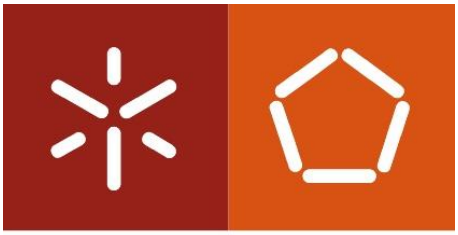


Universidade do Minho

Departamento de Informática

Mestrado em Engenharia Informática

Criptografia e Segurança de Informação

Tecnologias de Segurança
Trabalho Prático 2

Grupo Nº11

Ariana Lousada (PG47034) Luís Carneiro (PG46541)
Rui Cardoso (PG42849)

4 de maio de 2022

Conteúdo

Resumo

Este relatório contém a resolução do segundo trabalho prático inserido na unidade curricular Tecnologias de Segurança, pertencente ao perfil de Criptografia e Segurança da Informação inserido no Mestrado em Engenharia Informática da Universidade do Minho.

O presente documento descreve a análise relativa às ameaças, vulnerabilidades e *exploits* do serviço em desenvolvimento *mID*, assim como potenciais soluções para os riscos de impacto mais significativo para o projeto em questão.

Capítulo 1

Introdução

1.1 Contextualização

O alvo de análise deste projeto trata-se de um serviço de identificação pessoal digital e móvel - *mID*. Este serviço funciona com base em quatro principais componentes que comunicam e transferem dados entre si:

- Portador: Entidade externa que interage com a aplicação móvel que armazena dados de documentos de identificação e elementos utilizados pela aplicação do leitor para verificação de integridade e autenticidade.
- Verificador: Entidade externa que interage com a aplicação móvel na qual um verificador estabelece conexões com um portador e solicita atributos para o identificar.
- Smartphones iOS/Android: Dispositivos que contêm aplicações mID(do portador) ou aplicações leitoras(do verificador).
- Entidade emissora: Entidade que emite e confere autenticidade a um documento de identificação pessoal. Também é responsável pela disponibilização de mecanismos que garantam a autenticidade e integridade dos documentos digitais transmitidos.

Cada um destes componentes utiliza diferentes protocolos de estabelecimento de conexão, assim como diversos serviços que irão também ser alvo de análise ao longo do documento.

1.2 Objetivo

O principal objetivo deste projeto consiste numa análise detalhada de cada componente pertencente ao sistema, assim como a avaliação das ferramentas e serviços por ele utilizados.

Esta análise é feita de modo a que seja possível detetar vários tipos de ameaças, *exploits* ou vulnerabilidades que possam ser inseridas sem intenção no projeto durante a sua construção. Esta deteção irá permitir uma correção destas vulnerabilidades numa fase mais recente no projeto, de modo a garantir os requisitos de integridade e confiabilidade do serviço.

Após análise de vulnerabilidades e modelação de ameaças, serão apresentadas algumas sugestões de modificação do projeto de modo a melhorar a segurança do serviço.

1.3 Estrutura do relatório

Este documento encontra-se dividido em várias secções:

- Catalogação de vulnerabilidades e *exploits* (2): Secção na qual são abordadas as várias vulnerabilidades dos vários protocolos e serviços utilizados por cada componente do sistema.
- Catalogação de fraquezas típicas (3): Secção na qual são abordadas vulnerabilidades típicas encontradas neste tipo de sistema.
- Modelação de sistema (4): Apresentação de um diagrama representativo do sistema.

- Modelação de ameaças (5): Apresentação de vários modelos de ameaças referentes a cada componente do sistema.
- Análise de risco (6): Secção na qual é analisada em geral a gravidade dos riscos de momento presentes no sistema.
- Propostas de Soluções para Aumento de Segurança (7): Proposta de várias estratégias de melhoria do serviço e das suas componentes.
- Referências (8): Referências utilizadas ao longo do desenvolvimento do trabalho prático.

Capítulo 2

Catalogação de vulnerabilidades e *exploits*

Para iniciar a análise de vulnerabilidades e *exploits* do sistema, foram analisados os protocolos e serviços utilizados por cada entidade consultando as suas fraquezas publicadas no CVE.

2.1 Aplicação *mID* (aplicação do portador) e Aplicação leitora (aplicação do verificador)

As aplicações móveis do portador e do leitor foram desenvolvidas para sistemas operativos Android e IOS. Para ver possíveis vulnerabilidades neste aspeto, consultou-se a lista das 10 vulnerabilidades de segurança mais recentes de cada um destes sistemas operativos.

Google » Android : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2022-25822	416			2022-03-10	2022-03-16	4.9	None	Local	Low	Not required	None	None	Complete
An use after free vulnerability in sdp driver prior to SMR Mar-2022 Release 1 allows kernel crash.														
2	CVE-2022-25820	307			2022-03-10	2022-03-16	2.1	None	Local	Low	Not required	Partial	None	None
A vulnerable design in fingerprint matching algorithm prior to SMR Mar-2022 Release 1 allows physical attackers to perform brute force attack on screen lock password.														
3	CVE-2022-25818	119		Exec Code Overflow	2022-03-10	2022-03-16	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Improper boundary check in UWB stack prior to SMR Mar-2022 Release 1 allows arbitrary code execution.														
4	CVE-2022-25817	287			2022-03-10	2022-03-16	2.1	None	Local	Low	Not required	None	Partial	None
Improper authentication in One UI Home prior to SMR Mar-2022 Release 1 allows attacker to generate pinned-shortcut without user consent.														
5	CVE-2022-25816	287			2022-03-10	2022-03-16	2.1	None	Local	Low	Not required	None	Partial	None
Improper authentication in Samsung Lock and mask apps setting prior to SMR Mar-2022 Release 1 allows attacker to change enable/disable without authentication														
6	CVE-2022-25815				2022-03-10	2022-03-16	4.6	None	Local	Low	Not required	Partial	Partial	Partial
PendingIntent hijacking vulnerability in Weather application prior to SMR Mar-2022 Release 1 allows local attackers to perform unauthorized action without permission via hijacking the PendingIntent.														
7	CVE-2022-25814				2022-03-10	2022-03-16	4.6	None	Local	Low	Not required	Partial	Partial	Partial
PendingIntent hijacking vulnerability in Wearable Manager Installer prior to SMR Mar-2022 Release 1 allows local attackers to perform unauthorized action without permission via hijacking the PendingIntent.														
8	CVE-2022-24932	425			2022-03-10	2022-03-17	2.1	None	Local	Low	Not required	None	Partial	None
Improper Protection of Alternate Path vulnerability in Setup wizard process prior to SMR Mar-2022 Release 1 allows physical attacker package installation before finishing Setup wizard.														
9	CVE-2022-24931	863			2022-03-10	2022-03-17	4.6	None	Local	Low	Not required	Partial	Partial	Partial
Improper access control vulnerability in dynamic receiver in ApkInstaller prior to SMR MAR-2022 Release allows unauthorized attackers to execute arbitrary activity without a proper permission														
10	CVE-2022-24929				2022-03-10	2022-03-16	2.1	None	Local	Low	Not required	None	Partial	None
Unprotected Activity in AppLock prior to SMR Mar-2022 Release 1 allows attacker to change the list of locked app without authentication.														

Figura 2.1: Lista das 10 vulnerabilidades mais recentes do sistema operativo Android.

