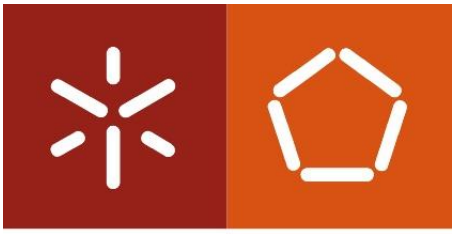


Universidade do Minho

Mestrado Integrado em Engenharia Informática

Segurança de Sistemas Informáticos

Relatório do Trabalho Prático 1

Grupo 11
Lázaro Pinheiro (A86788)
Luis Ramos (A83930)

1 de julho de 2021

Conteúdo

Lista de Figuras

1 Introdução

O presente relatório desenvolve-se no âmbito da Unidade Curricular Segurança de Sistemas Informáticos, tendo como principal objectivo identificar e clarificar potenciais incidentes de segurança a que um sistema de identificação digital e móvel poderá estar exposto.

Como era requisito do enunciado, este relatório comporta a descrição e modelação da arquitectura e entidades do sistema e, posteriormente, para cada entidade serão apresentadas possíveis ameaças e as respectivas soluções objectivando a sua mitigação.

Com a realização deste trabalho prático, o grupo objectiva uma melhor compreensão e aplicação dos conteúdos teóricos abordados em contexto sala de aula, tais como *Exploits*, *Threat Modelling*, *Vulnerabilities* e *Weaknesses*.

2 Arquitectura e Entidades do Sistema

O sistema em estudo consiste num sistema de identificação digital e móvel, o qual permitirá ao cidadão identificar-se a partir de um documento de identificação digital armazenado no seu *smartphone*, o que permitirá a substituição dos documentos de identificação pessoais físicos, podendo agregar vários documentos de identificação numa única aplicação facilmente transportável. Com muita frequência o ser humano sai de casa sem a carteira, onde até então estavam alojados os seus documentos de identificação, mas com muito menos frequência o *smartphone* fica esquecido, pois vivemos numa Era tecnológica e digital.

2.1 *System Model*

Para facilitar a compreensão do problema em mãos, a modelação do sistema em questão torna-se fulcral, uma vez que simplifica a perceção do mesmo, dado que facilita a abstração da implementação, contudo concretiza os pontos obscuros da segurança. Assim sendo, para realizar a modelação do sistema recorreu-se a um Diagrama de Fluxo de Dados, apresentado de seguida.

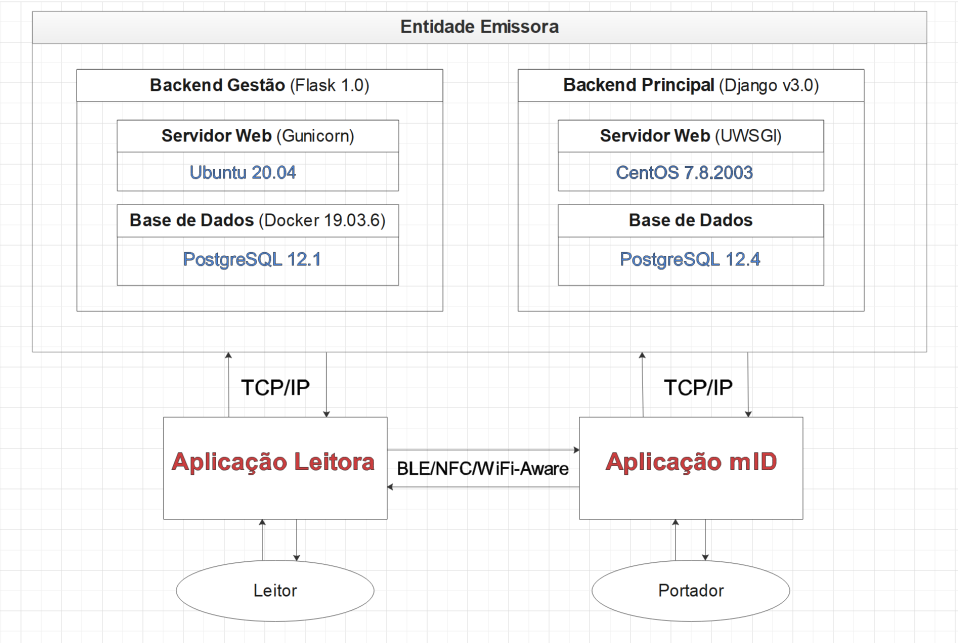


Figura 1: Diagrama de Fluxo de Dados

2.2 Entidades do Sistema

As entidades presentes no sistema alvo da análise são:

- Portador, pessoa ou entidade que é proprietária do documento de identificação;
- Verificador, pessoa ou entidade a quem se faz prova de identidade;
- Emissor, entidade com poder de emitir e conferir autenticidade a um documento de identificação pessoal.

