Universidade do Minho

Mestrado em Engenharia Informática

Tecnologias de segurança

Trabalho prático 2

Grupo 1

Rui Carlos Azevedo Carvalho - PG47633
Daniel Barbosa Miranda - PG47123
Ana Luísa Lira Tomé Carneiro - PG46983

Contents

Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular Tecnologias de Segurança e tem como objetivo analisar um sistema de identificação digital móvel no sentido de descobrir potenciais vulnerabilidades, problemas e fraquezas aos quais as componentes do sistema podem estar expostas. Nesta análise foi também avaliado o risco associada a cada fraqueza prevista assim como possíveis formas de solucionar o problema.

O sistema a analisar é composto por uma infraestrutura de suporte (entidade emissora) que vai emitir o documento de identificação a ser usado pelo o utilizar e vai atualizar esse documento periodicamente. O sistema é também composto pela aplicação mID do portador onde será armazenado o documento de identificação digital do próprio e pela aplicação do leitor que vai validar, caso seja necessário, o documento de identificação do portador. A cada uma destas componentes foi aplicado uma metodologia de análise de potenciais baseada no STRIDE, sendo que para a entidade emissora também foi aplicada a catalogação de vulnerabilidades associadas a componentes de software da entidades, visto que esta já se encontra implementada.

Metodologia seguida

O processo que foi utilizado para realizar a análise deste sistema inicia-se pelo desenvolvimento de um modelo de ameaças orientado ao software onde se a aplicou o STRIDE (*spoofing*, *tampering*, *information disclosure*, *DoS* e *elevation of privileges*) a cada uma das componentes do sistema. Para cada uma das categorias do STRIDE verificou-se quais são as fraquezas que o sistema possui que estão relacionadas com a categoria em questão assim como todas as ameaças que podem surgir através dessas fraquezas. Para além disto, tentamos indicar possíveis medidas para mitigar cada uma destas, sendo que, também se procedeu à análise do seu risco. Esta última consiste em determinar um valor para o risco da fraqueza se verificar no sistema em produção sendo atribuído um valor que vai de 1 a 5. Para além disto, atribui-se também um valor ao impacto da fraqueza também numerado de 1 a 5. Por conseguinte, a criticidade é obtida pela soma do valor do risco e do impacto, sendo que, através deste valor é possível indicar quais as fraquezas que devem ser eliminadas com mais urgência.

Por fim, procedeu-se à catalogação de vulnerabilidades conhecidas para o *software* utilizado pela entidade emissora e à enumeração dos pontos mais críticos do sistema.

Modelação de ameaças orientadas ao software

3.1 Aplicação do portador

Esta componente do sistema é uma aplicação em *smartphone* que todos os utilizadores possuem para terem acesso a documentos que permitem a sua identificação digital móvel. Os portadores obtém este documento a partir da entidade emissora do documento após a sua autenticação na emissora. O documento é armazenado no dispositivo do aplicador e vai ser alvo de atualizações periódicas por parte da emissora. Este documento será utilizado para fornecer à aplicação leitora uma forma de identificação digital móvel segura e eficiente.

De seguida é apresentado uma análise detalhada das possíveis fraquezas desta identidade, assim como possíveis ameaçadas que podem ocorrer caso as fraquezas apresentadas não sejam mitigadas. É também apresentado nas secções seguintes uma avaliação e análise do risco associado a cada uma das fraquezas apresentadas.

3.1.1 Spoofing

Os ataques de *Spoofing* são normalmente provocados pela falsificação de dados onde uma entidade identifica-se com sucesso como outra identidade para obter vantagens ilegitimas sobre a vítima. Desta forma, uma entidade maliciosa mascara-se como uma entidade de confiança para tentar obter informação confidencial que que outra entidade lhe envia a pensar serem seguras.

Fraqueza: Autenticação da entidades utilizando parâmetros gerados no estabelecimento da comunicação

Ameaça A ameaça principal associada a ataques de *spoofing* está na falsificação dos dados enviados por um atacante, fazendo com que a vítima pense que estes dados foram enviados por uma entidade de confiança. Caso a vítima não identifique corretamente a entidade que lhe está a enviar os dados é bem possível que esta estabeleça uma comunicação com uma entidade maliciosa permitindo o envio de informação sensível que possa ser roubada pelo atacante.

Forma de Mitigação Esta ameaça pode ser provocada através da autenticação das entidades na comunicação utilizando exclusivamente parâmetros gerados pela conexão como endereços IP, ID de conexão BLE ou NFC, entre outros. Contudo pode ser resolvida se a aplicação implementar métodos seguros de autenticação e verificação para

todo o acesso remoto, como por exemplo o uso de *passwords* e assinaturas digitais. Esta métricas permitem uma autenticação forte e verificação da identidade que lhe enviou a mensagem.

Análise de Risco Esta fraqueza como está prevista ser mitigada quando é expresso nos requisitos que é necessário garantir mecanismos robustos de autenticação, a confidencialidade e a integridade dos dados transferidos para o dispositivo do portador, então o risco associado a esta fraqueza é baixo. Já o impacto desta fraqueza é médio-alto visto que estes ataques impedem o funcionamento de todo os sistema de forma confidencial e íntegra. Desta forma temos:

- RISCO: 1
- IMPACTO: 4
- CRITICIDADE: 1 + 4 = 5

Fraqueza: Armazenamento das credencias do portador de forma não segura

Ameaça Uma das ameças associadas a ataques de *spoofing* existe quando uma entidade maliciosa consegue obter as credências de autenticação do portador da aplicação, visto estas estarem armazenadas no sistema sem estarem cifradas. Isto pode acontecer, caso a *password* seja armazenada em *plaintext* ou facilmente identificada descobrindo padrões entre as *passwords* armazenadas. Desta forma, a entidade maliciosa consegue aceder à conta do portador permitido que esta entidade consiga realizar operações e funcionalidades em nome do portador, ou seja, consegue-se fazer passar pelo portador e enganar as restantes entidades como o leitor e a emissora.

Forma de Mitigação Esta ameaça pode ser provocada por credências de autenticação de um portador não estarem armazenadas de forma segura, mas pode ser resolvida através da implementação implementar mecanismos que cifram de forma segura as *passwords* utilizando algoritmos de PBKDFs (*Password-based Key Derivation Functions*) que derivam uma *hash* de uma *password* utilizando uma função pseudo-aleatória, sendo essa a *hash* armazenada no sistema. Segundo o que é recomendado para o armazenamento de *passwords* não se deve utilizar funções de HASH simples, pois estas podem ser alvo de ataques de dicionário onde um atacante consegue descobrir padrões entre as *passwords* mais utilizadas por utilizadores e a *passaword* que pretende obter. Com a utilização de PBKDFs que a partir da *password* e de *salt* pseudo-aleatório derivam uma *hash* sempre diferente, isto é, *passwords* iguais derivam *hashs* diferentes. Assim, para se obter a *password* é necessário ter acesso ao valor do *salt*, acrescentando uma camada extra de segurança. Desta forma, torna-se difícil para um atacante descobrir qual a *password* que o portador usa para autenticação reduzindo o perigo de entidades maliciosas acederem às credências do portador e conseguirem obter os mesmos diretos na aplicação que este.

Análise de Risco Esta fraqueza não está prevista ser mitigada, pois não é expresso nos requisitos a necessidade de utilizar mecanismos de armazenamento de *passwords* de autenticação. Desta forma o risco associado a esta fraqueza é médio-alto, já o impacto

desta fraqueza é médio-alto visto que a realização de operações em acesso a dados sensíveis pode trazer grande impacto para todo o sistema. Desta forma temos:

- RISCO: 4
- IMPACTO: 4
- CRITICIDADE: 4 + 4 = 8

Fraqueza: Autenticação do portador na aplicação fraca

Ameaça Uma das ameças associadas a ataques de *spoofing* existe quando uma entidade maliciosa consegue obter as credências de autenticação do portador da aplicação, visto estas serem de fracas e de fácil obtenção. Isto pode acontecer, caso a *password* seja fraca. Desta forma, a entidade maliciosa consegue aceder à conta do portador permitido que esta entidade consiga realizar operações e funcionalidades em nome do portador, ou seja, consegue-se fazer passar por portador e enganar as restantes entidades como o leitor e a emissora.

Forma de Mitigação Esta ameaça pode ser provocada por credências de autenticação de um portador fracas e fáceis de obter, mas pode ser resolvida através da implementação do *Multi-Factor Authentication* (MFA) que permite uma autenticação por dois fatores normalmente usado em aplicações que lidam com transferências bancárias e dados sensíveis. Além disso, é também necessário obrigar ao utilizador utilizar como credencial uma *password* forte que seja difícil de adivinhar ou encontrar. Desta forma, é possível que a autenticação do portador seja bastante forte reduzindo o perigo de entidades maliciosas acederem às credencias do portador e conseguirem obter os mesmos diretos na aplicação que este.

Análise de Risco Esta fraqueza está prevista ser mitigada, pois é expresso nos requisitos a necessidade de utilizar uma autenticação robusta. Desta forma o risco associado a esta fraqueza é baixo, já o impacto desta fraqueza é médio-alto visto que a realização de operações em acesso a dados sensíveis pode trazer grande impacto para todo o sistema. Desta forma temos:

- RISCO: 1
- IMPACTO: 4
- CRITICIDADE: 1 + 4 = 5

3.1.2 Tampering

Os ataques de *tampering* dizem respeito à modificação de mensagens enviadas entre duas entidades. No caso da aplicação portador como esta troca mensagens com o emissor através de uma comunicação TCP/IP e com o portador através de uma comunicação por BLE, NFC ou Wifi-Aware, é necessário a implementação de métodos que permitem ao sistema contornar tais ataques.

Fraqueza: Transferência de dados não autenticados entre entidades

Ameaça A ameaça principal associada a ataques *tampering* está na modificação das mensagens enviadas entre as entidades. Assumindo que os canais TCP/IP e canais de comunicação entre o portador e o leitor não são seguros, é possível que um atacante intercete uma mensagem enviada neste canais e a altere mesmo sendo esta cifrada. Isto pode provocar alteração dos dados dos documentos que são enviados para do portador para o leitor em modo offline ou a alteração do token que é enviado em modo online. Além disso também é possível a alteração de dados de autenticação que são enviados para a emissora.