Apple » Iphone Os : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2022-22671				2022-03-18	2022-03-24	2.1	None	Local	Low	Not required	Partial	None	None
	An authentication issue was addressed with improved state management. This issue is fixed in iOS 15.4 and iPadOS 15.4. A person with physical access to an iOS device may be able to access photos from the lock screen.													
2	CVE-2022-22670				2022-03-18	2022-03-24	4.3	None	Remote	Medium	Not required	None	Partial	None
	An access issue was addressed with improved access restrictions. This issue is fixed in tvOS 15.4, iOS 15.4 and iPadOS 15.4, watchOS 8.5. A malicious application may be able to identify what other applications a user has installed.													
3	CVE-2022-22667	416		Exec Code	2022-03-18	2022-03-24	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
	A use after free issue was addressed with improved memory management. This issue is fixed in iOS 15.4 and iPadOS 15.4. An application may be able to execute arbitrary code with kernel privileges.													
4	CVE-2022-22666	787		Mem. Corr.	2022-03-18	2022-03-24	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
	A memory corruption issue was addressed with improved validation. This issue is fixed in tvOS 15.4, iOS 15.4 and iPadOS 15.4, watchOS 8.5. Processing a maliciously crafted image may lead to heap corruption.													
5	CVE-2022-22659			+Info	2022-03-18	2022-03-24	4.0	None	Remote	Low	???	Partial	None	None
	A logic issue was addressed with improved state management. This issue is fixed in iOS 15.4 and iPadOS 15.4. An attacker in a privileged network position may be able to leak sensitive user information.													
6	CVE-2022-22653	20			2022-03-18	2022-03-24	5.0	None	Remote	Low	Not required	Partial	None	None
	A logic issue was addressed with improved restrictions. This issue is fixed in iOS 15.4 and iPadOS 15.4. A malicious website may be able to access information about the user and their devices.													
7	CVE-2022-22652	668			2022-03-18	2022-03-26	3.6	None	Local	Low	Not required	Partial	Partial	None
	The GSMA authentication panel could be presented on the lock screen. The issue was resolved by requiring device unlock to interact with the GSMA authentication panel. This issue is fixed in iOS 15.4 and iPadOS 15.4. A person with physical access may be able to view and modify the carrier account information and settings from the lock screen.													
8	CVE-2022-22643				2022-03-18	2022-03-24	5.0	None	Remote	Low	Not required	None	Partial	None
	This issue was addressed with improved checks. This issue is fixed in iOS 15.4 and iPadOS 15.4, macOS Monterey 12.3. A user may send audio and video in a FaceTime call without knowing that they have done so.													
9	CVE-2022-22642			Bypass	2022-03-18	2022-03-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	This issue was addressed with improved checks. This issue is fixed in iOS 15.4 and iPadOS 15.4. A user may be able to bypass the Emergency SOS passcode prompt.													
10	CVE-2022-22641	416		+Priv	2022-03-18	2022-03-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
	A use after free issue was addressed with improved memory management. This issue is fixed in tvOS 15.4, iOS 15.4 and iPadOS 15.4, macOS Monterey 12.3. An application may be able to gain elevated privileges.													

Figura 2.2: Lista das 10 vulnerabilidades mais recentes do sistema operativo iOS.

Como é possível observar através das figuras anteriores talvez será necessário implementar métodos e estratégias de segurança extras para a versão da aplicação do portador para o sistema operativo iOS, tendo em conta os dados das suas vulnerabilidades mais recentes.

Para além disto, as aplicações estabelecem conexões via TCP/IP. Após uma pesquisa por parte da equipa de trabalho para encontrar vulnerabilidades graves recentes relativas à utilização deste protocolo nos sistemas operativos usados, não foram encontradas nenhumas relevantes.

Esta aplicação também pode utilizar uma das seguintes tecnologias:

- BLE(*Bluetooth Low Energy*)
  Após uma pesquisa relativa às vulnerabilidades de segurança desta tecnologia as mais revelantes encontradas foram:
  - `CVE-2019-2102`: Na especificação da tecnologia, caso uma LTK *hardcoded* seja utilizada, pode possibilitar a injeção de *keystrokes* em sistema Android devido à utilização de estratégias criptográficas impróprias. Esta vulnerabilidade foi detetada em junho de 2019 e tem um score CVSS 8.3.
  - `CVE-2019-17517`: Esta vulnerabilidade pode permitir que atacantes próximos do dispositivo causem um *buffer overflow* através de um pacote *Link Layer* criado. Esta vulnerabilidade foi detetada em fevereiro de 2020 e tem um score CVSS 6.1.
  - `CVE-2020-15531`: Esta vulnerabilidade pode permitir execução remota de código. Esta vulnerabilidade foi detetada em agosto de 2020 e tem um score CVSS 5.8.
- NFC(Near Field Communication)
  Após uma pesquisa relativa às vulnerabilidades de segurança desta tecnologia a mais revelante encontrada foi:
  - `CVE-2017-17225`: Vulnerabilidade relativa ao sistema operativo Android de *buffer overflow*. Um atacante pode utilizar um leitor de cartões NFC para inserir dados maliciosos num dispositivo móvel. Um *exploit* com sucesso desta vulnerabilidade pode levar a um *restart* do sistema ou execução de código remota. Esta vulnerabilidade foi detetada em março de 2018 e tem um score CVSS 8.3.
- Wifi-Aware
  A equipa de trabalho não foi capaz de encontrar nenhuma vulnerabilidade publicada desta tecnologia.

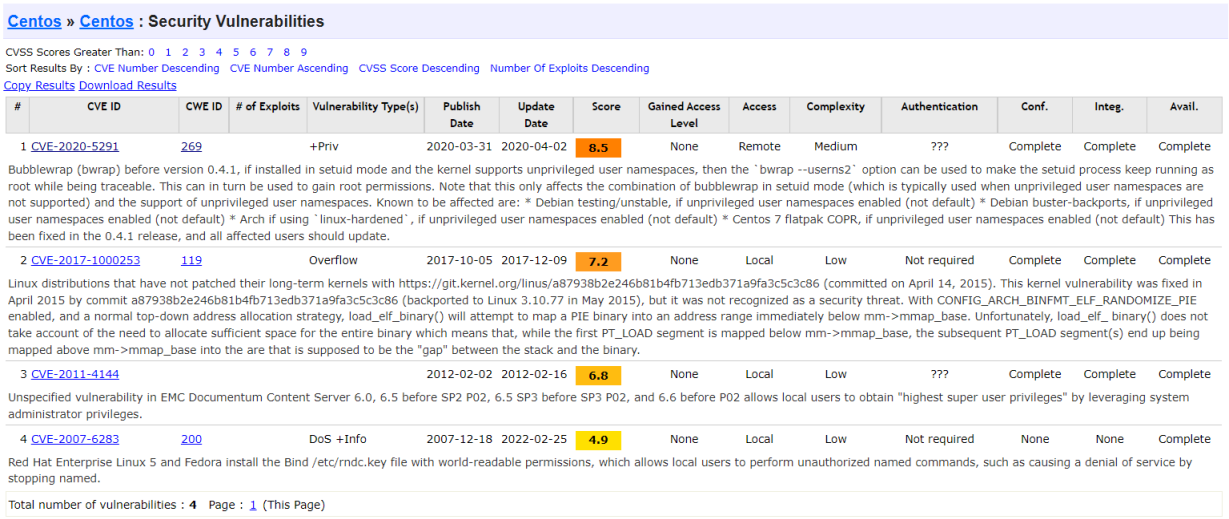
Em suma, o NFC e o Wifi-Aware serão as tecnologias mais seguras para usar no serviço em desenvolvimento, tendo em especial atenção a implementação da versão para dispositivos iOS, que são mais propensos a ameaças de segurança(de acordo com dados mais recentes).

2.2 Entidade emissora (backend do sistema)

Uma vez que esta entidade é a principal responsável pelos mecanismos que garantem a autenticidade e integridade dos documentos digitais, esta é a mais importante do sistema.

Ao longo desta secção vão ser analisados em termos de vulnerabilidades conhecidas todos os serviços que compõem a infra-estrutura desta entidade:

- CentOS 7.8.2003
  Este sistema operativo está a ser utilizado de momento no servidor web. Após a consulta da lista de vulnerabilidades no CVE do sistema operativo CentOS, verificou-se que não tem recentemente vulnerabilidades graves de segurança.