2.3 Requisitos das Entidades

De seguida, apresentam-se os requisitos e quais as especificações de cada entidade do sistema em análise.

2.3.1 Portador e Verificador

O Sistema estará ao alcance de qualquer pessoa desde que possua um dispositivo pessoal cujo sistema operativo seja:

- *Android;*
- *iOS.*

Neste, o utilizador efectuará a instalação de uma aplicação dependendo do seu "papel"no sistema, isto é, se o utilizador desempenhar a função de Verificador instalará a aplicação leitora, caso contrário o utilizador é um Portador e deverá instalar a aplicação *mID*.

Note-se que a condição do sistema operativo que o dispositivo possuí é necessária mas não suficiente. O dispositivo deve ainda suportar os seguintes protocolos de comunicação:

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

Protocolos estes que são utilizados na comunicação entre a aplicação do Portador e a aplicação do Verificador.

2.3.2 Emissora

A entidade emissora é dividida em dois grandes serviços, onde cada um deles possuí certos componentes. De seguida, para cada serviço são listados os seus componentes e a especificação de cada um deles no seguinte formato "componente : especificação".

- Serviço Principal do Sistema:
  - *Backend* : *Django v3.0*
  - Base de dados : *PostgreSQL 12.4*
  - Servidor *web*: *UWSGI*
  - Sistema operativo : *CentOS 7.8.2003*
- Serviço de Gestão do Sistema:
  - *Backend* : *Flask 1.0*
  - Servidor *web*: *Gunicorn*
  - Sistema operativo : *Ubuntu 20.04*
  - Virtualizador: *Docker 19.03.6*
    - Base de dados : *PostgreSQL 12.1*

De realçar que a comunicação entre cada aplicação e entidade emissora é realizada via rede pública recorrendo a tecnologias suportadas pela arquitectura ***TCP/IP***.

2.4 Interacções entre as Entidades

Para facilitar a compreensão do sistema em estudo, o grupo decidiu modelar as interações entre as entidades recorrendo a *Use Cases*.

2.4.1 Autenticação do Portador na Infraestrutura da Entidade Emissora

Cenário: "Na primeira vez que o portador usa a aplicação, esta conecta-se com a infra-estrutura de uma entidade emissora de documentos (usando comunicação TCP/IP) para o download de todos os dados associados a este documento. Para isso, o cidadão autentica-se ao respectivo serviço por forma a iniciar a transferência do seu documento."

Ator: Portador

Pré-condição: O Ator não está autenticado e este encontra-se *on-line*.

Pós-condição: A Aplicação *mID* possuí todos os dados associados a um documento.

Fluxo Normal:

1. O Ator autentica-se na Entidade Emissora;
2. A Entidade Emissora valida a autenticação e permite o download do documento requerido;
3. A aplicação *mID* efetua o *download* e posterior armazenamento.

2.4.2 Atualização dos dados da Aplicação do Portador

Cenário: "... Esta operação repete-se periodicamente para eventuais atualização de dados, neste caso, sem recorrer a uma autenticação explicita ao sistema."

Ator: Portador

Pré-condição: O Ator está autenticado, encontra-se *on-line* e existe uma atualização de dados.

Pós-condição: A Aplicação do *mID* atualiza os dados associados ao documento.

Fluxo Normal:

1. A aplicação *mID* à Entidade Emissora;
2. A Entidade Emissora aprova a transferência do documento em questão;
3. A aplicação *mID* efetua o download e posterior atualização dos dados do documento.

2.4.3 Prova de Identidade

Cenário: Todo o capítulo "*Descrição do sistema em desenvolvimento*"

Ator: Portador, Verificador e Emissora

Pré-condição: "O Portador está disposto a fazer a prova de identidade e tanto o Verificador como o Portador possuem as aplicações instaladas nos seus dispositivos pessoais e encontram-se autenticados na infraestrutura da entidade emissora."