Forma de Mitigação Esta ameaça pode ser provocada através de uma transferência de dados sem autenticação entre o portador e a emissora e entre o portador e o leitor, mas pode ser resolvida através da utilização de um HMAC (*hash message authentication code*) forte, como por exemplo o SHA-256 com um comprimento de chave 64 bytes, que permite a criação de um código único (*hash*) da mensagem a enviar. Esta *hash* será enviada no canal não seguro juntamente com a mensagem. Do lado da entidade de destino, esta vai confirmar se a *hash* que recebeu e a *hash* da mensagem recebida são iguais. Caso forem então significa que a mensagem não foi alterada. Caso não sejam iguais, então a entidade descarta mensagem pois é possível que tenha havido uma ataque de *tampering*.

Análise de Risco Esta fraqueza como está prevista ser mitigada quando é expresso nos requisitos que a interação entre o dispositivo do portador e as restantes entidades deve garantir a confidencialidade e integridade dos dados transmitidos, então o risco associado a esta fraqueza é baixo. Já o impacto desta fraqueza é médio-baixo visto que as alterações desta mensagens não impedem o funcionamento do sistema, mas tem impacto na integridade e confiança dos dados enviados. Desta forma temos:

- RISCO: 1
- IMPACTO: 2
- CRITICIDADE: 1 + 2 = 3

Fraqueza: Modificação do documento por entidades não autorizadas é permitido

Ameaça Outra das ameaças associadas a ataques de *tampering* está na modificação do documento armazenado no dispositivo do portador por parte de entidade maliciosas. É possível que o documento que se encontre armazenado no portador seja acedido por uma outra entidade e modifique o seu conteúdo, quebrando a integridade e autenticidade do documento, tal como era previsto não acontecer nos requisitos de sistema.

Forma de Mitigação Esta ameaça pode ser provocada através do armazenamento do documento com permissões que permitam que os utilizadores do dispositivo modifiquem o documento armazenado. Esta fraqueza pode ser resolvida com o uso de permissões para os utilizadores do sistema através de permissões *read-only* para todos os utilizadores do dispositivo e permissões *write* para a aplicação e para o portador pois

são as únicas entidades que tem o direito de eliminar o ficheiro, sendo que a aplicação ainda tem o direito de criar um novo documento atualizado que lhe seja enviado pela a emissora. Desta forma é possível que os *users* do dispositivo não consigam modificar o documento visto que não tem permissões para tal.

Análise de Risco Esta fraqueza como está prevista ser resolvida quando é expresso nos requisitos que se deve garantir a integridade e autenticidade dos dados armazenados no dispositivo do portador, então o risco associado a esta fraqueza é baixo. Já o impacto desta fraqueza é médio visto que as alterações destes documentos impedem de realizar a sua funcionalidade que é servir de identificação autêntica do portador. Desta forma temos:

- RISCO: 1
- IMPACTO: 3
- CRITICIDADE: 1 + 3 = 4

Fraqueza: Armazenamento do documento não autenticado pela emissora

Ameaça Outra das ameaças associadas a ataques de *tampering* está na identificação da modificação do documento armazenado no dispositivo do portador por parte de entidade maliciosas. É possível que o documento que se encontre armazenado no portador seja acedido por uma outra entidade e modifique o seu conteúdo, quebrando a integridade e autenticidade do documento, tal como era previsto não acontecer nos requisitos de sistema.

Forma de Mitigação Esta ameaça pode ser provocada através do armazenamento do documento não autenticado pela emissora, que pode ser resolvido com o uso de uma assinatura digital, geradas através dos algoritmos DSA ou EdDSA, por parte da emissora. Assim, todos os documentos que cheguem ao portador vindos da emissora devem ser assinados por esta, pois os dados quando forem lidos pelo leitor são verificados de forma a saber se o documento foi autenticado e validado pela emissora ou se os dados do documento foram adulterados por uma entidade maliciosa. Desta forma é possível identificar a alteração do documento de identificação digital.

Análise de Risco Esta fraqueza como está prevista ser resolvida quando é expresso nos requisitos que se deve garantir a integridade e autenticidade dos dados armazenados no dispositivo do portador, então o risco associado a esta fraqueza é baixo. Já o impacto desta fraqueza é médio-alto visto que não identificação das alterações destes documentos impedem o leitor e o documento de realizar a sua funcionalidade que é servir verificar a autenticidade do documento e servir de identificação autêntica do portador. Desta forma temos:

- RISCO: 1
- IMPACTO: 4
- CRITICIDADE: 1 + 4 = 5

Fraqueza: As interações entre o portador e as entidades não estão validadas pelo portador

Ameaça Uma outra ameaça associada a ataques *tampering* está na modificação das operações realizadas entre o portador e os diversos leitores. É possível que um atacante ou o próprio leitor consiga alterar a interação entre ele e o portador, comprometo a auditoria das interações efetuadas, tal como era pedido nos requisitos. Por exemplo é possível que o leitor altere o *token* a enviar para a emissor com os atributos que o portador permitiu que este obtivesse, podendo o leitor obter mais informação em relação ao portador do que aquela que este validou.

Forma de Mitigação Esta ameaça pode ser provocada através da não validação e autenticação das interações realizadas pelo portador por parte deste. Esta fraqueza pode ser resolvida através do uso de assinaturas digitais que assinem a interação realizada, fazendo com que qualquer alteração nesta interação seja facilmente identificado. Desta forma é possível que a emissor consiga verificar se os atributos pedidos no *token* forma validados pelo o portador ou não.

Análise de Risco Esta fraqueza está prevista ser mitigada, pois é expresso nos requisitos que o portador deve auditar as interações entre o leitor. Desta forma o risco associado a esta fraqueza é baixo, já o impacto desta fraqueza é baixo visto que vão existir ativos como o *token* que vão deixar de conseguir implementar na íntegra a sua funcionalidade.Desta forma temos:

- RISCO: 1
- IMPACTO: 2
- CRITICIDADE: 1 + 2 = 3

Fraqueza: Armazenamento de todos os *logs* no dispositivo do portador

Ameaça Por fim, uma ameaça associada a ameaça e fraqueza apresentadas anteriormente está na modificação dos *logs* armazenados no dispositivo do portador. Como estes *logs* ao estarem armazenados fora do sistema são suscetíveis de serem eliminados ou destruídos pelo portador ou por entidades maliciosas que entrem no seu dispositivo, comprometendo a auditoria das operações e interações entre as entidades.

Forma de Mitigação De forma de acrescentar um camada de segurança extra é importante que todos os *logs* de operações e interações não sejam armazenadas nos próprios dispositivos que as fizeram, mas sim numa entidade externa. Como forma de não permitir a entidades maliciosas a acederam a informação sensível em relação às operações realizadas pelo o próprio dispositivo que estão a aceder é necessário que, para além da assinatura digital que permite a validação das operações, sempre que o dispositivo do portador tenha acesso online estes armazenem os seus *logs* na entidade emissora. Desta forma os *logs* de operações que são realizadas offline são temporariamente armazenadas no dispositivo do portador para posteriormente serem armazenados na emissora quando o portador se encontrar online. Assim, torna a auditoria das

operações mais segura e íntegra se todas estas operações tiverem armazenadas num sistema controlado e seguro como é o caso da emissora.

Análise de Risco Esta fraqueza não está prevista ser resolvido, pois não é expresso nos requisitos onde os *logs* das operações e interações devem ficar armazenados. Desta forma o risco associado a esta fraqueza é alto, já o impacto desta fraqueza é médio-baixo visto que alterações nos *logs* não impedem o funcionamento do sistema, mas tem impacto na integridade e confiança dos dados armazenados. Desta forma temos:

- RISCO: 4
- IMPACTO: 2
- CRITICIDADE: 4 + 2 = 6

3.1.3 Repudiation

Os ataques de *repudiation* dizem respeito à possibilidade de entidades maliciosos de modificarem a identidade das ações de outras entidades. Desta forma é possível que um utilizador realize uma ação e que essa ação esteja identificada como se fosse realizada por outra entidade. É por isso, necessário que o sistema implemente métodos de forma a estabelecer a *non-repudiation* das ação realizadas em todo sistema.

Fraqueza: Transferência de dados não assinados entre o portador e as entidades

Ameaça A ameaça principal associada a ataques de *repudiation* está no facto da aplicação do portador não conseguir rastrear as atividades ilegais enviadas a partir deste, tornado assim impossível para o leitor ou emissor identificar quem é que realizou o ataque. Desta forma é possível que um atacante consiga enviar dados maliciosos para o leitor ou para o emissor a partir do portador, sem que estas entidades se percebam que os dados foram enviados por um entidade maliciosa.

Forma de Mitigação Esta ameaça pode ser provocada pelo envio de dados para o leitor ou emissora que não sejam assinados pelo portador, mas pode ser mitigada através do uso de assinaturas digitais que assinem os dados a serem transferidos entre entidades. A assinatura digital utiliza um certificado digital que devem ser acedido pelas as entidades do sistema de forma a obter a chave pública da assinatura e assim confirmar a identidade da entidade que lhe está a transferir dados. Desta forma é possível que as entidades consigam rastrear as atividades ilegais e legais e as respetivas entidades que realizaram essas atividades.

Análise de Risco Esta fraqueza como não está prevista ser mitigada, pois não é expresso nos requisitos que a interação entre o dispositivo do portador e as restantes entidades deve garantir a não-repudiação dos dados transmitidos, então o risco associado a esta fraqueza é médio. Já o impacto desta fraqueza é médio visto que não ser possível identificar as atividades ilegais pode permitir ao atacante aproveitar o sistema para obtenção de informação sensível e privada assim como outros ataques mais impactantes para todo o sistema sem que este seja apanhado. Desta forma temos:

- RISCO: 3
- IMPACTO: 3
- CRITICIDADE: 3 + 3 = 6

3.1.4 Information disclosure

Os ataques de *information disclosure* existem quando os atacantes exploram formas de aceder a informação privada e sensível. É possível que tanto um documento que esteja armazenado numa base dados como mensagens enviadas entre as entidades sejam alvos de ataques que visam vazar os dados sensíveis contidos neste documentos ou mensagens.