Centos » Centos : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2020-5291	269		+Priv	2020-03-31	2020-04-02	8.5	None	Remote	Medium	???	Complete	Complete	Complete
	Bubblewrap (bwrap) before version 0.4.1, if installed in setuid mode and the kernel supports unprivileged user namespaces, then the `bwrap --userns2` option can be used to make the setuid process keep running as root while being traceable. This can in turn be used to gain root permissions. Note that this only affects the combination of bubblewrap in setuid mode (which is typically used when unprivileged user namespaces are not supported) and the support of unprivileged user namespaces. Known to be affected are: * Debian testing/unstable, if unprivileged user namespaces enabled (not default) * Debian buster-backports, if unprivileged user namespaces enabled (not default) * Arch if using `linux-hardened`, if unprivileged user namespaces enabled (not default) * Centos 7 flatpak COPR, if unprivileged user namespaces enabled (not default) This has been fixed in the 0.4.1 release, and all affected users should update.													
2	CVE-2017-1000253	119		Overflow	2017-10-05	2017-12-09	7.2	None	Local	Low	Not required	Complete	Complete	Complete
	Linux distributions that have not patched their long-term kernels with https://git.kernel.org/linus/a87938b2e246b81b4fb713edb371a9fa3c5c3c86 (committed on April 14, 2015). This kernel vulnerability was fixed in April 2015 by commit a87938b2e246b81b4fb713edb371a9fa3c5c3c86 (backported to Linux 3.10.77 in May 2015), but it was not recognized as a security threat. With CONFIG_ARCH_BINFMT_ELF_RANDOMIZE_PIE enabled, and a normal top-down address allocation strategy, load_elf_binary() will attempt to map a PIE binary into an address range immediately below mm->mmap_base. Unfortunately, load_elf_ binary() does not take account of the need to allocate sufficient space for the entire binary which means that, while the first PT_LOAD segment is mapped below mm->mmap_base, the subsequent PT_LOAD segment(s) end up being mapped above mm->mmap_base into the are that is supposed to be the "gap" between the stack and the binary.													
3	CVE-2011-4144				2012-02-02	2012-02-16	6.8	None	Local	Low	???	Complete	Complete	Complete
	Unspecified vulnerability in EMC Documentum Content Server 6.0, 6.5 before SP2 P02, 6.5 SP3 before SP3 P02, and 6.6 before P02 allows local users to obtain "highest super user privileges" by leveraging system administrator privileges.													
4	CVE-2007-6283	200		DoS +Info	2007-12-18	2022-02-25	4.9	None	Local	Low	Not required	None	None	Complete
	Red Hat Enterprise Linux 5 and Fedora install the Bind /etc/rndc.key file with world-readable permissions, which allows local users to perform unauthorized named commands, such as causing a denial of service by stopping named.													

Total number of vulnerabilities : 4 Page : 1 (This Page)

Figura 2.3: Vulnerabilidades do sistema operativo CentOS

Contudo, uma vez que este sistema operativo está a ser utilizado para o servidor web, existe uma possibilidade de que esteja a ser utilizado o *webpanel* do CentOS.

1	CVE-2021-31324	77	Exec Code	2021-05-18	2021-05-24	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
The unprivileged user portal part of CentOS Web Panel is affected by a Command Injection vulnerability leading to root Remote Code Execution.													
2	CVE-2021-31316	89	Sql	2021-05-18	2021-05-24	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
The unprivileged user portal part of CentOS Web Panel is affected by a SQL Injection via the 'idsession' HTTP POST parameter.													
3	CVE-2020-15628	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_mail_autoreply.php. When parsing the user parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9710.													
4	CVE-2020-15627	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_mail_autoreply.php. When parsing the account parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9738.													
5	CVE-2020-15626	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_dashboard.php. When parsing the term parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9730.													
6	CVE-2020-15625	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_add_mailbox.php. When parsing the username parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9729.													
7	CVE-2020-15624	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_new_account.php. When parsing the domain parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9727.													
8	CVE-2020-15623	749	Exec Code	2020-07-28	2020-07-29	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
This vulnerability allows remote attackers to write arbitrary files on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_mod_security.php. When parsing the archivo parameter, the process does not properly validate a user-supplied path prior to using it in file operations. An attacker can leverage this vulnerability to execute code in the context of root. Was ZDI-CAN-9722.													
9	CVE-2020-15622	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_mail_autoreply.php. When parsing the search parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9712.													
10	CVE-2020-15621	89	Sql	2020-07-28	2020-07-29	7.8	None	Remote	Low	Not required	Complete	None	None
This vulnerability allows remote attackers to disclose sensitive information on affected installations of CentOS Web Panel cwp-e17.0.9.8.923. Authentication is not required to exploit this vulnerability. The specific flaw exists within ajax_mail_autoreply.php. When parsing the email parameter, the process does not properly validate a user-supplied string before using it to construct SQL queries. An attacker can leverage this vulnerability to disclose information in the context of root. Was ZDI-CAN-9711.													

Figura 2.4: Vulnerabilidades do *webpanel* do CentOS

Através da análise das 10 vulnerabilidades mais recentes pode-se concluir que este *panel* não deve ser utilizado. Estas vulnerabilidades tratam-se na maioria de execução remota de código ou de injeções de SQL. Este tipo de vulerabilidades podem por em causa a integridade e a confiabilidade dos documentos digitais, que são dois dos requisitos mais importantes do sistema.

Desde que o *webpanel* não seja utilizado, o sistema CentOS pode ser utilizado.

- Django v3.0
  Este serviço é utilizado como *backend* principal do sistema. A única vulnerabilidade detetada desta versão do Django permite *email spoofing*, mas apenas tem impacto parcial de integridade. Para além disto, existem estratégias que podem ser aplicadas para contornar esta vulnerabilidade[1].

  Com isto em mente, o Django pode ser utilizado como backend principal.

- UWSGI
  Este serviço é utilizado como servidor web do sistema.

  Após uma pesquisa de vulnerabilidades por parte da equipa de trabalho, não foram encontradas vulnerabilidades graves para o sistema em questão.

- PostgreSQL
  Este serviço é utilizado como base de dados principal(v.12.4) e como base de dados de gestão(v.12.1).

  Após a análise das vulnerabilidades mais recentes do PostgreSQL, apenas se encontraram, na **maioria**, vulnerabilidades de score CVSS menor ou igual a 5.0. Para além disto, estas vulnerabilidades mais recentes apenas afetam versões mais antigas do serviço, como a versão 7 e 8. Uma vez que só se utilizam as versões 12.1 e 12.4, não é necessário mudar o serviço para este constituinte da entidade emissora do sistema.

[1] Consultar https://django-ratelimit.readthedocs.io/en/stable/security.html

1	CVE-2021-32029	200	+Info	2021-10-08	2021-12-03	4.0	None	Remote	Low	???	Partial	None	None

A flaw was found in postgresql. Using an UPDATE ... RETURNING command on a purpose-crafted table, an authenticated database user could read arbitrary bytes of server memory. The highest threat from this vulnerability is to data confidentiality.

2	CVE-2021-32028			2021-10-11	2021-12-03	4.0	None	Remote	Low	???	Partial	None	None

A flaw was found in postgresql. Using an INSERT ... ON CONFLICT ... DO UPDATE command on a purpose-crafted table, an authenticated database user could read arbitrary bytes of server memory. The highest threat from this vulnerability is to data confidentiality.

3	CVE-2021-32027	119	Overflow	2021-06-01	2021-09-14	6.5	None	Remote	Low	???	Partial	Partial	Partial

A flaw was found in postgresql in versions before 13.3, before 12.7, before 11.12, before 10.17 and before 9.6.22. While modifying certain SQL array values, missing bounds checks let authenticated database users write arbitrary bytes to a wide area of server memory. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.

4	CVE-2021-23222	522		2022-03-02	2022-03-10	4.3	None	Remote	Medium	Not required	Partial	None	None

A man-in-the-middle attacker can inject false responses to the client's first few queries, despite the use of SSL certificate verification and encryption.

