universidade de aveiro estga escola superior de tecnologia e gestão de águeda

Relatório

Trabalho prático de Segurança Informática

Proteção de aplicações

Grupo 11:

João Novo – 84205

Rui Duarte – 29979

Águeda, 29 de dezembro de 2018

Licenciatura em Tecnologias da Informação

3º Ano – 1º Semestre

Índice

1. Introdução

1.1. Tema

O objetivo deste trabalho é o de desenvolver um sistema que permita distribuir aplicações de forma segura, garantindo que apenas são executadas pela combinação de utilizador e equipamento legítimos.

Desta forma o trabalho passa essencialmente pela implementação de mecanismos que garantam confidencialidade, integridade e autenticidade, aplicando os conhecimentos adquiridos ao longo da unidade curricular.

1.2. Objetivos

Contando com os pontos acima referidos, surgem os seguintes requisitos que a aplicação deve dar resposta:

- Criação de um ficheiro com um pedido de licença que inclua a identificação do utilizador, dados sobre a plataforma para a execução da aplicação e dados sobre a aplicação;
- Proteção (integridade, confidencialidade, autenticação, não repudiação) do pedido de licença;
- Validação do pedido de licença;
- Emissão da licença, com todos os dados que garantam que apenas uma aplicação legitima pode ser executada no sistema autorizado e pelo utilizador autorizado;
- Proteção (integridade, confidencialidade, autenticação, não repudiação) da licença emitida;
- Validação do documento da licença;
- Proteção contra execução da aplicação noutro sistema;
- Proteção contra a execução da aplicação por outro utilizador;
- Proteção contra a alteração da aplicação.

2. Aplicação

2.1. Cliente

O programa Cliente inicialmente verifica se existe o ficheiro de licença na máquina e, caso não exista, o programa recebe um input, se quer ou não fazer um pedido de licença.

Caso queira fazer um pedido de licença (Figura 1), necessita de ter o cartão de cidadão ligado a um leitor apropriado à máquina que deseja efetuar o pedido de licença, para assim conseguirmos obter os dados do cartão de cidadão e para o processo de assinatura e obtenção do certificado.

Figura 1 – Pedido de licença

Após termos os dados do cartão vamos buscar todos os dados que necessitamos para fazer o pedido de licença, os campos são os seguintes:

- Nome do Utilizador (Primeiro e Últimos nomes);
- Identificação Civil;
- Email;
- Dados da Máquina (CPU, Motherboard, GPU, MACs) – recorrendo a uma biblioteca chamada "jHardware";
- Nome da APP;
- Versão;
- Tamanho do programa;
- Chave Pública do Cliente.

Os diferentes tipos de validação da licença que temos são:

- Validação de Cartão de cidadão;
- Dados da máquina (é permitido a alteração de um componente);
- Nome e da versão do software;
- Se o programa foi alterado.

Caso já tenha a licença, o programa de seguida apresenta um menu onde pede ao utilizador se quer continuar para o programa (Figura 2), ou se quer ver os dados da licença (Figura 3).