Fraqueza: Transmissão de dados não cifrados entre o portador e as restantes entidades

Ameaça Umas das ameaças principais associadas aos ataques de *information disclosure* é o portador permitir que os dados enviados entre as entidades sejam facilmente acedidos por terceiros que tentem intercetar a comunicação entre o portador e as entidades. Esta ameaça pode provocar *data leak* dos dados enviadas entre o emissor, como por exemplo dados de autenticação e associados ao documento, e dos dados enviados para o leitor, como por exemplo os dados do documento e o *token*.

Forma de Mitigação Esta ameaça pode ser provocada pelo envio de dados para o leitor ou emissora que não sejam cifrados pelo portador, mas pode ser mitigada através do uso de cifras fortes e algoritmos de troca de chaves para permitir que o portador e a entidade de destino acordem numa chave para a decifrar a comunicação. Estas cifras podem ser o AES-256 no modo Galois Counter e para a troca de chaves podemos utilizar o algoritmo Elliptic Curve Diffie–Hellman (ECDH). Desta forma, o sistema consegue criar uma forma segura de transferir os dados sem que haja vazam da comunicação através de cifras e algoritmos de geração de chaves consideradas seguras e fortes.

Análise de Risco Esta fraqueza está prevista ser mitigada, pois é expresso nos requisitos que a interação entre o dispositivo do portador e as restantes entidades deve garantir a confidencialidade e integridade dos dados transmitidos. Desta forma o risco associado a esta fraqueza é baixo, já o impacto desta fraqueza é médio-alto visto que o *leak* de dados sensíveis nas transferências de dados associados ao documento de identificação digital ou associados à autenticação do portador na emissora podem ser nefastos para todo o sistema. Desta forma temos:

- RISCO: 1
- IMPACTO: 4
- CRITICIDADE: 1 + 4 = 5

Fraqueza: Armazenamento do documento e dos *logs* não cifrados no dispositivo do portador

Ameaça Da mesma forma que um atacante consegue intercetar informação que é transmitida entre o portador e outra entidade, é também possível que este aceda a informação sobre o documento e aos *logs* sobre as operações que se encontra armazenada no sistema, permitindo que dados privados e sensíveis sejam acedidos por entidades maliciosas.

Forma de Mitigação Esta ameaça pode ser provocada pelo o armazenamento do documento e dos *logs* não cifrados no dispositivo do portador, mas pode ser resolvida através do uso de cifras fortes que impeçam os atacantes de acederem à informação em *plaintext* sobre o documento e sobre as operações realizadas. Assim, uma entidade maliciosa que queira aceder aos dados de um portador pode conseguir aceder ao documento ou aos logs, mas como estes está cifrado o atacante não consegue saber a informação original. Contudo é possível ao portador ter acesso ao *plaintext* do documento e dos *logs* visto que este esta protegido por uma camada de autenticação, sendo que com as credencias do portador é possível aceder aos dados originais.

Análise de Risco Esta fraqueza não está prevista ser mitigada, pois não é expresso nos requisitos a necessidade de evitar o acesso de entidades não autorizadas aos dados reais contidos do dispositivo do portado. Desta forma o risco associado a esta fraqueza é médio-alto, já o impacto desta fraqueza é médio-alto visto que o *leak* de dados sensíveis associados ao documento de identificação digita de um portador e às operações realizadas por este pode trazer impacto para o sistema num todo. Desta forma temos:

- RISCO: 4
- IMPACTO: 4
- CRITICIDADE: $4 + 4 = 8$

3.1.5 DoS

Os ataques de *Denial of Service* são realizados de forma a tornar os recursos de um sistema indisponíveis para o seus utilizadores. Normalmente os alvos típicos deste tipo de ataques são servidores, pois há necessidade de estes estarem sempre disponíveis para que os utilizadores consigam aceder à informação nele contida. No caso da aplicação do portador, este tipo de ataques não seria preocupante pois uma aplicação não é geralmente alvo destes ataques visto que não há necessidade das aplicações do portador estarem sempre disponíveis para obtenção de informação. Estes ataques são geralmente direcionados para entidades como o emissor, pois há necessidade de este estar sempre disponível para atualizar documentos e comunicar com as entidades do sistema. Assim, estes ataques não são aplicados à entidade em questão.

3.1.6 Elevation of privilege

Um ataque de *elevation of privileges* ocorre quando uma entidade ganha direitos e privilégios que não devia estar disponíveis para essa entidade. Desta forma, os atacantes

conseguem ter acesso a funcionalidades, operações e ficheiros normalmente protegidos a entidades não autorizadas.

Fraqueza: O não tratamento de *inputs* enviados pelo portador

Ameaça Uma das ameças associadas a ataques de *elevation of privileges* existe quando uma entidade maliciosa consegue através do envio de *inputs* para o emissor provocar a execução de código arbitrário, acesso não autorizada à memória ou execução de *queries* SQL para conseguir aceder à base de dados ou a outras componentes da emissora e assim modificá-la. Desta forma, é possível ao portador malicioso aceder a permissões e direitos que não lhe estão autorizados.

Forma de Mitigação Esta ameaça pode ser provocada pelo não tratamento de *inputs* enviados pelo portador, mas pode ser resolvida através da verificação e validação dos dados recebidos pela emissora. Desta forma, é possível ao emissor impedir que o portador, aquando do envio da autenticação à emissora, lhe consiga enviar inputs maliciosos que permitam a esta ter acesso a operações e funcionalidades que não lhe são autorizados.

Análise de Risco Esta fraqueza não está prevista ser mitigada, pois não é expresso nos requisitos a necessidade de utilizar métodos de devam verificar os inputs do dados enviados pelo portador. Desta forma o risco associado a esta fraqueza é médio-alto, já o impacto desta fraqueza é médio-alto visto que a realização de operações a dados sensíveis pode trazer grande impacto ao funcionamento de todo o sistema. Desta forma temos:

- RISCO: 4
- IMPACTO: 4
- CRITICIDADE: 4 + 4 = 8

Fraqueza: O portador altera as permissões de acesso ao documento

Ameaça Uma das principais ameças associadas a ataques de *elevation of privileges* acontece quando uma entidade altera as permissões relacionadas com um ficheiro que foram estabelecidas pelo sistema para assim conseguir realizar operações não autorizadas. Neste caso pode ser possível ao portador, que tem acesso às permissões de administrador do dispositivo, alterar as permissões associadas ao documento de identificação para que este consiga modificar o documento.

Análise de Risco Esta fraqueza não está prevista ser mitigada, pois não é expresso nos requisitos a preocupação da alteração das permissões associadas ao documento por parte do administrador do dispositivo. Desta forma o risco associado a esta fraqueza é médio-alto, já o impacto desta fraqueza é médio-alto visto que a modificação do documento pdoe colocar em causa a integridade. mas é identificável a sua modificação através do uso de assinaturas digitais por parte do emissor. Desta forma temos:

- RISCO: 4

- IMPACTO: 4
- CRITICIDADE: 4 + 4 = 8

3.2 Aplicação do verificador

Outro elemento integrante do sistema é a aplicação do verificador que funciona em vários sistemas operativos desde que sejam compatíveis com os protocolos de comunicação utilizados. Esta aplicação comunica com o portador para verificar uma lista de atributos que fazem parte do documento de identificação digital do utilizador.

Tal como realizado para a aplicação do portador, segue-se a aplicação do STRIDE a esta entidade.

3.2.1 Spoofing

Fraqueza: A *password* do utilizador do dispositivo onde a aplicação está instalada não é cifrada antes de ser guardada.

Ameaças Caso um atacante consiga a *password* armazenada em *plain text* associada à aplicação do verificador então pode entrar na conta de um leitor e utilizá-la para obter dados de portadores. Para além disto, o sistema de *logs* iria registar que a conta que foi roubada é que realizou a operação.

Forma de mitigação Para mitigar esta fraqueza deve-se aplicar um *hash*, utilizando uma PBKDFs (*Password-based Key Derivation Functions*), à *password* sendo este o elemento que fica guardado assim como o *salt* da mesma. De seguida, para verificar a *password* do utilizador realiza-se o *hash* da *password* de *input* e compara-se com que o que foi guardado inicialmente.

Análise de risco De facto, armazenar o *hash* de uma *password* é uma prática comum na construção de sistemas que necessitem de um mecanismo de autenticação então o risco de ocorrência é baixo. Já o seu impacto é relativamente alto dado que ainda seria necessário convencer portadores a partilharem dados. Para além disso, também há a hipótese deste atacante aceder aos *logs* sobre operações que a conta roubada tenha participado.

- RISCO: 1
- IMPACTO: 3
- CRITICIDADE: 1 + 3 = 4

3.2.2 Tampering

Fraqueza: Os dados dos portadores ou *token* que é enviado pelo portador e que ficam armazenado, ainda de forma temporária , no leitor podem ser modificados por outras aplicações do sistema

Ameaças Tal como foi mencionado no caso do portador, ao ser possível que aplicações maliciosas possam modificar estes dados leva a que seja quebrado o requisito de segurança relacionado com a integridade dos dados em causa o correto funcionamento do sistema.

Forma de mitigação Os ficheiros relativos a estes documentos devem possuir as permissões corretas em que apenas permite que os dados apenas sejam lidos, ou seja, o ficheiro não possui permissão de escrita.

Análise de risco Em termos de risco e impacto este é exatamente igual ao que está presente no portador.

- RISCO: 1
- IMPACTO: 3
- CRITICIDADE: 1 + 3 = 4

Fraqueza: Não se realiza a verificação de integridade dos dados que a aplicação recebe.

Ameaças Uma das ameaças relacionada com esta fraqueza aponta para possibilidade dos dados poderem ser alterados durante a transmissão dos dados, seja por serem corrompidos, seja por serem modificados por um atacante. Desta forma os dados certificado enviado pela entidade emissora para o leitor pode ser também ele alterado na sua transmissão. Caso este último caso se verifique, o certificado que fica guardado no leitor não é o mesmo que fica guardado na entidade emissora, logo a utilização deste para assinaturas pode levar a incorreções quando comparadas com as assinaturas que se podem obter com o certificado presente na entidade emissora. Para além disto, os dados que a entidade emissora envia sobre um determinador portador podem também ser alterados.

