Escola de Engenharia
Universidade do Minho

DSS *Demo Web App*

Projeto de Desenvolvimento 2020/2021
Engenharia de Segurança

29/6/2021

Grupo 3:

Constança Elias - PG42820
Diogo Rio - A84752
Filipe Freitas - PG42828
Maria Araújo - PG42844

Estrutura do Relatório

- **DSS *Demo Web App***
- **Solução Desenvolvida**
  - Cartão de Cidadão
  - Chave Móvel Digital
  - Fonte de *timestamp* do cartão de cidadão
  - Guardar configurações do utilizador
- **Técnicas de Desenvolvimento de *Software* Seguro**
- **Conclusão**

DSS (*Digital Signature Services*)

- É um projeto *open-source,* em Java, que permite assinar, validar e extender assinaturas eletrónicas avançadas (AdES).
- As principais funcionalidades são:
  - assinatura de documentos em diversos formatos (XML, PDF, ODT, TXT, ZIP...);
  - três formatos principais de assinatura digital (XAdES, CAdES e PAdES);
  - validação de certificados.

DSS Demo Web App

- A DSS Demo WebApp consiste numa integração da *framework* DSS, sendo um exemplo concreto da utilização da mesma.
- Principais funcionalidades:
  - assinar documentos
  - extender uma assinatura
  - validar uma assinatura
  - validar um certificado

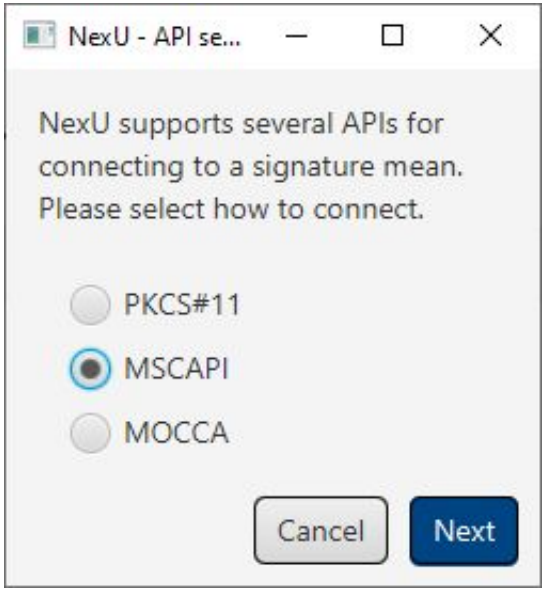
CEF Digital
Connecting Europe
DSS Demonstration WebApp

Solução Desenvolvida

- Integração das funcionalidades pedidas:
  - Integração do Cartão de Cidadão
    - Utilização do NexU
  - Integração da Chave Móvel Digital
  - Utilização do *Timestamp* do Cartão de Cidadão
  - Guardar configurações e dados do utilizador

Integração do Cartão de Cidadão

- Em *Windows*, o NexU já permite assinaturas com o Cartão de Cidadão português.
- Basta utilizar a API *MSCAPI* do *Windows* para interação com *Smart Cards* (não existe ligação automática com o CC português)



Timestamp do Cartão de Cidadão

- Utilizado em todos os tipos de ficheiros que a aplicação permite assinar.
- Foi necessário alterar a configuração do ***Timestamp Protocol (TSP)***, de modo a utilizar o servidor do Cartão de Cidadão português.
- No ficheiro *tsp-config.xml* alterou-se o *bean* que corresponde à fonte do *timestamp*.

Alteração do servidor do *timestamp*

```
22  <bean id="tspSource" class="eu.europa.esig.dss.service.tsp.OnlineTSPSource">
23    <constructor-arg name="tspServer" value="http://ts.cartaodecidadao.pt/tsa/server" />
24    <constructor-arg name="dataLoader" ref="tspDataLoader" />
25  </bean>
26
27  <bean id="tspDataLoader" class="eu.europa.esig.dss.service.http.commons.CommonsDataLoader">
28    <constructor-arg name="contentType" value="application/timestamp-query" />
29  </bean>
```

Integração da Chave Móvel Digital

- Com base no projeto desenvolvidos por alunos do ano passado.
- Foi necessário adicionar uma dependência.
- Foi criada uma nova área na página de aplicação.