5	CVE-2021-23214	89	Sql	2022-03-04	2022-03-15	5.1	None	Remote	High	Not required	Partial	Partial	Partial

When the server is configured to use trust authentication with a clientcert requirement or to use cert authentication, a man-in-the-middle attacker can inject arbitrary SQL queries when a connection is first established, despite the use of SSL certificate verification and encryption.

6	CVE-2021-20229	863		2021-02-23	2021-06-09	4.0	None	Remote	Low	???	Partial	None	None

A flaw was found in PostgreSQL in versions before 13.2. This flaw allows a user with SELECT privilege on one column to craft a special query that returns all columns of the table. The highest threat from this vulnerability is to confidentiality.

7	CVE-2021-3677			2022-03-02	2022-04-08	4.0	None	Remote	Low	???	Partial	None	None

A flaw was found in postgresql. A purpose-crafted query can read arbitrary bytes of server memory. In the default configuration, any authenticated database user can complete this attack at will. The attack does not require the ability to create objects. If server settings include max_worker_processes=0, the known versions of this attack are infeasible. However, undiscovered variants of the attack may be independent of that setting.

8	CVE-2021-3393	209	+Info	2021-04-01	2021-06-04	3.5	None	Remote	Medium	???	Partial	None	None

An information leak was discovered in postgresql in versions before 13.2, before 12.6 and before 11.11. A user having UPDATE permission but not SELECT permission to a particular column could craft queries which, under some circumstances, might disclose values from that column in error messages. An attacker could use this flaw to obtain information stored in a column they are allowed to write but not read.

Figura 2.5: Vulnerabilidades mais recentes do PostgreSQL

- Ubuntu 20.04 O Ubuntu é utilizado como sistema operativo do serviço de gestão do sistema.

  Após a análise das vulnerabilidades mais recentes deste sistema operativo no CVE, encontraram-se as seguintes vulnerabilidades:

1	CVE-2021-44420	287	Bypass	2021-12-08	2022-02-22	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

In Django 2.2 before 2.2.25, 3.1 before 3.1.14, and 3.2 before 3.2.10, HTTP requests for URLs with trailing newlines could bypass upstream access control based on URL paths.

2	CVE-2021-3491	787	Exec Code Overflow Bypass	2021-06-04	2021-09-14	7.2	None	Local	Low	Not required	Complete	Complete	Complete

The io_uring subsystem in the Linux kernel allowed the MAX_RW_COUNT limit to be bypassed in the PROVIDE_BUFFERS operation, which led to negative values being usedin mem_rw when reading /proc/<PID>/mem. This could be used to create a heap overflow leading to arbitrary code execution in the kernel. It was addressed via commit d1f82808877b ("io_uring: truncate lengths larger than MAX_RW_COUNT on provide buffers") (v5.13-rc1) and backported to the stable kernels in v5.12.4, v5.11.21, and v5.10.37. It was introduced in ddf0322db79c ("io_uring: add IORING_OP_PROVIDE_BUFFERS") (v5.7-rc1).

3	CVE-2020-15811	444	Http R.Spl. Bypass	2020-09-02	2021-03-04	4.0	None	Remote	Low	???	None	Partial	None

An issue was discovered in Squid before 4.13 and 5.x before 5.0.4. Due to incorrect data validation, HTTP Request Splitting attacks may succeed against HTTP and HTTPS traffic. This leads to cache poisoning. This allows any client, including browser scripts, to bypass local security and poison the browser cache and any downstream caches with content from an arbitrary source. Squid uses a string search instead of parsing the Transfer-Encoding header to find chunked encoding. This allows an attacker to hide a second request inside Transfer-Encoding: it is interpreted by Squid as chunked and split out into a second request delivered upstream. Squid will then deliver two distinct responses to the client, corrupting any downstream caches.

4	CVE-2020-15810	444	Bypass	2020-09-02	2021-03-17	3.5	None	Remote	Medium	???	None	Partial	None

An issue was discovered in Squid before 4.13 and 5.x before 5.0.4. Due to incorrect data validation, HTTP Request Smuggling attacks may succeed against HTTP and HTTPS traffic. This leads to cache poisoning. This allows any client, including browser scripts, to bypass local security and poison the proxy cache and any downstream caches with content from an arbitrary source. When configured for relaxed header parsing (the default), Squid relays headers containing whitespace characters to upstream servers. When this occurs as a prefix to a Content-Length header, the frame length specified will be ignored by Squid (allowing for a conflicting length to be used from another Content-Length header) but relayed upstream.

5	CVE-2020-15707	362	Exec Code Overflow Bypass	2020-07-29	2021-09-13	4.4	None	Local	Medium	Not required	Partial	Partial	Partial

Integer overflows were discovered in the functions grub_cmd_initrd and grub_initrd_init in the efilinux component of GRUB2, as shipped in Debian, Red Hat, and Ubuntu (the functionality is not included in GRUB2 upstream), leading to a heap-based buffer overflow. These could be triggered by an extremely large number of arguments to the initrd command on 32-bit architectures, or a crafted filesystem with very large files on any architecture. An attacker could use this to execute arbitrary code and bypass UEFI Secure Boot restrictions. This issue affects GRUB2 version 2.04 and prior versions.

6	CVE-2020-15706	362	Exec Code Bypass	2020-07-29	2021-05-01	4.4	None	Local	Medium	Not required	Partial	Partial	Partial

GRUB2 contains a race condition in grub_script_function_create() leading to a use-after-free vulnerability which can be triggered by redefining a function whilst the same function is already executing, leading to arbitrary code execution and secure boot restriction bypass. This issue affects GRUB2 version 2.04 and prior versions.

7	CVE-2020-15705	347	Bypass	2020-07-29	2021-09-21	4.4	None	Local	Medium	Not required	Partial	Partial	Partial

GRUB2 fails to validate kernel signature when booted directly without shim, allowing secure boot to be bypassed. This only affects systems where the kernel signing certificate has been imported directly into the secure boot database and the GRUB image is booted directly without the use of shim. This issue affects GRUB2 version 2.04 and prior versions.

8	CVE-2020-15078	287	Bypass +Info	2021-04-26	2021-12-10	5.0	None	Remote	Low	Not required	Partial	None	None

OpenVPN 2.5.1 and earlier versions allows a remote attackers to bypass authentication and access control channel data on servers configured with deferred authentication, which can be used to potentially trigger further information leaks.

9	CVE-2020-13777	327	Bypass	2020-06-04	2020-06-19	5.8	None	Remote	Medium	Not required	Partial	Partial	None

GnuTLS 3.6.x before 3.6.14 uses incorrect cryptography for encrypting a session ticket (a loss of confidentiality in TLS 1.2, and an authentication bypass in TLS 1.3). The earliest affected version is 3.6.4 (2018-09-24) because of an error in a 2018-09-18 commit. Until the first key rotation, the TLS server always uses wrong data in place of an encryption key derived from an application.

10	CVE-2020-11934	668	Bypass	2020-07-29	2020-08-05	1.9	None	Local	Medium	Not required	None	Partial	None

It was discovered that snapctl user-open allowed altering the $XDG_DATA_DIRS environment variable when calling the system xdg-open. OpenURL() in usersession/userd/launcher.go would alter $XDG_DATA_DIRS to append a path to a directory controlled by the calling snap. A malicious snap could exploit this to bypass intended access restrictions to control how the host system xdg-open script opens the URL and, for example, execute a script shipped with the snap without confinement. This issue did not affect Ubuntu Core systems. Fixed in snapd versions 2.45.1ubuntu0.2, 2.45.1+18.04.2 and 2.45.1+20.04.2.

Figura 2.6: Vulnerabilidades mais recentes do Ubuntu(versão 20.04)

  Dado que a vulnerabilidade mais recente encontrada foi em 2021 e que este sistema operativo se trata uma das distribuições mais utilizadas atualmente, em princípio a sua utilização é segura.