Por fim, tanto o *token* como a lista de atributos que o verificador recebe podem ser alterados o que pode levar a que o leitor obtenha informação errada sobre o portador.

Forma de mitigação De forma a mitigar esta fraqueza pode-se utilizar um HMAC junto da mensagem enviada pelo canal de comunicação ou utilizar uma cifra autenticada (AEAD) como o AES no modo GCM. Ao receber a mensagem deve voltar a gerar-se o hash ou a tag de autenticação e comparar com aquela que foi recebida. Caso não sejam iguais então os dados foram alterados.

Análise de risco Como a verificação de integridade entre dispositivos e a entidade emissora fazem parte dos requisitos da norma então a probabilidade da fraqueza estar presente em produção é mais baixa. O impacto desta pode ser elevado pois os dados obtidos sobre o portador podem levar a que o leitor não obtenha os dados corretos do portador, tal como a data de nascimento, nome entre outros.

- RISCO: 1
- IMPACTO: 3
- CRITICIDADE: 1 + 3 = 4

Fraqueza: O *token* que contém os atributos é enviado para o verificador, numa operação *online*, sem estar assinado.

Ameaças O verificador pode pedir mais dados à entidade emissora do que aqueles que o portador que lhe enviou o *token* autorizou inicialmente, através da modificação da lista de atributos a serem pedidos à entidade emissora.

Forma de mitigação A aplicação do portador deve assinar o *token* antes deste ser enviado para o verificador. Sugere-se a utilização de `EcDSA` ou `DSA` como esquema de assinaturas. Para isto, deve confirmar a assinatura do *token* junto da entidade emissora.

Análise de risco Dado que existem requisitos de segurança relacionados com a autenticidade dos dados trocados nos canais de comunicação é pouco provável que se verifique esta fraqueza no sistema em produção. A exploração desta fraqueza pode ser bastante impactante, dado que um verificador pode obter qualquer dado do portador, logo é necessário garantir que esta não se verifica no sistema em produção.

- RISCO: 2
- IMPACTO: 4
- CRITICIDADE: 2 + 4 = 6

3.2.3 Repudiation

Fraqueza: O *log* de uma operação *offline* não é armazenada no dispositivo do verificador

Ameaças Numa operação *offline* caso o a aplicação do leitor não armazene um *log* desta operação e seja apenas a aplicação do portador a fazê-lo é possível que o portador elimine este registo do seu dispositivo. Desta forma, caso a operação em causa necessite de ser auditada no futuro não há forma de o fazer o que pode levar a que o portador negue estar envolvido nesta operação.

Forma de mitigação No modo offline o verificador, como elemento interveniente numa operação deve guardar um registo de que a operação ocorreu assim como deve armazenar outras informações que sejam relevantes para esta operação.

Análise de risco Como o requisito de segurança sobre a auditoria da aplicação do leitor não está formulado de forma detalhada é possível que esta fraqueza esteja presente na implementação final. Para além disto, dado a facilidade de explorar a fraqueza e a suas consequências em que pode não ser possível encontrar o portador envolvido numa operação maliciosa, o seu impacto pode ser significante.

- RISCO: 3
- IMPACTO: 3
- CRITICIDADE: 3 + 3 = 6

Fraqueza: O *log* de uma operação *offline* é armazenado sem ser assinado pelo portador

Ameaças Caso os *logs* das operações *offline* estejam a ser guardados na aplicação do verificador e estes não forem assinados pelo portador então o verificador pode alterar o *log* sem ser detetado, pois não há forma de comprovar que este foi alterado. Desta forma, não se pode confiar na veracidade destes *logs*, ou seja, um verificador malicioso pode, por exemplo, alterar o *log* de modo a que represente uma operação maliciosa que envolva um portador e assim atribuir a culpa a esse portador.

Forma de mitigação O portador deve utilizar o seu certificado para assinar todas os registos que são guardados no dispositivo do verificador sobre operações que sejam efetuadas no modo *offline*.

Análise de risco Tal como na fraqueza anterior, não existe uma descrição detalhada do requisito de segurança relacionado com a auditoria das operações, logo esta fraqueza pode estar presente no sistema em produção. O impacto é igual ao caso anterior no sentido em que um verificador pode alterar estes registos.

- RISCO: 3
- IMPACTO: 3
- CRITICIDADE: 3 + 3 = 6

3.2.4 Information disclosure

Fraqueza: As mensagens enviadas nos canais de comunicação que envolvem o verificador não são confidenciais (cifradas).

Ameaças Existem várias ameaças relacionadas com o facto do canal de comunicação entre o verificador e a entidade emissora e entre o verificador e o portador. Primeiramente, é possível que um atacante consiga intercetar mensagens trocadas entre as várias entidades que comunicam com o leitor e pode o obter o *token* que é enviado pelo portador numa operação online, dados obtidos pelo leitor provenientes da entidade emissora entre outro tipo de dados.

Forma de mitigação Cifrar as mensagens que são trocadas nos canais de comunicação que envolvem o verificador. para isto pode-se utilizar um esquema de cifra autenticada (`AEAD`) como o `AES` no modo `GCM` com chaves com pelo menos 2048 bits. Pode se utilizar um esquema de cifra simétrica (acordo de chave feito com `Diffie-Hellman`) ou então um de chave pública.

Análise de risco: Segundo os requisitos está prevista a implementação de um canal que garante confidencialidade dos dados nos canais de comunicação que envolvem o verificador, logo a probabilidade desta fraqueza se verificar é baixa, ainda que mesmo que esteja implementado é necessário ter em atenção no algoritmo utilizado, tamanho das chaves entre outros parâmetros. No entanto, o impacto desta ocorrer que seja porque não há qualquer implementação ou porque a implementação não segue padrões de segurança atualizados torna possível a que um atacante obtenha informação sensível acerca das pessoas que utilizam a aplicação e que está presente no documento de identidade digital, por exemplo.

- RISCO: 2
- IMPACTO: 5
- CRITICIDADE: 2 + 5 = 7

Fraqueza: O *token* de autorização pode ser reutilizado.

Ameaças O facto de um verificador poder reutilizar um *token* que lhe foi enviado por um portador durante uma operação *online* permite a que este volte a pedir os dados à entidade emissora mesmo depois da operação já ter ocorrido, ou seja, obtém dados acerca do portador, não sendo fornecida autorização por parte deste último, após a primeira operação. Isto pode levar a que o leitor tenha sempre acesso a atualizações da informação presente no documento de identificação digital.

Forma de mitigação Para eliminar esta fraqueza o *token* deverá ser implementado de forma a que seja de utilização única, ou estabelecer um *timeout* para que deixe de ser válido ao fim de algum tempo.

Análise de risco: Como não há requisitos de segurança explícitos sobre este aspeto é provável que o sistema apresente esta fraqueza após ser implementado. No entanto apesar do leitor ter acesso a informação não autorizada o impacto desta fraqueza como outras citadas anteriormente.

- RISCO: 4
- IMPACTO: 2
- CRITICIDADE: 4 + 2 = 6

3.2.5 DoS

Em relação ao aspetos das fraquezas que podem provocar ataques de *denial of service* observa-se o mesmo que no portador, logo não foram identificadas quaisquer fraquezas em relação a esta categoria.

3.2.6 Elevation of privilege

Fraqueza: O tratamento dos dados que recebe como *input* de outras entidades (portador, p.e) não é feito de forma correta.

Ameaças Caso a aplicação do verificador não fizer um tratamento correto dos inputs que recebe de outras entidades é possível que este possa sofrer um ataque onde executa código arbitrário e malicioso o que pode levar, por exemplo, ao redirecionamento de *tokens* provenientes de portadores ou de dados do documento de identificação digital que é enviado também pelo portador. Para além disso, é possível que o certificado que utiliza para assinar as mensagens seja também roubado por uma entidade externa através deste mesmo ataque.

Forma de mitigação O código desenvolvido no que toca ao *handle* de *inputs* provenientes dos canais de comunicação com outras entidades devem verificar se estes são corretos e caso não sejam devem ser descartados.

Análise de risco Existe uma grande probabilidade desta fraqueza se verificar no sentido em que facilmente pode existir um caso na verificação e validação de *inputs* que não está implementada da a grande variedade de *inputs* que pode receber. O seu impacto é grande, pois pode obter informação armazenada sobre portadores, *logs*, ou até mesmo o certificado que a aplicação utiliza para se autenticar.

- RISCO: 4
- IMPACTO: 5
- CRITICIDADE: 4 + 5 = 9

3.3 Entidade emissora

A última componente integrante do sistema trata-se do *backend* do mesmo e, implementa mecanismos capazes de emitir e averiguar a autenticidade e integridade do documento digital previamente mencionado, tanto no modo online como offline.

3.3.1 Alterações à aplicação

Um requisito que serviria como uma melhoria ao sistema seria a implementação de um Syslog na entidade emissora, que seria responsável por receber e organizar todos os logs resultantes das operações de cada entidade no sistema.

O Syslog deveria ser distribuído de modo a estar preparado para receber logs de diferentes entidades simultaneamente e, também deveria estar protegido contra Denial

of Service de modo a agir de forma contínua para aquilo que foi feito. Além disso seria vital a existência de um mecanismo de sincronização para a situação em que quando uma outra entidade realiza uma operação offline seja guardado um log local e, assim que ficar online seja realizado o envio desse log à entidade emissora, para ser armazenado no Syslog. Por fim, deveria também haver mecanismos de proteção contra o envio de falsos logs por parte um hipotético atacante.

3.3.2 Spoofing

Fraqueza: Falta de mecanismos robustos de verificação no processo de autenticação inicial

Ameaças Um portador pode fazer-se passa por outro ao tentar efetuar a autenticação inicial na entidade emissora para obtenção do certificado.

Forma de mitigação Para países que cuja existência de chaves móveis digitais se verifique, deverá ser obrigatório efetuar autenticação usando as mesmas, para que a entidade emissora tenha a certeza que quem se está a tentar autenticar seja de facto legítimo.

