Universidade do Minho
Mestrado Integrado em Engenharia Informática
SSI-Trabalho Prático 1

João Bernardo Freitas a74814
Rui Pereira pg42853

20 de Novembro 2020

Contents

1 Introdução

Neste relatório vamos descrever os passos que seguimos para realizar as duas primeiras etapas da modelação de ameaças para o trabalho prático nº1 da cadeira de Segurança de Sistemas Informáticos.
Mais especificamente vamos começar por dar um *overview* ao projeto em si, vamos realizar a modelação do sistema e por fim vamos identificar as ameaças bem como potenciais soluções para estas mesmas.

2 Sistema em Desenvolvimento

O sistema em desenvolvimento **mID** visa a desmaterialização de documentos de identificação pessoal, ou seja, um utilizador poderia substituir um documento pessoal físico, por exemplo a carta de condução, por uma versão digital no seu *smartphone*.

Este sistema consiste em 3 componentes principais:

- **Aplicação portadora-mID**
- **Aplicação leitora**
- **Entidade Emissora**

2.1 Aplicação portadora

Esta componente corresponde à aplicação móvel para sistemas operativos **Android** e **IOS** que irá armazenar todos os dados relativos a um documento de identificação bem como todos os elementos que irão ser utilizados pela aplicação leitora que irá verificar a integridade e autenticidade destes mesmos.

Para o utilizador fazer o *download* dos seus documentos a aplicação portadora conecta-se, através do protocolo **TCP-IP**, com a infra-estrutura da entidade emissora de forma a obter todos os dados associados a um documento. Naturalmente que este utilizador necessita de estar autenticado no respetivo serviço de forma realizar a transferência na primeira vez que esta é feita.

Esta operação realizara-se-à periodicamente para atualização de dados sendo que não será necessário nova autenticação após a primeira transferência.

Os dados são armazenados em formato **JSON - JavaScript Object Notation**, sendo que é importante a garantia de autenticidade e integridade destes mesmos.

Todas as **provas de identidade**, realizadas entre a aplicação portadora e leitora, podem ser realizadas tanto em modo *Offline*, onde o dispositivo do portador transfere os atributos de identificação diretamente para o dispositivo leitor, assim como os dados necessários para a sua verificação, e *Online*, onde o dispositivo leitor transfere um token de autorização para que o verificador consulte diretamente a entidade emissora do documento.

Em ambos os modos deve ser a aplicação portadora a iniciar o processo através de um **código QR** sendo que a transferência em si iria ser realizada através de uma das seguintes tecnologias:

- **BLE - Bluetooth Low Energy**
- **NFC - Near Field Communication**
- **WiFi-Aware**

Uma vez que esta comunicação seja estabelecida o *verificador* envia um pedido contendo os identificadores desejados ao qual o utilizador pode aceitar uma transferência na totalidade ou apenas de um subconjunto destes mesmos. Toda este comunicação é suportada por mensagens codificados no formato **CBOR - Concise Binary Object Representation**.

Naturalmente que é necessário garantir que esta interação mantenha a confidencialidade e integridade dos dados transferidos bem como que haja uma maneira de auditar as interações ocorridas com leitores.

2.2 Aplicação leitora

Tal como a componente anterior, esta será uma aplicação móvel para sistemas operativos **Android** e **IOS**, ou qualquer outro dispositivo que suporte os 3 protocolos de comunicação previamente mencionados. Esta aplicação irá permitir que um *verificador* estabeleça comunicação com um *portador* para fazer a transferência dos atributos necessários para identificar um portador, como por exemplo confirmar a sua idade.

Esta aplicação deverá suportar todos os protocolos mencionados previamente bem como o modo de funcionamento **Online** e **Offline**. Por fim deve também permitir a auditoria das operações o que leva á necessidade da aplicação, ou seu utilizador, estar também autenticada.

No modo **Online** de operação, é preciso garantir que a aplicação leitora não é capaz de alterar a lista de atributos autorizados pelo portador antes da consulta à infra-estrutura da entidade emissora.

2.3 Entidade Emissora

Por fim temos a entidade emissora, que corresponde às entidades com poder de emitir e conferir a autenticidade de um documento, como tal esta entidade é que é responsável por prover os mecanismos que garantem a autenticidade e integridade dos documentos digitais.
Como foi descrito nas secções anteriores, esta entidade comunica com as outras duas através da rede pública recorrendo ao protocolo **TCP/IP**.

Esta componente ainda está em testes preliminares ou seja a lista destes ainda não é final e, como tal, está sujeita a alterações.