Integração da Chave Móvel Digital

- Esta nova página tem 3 estados diferentes:
  - **Página Principal** – contém a introdução dos dados da CMD e da assinatura a gerar;
  - **Página de OTP** – onde é se introduz o OTP recebido;
  - **Página de *Download*** – onde é feito o *download* do ficheiro assinado.
- Validação das entradas utilizando a *framework Spring*.
- Funciona para todos os tipos de validação e *hash* disponíveis.

Login na DSS Web App

- Último requisito do projeto: guardar dados e configurações do utilizador de modo a que quando fizesse *login* pudesse usar os seus dados perante a CMD.
- Optou-se por guardar apenas o nº de telemóvel no lado do cliente, para cumprir com as leis de proteção de dados pessoais
- Fica guardado na *localStorage* do *browser* do cliente.
- Caso já esteja lá guardado é recuperado automaticamente.
- O utilizador escolhe se quer guardar o número através de uma *checkbox.*

Técnicas de Desenvolvimento de Software Seguro

- OWASP Software Assurance Maturity Model (SAMM)
- Microsoft Security Development Lifecycle (SDL)
- OWASP Application Security Verification Standard
- PIA
- *Buffer Overflow*
- Vulnerabilidade de inteiros
- Validação de *Input*

Microsoft SDL

Este modelo de desenvolvimento de *software* seguro divide-se em 7 fases.

- **Fase de Formação**
  - Todos os elementos do grupo adquiriram conhecimentos a nível de segurança de software no perfil de Criptografia e Segurança da Informação.
- **Fase de Requisitos**
  - Garantir que as normas definidas pelo RGPD são cumpridas;
  - Seguir o standard de verificação de segurança de aplicações *(OWASP Application Security Verification Standard)*

Microsoft SDL

- **Fase de Desenho**

  Procurou-se identificar possíveis riscos para o sistema.

  - ***Denial of Service***

    - Os invasores podem congestionar a rede com ruído que pode provocar colisões de sinal e produção de erros.
    - Se a rede for sobrecarregada por utilizadores da aplicação pode levar a uma desconexão do serviço.

Microsoft SDL

- **Fase de Codificação**
  - Teve-se em atenção o processo de codificação, procurando adotar boas práticas.
- **Fase de Verificação**
  - Revisão e testes manuais das funcionalidades, procurando descobrir alguma abertura no sistema que necessitasse de ser corrigido.
- **Fases de Publicação e de Resposta**

OWASP *Application Security Verification Standard*

Tendo em conta a lista *standard* mais recente, (2017).

- **A1 - Injeção de Código**
  - Adequada validação de *input.*
- **A2 - Quebra de autenticação**
  - Contacto telefónico não é uma informação que por si só traga vantagens a um atacante.
- **A3 - Exposição de dados sensíveis**
  - Não foram armazenados dados confidenciais.

OWASP *Application Security Verification Standard*

- **A6 - Configuração incorrecta de segurança**
  - Procurou-se que os erros do sistema não sejam expostos ao utilizador.
- **A7 - *Cross-Site Scripting***
  - Todos os dados recebidos são validados e nenhuma informação é enviada para o *browser* sem ter sido previamente verificada e filtrada.

Outras práticas de segurança

- **PIA**
  - Para demonstrar *compliance* com o RGPD (Regulamento Geral de Proteção de Dados)

- ***Buffer Overflow***
  - Programas em Java, como é o caso da DSS WebApp, não são vulneráveis a problemas de *Buffer Overflow.*

Outras práticas de segurança

- **Vulnerabilidade de Inteiros**
  - Não se aplica
  - Nº de telefone, Pin da CMD e o código de confirmação guardados numa *string*