Análise de risco De acordo com os requisitos, a mitigação desta fraqueza não está prevista, pelo que o seu risco deverá ser médio alto. Além disso o seu impacto também seria alto na medida em que, se aplicado em grande escala, seria comprometida uma enorme quantidade de dados.

- RISCO: 4
- IMPACTO: 5
- CRITICIDADE: $4 + 5 = 9$

3.3.3 Tampering

Fraqueza: Falta de verificação da integridade dos dados recebidos na comunicação.

Ameaças O atacante poderá interceptar a comunicação entre a entidade emissora e uma outra entidade e criar um servidor que captura todo o tráfego e faz "sniffing" dos pacotes. Nesse servidor os dados serão modificados e enviados para esta segunda entidade, que por sua vez, recebe a informação incorreta.

Forma de mitigação Tal como no caso do verificador (3.2.2), poderá usar-se um HMAC junto da mensagem enviada pelo canal de comunicação ou utilizar uma cifra autenticada (AEAD) como o AES no modo GCM.

Análise de risco A mitigação desta fraqueza está prevista nos requisitos da norma, pelo que o seu risco deverá ser muito baixo. Apesar disso o impacto do mesmo na sua ocorrência é elevado dado que a informação recebida pelas restantes entidades não iria estar correta.

- RISCO: 1
- IMPACTO: 5
- CRITICIDADE: 1 + 5 = 6

Fraqueza: Validação imprópria dos inputs recebidos de entidades externas

Ameaças A receção de certos inputs na comunicação com as restantes entidades pode levar à realização operações indesejáveis no sistema, na medida em que um atacante pode modificar informações armazenadas na base de dados e, consequentemente a informação enviada para os restantes utilizadores.

Um exemplo destes inputs são *SQL injections* na qual utilizando *SQL statements* é possível modificar a informação como for desejado.

Forma de mitigação Validação de *Input* do seguinte modo:

- Aplicação e verificação dos limites de qualquer *input* exterior a entidade emissora.
- Verificação de autorização para cada pedido.
- Limitação de caracteres especiais válidos.
- Os pedidos deverão corresponder a *queries* parametrizadas ou procedimentos armazenados

Análise de risco Uma vez que esta a mitigação desta fraqueza está prevista nos requisitos da norma, dado que é *Secure by design*, o seu risco será bastante reduzido. Ainda assim, na hipotética ocorrência o seu impacto seria elevado uma vez que a informação que estaria a ser fornecida ao utilizador estaria adulterada.

- RISCO: 1
- IMPACTO: 5
- CRITICIDADE: 1 + 5 = 6

3.3.4 Repudiation

Fraqueza: Não existência de um Syslog.

Ameaças Se não houver armazenamento de *logs* num *Syslog* não será possível auditar as operações que ocorreram entre as entidades e, consequentemente no caso de um ataque não será possível identificar o seu autor.

Forma de mitigação Implementação do *Syslog* anteriormente mencionado.

Análise de risco Tendo em conta que foi mencionado que o *Syslog* deveria ser implementado, o risco será baixo uma vez que a fraqueza seria reduzida. No caso em que esta situação se verificaria, o impacto seria médio uma vez que identificar um hipotético atacante não significaria necessariamente que o sistema não acabasse comprometido, o que de facto seria algo prejudicial, mas, no entanto, não está associado a esta fraqueza.

- RISCO: 1
- IMPACTO: 3
- CRITICIDADE: 1 + 3 = 4

3.3.5 Information disclosure

Fraqueza: Inexistência de um sistema contra TCP SYN *flood attacks*.

Ameaças Um atacante poderá interceptar a comunicação entre outra entidade e a entidade emissora e obter os dados transmitidos entre ambas.

Forma de mitigação Utilizar um mecanismo capaz de mitigar TCP SYN *flood attacks*, como por exemplo, o `TCP Intercept`.

Análise de risco A probabilidade desta fraqueza se verificar é médio alta dado que requer alguns conhecimentos protocolares e não está prevista nos requisitos. Por sua vez a sua ocorrência A mitigação desta fraqueza não está prevista nos requisitos, pelo que o seu risco será alto. Por sua vez, o impacto da mesma poderia ser elevado, caso esta fosse explorada em grande escala, dado que seriam comprometidos dados pessoais.

- RISCO: 4
- IMPACTO: 5
- CRITICIDADE: 4 + 5 = 9

Fraqueza: Gestão imprópria de erros/exceções

Ameaças Um atacante poderá fornecer um *input* que dará *trigger* em erros/exceções na entidade emissora e partir dessas erros poderá ser possível obter informações do funcionamento do sistema ou até mesmo

Forma de mitigação Criar um *handler* para erros/exceções, de modo que não sejam comprometidas informações da implementação.

Análise de risco A mitigação desta fraqueza não está prevista nos requisitos, mas serão necessários *inputs* bastante especializados para a causar a sua ocorrência pelo que o seu risco será médio. Quanto ao seu o seu impacto este seria médio, dado que a exposição de dados é muito reduzida.

- RISCO: 3

- IMPACTO: 3
- CRITICIDADE: 3 + 3 = 6

Fraqueza: Validação imprópria dos *inputs* recebidos de entidades externas

Ameaças Um determinado conjunto de *inputs* a ser recebido das comunicações à entidade emissora poderá levar ao comprometimento do sistema, na medida em que os dados contidos no mesmo são expostos a entidades maliciosas ou até mesmo publicamente se as entidades maliciosas em questão o desejarem.

Um exemplo deste tipo de *inputs* são *buffer overflows* que podem fazer com que seja executado código arbitrário no sistema, código esse que servirá para obter os ditos dados.

Outro exemplo são *SQL injections*, que através de *SQL statements* poderão causar o envio, por parte da entidade emissora, da informação que desejarem.

Forma de mitigação Validação de *Input* do seguinte modo:

- Aplicação e verificação dos limites de qualquer *input* exterior a entidade emissora.
- Verificação de autorização para cada pedido.
- Limitação de caracteres especiais válidos.
- Os pedidos deverão corresponder a *queries* parametrizadas ou procedimentos armazenados

Análise de risco A mitigação desta fraqueza está prevista nos requisitos da norma, dado que é *Secure by design*, pelo que o seu risco será bastante reduzido. Ainda assim, na eventualidade da sua ocorrência o impacto seria elevado, uma vez que haveria exposição de dados privados.

- RISCO: 1
- IMPACTO: 5
- CRITICIDADE: 1 + 5 = 6

3.3.6 DoS

Fraqueza: Falta de controlo da quantidade de pedidos de dados e a sua atualização.

Ameaças Um atacante poderá simular diversos pedidos de dados de modo a congestionar a base de dados e prevenir que o sistema satisfaça as necessidades dos restantes utilizadores. Além disso, a receção desses diversos pedidos também poderá resultar numa maior consumo de recursos a nível de rede, pelo que poderia reduzir drasticamente o fluxo de comunicação com as restantes entidades, diminuindo a disponibilidade do sistema.

Forma de mitigação Limitar a quantidade de pedidos com sobre os quais o sistema está a realizar operações.

Análise de risco Esta fraqueza não está prevista nos requisitos pelo que apresenta um risco médio-alto. Quanto ao seu impacto este é alto pelo facto de impossibilitar o funcionamento do sistema.

- RISCO: 4
- IMPACTO: 5
- CRITICIDADE: 4 + 5 = 9

Fraqueza: Validação imprópria dos inputs recebidos de entidades externas

Ameaças A receção de certos *inputs* pode levar ao *crash* ou *slowdown* do sistema, uma vez que certos processos tornam-se indisponíveis para continuar a fornecer o seu serviço habitual. É de notar que estes *inputs* geralmente estão mais orientados para causar a corrupção da memória ou o aumento do consumo da mesma ou simplesmente o aumento da utilização do CPU dos ditos processos.

Um exemplo destes inputs é *buffer overflow* que corrompe a *stack* de execução, originando portanto o problema em questão, podendo comprometer a disponibilidade do sistema.

Forma de mitigação Validação de *Input* do seguinte modo:

- Aplicação e verificação dos limites de qualquer *input* exterior a entidade emissora.
- Verificação de autorização para cada pedido.
- Limitação de caracteres especiais válidos.
- Os pedidos deverão corresponder a *queries* parametrizadas ou procedimentos armazenados

Análise de risco A mitigação desta fraqueza está prevista nos requisitos da norma, dado que é *Secure by design*, pelo que o seu risco será bastante reduzido. Ainda assim, na eventualidade da sua ocorrência o impacto seria elevado, uma vez que se verificaria o comprometimento da disponibilidade do sistema, não sendo capaz de fornecer os seus serviços.

- RISCO: 1
- IMPACTO: 5
- CRITICIDADE: 1 + 5 = 6

3.3.7 Elevation of privilege

Fraqueza: Validação imprópria dos inputs recebidos de entidades externas

Ameaças A entidade emissora pode ficar suscetível a inputs provenientes de entidades exteriores na medida em que estes inputs poderão fazer com que o atacante modifique os seus privilégios ou ganhe acesso a propriedades de leitura e escrita, resultando possivelmente num controlo total sobre a base de dados. Esta elevação de privilégios poderá ser tanto vertical, quando o atacante aumenta as suas permissões ou acessa uma conta com permissões superiores, como horizontal, quando o atacante adquire acesso a outras contas do mesmo nível de permissões dele.

Exemplos destes inputs são *buffer overflows*, que corrompem a *stack* permitindo a execução de código arbitrário, e *SQL injections* que utilizam *SQL statements* que são executados na base de dados.

Forma de mitigação Validação de *Input* do seguinte modo:

- Aplicação e verificação dos limites de qualquer *input* exterior a entidade emissora.
- Verificação de autorização para cada pedido.
- Limitação de caracteres especiais válidos.
- Os pedidos deverão corresponder a *queries* parametrizadas ou procedimentos armazenados

Análise de risco Uma vez que esta a mitigação desta fraqueza está prevista nos requisitos da norma, dado que é *Secure by design*, o seu risco será bastante reduzido. Ainda assim, caso esta ameaça ocorresse o seu impacto seria alto uma vez que um atacante poderia controlar toda a informação presente na base de dados, através da elevação de privilégios.