Pós-condição: "A prova de identidade foi realizada com sucesso"

Fluxo Normal:

1. O Portador disponibiliza um *QR Code* gerado pela aplicação *mID*;
2. O Verificador efetua a leitura do *QR Code* disponibilizado pelo Portador;
3. É estabelecido um canal de comunicação entre a aplicação do Portador e a do Verificador;
4. O Verificador decide que o modo de operação será *on-line*;
5. O dispositivo leitor transfere um *token* de autorização da aplicação do portador;
6. O verificador a partir do *token* transferido consulta diretamente a entidade Emissora;
7. A entidade Emissora permite o acesso do Verificador aos dados do Portador;
8. O Verficador consulta os dados aos quais o Portador lhe permitiu acesso.

Fluxo Alternativo:

4.1 O Verificador decide que a operação irá decorrer no modo *off-line*;

4.2 O Verificador envia um pedido contendo os identificadores dos atributos que deseja;

4.3 O Portador seleciona os atributos que pretende disponibilizar;

4.4 O dispositivo do portador transfere os atributos de identificação e dados necessários para a sua verificação diretamente para o dispositivo leitor;

4.5 Volta ao passo 8.

3 *Find and Address Threats*

Esta secção é dedicada à modelação de possíveis ameaças ao sistema em análise. De seguida é descrita a modelação de ameaças, por forma a uma melhor compreensão das possíveis ameaças das entidades do sistema.

O que é?

Hoje em dia existem várias definições, tais como a que *Microsoft* cita, onde refere que *"It's an engineering technique you can use to help you identify threats, attacks, vulnerabilities, and countermeasures that could affect your application."*[1]. Outros autores referem-se à modelação de ameaças como "Threat modeling is the use of abstractions to aid in thinking about risks."[2] Tendo em conta estas descrições a modelação de ameaças é um processo de análise complexo que visa procurar o que pode estar errado no sistema a construir e a idealizar. Esta é orientada aos ativos, atacantes e *software*.

Porquê?

O modelo de ameaças ajuda a travar e a prevenir ameaças, fortalecendo um produto final com segurança.

Como?

Recorrendo à abstração das ameaças, tentando encaixar-las num sistema de classes desenvolvido pela Microsoft intitulado por **STRIDE**(acrónimo de cada componente apresentadas sucintamente de seguida), ou seja, um modelo de ameaças usado para ajudar a raciocinar e encontrar ameaças num sistema. O modelo **STRIDE** é usado em conjunto com um modelo do sistema(desenvolvido na secção 2.1) resultando numa análise completa dos processos, armazenamento de dados, fluxos de dados e limites de confiança.

- *Spoofing*:
  - Descrição - Fingir ser algo ou alguém que não o próprio;
  - Propriedade violada - Autenticação;
- *Tampering*:
  - Descrição - Modificar dados em disco, na rede ou em memória volátil;
  - Propriedade violada - Integridade;
- *Repudiation*:
  - Descrição - Ato de recusar a autoria de algo que aconteceu;

  - Propriedade violada - Não Repúdio;

- *Information Disclosure*:
  - Descrição - Divulgar informações a uma entidade que está não autorizada a ter acesso a esta;
  - Propriedade violada - Confidencialidade;

- *Denial of service*:
  - Descrição - Consumir desnecessariamente recursos necessários a um serviço com o intuito de os esgotar;
  - Propriedade violada - Disponibilidade;

- *Elevation of privilege*:
  - Descrição - Permitir que uma entidade faça algo que não está autorizada a fazer;
  - Propriedade violada - Autorização;

3.1 Aplicação *mID* (aplicação do portador)

A aplicação *mID*, denominada também de aplicação do portador, é, como o próprio nome sugere, a aplicação que irá permitir ao utilizador usufruir do serviço e das funções que esta dispõe.

Esta aplicação comunica com a aplicação leitora através de um canal **BLE**, **NFC** ou **WiFi-Aware** e comunica com a entidade emissora através de um canal **TPC/IP**.

A primeira vez que o utilizador interage com a aplicação, esta conecta-se com a infra-estrutura da entidade emissora de modo a realizar o download de todos os dados associados ao respectivo documento, sendo estes dados transferidos no formato **JSON**.