- **OS do servidor web:** CentOS 7.8.2003
- **Backend principal:** Django v3.0
- **Servidor Web:** UWSGI
- **BD:** PostgreSQL 12.4
- **SO do serviço de gestão do sistema:** Ubuntu 20.04
- **Backend de gestão:** Flask 1.0
- **Servidor Web:** Gunicorn
- **BD de dados de gestão:** PostgreSQL 12.1 (a correr num container **Docker** versão 19.03.6)

3 Modelação de Ameaças

A modelação de ameaças é um processo que visa a identificação de vulnerabilidades no sistema de forma aos defensores deste mesmo desenvolverem estratégias de combate a eventuais ataques. Para facilitar este combate vamos identificar os ativos a proteger, eventuais vetores de ataque e suas eventuais soluções.

A modelação de ameaças está estruturada em 4 fases:

- **Modelação do sistema**- Esta fase consiste em modelar o sistema através de **diagramas de estado**, ***swim lane*** e/ou **diagramas de fluxo de estados** o que nos irá dar uma melhor visão do sistema em si.
- **Identificação de ameaças**- Esta fase consiste, tal como o nome indica, em identificar as vulnerabilidades do sistema e vetores de ataque com ajuda dos diagramas desenvolvidos na fase anterior.
- **Resolução de ameaças**- Nesta fase espera-se que sejam propostas soluções para as vulnerabilidades identificadas na fase anterior
- **Validação**- Nesta fase espera-se que as soluções propostas nas fases anteriores sejam validadas.

4 Modelação do sistema

Nesta secção será apresentada a primeira etapa de **Modelação de Ameaças**, mais especificamente **Modelação do Sistema**. Nesta primeira etapa decidimos modelar o sistema através de *Data Flow Diagrams* ou *DFDs*, de *Swim Lane Diagrams* ou *SLDs* e *Trust Boundaries Diagrams* ou *TBDs*. Os DFDs permitem-nos fazer uma observação geral do sistema e de cada componente enquanto que os SLDs favorecem a interação entre apenas duas partes do sistema facilitando a identificação de possíveis ameaças, finalmente temos os TBDs que nos indicam quando os dados do programa mudam de domínio de confiança.

4.1 Data Flow Diagram

Através de um **Diagrama de Fluxo de Dados** podemos verificar todos os fluxos de dados de um sistema, por exemplo, os fluxos de dados unidireccionais entre a entidade emissora e aplicação portadora/leitora ou os fluxos de dados bidireccionais entre a aplicação portadora e leitora.

Com este diagrama podemos já começar a identificar eventuais vulnerabilidades no sistema como falsificação de dados e/ou alterações no hardware.

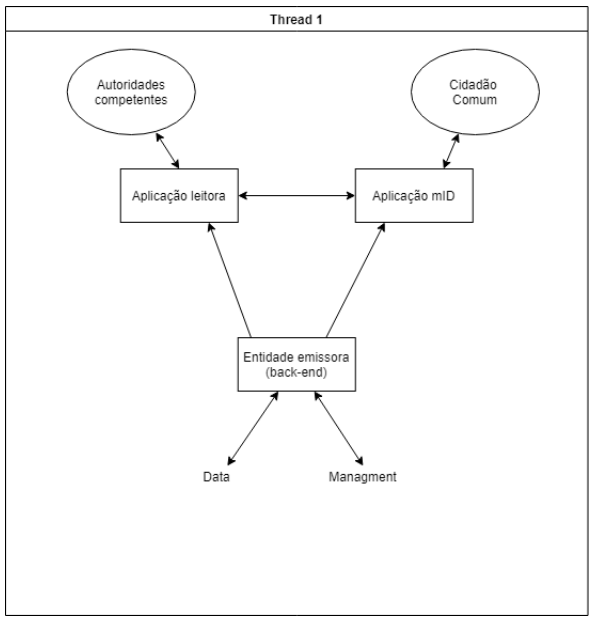

Figure 1: Data Flow Diagram

4.2 Swim Lane Diagram

Nos seguintes diagramas podemos observar as interações entre os diversos componentes do sistema, como por exemplo o envio de dados entre o portador e o leitor.

4.2.1 Comunicação Portador -> Leitor

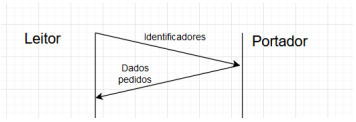

Figure 2: Swim Lane Leitor-Portador/Leitor

4.2.2 Comunicação Emissor -> Portador/Leitor

Após análise do documento fornecido chegamos à conclusão que a comunicação entre o emissor e o portador e a comunicação entre o emissor e o leitor são, em termos de segurança, semelhantes. Como tal o seguinte diagrama irá ser utilizado para representar essas comunicações.