- RISCO: 1
- IMPACTO: 5
- CRITICIDADE: 1 + 5 = 6

Catalogação de Vulnerabilidades

Nesta catalogação analisou-se as vulnerabilidades mais recentes para as diversas componentes que se encontram implementadas na entidade emissora. Como as entidades do portador e do emissor ainda não forma desenvolvidas, esta catalogação focou-se na análise e catalogação de possíveis ameaças e vulnerabilidades associadas às versões e software instalado na emissora.

4.1 CentOS

4.1.1 Denial of service

CVE-2017-5972

Esta vulnerabilidade está associada a uma falha na implementação de um mecanismo que protege ataques que utilizem SYN cookies durante conexões rápidas ao servidor. Desta forma, um atacante pode enviar uma grande quantidade de pacotes SYN o que provoca um DoS à entidade emissora [2]. A exploração desta vulnerabilidade pode levar a que não se possam realizar operações *online* de forma correta entre o portador e o verificador. Para além disto, não permite a atualização do certificado das aplicações nem o tratamento correto dos *logs* de modo a proteger a sua integridade. Por todas estas razões é uma vulnerabilidade que pode ter um grande impacto na forma de funcionamento da entidade emissora.

CVSS v3.1 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
Impact Score: 3.6
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): None
Integrity (I): None
Availability (A): High

Figure 4.1: Vetor de ataque da vulnerabilidade CVE-2017-5972

4.1.2 Denial of service, Elevation of privileges

CVE-2018-17977

A vulnerabilidade que se segue permite a que utilizadores locais possam executar ataques que levam a que o sistema não possa processar mais pedidos através da forma incorreta como o sistema lida com pacotes "IPPROTO_AH" e "IPPROTO_IP", o que leva a que os utilizados elevem as suas permissões no sistema para que possam executar aplicações que ocupem uma grande percentagem da memória do sistema [3]. Tal como no caso anterior as consequências da exploração desta vulnerabilidade podem ser bastante impactantes o que revela uma grande ameaça ao bom funcionamento do sistema.

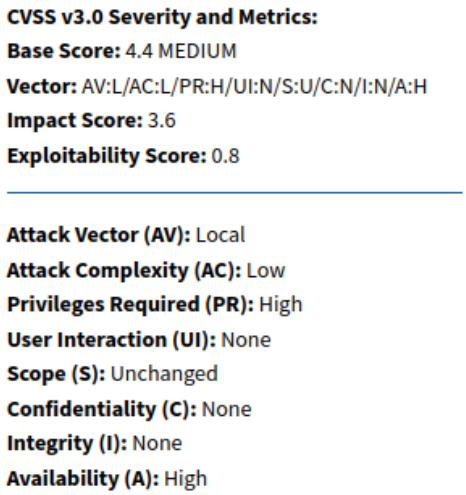

Figure 4.2: Vetor de ataque da vulnerabilidade CVE-2018-17977

4.2 Django

4.2.1 Information Disclosure

CVE-2021-28658

No Django, a classe MultiPartParser permite efetuar o "parse" de conteúdo de formulários HTML, incluindo upload de ficheiros.

Nesta vulnerabilidade a classe mencionada era suscetível a ataques de travessia de diretorias na medida em que era possível dar upload de ficheiros com nomes adequados para tal situação, nomeadamente recorrendo ao uso de "../". Ao efetuar esses ataques era possível adquirir acesso não autorizado a informações contidas nas diretorias da entidade emissora.

CVSS v3.1 Severity and Metrics:
Base Score: 5.3 MEDIUM
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Impact Score: 1.4
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): Low
Integrity (I): None
Availability (A): None

Figure 4.3: Vetor de ataque da vulnerabilidade CVE-2021-28658

CVE-2021-3281

Esta vulnerabilidade ocorria com o métodoIn Django 2.2 before 2.2.18, 3.0 before 3.0.12, and 3.1 before 3.1.6, the django.utils.archive.extract utilizando nos comandos "startapp –template" e "startproject –template", e permitia ataques de travessia de diretorias através de arquivos com o caminho completo ou o caminho relativo. À semelhança do anterior, ao efetuar esses ataques era possível adquirir acesso não autorizado a informações contidas nas diretorias da entidade emissora.

CVSS v3.1 Severity and Metrics:
Base Score: 5.3 MEDIUM
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
Impact Score: 1.4
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): None
Integrity (I): Low
Availability (A): None

Figure 4.4: Vetor de ataque da vulnerabilidade CVE-2021-3281

4.2.2 Elevation of Privileges, Information disclosure

CVE-2020-24584

Na situação em que era usado a versão do backend em questão, juntamente com *python* 3.7+, diretorias de nível intermédio da cache do sistema de ficheiros utilizavam a *user mask* pré-definida pelo sistema em vez de "0o077". Esta última máscara assegura-se que todos os ficheiros e diretorias sejam criados sem qualquer permissão para Grupo e

Outros, ou seja através do *backend* da entidade emissora poderia haver uma ameaça à confiabilidade dos dados.

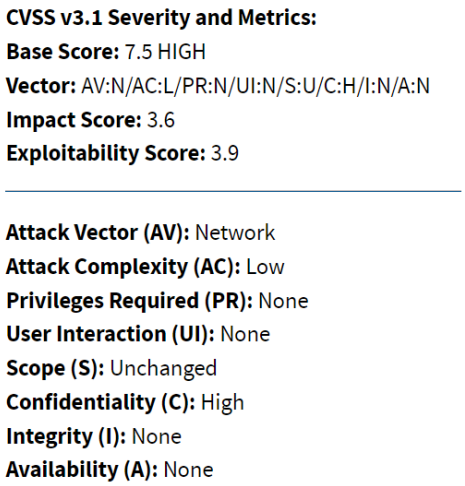

Figure 4.5: Vetor de ataque da vulnerabilidade CVE-2020-24584

4.3 UWSGI

4.3.1 Denial of service

CVE-2021-36160

Esta vulnerabilidade está ligada com o caso onde a partir de um Uri-path especificamente feito para o propósito pode ler a memória do servidor fora daquela que lhe foi alocada e fazer com que o servidor vá abaixo [12]. Desta forma, é possível causar um ataque de *denial of service* à entidade emissora tornando esta incapaz de fornecer serviço às aplicações do portador e do verificador.

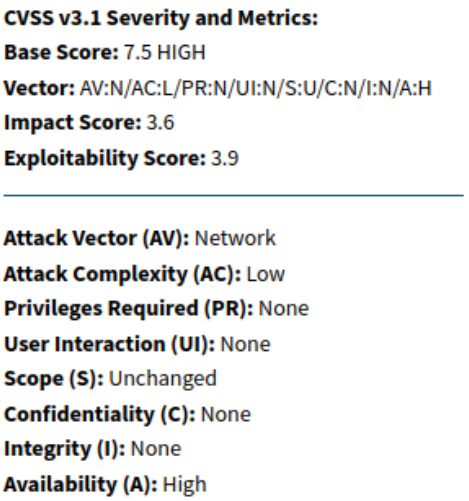

Figure 4.6: Vetor de ataque da vulnerabilidade CVE-2021-36160

4.3.2 Information disclosure

CVE-2020-11984

Esta vulnerabilidade permite a realiza de um *buffer overflow*, caso esteja a ser utilizado o mod_proxy_uwsgi, que permite obter informação presente neste servidor [13]. Ora, a exploração desta vulnerabilidade pode levar a que a informação armazenada na entidade emissora, tal como informação sobre certificados, dados de identificação de portadores ou *logs* podem ser capturados por um atacante.

CVSS v3.1 Severity and Metrics:
Base Score: 9.8 CRITICAL
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Impact Score: 5.9
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

Figure 4.7: Vetor de ataque da vulnerabilidade CVE-2020-11984

CVE-2018-7490

Esta vulnerabilidade está relacionada com a utilização da opção "–php-docroot" para realizar uma verificação do tipo "DOCUMENT_ROOT" que permite a exploração da diretoria de ficheiros do servidor [14]. Desta forma é possível que o atacante obtenha sobre informação sensível armazenada na entidade emissora, tal como acontecia com a vulnerabilidade anterior.

CVSS v3.0 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Impact Score: 3.6
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): None
Availability (A): None

Figure 4.8: Vetor de ataque da vulnerabilidade CVE-2018-7490

4.4 PostgreSQL

4.4.1 Spoofing, Tampering, Information Disclosure, Denial of Service, Elevation of Privileges

CVE-2021-32027

Nesta vulnerabilidade, um atacante autenticado na base de dados da entidade emissora podia escrever bytes arbitrários na memória do servidor, de modo que adquiria capacidades de aceder a informações nela contidas. Ademais, se o atacante desejasse, ainda era possível adicionar ou modificar informações nela contidas assim como impedir o seu funcionamento.

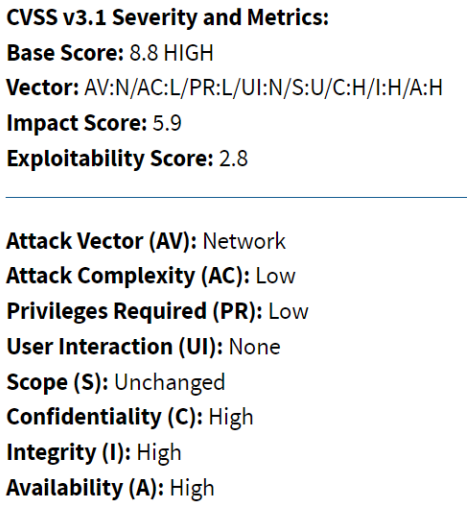

Figure 4.9: Vetor de ataque da vulnerabilidade CVE-2021-32027

4.4.2 Information Disclosure, Elevation of Privileges, Tampering

CVE-2020-25696

Esta vulnerabilidade está ligada com o terminal interativo `psql`, em que é possível, através da realização de *queries* a um servidor comprometido é possível executar código arbitrário nesse servidor [11]. A partir daí é possível realizar ataques que impeçam o funcionamento do sistema, comprometendo o fornecimento de serviço às aplicações do portador e do leitor. Para além disso, é possível também eliminar ou alterar dados que estejam presentes na base de dados da entidade emissora.