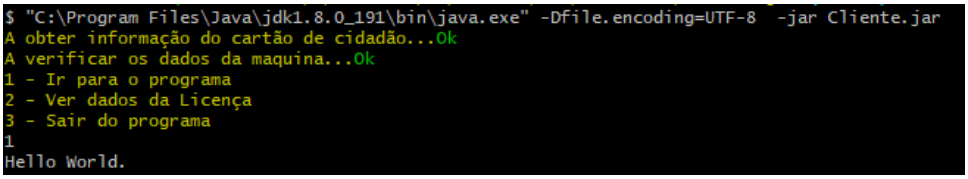

Figura 2 – Programa principal

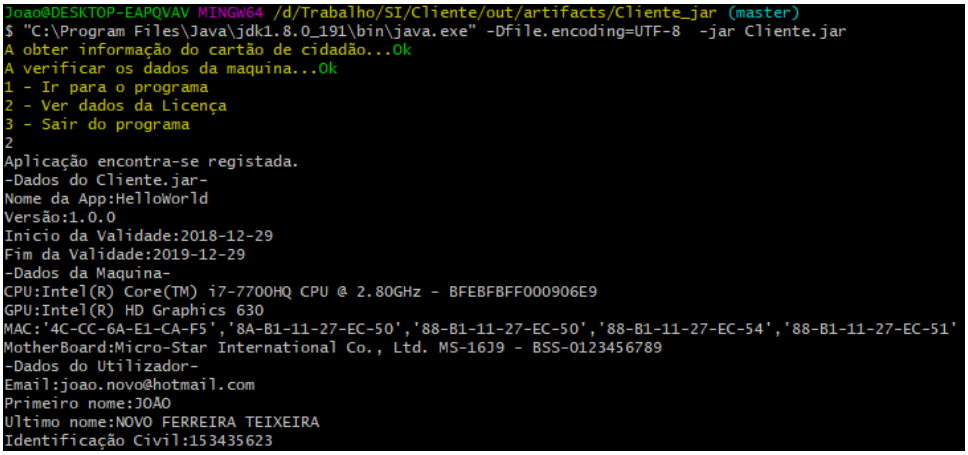

Figura 3 – Dados da licença

2.2. Servidor (Gerador de Licenças)

Do lado do Gerador de licenças, ele tem apenas três comandos, um que é o "-help", outro que é o "-gerar {nome do ficheiro}" e outro que é o "-list"

O "-help" (Figura 4), apenas mostra no ecrã os comandos e como os devemos usar. Caso não seja introduzido nenhum comando ele por omissão executa o comando "-help"

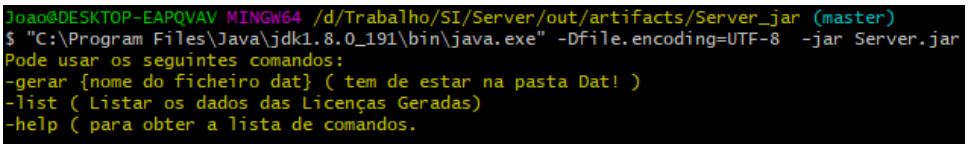

Figura 4 - Comando Help

Enquanto o "-gerar" precisa de ter a pasta "Dat" criada na diretoria onde tem o gerador de licenças e os ficheiros ".dat" que vêm do lado do cliente que são os pedidos de licença devem ir para dentro dessa pasta. Quando se usa o comando "-gerar" deve apenas pôr se o nome do ficheiro, sem o ".dat" exemplo "Gerador.jar -Dfile.encoding=UTF-8 -gerar 153635623".

O que o método "-gerar" (Figura 5) vai fazer é desencriptar o pedido de licença e ver se tudo bate certo verifica se o pedido é válido e após verificar tudo, gera uma licença de um ano com início na data onde foi feita a licença e o fim da licença é a soma de mais um ano.
E por fim dá-nos um output para um ficheiro com o mesmo nome do ".dat" só que a extensão muda para ".lic" e vai para a pasta "Lic".

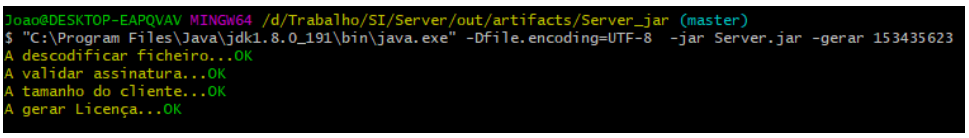

Figura 5 - Comando Gerar

No comando "-list" (figura 6) são listadas todas as licenças criadas até ao momento, mostrando assim nome do ficheiro, nome da APP, versão, Nome do Utilizador e a validade.

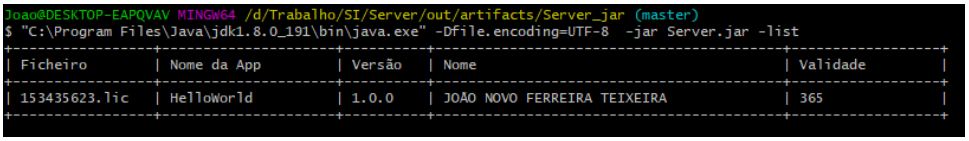

Figura 6 - Comando Listar

3. Mecanismos de segurança

Os mecanismos de segurança implementados nesta aplicação e a seguir descritos servem o propósito de assegurar a confidencialidade, integridade, autenticação e a não repudiação dos dados e da licença.

Os algoritmos de cifra assimétrica são computacionalmente mais complexos que cifra simétrica. Mas a distribuição da chave pública é mais prática que a chave secreta. Como tal, é utilizado um sistema de cifra híbridas que, na prática, reúnem o melhor de dois mundos. O que este sistema faz é gerar a chave secreta, chamada de chave de sessão; usar a cifra assimétrica para cifrar a chave secreta; usar a cifra simétrica e a chave secreta para os restantes dados. Para as cifras simétricas foi utilizado o algoritmo AES (Advanced Encription Standard), norma dos EUA com chaves de 128bits. Para as cifras assimétricas foi utilizado o algoritmo RSA (Rivest Shamir Adleman) que é o algoritmo de cifra em chave pública mais utilizado.