- Flask 1.0

  Este serviço é utilizado como *backend* de gestão.

  Após uma pesquisa por várias vulnerabilidades, a equipa de trabalho deparou-se com uma grave vulnerabilidade de segurança na versão do primeiro lançamento da tecnologia, que permitia execução de código remota. Contudo, esta vulnerabilidade foi corrigida na versão 1.0, que é a que está a ser utilizada de momento no sistema. Com isto em conta, em princípio esta tecnologia pode continuar a ser utilizada.

- Gunicorn

  Este serviço é utilizado como segundo servidor web do sistema.

  Após uma pesquisa no CVE por potenciais vulnerabilidades de segurança, a equipa de trabalho não encontrou nenhuma fraqueza significativa. Visto isto, este serviço em principio pode continuar a ser utilizado.

- Docker 19.03

  A base de dados de gestão do sistema está contida num container Docker.

  Após uma pesquisa no CVE por potenciais vulnerabilidades de segurança, não foram encontradas vulnerabilidades graves da versão 19.03 que é utilizada pelo sistema.

Capítulo 3

Catalogação de fraquezas típicas

Como já referido anteriormente, o objeto de análise consiste numa espécie de sistema distribuído constituído por várias entidades, cada uma com a sua própria função.

Conexões

Este tipo de sistemas são muitas vezes alvos de ataques às conexões que interligam os diferentes componentes. Este tipo de ataques pode comprometer a informação transportada no sistema, nomeadamente a sua integridade e confidencialidade.

Para além disto através do acesso às conexões os atacantes podem provocar indisponibilidade dos serviços através de ataques de Negação de Serviço.

Componentes do sistema

Para além das conexões, os atacantes também se podem focar em cada componente em particular para danificar o sistema e/ou os seus dados. Relativamente a componentes como o Portador, existe a possibilidade de *Spoofing*. Um atacante pode, por exemplo, realizar uma chamada a um cidadão e passar por um trabalhador relacionado com o sistema para extrair informação para uso impróprio.

Em componentes como o Verificador, um atacante pode ter acesso a dados de Portadores, podendo também utilizar também essa informação para uso impróprio. Para além disto, caso o atacante tenha acesso suficiente, pode alterar os atributos recebidos e assina-los como se a mID fosse válida.

Em entidades do género da entidade emissora, podem ser alterados dados armazenados diretamente nas bases de dados(*tampering*).

Arquitetura

Para além de ataques às componentes do sistema e às conexões entre estes, existem vulnerabilidades que podem ter origem na própria arquitetura do projeto. Estas vulnerabilidades podem estar associadas a componentes que não deviam estar no sistema, ou cujas ligações são mal implementadas no contexto do programa.

Capítulo 4

Modelação de sistema

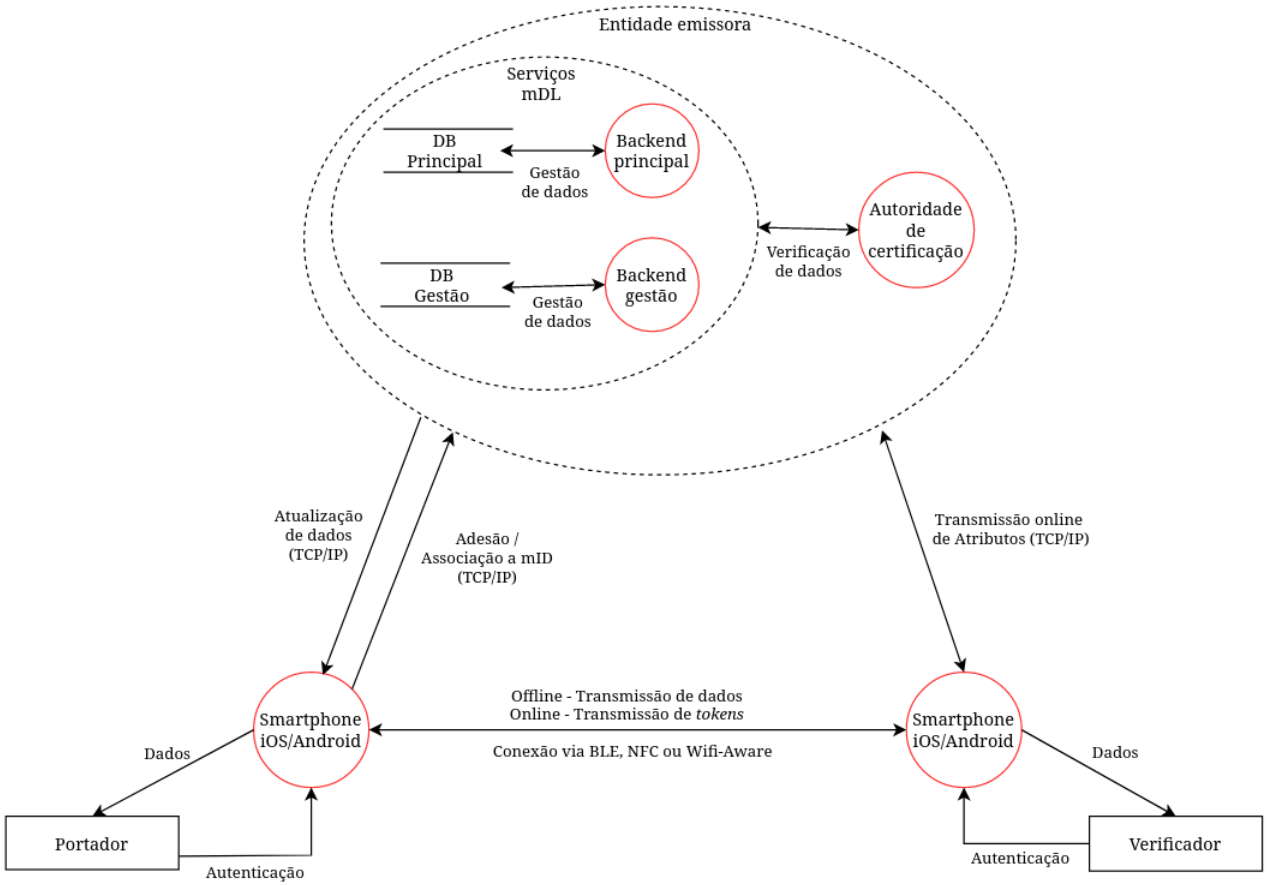


Figura 4.1: *Data Flow Diagram* do sistema mID

Capítulo 5

Modelação de ameaças

De acordo com os requisitos do sistema, a confiabilidade, privacidade e segurança dos dados e da infraestrutura tratam-se dos pontos mais importantes do serviço. Com isto em conta, a equipa de trabalho decidiu construir um modelo de ameaças orientado aos atacantes e ao software, com base no modelo `STRIDE` (*Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilegdes*).

5.1 Portador

O portador trata-se de uma entidade externa que interage com um dispositivo iOS ou Android que contém a aplicação mID.

Spoofing

O portador, uma vez que se trata de um cidadão com a aplicação mID, pode ser alvo de *Spoofing*. Como já mencionado no capítulo (3), um atacante pode entrar em contacto com o cidadão e passar por um funcionário da empresa do sistema, extraindo dados pertencentes ao sistema. Estes dados podem ser por exemplo credenciais do cidadão, que lhe permite o acesso à sua aplicação.

Uma maneira de contornar esta vulnerabilidade seria investir numa implementação de um sistema de autenticação seguro, através de *tokens*, características físicas do portador(como por exemplo impressão digital), entre outros.

Tampering

Um atacante com acesso à aplicação de um determinado portador tem acesso a ficheiros *JSON* após atualizações de dados provenientes da entidade emissora. Estes ficheiros podem então ser alterados e enviados ao Verificador. Caso este envio seja feito em modo *offline*, não existe maneira de verificar se os dados estão atualizados, o que permite a inserção de dados falsos com sucesso no sistema.

