Guião de Demonstração

Considerações:

→ Infelizmente, por motivos de falta de tempo, não conseguimos testar e corrigir possíveis erros na parte referente à assinatura digital. O código foi implemente segundo a lógica teórica.Para tratar da assinatura digital criamos os handlers SignatureHandler e SignatureCheckHandler na pasta security.
→ Considerámos que o LoggingHandler se encontra já instalado no computador que irá correr o nosso projecto, sendo usado neste projecto como uma dependência.
→ Decidimos copiar o ficheiro CertUtil, disponibilizado pelos professores, para o nosso projecto e adaptá-lo para o correcto funcionamento deste projecto.

Instruções:

Para obter o projecto do repositório Git utilizar o comando:
→ git clone https://github.com/tecnico-distsys/T06-Komparator/

Para limpar e instalar/compilar o projecto sem correr os testes IT utilizar o comando na pasta T06_Komparator:
→ mvn clean install -DskipITs

Antes de executar o projecto é necessário garantir que o juddi se encontra ligado e as dependências para o Logging Handler se encontram instaladas.

Para executar o projecto deve-se:
Entrar na pasta supplier-ws e correr três Suppliers:
→ mvn exec:java
→ mvn exec:java -Dws.i=2
→ mvn exec:java -Dws.i=3

Entrar na pasta mediator-ws e correr o Mediator:
→ mvn exec:java

Para correr testes ITs e verificar a troca de mensagens SOAP entre cliente e servidor:
Entrar na pasta mediator-ws-cli e correr o comando:
→ mvn verify

Para testar a segurança entre o canal mediator-ws e o mediator-ws-cli em relação à cifra/decifra do cartão de crédito deve-se entrar na pasta mediator-ws-cli e executar o seguinte comando para apenas executar um teste e observar as mensagens SOAP:
→ mvn test -Dtest=BuyCartIT#buyCartSucessTest

Para testar a segurança entre supplier-ws e supplier-ws-cli em relação à assinatura digital pode-se fazer a chamada do método ping entrando na pasta mediator-ws-cli e executando o comando:
→ mvn exec:java

Caso 1: Demonstar funcionamento normal do programa
→ Obrigatório: Configuração do LoggingHandler para imprimir as mensagens SOAP antes e depois de adicionados os elementos de segurança ao cabeçalho das mensagens.

Caso 2: Demonstrar falha na frescura
→ Obrigatório: Configurar o handler TimeStamp
→ Sugestão: Fazer com que o programa após adicionado os cabeçalhos fique a dormir por um tempo superior a 3 segundos.

Caso 3: Demonstrar falha na unicidade
→ Obrigatório: Configurar o handler IdHandler
→ Sugestão: Obter um número aleatório e força-lo a que ele seja repetido no cabeçalho da mensagem.

Caso 4: Demonstrar resistência a um ataque
→ Sugestão: Simular a alteração não autorizada da mensagem, por exemplo, para mudar o preço devolvido por um fornecedor

Caso 5: Demonstrar falha na autenticidade
→ Sugestão: Simular alteração não autorizada da mensagem em relação ao campo referente ao identificador do emissor que se encontra no cabeçalho SOAP. Por exemplo, alterar para um emissor que não existe

Caso 6: Demonstrar falha na integridade
→ Sugestão: Simular a alteração não autorizada da mensagem em relação ao campo assinatura que se encontra no cabeçalho SOAP

Caso 7: Demonstrar falha na cifra
→ Sugestão: Simular alteração não autorizada da mensagem em relação ao número de cartão de crédito após cifrado, acrescendo por exemplo algum número.