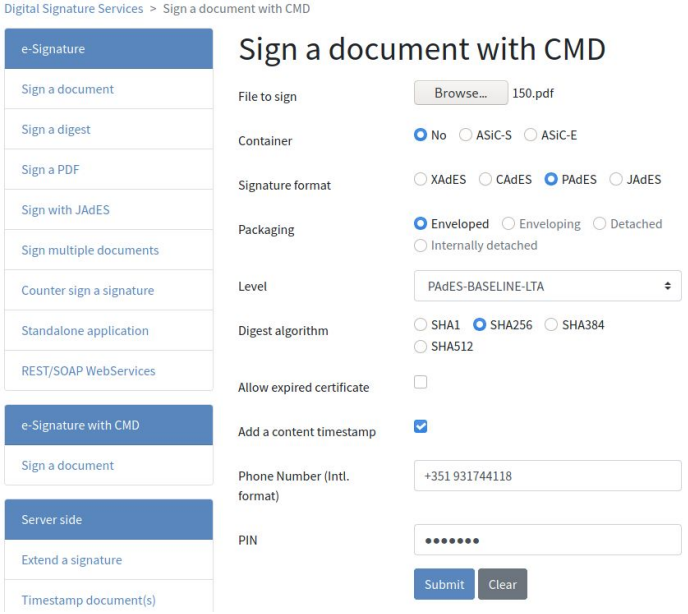
- **Validação de *Input***
  - Validação do nº de telemóvel, o PIN da CMD associada e o código OTP recebido no telemóvel.

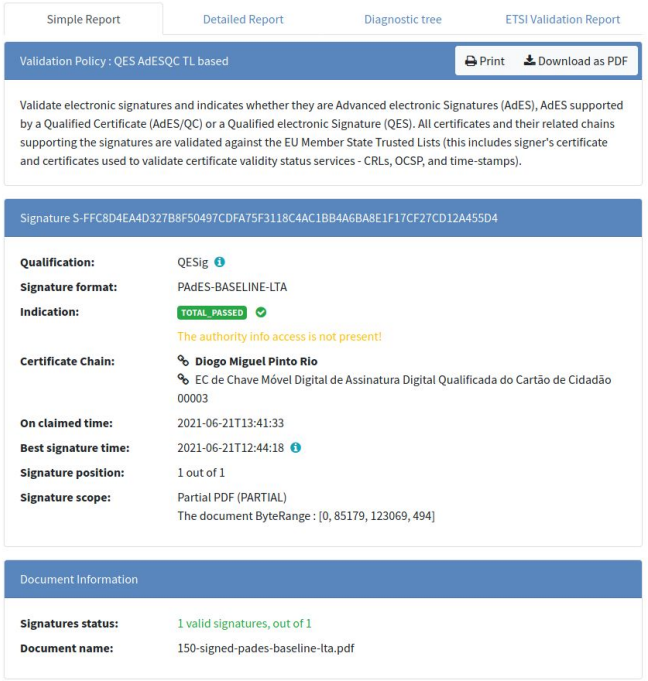
Instalação da *App*

- Requisitos
- Processos de instalação detalhados no relatório.
    - Em *Windows* e *Linux*

- **Nota:** A funcionalidade de assinatura com Cartão de Cidadão apenas está disponível em Windows.

Utilização da *App*





Conclusão

- Este projeto permitiu **integrar/utilizar APIs** e código de terceiros de modo a simplificar o processo de desenvolvimento ou **aumentando a sua segurança**.

- Parte **mais desafiante**: instalação da *DSS Demo WebApp*
    - falhas na compilação das várias dependências que eram necessárias para construir as várias partes que compõem a aplicação

- A nível de **segurança**, foram tidos em conta as estratégias e recomendações que foram lecionadas ao longo da unidade curricular.

Conclusão

- **Trabalho futuro:**
  - **integrar sessões** na plataforma, para permitir guardar outros dados além do seu contacto, como um histórico das operações realizadas (para poder reutilizar acções);
  - **Implementar diretamente a configuração**, no NexU, para a assinatura com o Cartão de Cidadão, sem ser necessário configurá-la nenhuma vez.
  - Implementar as outras **páginas da aplicação** *DSS WebApp* no separador do **CMD.**
  - Aplicação de outras estratégias de *software* seguro, como o recurso a programas de detecção de **vulnerabilidades de *software.***