3.1. Criação do ficheiro de pedido de licença (cliente)

Após a leitura correta e obtenção dos dados da máquina e do utilizador (através da leitura dos dados do Cartão de Cidadão), os mesmos são assinados através de uma assinatura digital do Cartão de Cidadão garantindo a autenticação, a integridade e o não repúdio dos dados. Ao texto em claro assinado é-lhe adicionado a assinatura e o respetivo certificado.

De forma a que a licença gerada no servidor consiga ser decifrada no cliente é gerado um par de chaves e a sua respetiva chave pública adicionada aos dados acima referidos

De seguida, é gerada uma chave simétrica que é utilizada para cifrar o conteúdo acima mencionado, garantindo a confidencialidade. De seguida a chave simétrica é cifrada assimetricamente com a chave pública (pré-gerada) do gerador de licenças. Desta forma os princípios de confidencialidade, integridade, autenticação e não repúdio são assegurados.

De salientar ainda, que toda a informação é codificada em Base64.

3.2. Criação do ficheiro de licença (gerador de licenças)

O servidor recebe o ficheiro contendo os dados da licença e com a sua chave privada pré-gerada, consegue desencriptar a chave simétrica, podendo aceder aos dados, após a utilização desta mesma chave.

Após esta operação, valida o certificado que se traduz pela construção e subsequente validação do caminho de certificação descritos mais em pormenor no capítulo 3.3 e 3.4. De seguida é efetuada a validação da assinatura. É ainda feita a comparação do tamanho programa contido no ficheiro recebido, com o do programa original, de forma a impedir qualquer tipo de modificação.

Aos dados recolhidos é-lhe adicionado o período de validade da licença.

A nível de mecanismos de segurança o processo de criação do ficheiro de licença é similar ao do pedido do mesmo.

Os dados são encriptados simetricamente e a chave gerada é encriptada assimetricamente com a chave pública do cliente. A grande diferença reside no processo de assinatura. Ao invés de ser assinado com o cartão de cidadão, este ficheiro é assinado com uma assinatura digital, sendo adicionado ao ficheiro final a assinatura e respetiva chave pública, de forma à mesma poder ser verificada/validada no cliente.

3.3. Construção do caminho de certificação

O caminho de certificação é o conjunto de certificados de confiança, desde o certificado ancora/raiz de confiança, até ao certificado do utilizador final (neste caso, o certificado alvo que queremos validar).

Para construir o caminho de certificação, para além do certificado alvo que queremos validar, precisamos de um certificado ancora/raiz de confiança e dos certificados intermédios. Neste caso, a máquina encarregue de gerar as licenças contém o certificado ancora/raiz de confiança e o conjunto de todos os possíveis certificados intermédios.

Com base no certificado alvo que queremos validar vamos ver qual a sua hierarquia de certificação. De seguida, coloca-se cada certificado seguinte da hierarquia, no caminho de certificação.

Antes de qualquer certificado ser colocado no caminho de certificação, é verificado se está dentro do prazo de validade. Se não estiver, será apresentada uma mensagem de erro.

Ainda antes de colocar o certificado ancora/raiz de confiança para o caminho de certificação, verifica-se se este foi assinado por ele próprio. Se assim não for, significa que não é um certificado ancora/raiz de confiança e será apresentada uma mensagem de erro.

3.4. Validação do caminho de certificação

A validação de um caminho de certificação passa por verificar se as assinaturas de todos os certificados estão válidas e se nenhum dos certificados foi revogado, isto é, deixou de estar válido antes do fim do seu prazo de validade.

Depois de termos um caminho de certificação construído, verificam-se as assinaturas dos certificados, e verifica-se ainda se algum dos certificados foi revogado. Caso algo não esteja em conformidade. Se não estiver, será apresentada uma mensagem de erro.

Para verificar se um certificado foi revogado ou não, e utilizado o Online Certificate Status Protocol (OCSP). Caso não haja ligação disponível à internet, será apresentada uma mensagem de erro.

Não se verificando nenhum tipo de problema considera-se a validação como positiva.

4. Dificuldades e soluções encontradas

O parâmetro "Dfile.encoding=UTF-8" tem de ser utilizado na execução das aplicações, de forma a garantir que as mensagens sejam apresentadas com a correta acentuação e caracteres especiais.

Os métodos de encriptação simétrico e assimétrico passaram por variadas versões de código, ocorrendo vários erros na fase de desencriptação, principalmente no tamanho das chaves. Na versão "final" esse código foi complemente reformulado, eliminando os erros e criando uma versão bastante mais "clean".

Existiram dificuldades iniciais na leitura das chaves proveniente de ficheiros, resolvido através da codificação e consequente descodificação em Base64.