A utilização do *JWS - JSON Web Signature* ajuda a reduzir significativamente este tipo de risco.

Repudiation

Um Portador pode utilizar a vulnerabilidade descrita anteriormente para alterar os seus dados e negar que o fez(isto numa conexão offline). Esta vulnerabilidade pode ser corrigida também com a implementação de métodos de criptografia anteriormente referidos.

Information Disclosure

Como já referido nas secções de *Spoofing* e *Tampering*, um atacante pode ganhar acesso aos ficheiros JSON e retirar informação. Este risco pode também ser corrigido através da implementação de métodos de criptografia para os ficheiros *JSON*.

Elevation of Priviledge

A possibilidade de acesso à conta de um Portador(através do caso possível de *Spoofing* anteriormente descrito) permite ao atacante ganhar algum tipo de estatuto no sistema que não tinha antes.

Esta vulnerabilidade pode ser corrigida através da implementação de um sistema de autenticação seguro, como descrito na secção de *Spoofing*.

5.2 Verificador

O verificador trata-se de uma entidade externa que interage com um dispositivo iOS ou Android que contém a aplicação leitora.

Spoofing

Um atacante, caso tenha ao seu dispor dados a cerca de um dado Verificador, pode tentar obter acesso ao sistema através de um ataque de força bruta.

Para corrigir esta vulnerabilidade recomenda-se aplicar uma camada extra de segurança, como uma firewall de modo a evitar casos de autenticação incorreta.

Information Disclosure

Caso a vulnerabilidade exposta na secção de *spoofing* seja explorada, o atacante pode ter acesso a informação de um ou mais portadores, caso a aplicação do verificador em causa verifique mais do que um.

A recomendação de solução do ponto anterior também acaba por mitigar este risco de *information disclosure*.

Elevation of Priviledge

O atacante, caso obtenha acesso à conta de um Verificador, ganha um estatuto no sistema que antes não tinha.

A recomendação de solução mencionada no pontode *spoofing* pode ser também utilizada para mitigar este tipo de risco.

5.3 Smartphone iOS/Android

Os dispositivos de sistemas operativos iOS e Android são utilizados como *hosts* das aplicações mID (do portador) e leitora (do verificador). Em ambas as aplicações são armazenados e transferidos dados bidirecionalmente.

Após o estabelecimento da conexão é efetuado um *download* inicial de todos os documentos que dizem respeito ao Portador em questão através de uma comunicação TCP/IP da entidade emissora em formato *JSON*. O mesmo tipo de comunicação é também utilizada para uma atualização de dados periódica.

Para o estabelecimento da conexão entre as aplicações o portador faz *scan* de um *QR Code*. O portador pode então enviar dados através de *BLE (Bluetooth Low Energy), NFC - Near Field Communication ou Wifi-Aware*. Após um pedido proveniente do verificador, o portador envia os dados solicitados através de mensagens no formato *CBOR (Concise Binary Object Representation)*.

Spoofing

Um atacante pode ser capaz de trocar um IP de um destes dispositivos com o seu próprio(*IP-Spoofing*), o que atribui acesso a documentos de portadores.

Uma possível solução para esta vulnerabilidade seria a implementação de uma *blacklist* de endereços IP suspeitos.

Para além disto, no caso de ser utilizado o *BLE* para transferência de ficheiros, um atacante pode copiar os *advertising packets* e alterá-los inserindo o seu endereço MAC.

Uma vez que estes *packets* são transmitidos no seu formato normal, bastaria implementar um algoritmo de cifra em cada pacote, como por exemplo o CCM-AES.

No caso da utilização do *NFC* um atacante (que esteja na área de conexão de um destes dispositivos) pode fingir que é um dos dispositivos a receber dados. Isto é possível devido à maneira que o *NFC* estabelece conexões: automáticas com *tags* próprias.

No caso da utilização do *Wifi-Aware* o atacante também pode fingir ser um verificador ou portador através de *tampering* dos dados.

De modo a que este risco seja mitigado, poderia-se implementar uma política de código de segurança, palavra passe ou *token* entre os dispositivos.

Tampering / Information Disclosure

Como já referido na secção 3, um atacante pode intercetar as conexões TCP/IP entre os dispositivos, ler os pacotes de dados transportados e alterar a sua informação. Uma vez que não existe uma autenticação do sistema sempre que existe uma atualização de dados, um atacante pode intercetar a conexão de modo a retirar ou alterar a informação enviada da entidade emissora para a aplicação do portador.

Uma possível solução para redução deste risco seria utilizar um protocolo de segurança como o *TLS - Transport Layer Security* de modo a inserir uma camada extra de segurança nas conexões entre dispositivos. Também se poderia inserir um método de autenticação para as atualizações de dados.

Caso seja utilizado o *BLE* existe a possibilidade de ataques do tipo *man-in-the-middle*. Isto pode permitir (a um atacante em proximidade do dispositivo) a inserção de informação incorreta ou até maliciosa para o sistema.

Uma possível solução para evitar este tipo de ataques seria a implemetação de um prótocolo do género do *OOB - Out of Band Pairing Method*, que permite a troca de pacotes na conexão através de um método diferente. Neste caso seria o *NFC* ou o *Wifi-Aware*.

Caso seja utilizado o *NFC* um atacante pode tentar aceder à troca de dados através de *Eavesdropping* (detetando as frequências da própria comunicação) com um *RFID jammer*. Se o atacante não for capaz de alterar os dados, este pode, por exemplo, gravar a comunicação entre os dois dispositivos e guardar essa informação para partilhar com outras fontes externas.

Apesar deste ataque ter uma taxa de sucesso mais reduzida, pode na mesma ser corrigido através da implementação de um método que detetasse sempre a frequência correta do dispositivo recetor.

Caso seja utilizado o *Wifi-Aware* também podem-se detetar casos de ataques do tipo *man-in-the-middle*.

Neste caso, uma vez que esta tecnologia funciona com base no protocolo TCP/UDP, poderia ser implementado o protocolo TLS, já mencionado anteriormente.

Para além das tecnologias utilizadas para a troca de pacotes, um atacante pode retirar e alterar informação das mensagens *CBOR* trocadas.

Uma vez que o sistema utiliza o formato *COSE*(formato *CBOR* que cifra e assina os dados contidos em cada uma), este risco é reduzido significativamente.

Repudiation

Caso a comunicação entre os dispositivos for efetuada através de *Wifi-Aware*, a identidade do atacante não é conhecida. Isto permite a negação de ações realizadas por parte do atacante.

Denial of Service

Um dos ataques mais bem sucedidos conhecidos do protocolo TCP/IP é o *SYN Flood attack*. Uma vez que a primeira conexão estabelecida envolve o portador e a entidade emissora, este tem de enviar um pacote de sincronização e receber um *acknowledgment*. Um atacante pode provocar indisponibilidade dos serviços da entidade emissora, enviando pacotes de sincronização a uma velocidade que seja maior à capacidade de resposta dos servidores desta - um ataque de negação de serviço.

A implementação de *firewalls* ou de uma política de *4-way-handshake* poderia mitigar este risco.

Caso seja utilizado o *BLE* para transferência de dados, um atacante pode enviar vários pedidos de conexão rapidamente e impedir o portador e verificador de comunicarem entre si.

Caso seja utilizado o *Wifi-Aware*, um atacante pode dessincronizar os canais das entidades em comunicação. Isto impossibilita a comunicação com outros dispositivos através desta tecnologia.

A implementação de uma *firewall* também poderia mitigar este tipo de risco.

Elevation of Priviledge

Como já foi mencionado anteriormente na secção *Tampering / Information Disclosure*, caso o atacante ganhe acesso a um dos dispositivos do verificador ou do portador, este passa a fazer parte do sistema, podendo interagir com os seus componentes.