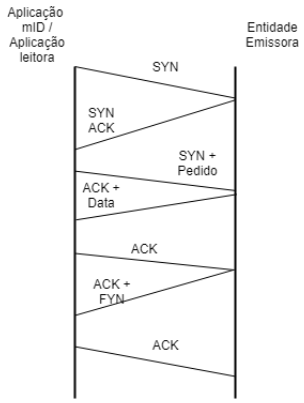

Figure 3: Swim Lane Emissor-Portador/Leitor

4.3 Trust Boundary Diagram

Os Diagramas de Fronteiras de Confiança são utilizados para identificar os pontos de um sistema onde os dados do programa mudam de contexto, permitindo então a identificação de entradas/saídas de dados de/para o sistema em questão.

Como podemos observar através do seguinte **TBD**, tal acontece quando dados são enviados da entidade emissora, o **Back-End** do sistema, para a aplicação portadora/leitora, o **Front-End** do sistema.

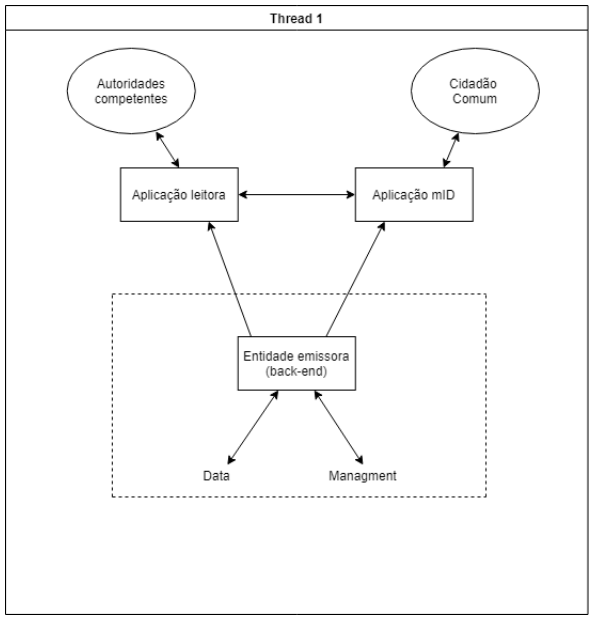

Figure 4: Trust Boundary Diagram

5 Identificação de ameaças

Esta fase consiste em identificar os ativos e as ameaças a que estes podem estar sujeitos. Em seguida estão os componentes que, na nossa opinião, são mais vulneráveis a ataques e, como tal, merecem mais a nossa atenção.

5.1 Activos a proteger

Neste sistema há três componentes principais, **Entidade Emissora, Aplicação Portadora e Aplicação Leitora**, sendo que o primeiro é o que assegura a integridade e veracidade de todos os dados do sistema e os últimos dois são os componentes mais vulneráveis a ataques, visto que são os dois componentes que fazem contacto com entidades externas humanas.

Devido a este factor, podemos afirmar que a **Aplicação Portadora e Aplicação Leitora** são componentes que têm de cumprir os requisitos de segurança básicos como por exemplo *integridade, autenticação e disponibilidade.*

Para além de proteger os componentes físicos do sistema é também necessário considerar que a imagem que o sistema transmite para os eventuais utilizadores é um activo a proteger, ou seja, é necessário que os eventuais clientes tenham confiança na aplicação.

5.2 STRIDE

Para identificarmos as potenciais ameaças à segurança do sistema seguimos o modelo leccionado nas aulas criado pela *Microsoft*, **STRIDE**.

- **Spoofing**- Utilização de identidade falsa para ganhar acesso ao sistema.
- **Tampering**- Modificação de dados não autorizada no sistema.
- **Repudiation**- Recusa de certas acções pelo utilizador.
- **Information Disclosure**- Exposição de dados não autorizada de uma entidade.
- **Denial of Service**- Tornar o sistema indisponível.
- **Elevation of Privilege**- Ganhar privilégios ou permissões que permitam fazer algo no sistema que não deveria conseguir fazer normalmente.

De seguida vamos ver as ameaças para cada componente bem como será possível proteger contra estas.