A obtenção do hardware foi sistema foi, também, uma dificuldade, tendo sido resolvida recorrendo a uma biblioteca chamada "jHardware". No entanto, em relação às placas de rede, o mesmo não filtrava as placas de redes virtuais. Isto foi resolvido criando um método que filtra os endereços MAC permitindo obter apenas os endereços físicos, eliminando os eventuais virtuais. De salientar, ainda, que, apesar do nível de funcionalidade da biblioteca ser excelente, o desempenho da mesma fica um pouco aquém do esperado (demora entre 20 a 30 segundos a recolher as informações do sistema).

Verificámos que aos certificados convertidos em string teriam de lhe ser adicionados tags como abaixo apresentado, caso contrário a sua leitura era impossibilitada.

```
-----BEGIN CERTIFICATE-----\n"+textoCertB64+"-----END CERTIFICATE-----
```

A validação do certificado do cartão foi efetuada em termos de período de validade, mas a nível da validação do caminho de certificação a implementação não ficou completamente funcional.

Em caso de falha na verificação de alguns componentes, é invocado o método "System.exit(0)" que deveria sair do programa sem mais nenhum tipo de output mas, no nosso caso, devolve-nos sempre um "erro fatal" (fig. 7).

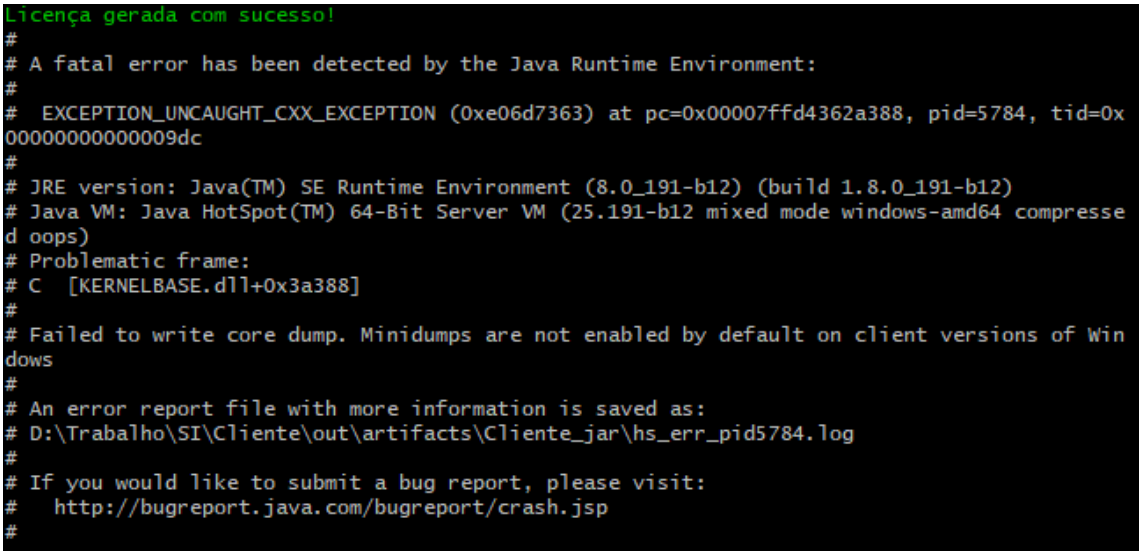

Figura 7 - Erro fatal

5. Conclusão

O trabalho realizado revelou-se um bom caso de estudo e oportunidade de aprendizagem relativamente à implementação de mecanismos de segurança na criação de licenças.

Assim, sumarizam-se os aspetos essenciais explorados durante a realização do projeto:

- Implementação de um sistema de criação e gestão de licenças
- Implementação de mecanismos de segurança que assegurem a confidencialidade, autenticação e integridade.

A elaboração deste projeto proporcionou uma excelente oportunidade para consolidar temas desenvolvidos ao longo da cadeira de Segurança Informática, bem como um desafio que maturou uma ética de trabalho já existente dos diversos elementos do grupo.

6. Bibliografia

Documentação Java: https://docs.oracle.com/javase/8/

Documentação Cartão de Cidadão: https://www.autenticacao.gov.pt/cc-documentacao

André Zuquete – Segurança 18-19: http://sweet.ua.pt/andre.zuquete/aulas/Seguranca/18-19/index-work.html

jHarware: https://github.com/profesorfalken/jHardware

TutorialsPoint Java Crytography Tutorial: https://www.tutorialspoint.com/java_cryptography/index.htm

JAVA in a nutshell – javax.crypto Package: https://docstore.mik.ua/orelly/java-ent/jnut/ch26_01.htm

7. Anexos

A este relatório anexam-se os seguintes ficheiros:

- SI_Client.zip,
- SI_Server.zip