5.4 Entidade Emissora

A entidade emissora é o componente que confere a autenticidade de um documento de identificação pessoal, disponibilizando também mecanismos que garantem a autenticidade e integridade dos documentos transferidos(quer em modo *online* como *offline*). Esta entidade pode estabelecer conexões com os Portadores, de modo a transferir atualizações de dados ou com verificadores, de modo a verificar a autenticidade dos documentos obtidos.

Spoofing

Como mencionado anteriormente, a entidade emissora comunica com os restantes componentes do sistema através de conexões TCP/IP, o que possibilita a um atacante fazer *IP-Spoofing*, no qual este troca um dos endereços IP pertencentes ao sistema pelo seu próprio.

Para reduzir este risco, poderia-se implementar uma *blacklist* de endereços IP suspeitos, de modo a que estes fossem imediatamente bloqueados, impossibilitando o seu acesso a dados do sistema.

Tampering / Information Disclosure

Caso um atacante seja capaz de intercetar uma das conexões TCP/IP entre a entidade emissora e outro qualquer componente do sistema, este poderá ler(e divulgar posteriormente) ou alterar os pacotes enviados/recebidos pela entidade.

Para reduzir este risco poderia ser implementado um protocolo de segurança como por exemplo o *TLS* de modo a adicionar uma camada de segurança a estas conexões.

Um atacante também pode obter acesso às bases de dados principal ou à de gestão da entidade emissora e alterar os registos diretamente ou extraí-los para posterior divulgação não autorizada.

De modo a assegurar a integridade dos dados armazenados na entidade emissora poderia ser implementado um mecanismo de *backups* periódicas dos dados, ou até mesmo armazenar porções de dados em outros componentes do sistema(redundância de dados). Também se podem criar utilizadores com funções específicas na base de dados com autenticação necessária por código de segurança ou password.

Repudiation

Se um atacante for capaz de alterar os *logs* do sistema da entidade emissora, este pode fazer alterações e eliminar o seu registo.

É recomendável que este tipo de ficheiros tenham permissões muito restritas, como por exemplo autorizar apenas o *append* nestes ficheiros. Assim, apenas é possível adicionar informação e o atacante já não é capaz de eliminar ou alterar *logs* relativos às duas alterações.

Denial of Service

Tal como mencionado na descrição de ameaças dos dispositivos iOS/Android, um atacante pode fazer um ataque de negação de serviço através de um *SYN flood attack*.

Este risco poderia ser reduzido com a implementação de *firewalls* e/ou de uma comunicação *4-way-handshake*.

Elevation of Priviledge

Caso o atacante tenha acesso a código de *backend* do sistema este poderá executar ações como administrador.

De modo a que o atacante nunca ganhe acesso de administrador, é necessária a implementação de mecanismos de autenticaçao e de autorização seguros, assim como uma distribuição de diferentes níveis de privilégios por vários utilizadores diferentes, para que nunca seja possível a um atacante provocar uma alteração global ao sistema.

5.5 Resumo de ameaças do sistema

Ameaças	Portador	Verificador	Smartphone iOS/Android	Entidade Emissora
Spoofing	✗	✗	✗	✗
Tampering	✗	-	✗	✗
Repudiation	✗	-	✗	✗
Information Disclosure	✗	✗	✗	✗
Denial of Service	-	-	✗	✗
Elevation of Privilege	✗	✗	✗	✗

Tabela 5.1: Resumo de ameaças do sistema *mID*

Capítulo 6

Análise de risco

Na construção do serviço foram estabelecidos vários requisitos que devem ser cumpridos:

- Segurança por *design*
- Confiabilidade
- Integridade
- Interoperabilidade
- Privacidade
- Utilizável mesmo sem conectividade com a infraestrutura
- Flexíbilidade

Da análise de riscos, vulnerabilidades e ameaças feita ao longo deste documento foram encontradas fraquezas que podem por nomeadamente os requisitos de Confiabilidade, Privacidade e Integridade em causa.

Muitas destas vulnerabilidades principalmente do tipo *Information Disclosure* e *Tampering* de dados. Estes riscos são mais elevados e propenços a *exploit* nos dispositivos móveis do sistema, nas aplicações mID do portador e leitora do verificador[1]. Nos dispositivos iOS/Android existe a possibilidade de ocorrerem ataques de *man-in-the-middle*, nomeadamente através do *BLE*.

Para além da *Information Disclosure* e *Tampering* é também necessário ter em conta os riscos de *Spoofing*, que podem consequentemente dar acesso ao atacante para a alteração e obtenção de dados do sistema e de *Denial of Service* que, num caso extremo, pode causar um *crash* total ou parcial da aplicação.

O *Spoofing* é mais provável ocorrer com um Portador ou Verificador, através do contacto direto com a pessoa e Dispositivos de aplicações, que podem ser alvos de *IP-Spoofing* ou de tentativa de conexão direta por parte do atacante.

Relativamente a ameaças de *Denial of Service*, apesar de não se tratarem de relações diretas com os requisitos do sistema, é necessário corrigir todas as vulnerabilidades que existam, uma vez que a própria arquitetura do sistema já é propensa a sobrecarga em modo online. Caso um atacante tenha conhecimento desta particularidade e a use para sua vantagem, pode levar a uma indisponibilidade temporária dos serviços do sistema.

As ameaças de *Elevation of Privilege* e de *Repudiation* podem ser contornadas corrigindo as vulnerabilidades dos pontos anteriores.

[1] De notar que também são possíveis ataques à própria entidade emissora. Contudo, estas fraquezas são de mais difícil *exploit*.

Capítulo 7

Propostas de Soluções para Aumento de Segurança

De modo a melhorar a segurança no sistema, é necessário ter em conta as principais vulnerabilidades que põem em causa os requisitos mais importantes referidos na secção anterior(6).

De modo a reduzir o risco de *Information Disclosure* e *Tampering* nos dispositivos iOS/Android recomenda-se que não se utilize o *BLE*, visto que é de fácil interceção por parte de um atacante como *man-in-the-middle*. É preferível que sejam utilizadas as tecnologias *NFC* e *Wifi-Aware*(de preferência o *NFC*, uma vez que as ligações são mais difíceis de interceção devido à manipulação necessária de frequências).

Apesar disto, caso seja mesmo necessária a utilização do BLE, recomenda-se fortemente a implementação de um protocolo do mesmo tipo do *OOB - Out of Band Pairing Method*, que permite que o dispositivo transmita pacotes com outra tecnologia automaticamente.

Para melhorar a segurança relativamente à utilização do *NFC*, pode ser implementado um método de deteção de frequências, de modo a detetar corretamente a frequência do dispositivo que deve receber os dados.

Para melhorar a segurança na utilização do *Wifi-Aware*, uma vez que esta tecnologia funciona com base de TCP/UDP, pode ser implementado um protocolo de segurança como o *TLS - Transport Layer Security*. Este protocolo também pode ser implementado nas conexões em modo *online* entre a entidade emissora e as restantes componentes do sistema, de modo a que o atacante não tenha acesso aos dados contidos nos pacotes transportados.

Para reduzir o risco de *Denial of Service* nas conexões TCP/IP em modo *online*, podem ser implementadas *firewalls* que filtrem pacotes SYN que não tenham obtido resposta. Para melhorar a sobrecarga na entidade emissora em modo online, são propostas algumas mudanças na arquitetura no sistema. É proposto que se retire a conexão direta entre a aplicação mID e a entidade emissora. Esta conexão pode permitir que atacantes tenham acesso mais facilitado através de *spoofing* a um cidadão Portador, que por norma são os principais alvos deste tipo de ataques. Visto isto, é sugerido que os verificadores sejam as únicas entidades a comunicar com a entidade emissora.

De modo a permitir a atualização de dados nas aplicações mID e auxiliar a recuperação de dados em caso de ataque direto às bases de dados da infraestrutura da entidade emissora, é sugerido utilizar redundância de dados - colocar os dados estritamente necessários aos portadores ligados a cada verificador, cifrados com métodos confiáveis de impossível decifra por ataques de força bruta e de díficil descoberta da chave de cifra. Para isto, também seria necessário estabelecer um número máximo de portadores ligados a cada verificador, de modo a não causar sobrecarga nestes nodos.