Todas as operações que correspondem a uma prova de identidade ocorre entre um dispositivo leitor e um dispositivo portador. A comunicação entre estes dois dispositivos é realizada por mensagens codificadas no formato **CBOR**.

Algumas garantias a ter:

- O canal de comunicação **TPC/IP** aberto para o download de todos os dados associados ao respectivo documento é seguro e protegido;
- É o verdadeiro utilizador a autenticar-se na aplicação para iniciar a devida transferência;
- A integridade dos dados armazenados no dispositivo do portador;

- Verificar se os dados transferidos são transmitidos pela entidade emissora, através da sua assinatura;
- O canal de comunicação **BLE**,**NFC**,**WiFi-Aware** aberto para a autenticação do utilizador é seguro e protegido;
- As operações efectuadas entre o leitor e o portador são seguras em qualquer um dos modos (on-line/off-line);
- O algoritmo de codificação das mensagens é seguro e não apresenta riscos para o respectivo conteúdo, garantindo a integridade e confidencialidade desses dados;

Vulnerabilidades:

- *[Spoofing]*:
  - Orientado aos Ativos:
    - Aceder à aplicação com as credenciais de outro utilizador;
  - Orientado aos Atacantes:
    - O dispositivo foi perdido ou roubado;
- *[Denial of Service]*:
  - Orientado aos Ativos:
    - Não consegues efetuar operações;
  - Orientado aos Atacantes:
    - Bloqueia a rede de comunicação, impedindo o utilizador de comunicar com a *backend* e vice-versa;
  - Orientado ao Software:
    - Os recursos do software são esgotados, não conseguindo dar resposta;
- *[Tampering]*:
  - Orientado aos Ativos:
    - Alterar os seus próprios dados;
  - Orientado aos Atacantes:
    - O atacante interpõe-se no meio das comunicações com o intuito de alterar os dados que são trocados entre as entidades, comprometendo assim a integridade dos dados;

Soluções:

- *[Tampering]*:
  - Orientado ao *Software*:
    - Para garantir a integridade dos dados armazenados, ou seja, garantir que eles não foram alterados, podemos recorrer a uma técnica que usa 3 ficheiros (texto, espelho e estrutura);

3.2 Aplicação leitora (aplicação do verificador)

A aplicação leitora é muito semelhante à aplicação *mID*, pois ambas suportam exatamente os mesmos protocolos de comunicação e o dispositivo onde esta é utilizada necessita de possuir um dos sistemas operativos especificados.

Esta aplicação permite ao Verificador estabelecer uma comunicação com um portador e, posteriormente, a requisição dos dados que pretende consultar, com a finalidade de identificar o Portador em questão. Note-se que a requisição dos dados a consultar poderá ser *on-line* ou *off-line*.

Algumas garantias a ter:

- O Verificador para executar uma operação de identificação de um Portador, tem que se encontrar autenticado na infraestrutura da entidade Emissora;
- O canal de comunicação criado entre as aplicações *mID* e portadora, usando os protocolos de comunicação supra citados, é seguro e protegido na comunicação ponto-a-ponto.
- Em qualquer modo de operação(*on-line*/*of-line*) a comunicação é fiável e segura;
- No modo *on-line*, a aplicação leitora não ser capaz de alterar a lista de atributos autorizados pelo Portador antes da consulta à infraestrutura da entidade emissora;
- O canal de comunicação **TPC/IP** aberto entre a aplicação leitora e a infraestrutura da entidade emissora para autenticação do Verificador é seguro e protegido;
- O canal de comunicação **TPC/IP** aberto entre a aplicação leitora e a infraestrutura da entidade emissora para o download dos dados a que o Portador lhe concedeu acesso, via *token* de autorização é protegido;
- o *token* recebido pela aplicação *mID* tem que ser válido.