5.2.1 Spoofing

Spoofing na Aplicação mID (Aplicação do portador)
Um atacante pode fazer com que o seu dispositivo transmita dados válidos pertencentes a outro utilizador. Uma forma de combater esta vulnerabilidade seria, por exemplo, adicionar algo que identificasse o dispositivo de um utilizador, ou seja, o atacante sem ter o dispositivo do outro utilizador não poderia usufruir da identificação dele.

Spoofing na Aplicação Leitora
Um atacante poderá falsificar o seu dispositivo de forma a tornar-se numa entidade leitora. Para precaver esta vulnerabilidade teria de se implementar algo que validasse a identificação do leitor.

Spoofing na Entidade Emissora
Um atacante pode, através de *DNS Spoofing* fazer com que a entidade leitora ou portadora tentem transferir os dados de um sistema "envenenado", o que iria dar acesso ao atacante às credenciais do leitor/portador. Para combater esta vulnerabilidade a entidade emissora tem de ter algo que a identifique como a entidade certa e os dados enviados para a autenticação devem ser encriptados.

5.2.2 Tampering

Tampering de Memoria na aplicação mID
Um documento, ao ser transferido, pode ser alterado enquanto está a atravessar a rede. Para precaver esta vulnerabilidade podemos adicionar um *checksum* ao documento de forma a verificar se o ficheiro que foi recebido é igual ao ficheiro que foi enviado.

A aplicação portadora mantém sempre uma cópia local dos documentos do utilizador, como é óbvio, esta cópia poderá ser alterada pelo utilizador para, por exemplo, colocar uma idade mais alta. Para combater esta vulnerabilidade o documento pode ser *hashed* o que dificultará qualquer edição e pode também ter um *checksum* para verificar se o ficheiro foi alterado.

5.2.3 Repudiation

Neutralizar o sistema de auditoria
Um atacante pode atacar os registos que são utilizados para auditoria. Uma forma de combater esta vulnerabilidade seria através de *checksums*, ficheiros encriptados e uma forma de comparar o ficheiro que a aplicação leitora criou para auditoria com o ficheiro que a aplicação portadora criou para auditoria.

5.2.4 Information Disclosure

Information Disclosure nos processos

Um atacante pode, através de casos de erro na aplicação mID ou leitora, tentar provocar uma corrupção na memória de forma a ganhar acesso a informações restritas. Esta ameaça pode ser combatida através da implementação de proteções contra *Memory Corruption*.

Information Disclosure nas bases de dados

Um atacante pode ganhar acesso a dados confidencias de um utilizador através do ficheiros *log* ou *temp*. Uma maneira de combater esta vulnerabilidade seria através da limpeza periódica desses ficheiros.

Um atacante pode ganhar acesso a dados confidencias de um utilizador através da memória *swap*. Esta vulnerabilidade pode ser neutralizada através de ferramentas de segurança nesta componente do sistema.

Information Disclosure nos fluxos de dados

Um atacante pode ganhar acesso a dados relevantes de um utilizador através de *packet sniffers* ou até se conseguir redirecionar o tráfego de forma a passar por uma máquina que este controle. Estas vulnerabilidades podem ser neutralizadas se os pacotes forem cifrados com pares de chaves públicas e privadas, ou seja, só o destino final é que pode decifrar as mensagens.

5.2.5 Denial of Service

Denial of Service contra processos

Um atacante pode provocar um elevado uso dos recursos computacionais da entidade emissora através de pedidos *Web* o que levará ao throttling ou até que a entidade emissora fique indisponível. Uma maneira de combater esta vulnerabilidade seria, por exemplo, através de um sistema que rejeite todos os pedidos *Web* que não pertençam a uma aplicação emissora ou portadora, sendo que também podiam ser implementadas ferramentas que controlem o uso de recursos computacionais de cada processo.

Denial of Service fluxos de dados

Um atacante pode injetar pacotes de modo a provocar uma congestão da rede da entidade emissora o que irá impedir o transporte de dados desta mesma. Esta ameaça pode ser combatida através de ferramentas de controlo de congestão e através da criação de vários servidores para a entidade emissora.

6 Conclusão

Na elaboração deste trabalho foi possível obter um conhecimento mais aprofundado sobre a modelação de um sistema e a identificação de possíveis ameaças a que este sistema possa estar sujeito.Foi possível também concluir que a modelação de um sistema pode ser abordado de várias formas e que por cada abordagem feita ao sistema é possível identificar possíveis ameaças diferentes. A nossa abordagem baseou-se mais nos ativos do sistema, mas outro tipo de abordagem diferente, poderia ser mais concretamente nos softwares utilizados. Em suma, acreditamos que os objetivos que nos foram propostos foram alcançados.