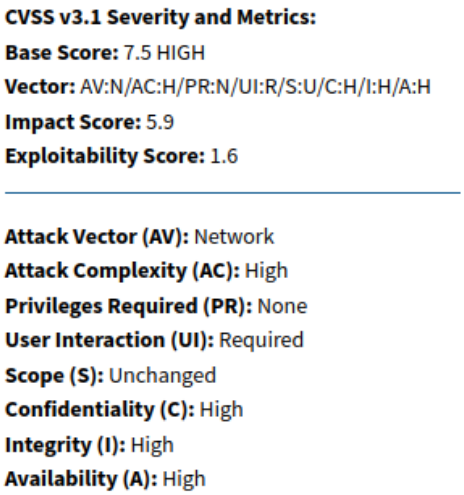

Figure 4.10: Vetor de ataque da vulnerabilidade CVE-2020-25696

4.4.3 Information Disclosure, Elevation of Privileges

CVE-2021-3393

Esta vulnerabilidade está relacionada com um utilizador da base de dados que possua permissão para realizar *updates* mas não permissões para realizar *selects* relacionadas com uma coluna em específicos poderia, eventualmente, criar *queries* que permitiriam obter valores dessa coluna através de mensagens de erro [10].

Ora, esta vulnerabilidade pode ser explorada num caso de elevação de privilégios de um atacante onde poderia executar código arbitrário com as permissões necessárias de modo a obter dados relativos a utilizadores do sistema que estejam ligados aos documentos de identificação digital.

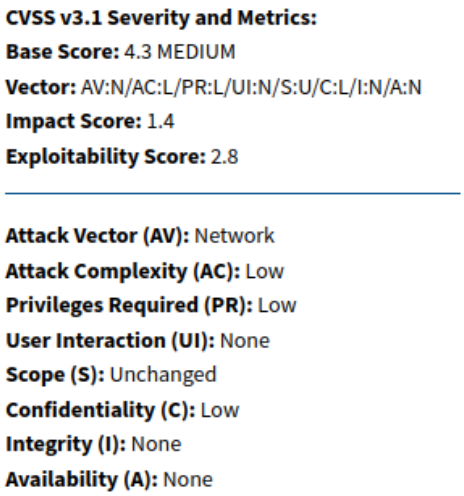

Figure 4.11: Vetor de ataque da vulnerabilidade CVE-2021-3393

4.5 Ubuntu

4.5.1 Denial of Service

CVE-2020-27351

No serviço "aptd daemon", responsável pela instalação e atualização de packages, foram encontrados vários *leaks*, de memória e de *file descriptor*, através dos quais era possível provocar *buffer overflow* e consequentemente o *crash* do serviço.

. Esta vulnerabilidade podia ser explorada por um atacante para consumir mais recursos ao ponto de causar DoS desse serviço, o que apesar de inconveniente teria um risco reduzido graças às suas carateristicas, nomeadamente o facto de que o atacante necessitaria de ser local, além de que também não impossibilitaria o sistema operativo do serviço de gestão do sistema da entidade emissora continuar com as suas restantes atividades.

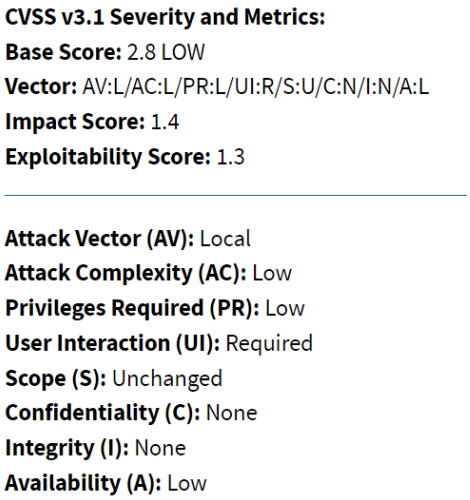

Figure 4.12: Vetor de ataque da vulnerabilidade CVE-2020-27351

4.5.2 Spoofing, Tampering, Information Disclosure, Denial of Service, Elevation of Privileges

CVE-2020-15708

A ferramenta de gerenciamento de tecnologias de virtualização da versão em questão do Ubuntu, `libvirt`, criava *sockets* de controlo com permissões de leitura e escrita, ou seja os programas eram corridos com permissões de administrador. Tendo isso em conta um atacante poderia usar isso a seu favor e reescrever ficheiros arbitrários ou executar código arbitrário com permissões de administrador, o que sem dúvida seria grave nos contextos da aplicação emissora, que representa um ponto de confidencialidade, integridade e disponibilidade.

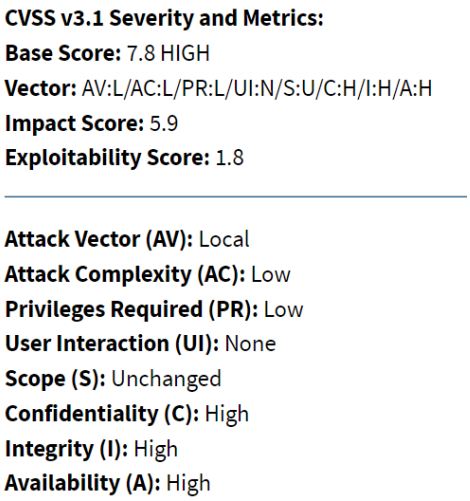

Figure 4.13: Vetor de ataque da vulnerabilidade CVE-2020-15708

4.6 Flask

O Flask é uma *framework* de web escrita em Python que fornece ferramentas úteis e funcionalidades para criar aplicações em Python facilmente. Na entidade emissora, a versão do Flask utilizada é o 1.0 e vai servidor como *backend* de gestão.

4.6.1 Elevation of Privileges, Information Disclosure e Tampering

CVE-2021-3306

A extensão Flask-Cashing utiliza uma serialização Pickle que pode levar a que um atacante consiga executar código arbitrário ou elevar os seus privilégios. Se um atacante ganhar acesso ao armazenamento de cache consegue contaminar este armazenamento com informação maliciosa ou até executar código python. Esta vulnerabilidade parece estar resolvida nas versões após 1.10.1 [1]

Esta vulnerabilidade pode permitir a um atacante aceder ao armazenamento cache da entidade emissora e obter e alterar dados relativos a utilizadores do sistema que estejam associados aos documentos de identificação digital através da execução de código arbitrário e elevação de privilégios.

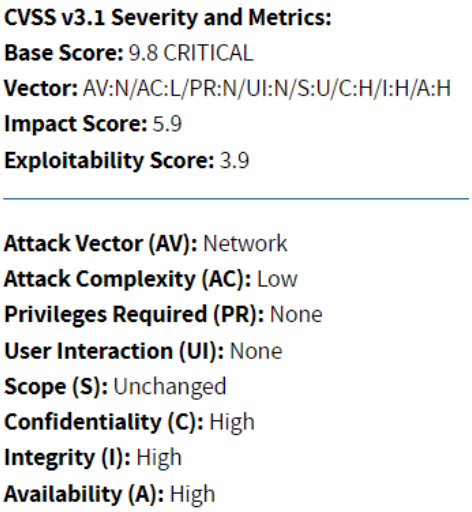

Figure 4.14: Vetor de ataque da vulnerabilidade CVE-2021-3306

4.7 Gunicorn

O Gunicorn é uma interface de um servidor Gateway HTTP em Python amplamente compatível com várias *frameworks web*, implementado de forma simples, leve nos recursos utilizados e bastante rápido. Na entidade emissora, é utilizado como um servidor web que pode apresentar diversas vulnerabilidades dependendo a versão a utilizar do Gunicorn.

4.7.1 Tampering

CVE-2018-1000164

Esta vulnerabilidade afeta a versão 19.4.5 e diz respeito à neutralização imprópria de sequências CRLF no cabeçalho de um pedido HTTP que pode resultar no servidor enviar um cabeçalho HTTP arbitrário. Isto é, é possível que um atacante consiga fazer com que o servidor enviei um HTTP response com um cabeçalho adulterado. Contudo esta vulnerabilidade está mitigada na versão 19.5.0 do Gunicorn. [4]

Esta vulnerabilidade pode permitir a um atacante fazer com o servidor enviar um cabeçalho HTTP alterado que faz com que este consiga enviar ao destino (portador ou leitor) informação adicional maliciosa que seja enviada nesse cabeçalho comprometendo o funcionamento e a integridade do sistema. Esta informação pode afetar as componestes do sistema e até provocar ameaças e problemas no portador ou leitor.

CVSS v3.0 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N
Impact Score: 3.6
Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): None
Integrity (I): High
Availability (A): None

Figure 4.15: Vetor de ataque da vulnerabilidade CVE-2018-1000164

4.7.2 Information Disclosure

CVE-2018-12564

Antes da versão 2018.5.post1. do Linaro LAVA, é possível que um utilizador maliciosos force o *lava-server-gunicorn* em fazer um download um ficheiro do sistema de ficheiros, se este for legível ao *lava-server-gunicorn* e for um yaml válido.[5]

Esta vulnerabilidade pode permitir a um atacante aceder a ficheiros de *logs*, operações e comunicações realizadas ou dados dos utilizadores que utilizam o sistema que sejam privados, sensíveis ao público.

CVSS v3.0 Severity and Metrics:
Base Score: 6.5 MEDIUM
Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
Impact Score: 3.6
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): Low
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): None
Availability (A): None

Figure 4.16: Vetor de ataque da vulnerabilidade CVE-2018-12564

CVE-2018-12563

Antes da versão 2018.5.post1. do Linaro LAVA, é possível que um utilizador maliciosos force um pedido HTTP que obrigar o lava-server-gunicorn a enviar um ficheiro qualquer que esteja armazenado no servidor, se este for legível ao lava-server-gunicorn e for um yaml válido.[6]

Tal como acontece com a vulnerabilidade anterior, esta também pode permitir que um atacante aceda a ficheiros de *logs*, operações e comunicações realizadas ou dados dos utilizadores que utilizam o sistema que sejam privados, sensíveis ao público.

