ou negar diálogos com certos clientes).

Dependendo do grau de integração e funcionalidade final, a utilização de uma aplicação popular também poderá conduzir a pontos extra. Todas estas situações devem ser discutidas com o docente com a devida antecedência.

Nota: os bónus apenas deverão ser considerados após terem sido concretizadas as demais funcionalidades, listadas na Secção 2.1.