Vulnerabilidades:

- *[Spoofing]*:
  - Orientado aos Ativos:
    - Aceder à aplicação com as credenciais de outro utilizador;
  - Orientado aos Atacantes:
    - O dipositivo foi perdido ou roubado;
- *[Tampering]*:
  - Orientado aos Ativos:
    - Alterar os dados a que o Portador lhe conferiu acesso;
    - Alterar os seus próprios dados;
  - Orientado aos Atacantes:
    - O Portador pretende alterar os seus dados, em seu benefício;
    - O atacante interpõe-se no meio das comunicações com o intuito de alterar os dados que são trocados entre as entidades, comprometendo assim a integridade dos dados;
  - Orientado ao Software:
    - A não utilização de um *software anti-tampering*, permite a injeção de código que irá ser executado e o qual poderá aceder e alterar dados, comprometendo assim o normal funcionamento da aplicação e a integridade dos dados transacionados;
- *[Repudiation]*:
  - Orientado ao Software:
    - O protocolo de comunicação **TCP/IP** e os 3 protocolos de comunicação entre as aplicações móveis não garantem não-repúdio;
- *[Information Disclousure]*:
  - Orientado aos Ativos:
    - O Verificador após obter os dados requeridos a um Portador pode, acidentalmente ao não, partilhar os dados que tem em mãos;
  - Orientado aos Atacantes:
    - Um atacante pode interpor-se entre as comunicações tanto no modo *on-line* como *off-line* obtendo todos os dados em transacção, comummente intitulado por "*man-in-the-middle*", no caso de não se usar comunicação segura e os dados não estarem encriptados;

        * Após o *Tampering* do dispositivo, o atacante ao ter acesso ao material criptográfico utilizado para encriptar e desencriptar os dados transacionados, pode interpor-se na comunicação e obter os dados transacionados que outrora se pensavam encriptados;
    - Orientado ao Software:
        * Um *developers* pode colocar inadvertidamente informações sensíveis ou dados em num local no dispositivo móvel que é facilmente acedido por outros aplicações.

- *[Denial of Service]*:
    - Orientado aos Ativos:
        * O Verificador desliga o *Wi-fi*, ou o *Bluetooth* ou o *NFC*, acidentalmente ou propositadamente;
    - Orientado aos Atacantes:
        * *Bluetooth*:
            · O envio de pacotes um pouco maiores, ou envio massivo de pacotes, fazendo com que o dispositivo alvo não consiga tratar de tráfego e comece a descartar-lo, bem como toda e qualquer conexão estabelecida ou tentativas de estabelecimento enquanto sob-ataque;
            · Geração de ruído, ou *jamming*, seja através da identificação da sequência de saltos da mesma ordem de frequência dos dispositivos envolvidos, seja através do preenchimento de boa parte do espectro com tráfego de ruído;
        * *NFC*:
            · Recorrendo a um transmissor de ondas de rádio, o atacante transmite ondas de frequências válidas no tempo correto de forma a perturbar a comunicação entre os dispositivos do Portador e Verificador, impedindo assim, que o Verificador receba um sinal que consiga compreender. Uma vez que o atacante não consegue adulterar os dados, apenas impede que os dados cheguem ao Verificador.
        * *Wifi-Aware*:
            · A transmissão lícita de tramas de Sincronização *Beacon* numa **NAN**(*Neighbor Awareness Networking*) é fundamental. Um atacante pode interromper o processo de sincronização, enviando tramas *Beacon* corrompidas, incluindo informações de *cluster* falsas. Devido ao facto das informações de sincronização são transmitidas por vários

dispositivos numa NAN, o processo é robusto o suficiente para superar as tramas *Beacon* maliciosas individuais e específicas. No entanto, uma transmissão contínua resultará numa negação de serviço;

Soluções:

- *[Spoofing]*:
  - Orientado aos Ativos:
    - Associar à aplicação leitora dados biométricos como complemento de autenticação;
    - A cada operação solicitar autenticação em qualquer um dos modos de operação(e.g; *on-line/off-line*);
  - Orientado aos Atacantes:
    - Colocar um mecanismo de autenticação no acesso à aplicação leitora(e.g; leitor biométrico, PIN ...);
    - Bloquear ou limpar as informações sensíveis remotamente;
    - A cada operação solicitar autenticação em qualquer um dos modos;
- *[Tampering]*:
  - Orientado aos Ativos:
    - Não permitir inserção de dados em nenhuma operação, a não ser na autenticação;
  - Orientado aos Atacantes:
    - Não permitir inserção de dados em nenhuma operação, a não ser na autenticação;
    - Utilizar encriptação dos dados trocados;
  - Orientado ao *Software*:
    - Utilização de *software anti-tampering*;
- *[Repudiation]*:
  - Orientado aos Ativos:
    - Utilizar um *token* de sessão por forma a garantir o não-repúdio, uma vez que o campo *Signature* consiste na cifra, usando o algoritmo da chave simétrica, do campo *Payload* garantindo assim a integridade dos dados e o não-repúdio;
  - Orientado ao *Software*:

        * Por forma a complementar o facto das comunicações não possuírem um mecanismo de não repúdio, utilizam-se mecanismos tais como:
            · Criptografia - O algoritmo de chave pública escolhido deve fornecer à aplicação o que é necessário em termos de funcionalidades e se a segurança provida é ideal para a aplicação bem como o desempenho;
            · **TLS** - *Transport Layer Security*;
            · **VPN** - *Virtual Private Network*;

- *[Information Disclousure]*:
    - Orientado aos Ativos:
        * Controlar o acesso à área de transferência para impedir funções de copiar e colar;
        * Bloquear capturas de ecrã;
    - Orientado aos Atacantes:
        * Todos os dados partilhados entre as entidades devem ser encriptados;
        * No caso de *tampering*, utilizar criptografia assimétrica.
        * No modo *on-line* o utilizar protocolo **TLS** com a finalidade de a comunicação decorrer no modo seguro;
    - Orientado ao Software:
        * Evitar armazenar dados sensíveis no dispositivo. Se necessário, verificar se se encontram encriptados e em seguida armazenar apenas em arquivos ou bases de dados.
        * O uso de um modelo baseado em *conteiners*.

- *[Denial of Service]*:
    - Orientado aos Ativos:
        * A aplicação leitora não permite ao Verificador que desligue qualquer modo de comunicação no decorrer de uma operação;
    - Orientado aos Atacantes:
        * *NFC*:
            · Este tipo de ataque supra citado, embora plenamente viável, pode ser também facilmente detetado, por meio de uma verificação do campo eletromagnético, além de poder ser impedido pelo uso de um canal de comunicação seguro.

3.3 Entidade Emissora (backend do sistema)

A entidade emissora é a entidade com poder de emitir e conferir autenticidade a um documento de identificação pessoal.

Esta entidade é também responsável por prover os mecanismos que garantem a autenticidade e integridade dos documentos digitais transmitidos tanto no modo *on-line*, quanto no modo *off-line*.

De seguida iremos analisar cada componente da estrutura da entidade, referida na secção 2.3.2.

- Django v3.0:
  - Foram encontradas **406** vulnerabilidades neste componente sendo que a maioria é de uma escala média *(6.0 Medium)*.
- Flask 1.0:
  - Foi encontrada apenas **1** vulnerabilidade neste componente sendo a sua escala alta *(7.5 High)*.
- PostgreSQL 12.4:
  - Foram encontradas apenas **2** vulnerabilidades neste componente sendo que as duas foram de uma escala alta *(7.3 High e 7.1 High)*.
- PostgreSQL 12.1:
  - Foram encontradas **471** vulnerabilidades neste componente sendo que a maioria é entre a escala média e a escala alta*(5.1 Medium e 7.5 High)*.
- Gunicorn:
  - Foram encontradas apenas **5** vulnerabilidades neste componente sendo que uma delas foi de uma escala bastante elevada *(9.1 Critical)*.
- UWSGI:
  - Foram encontradas apenas **4** vulnerabilidades neste componente sendo que duas delas foram de uma escala bastante elevada *(9.8 Critical)*.
- CentOS 7.8.2003:
  - Foram encontradas apenas **90** vulnerabilidades neste componente sendo que algumas delas foram de uma escala alta *(7.5 High)*.
- Ubuntu 20.04:

– Foi encontrada apenas **1** vulnerabilidade neste componente sendo a sua escala alta *(7.8 High)*.

De modo a garantir uma maior segurança decidiu-se que seria benéfico adicionar à infra-estrutra:

- Um algoritmo de encriptação para as comunicações, especialmente para a comunicação em **TCP/IP**, de modo a proteger todos os dados e toda a informação confidencial.

3.3.1 Análise de vulnerabilidades dos componentes

- Django v3.0
  - Análise da Descrição:

    Após uma pesquisa na base de dados de vulnerabilidades e exposições comuns, foi encontrada uma vulnerabilidade bastante preocupante. Essa vulnerabilidade permitia aos atacantes injetar SQL malicioso.