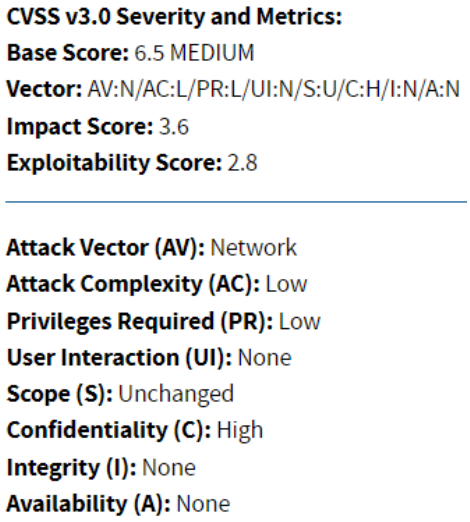

Figure 4.17: Vetor de ataque da vulnerabilidade CVE-2018-12563

4.8 DOCKER

O docker é um conjunto de produtos que funcionam como *Platform-as-a-service* que usam virtualização de nível operacional para entregar software em pacotes designados de *containers*. Os *containers* são isolados uns dos outros, mas é possível a comunicação entre eles sendo que cada *container* possui o seu próprio software, bibliotecas e ficheiros de configuração. Como estão ligados ao mesmo sistema operativo usam menos recursos que as máquinas virtuais. O *docker* 19.03.6 está a ser utilizado para armazenar a base de dados PostgreSQL 12.1 que vai ser utilizada como base de dados de gestão.

4.8.1 Tampering

CVE-2020-27534

Um ficheiro utilbinfmt_miscCheck.go que se encontra no Builder do Docker Engine chama o os.OpenFile com um *pathname* construído, temporariamente, com o primeiro argumento vazio através do ioutil.TempDir. Este nome do caminho torna inseguro as validações *qemu-check* do os.OpenFile. [7] Como as validações deixam de estar seguras é possível a modificação de parâmetros enviados para os.OpenFile, causando ameaças de *tampering*. Esta vulnerabilidade pode ser encontrada antes da versão 19.03.9 do Docker.

Com a mudança de parâmetros no OpenFile é possível que um atacante consiga alterar as permissões de acesso a ficheiros e com isso alterar ficheiros de configuração do docker que podem influenciar no *container* de a base de dados de gestão e consequentemente em todo o acesso à BD.

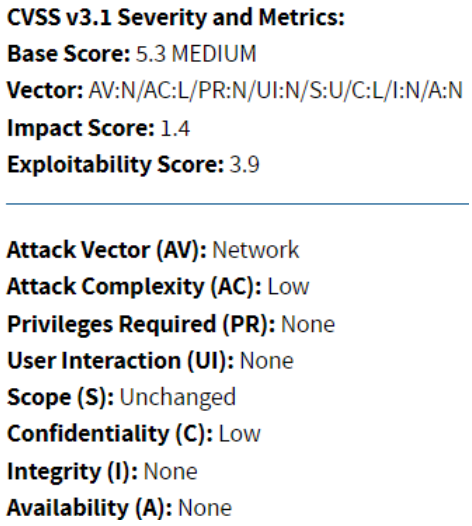

Figure 4.18: Vetor de ataque da vulnerabilidade CVE-2020-27534

4.8.2 Elevation of Privileges e Tampering

CVE-2021-21284

Nas versões do docker antes das versões 9.03.15 e 20.10.3 há uma vulnerabilidade que envolve a opção –userns-remap onde o acesso remapeado à root permite a um atacante obter os privilégios da root real. Quando é usada a opção –userns-remap, se o utilizador root no *namespace* remapeado tiver acesso ao sistema de ficheiro do host então um atacante consegue modificar os ficheiro "/var/lib/docker/¡remapping¿" criando ficheiro de escrita com privilégios estendidos. As versões 20.10.3 e 19.03.15 contêm *patches* que previnem atacantes de elevarem os seus privilégios.[8]

Com a elevação dos privilégios a root no docker é possível a um atacante tomar controlo sobre ficheiros de configuração que são criados para configurar *containers* e as imagens que neles correm, permitindo a entidades maliciosas escreverem novos ficheiros ou até altera-los de forma impedir o bom funcionamento do *container* que corre a a base de dados de gestão.

CVSS v3.1 Severity and Metrics:
Base Score: 6.8 MEDIUM
Vector: AV:A/AC:L/PR:L/UI:N/S:C/C:N/I:H/A:N
Impact Score: 4.0
Exploitability Score: 2.3

Attack Vector (AV): Adjacent
Attack Complexity (AC): Low
Privileges Required (PR): Low
User Interaction (UI): None
Scope (S): Changed
Confidentiality (C): None
Integrity (I): High
Availability (A): None

Figure 4.19: Vetor de ataque da vulnerabilidade CVE-2021-21284

4.8.3 Denial of Services

CVE-2021-21285

Nas versões do docker antes das versões 9.03.15 e 20.10.3 há uma vulnerabilidade onde o *pull* de uma imagem docker que esteja intencionalmente malformada pode ficar tornar o docker daemon indisponível, fazendo com que este deixe de conseguir funcionar corretamente. As versões 20.10.3 e 19.03.15 contém *patches* que previnem a indisponível do docker daemon.[9]

Caso um atacante explore as outras falhas apresentadas e consiga ter controlo sobre docker a partir da alteração e/ou criação de ficheiros de configuração do docker, pode ser possível que crie uma imagem intencionalmente malformada e crie um *container* com essa imagem que depois vai provocar a indisponibilidade do docker daemon e assim tornar os seus *containers* indisponível, ou seja, a tornar a base de dados de gestão indisponível.

CVSS v3.1 Severity and Metrics:
Base Score: 6.5 MEDIUM
Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H
Impact Score: 3.6
Exploitability Score: 2.8

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): None
Integrity (I): None
Availability (A): High

Figure 4.20: Vetor de ataque da vulnerabilidade CVE-2021-21285

, para além de toda a catalogação de vulnerabilidades presente no *software* que será utilizado na entidade emissora

Pontos críticos do sistema

Após toda a análise das fraquezas presentes nas várias entidades do sistema, das ameaças que podem surgir através destas e da verificação da criticidade que estas apresentam, é possível estabelecer quais as fraquezas que devem ser tratadas com mais urgência e quais os pontos mais críticos deste sistema.

De facto, deve-se dar mais atenção às fraquezas que apresentam um valor de criticidade maior de acordo com a análise feita anteriormente, sendo que, existem várias que não estariam englobadas nos requisitos de segurança presentes no documento de descrição do sistema. Desta forma, deve-se dar uma maior atenção sobre a forma como se tratam os *inputs* que são recebidos pelas várias entidades, esquemas de cifra utilizados para garantir os aspetos de autenticidade, integridade e confidencialidade dos dados trocados nos canais de comunicação, para além de garantir que se utilizam algoritmos e parâmetros (p.e, tamanho das chaves) seguros de forma a não comprometer os dados. Deve também existir uma atenção especial à forma como os *logs* devem ser armazenados, de acordo com algumas das formas de mitigação enunciadas anteriormente, tanto em operações *offline* como *online*, caso contrário pode não ser possível auditar várias operações que aconteceram entre entidades.

Além disto, os pontos mais críticos do sistema, são, de facto, os certificados utilizados para assinar mensagens de forma a garantir a autenticidade da informação, que vão desde os *logs* armazenados nos dispositivos, *tokens* de autorização e listas de atributos. Por conseguinte, os dados referentes ao documento de identificação digital que são armazenados no portador para enviar para o leitor no caso de uma operação *offline* devem também ser alvos de uma atenção redobrada para evitar que sejam capturados por atacantes o que pode levar a uma fuga de informação crítica do sistema. Por fim, a forma como se tratam os *logs* também deve ser implementada para que não exista uma entidade que possa repudiar alguma operação, tal como foi descrito na análise feita anteriormente.

Conclusão

A metodologia seguida permitiu identificar vários tipos de fraquezas presentes nas várias entidades do sistema assim como possíveis ameaças as quais estas podem ser alvo.

Para além disto, a análise de risco que foi feita nessa mesma secção permitiu identificar quais as fraquezas mais críticas, ou seja, aquelas com maior probabilidade de aparecer no sistema em produção e/ou as mais impactantes para que a equipa de desenvolvimento se possa focar mais na resolução das mesmas.

Por conseguinte, a catalogação das vulnerabilidades presentes no sistema da entidade emissora permite revelar à equipa que tipo de problemas é que esta pode estar sujeita. Ainda sobre a entidade emissora, foi possível estabelecer alterações relacionadas essencialmente à forma como esta deveria tratar dos *logs* do sistema de forma a que exista sempre a possibilidade de auditar as operações feitas no sistema.

Em suma, a metodologia aplicada permitiu identificar os ativos mais críticos do sistema, várias fraquezas e vulnerabilidades que a equipa se pode vir a deparar quando o sistema estiver implementado e ainda algumas formas de mitigar algumas das fraquezas encontradas.

Bibliography

[1] "CVE-2021-3306" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2021-33026 [Acedido em abril de 2022].

[2] "CVE-2017-5972" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2017-5972 [Acedido em abril de 2022].

[3] "CVE-2018-17977" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2018-17977 [Acedido em abril de 2022].

[4] "CVE-2018-1000164" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2018-1000164 [Acedido em abril de 2022].

[5] "CVE-2018-12563" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2018-12563 [Acedido em abril de 2022].

[6] "CVE-2018-12564" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2018-12564 [Acedido em abril de 2022].

[7] "CVE-2020-27534" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2020-27534 [Acedido em abril de 2022].

[8] "CVE-2021-21284 " [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2021-21284 [Acedido em abril de 2022].

[9] "CVE-2021-21285" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2021-21285 [Acedido em abril de 2022].

[10] "CVE-2021-3393" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2021-3393 [Acedido em abril de 2022].

[11] "CVE-2020-25696" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2020-25696 [Acedido em abril de 2022].

[12] "CVE-2021-36160" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2021-36160 [Acedido em abril de 2022].

[13] "CVE-2020-11984" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2020-11984 [Acedido em abril de 2022].

[14] "CVE-2018-7490" [online]. Disponível em: https://nvd.nist.gov/vuln/detail/CVE-2018-7490 [Acedido em abril de 2022].