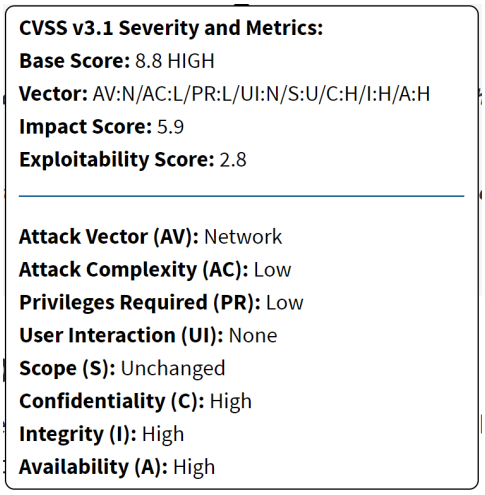
CVE-2020-9402 Detail

Current Description

Django 1.11 before 1.11.29, 2.2 before 2.2.11, and 3.0 before 3.0.4 allows SQL Injection if untrusted data is used as a tolerance parameter in GIS functions and aggregates on Oracle. By passing a suitably crafted tolerance to GIS functions and aggregates on Oracle, it was possible to break escaping and inject malicious SQL.

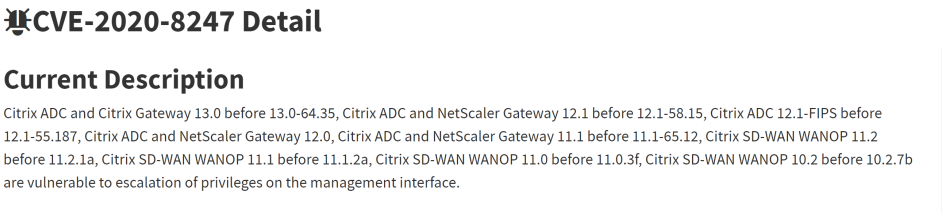
  - Análise do Impacto

    Como é possível verificar na figura abaixo, esta vulnerabilidade teve um impacto bastante notável sendo a sua **base score** de *8.8 High*



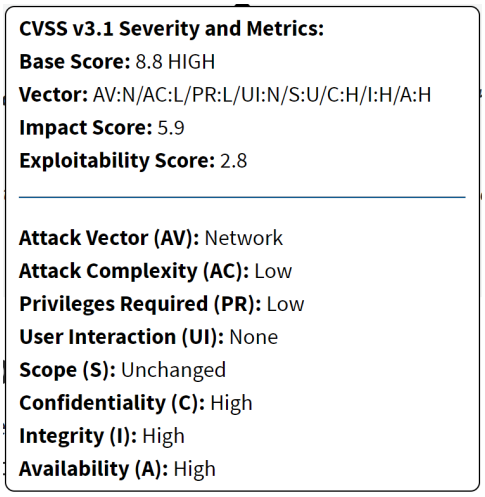
- PostgreSQL 12.1
  - Análise da Descrição:

    Após uma pesquisa na base de dados de vulnerabilidades e exposições comuns, foi encontrada uma vulnerabilidade bastante preocupante. Tal falha deixava a base de dados vulnerável a *Elevation of Privilege* na interface de gestão.



  - Análise do Impacto

    Como é possível verificar na figura abaixo, esta vulnerabilidade teve um impacto bastante notável sendo a sua **base score** de *8.8 High*



4 Conclusão

Culminada a elaboração deste trabalho prático, importa referir que a execução do mesmo permitiu aos elementos do grupo compreender a arquitetura e o modo de funcionamento do sistema alvo da análise.

A execução desta fase do trabalho prático permitiu uma melhor consolidação dos construtos teóricos através da análise dos mesmos no serviço de identificação digital e móvel.

Ao nível de dificuldades sentidas, importa referir a complexidade sentida no início da modelação do sistema.

Este trabalho prático tornou-se fulcral, pois a consolidação dos construtos teóricos aplicados a este sistema, torna-se uma base sólida que permitirá aprofundar os conceitos mais práticos da área da segurança dos sistemas informáticos.

No ímpeto geral o desenvolvimento desta fase do trabalho decorreu como planeado, alcançando os objetivos delineados pelo enunciado.

Referências

[1] Microsoft: Threat Modeling,
`https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling`

[2] Adam Shostack. *Threat Modeling: Designing for Security.* (WILEY)