Para uma melhor visualização da solução proposta, foi desenvolvido um novo *Data Flow Diagram*:

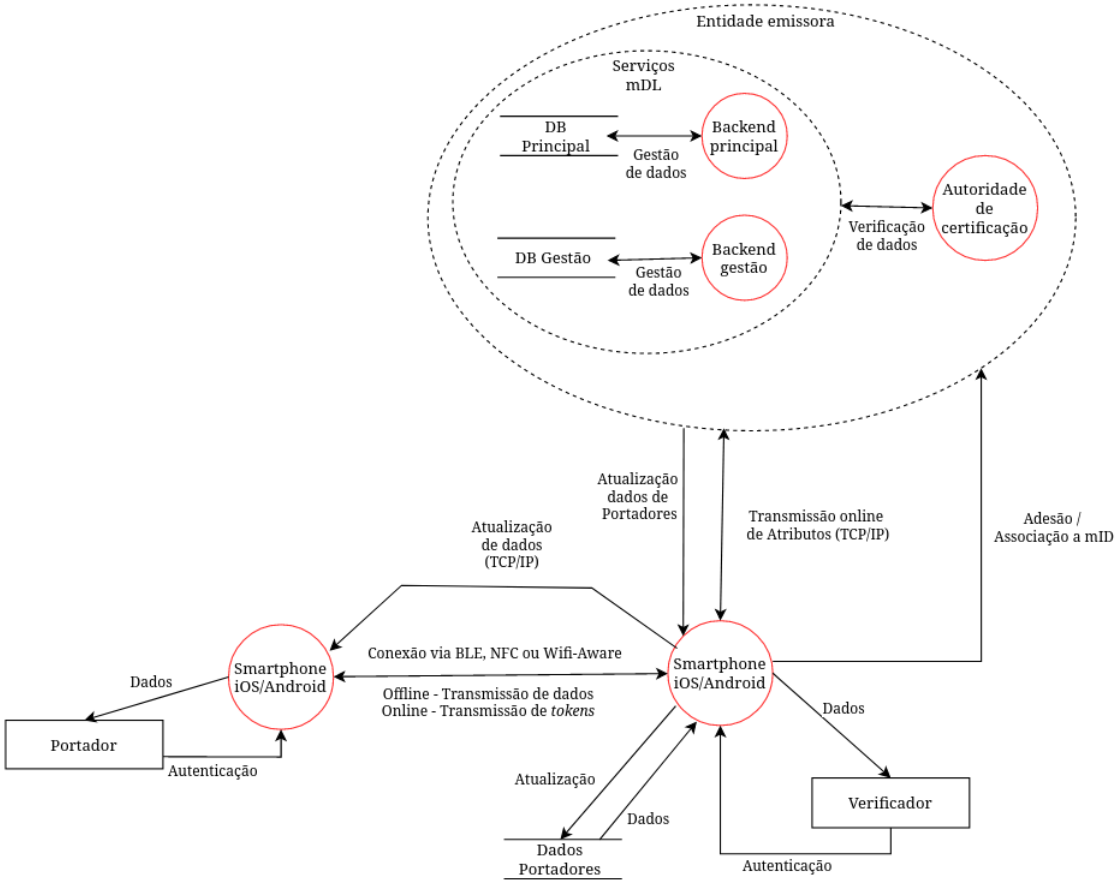


Figura 7.1: Proposta de arquitetura para o sistema mID

Deste modo, os pedidos de adesão iriam ser enviados para o verificador. Uma vez que cada verificador pode ter mais do que um portador associado, este podia juntar todos os pedidos de adesão num só, o que também iria tirar "peso"na parte da receção de pedidos na infraestrutura.

Por fim, em termos de serviços utilizados pela infraestrutura da entidade emissora podem ser utilizados os já propostos. De notar apenas que não se deve utilizar o *webpanel* do CentOS(*host* do servidor web), por apresentar vulnerabilidades de injeções de SQL e de execução de código remota.

Para melhorar a segurança nesta parte, aconselha-se ainda a utilização de uma WAF(Web Application Firewall). Este tipo de serviço poderia proteger o sistema contra ataques de *Cross-site Scripting* (XSS) e *SQL Injection*. De modo a não adicionar custos à empresa, aconselha-se a utilização de serviços da OWASP, como OWASP ModSecurity CRS, OWASP Coraza WAF e WASC OWASP Web Application Firewall Evaluation Criteria Project (WAFEC).

Para proteger portadores, também se aconselha a utilização de servidores *proxy* (de preferência um que também monitorize atividades online, de modo a permitir uma deteção mais fácil de ataques ao sistema). Um servidor proxy que poderá ser utilizado seria, por exemplo, o *Bright Data*.

Capítulo 8

Referências

- Catalogação de vulnerabilidades e *exploits*:
  - https://www.cvedetails.com/vulnerability-list.php?vendor_id=1224&product_id=19997&version_id=0&page=1&hasexp=0&opdos=0&opec=0&opov=0&opcsrf=0&opgpriv=0&opsqli=0&opxss=0&opdirt=0&opmemc=0&ophttprs=0&opbyp=0&opfileinc=0&opginf=0&cvsscoremin=0&cvssscoremax=0&year=0&cweid=0&order=1&trc=2563&sha=1bd76566e804bd0baf4aa6ef43598ed24565b5b6
  - https://www.cvedetails.com/vulnerability-list/vendor_id-49/product_id-15556/Apple-Iphone-Os.html
  - https://www.cvedetails.com/cve/CVE-2019-2102/
  - https://www.cvedetails.com/cve/CVE-2019-17517/
  - https://www.cvedetails.com/cve/CVE-2020-15531/
  - https://www.cvedetails.com/cve/CVE-2017-17225/
  - https://www.wi-fi.org/discover-wi-fi/wi-fi-aware
  - https://www.cvedetails.com/vulnerability-list/vendor_id-10167/product_id-18131/Centos-Centos.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-17565/Centos-webpanel.html
  - https://django-ratelimit.readthedocs.io/en/stable/security.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-10199/product_id-18211/version_id-625305/Djangoproject-Django-3.0.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-336/product_id-575/Postgresql-Postgresql.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-4781/product_id-20550/version_id-579251/opbyp-1/Canonical-Ubuntu-Linux-20.04.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-17891/product_id-45578/Gunicorn-Gunicorn.html
  - https://www.cvedetails.com/vulnerability-list/vendor_id-17201/product_id-57169/Palletsprojects-Flask.html
  - https://medium.com/swlh/hacking-flask-applications-939eae4bffed
  - https://palletsprojects.com/blog/flask-1-0-released/
  - https://www.cvedetails.com/vulnerability-list/vendor_id-13534/product_id-28125/Docker-Docker.html
- Catalogação de fraquezas típicas:
  - https://www.cs.purdue.edu/homes/bb/bhargava-vuln-threats.pdf
- Modelação de Ameaças:

  - https://www.quora.com/Isnt-it-useless-to-encrypt-a-JSON-object-when-it-is-passed-in-an-HTTPS-communication
  - https://www.novelbits.io/bluetooth-low-energy-advertisements-part-1/
  - https://developer.android.com/guide/topics/connectivity/wifi-aware
  - https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview
  - https://developer.android.com/guide/topics/connectivity/nfc
  - https://www.internetsociety.org/deploy360/tls/basics/
  - https://www.csa.gov.sg/singcert/alerts/multiple-vulnerabilities-in-bluetooth-low-energy-devices
  - https://resources.infosecinstitute.com/topic/near-field-communication-nfc-technology-vulnerabilities-and-principal-attack-schema/
  - https://tools.ietf.org/id/draft-ietf-cose-rfc8152bis-struct-00.html
  - https://www.oreilly.com/library/view/network-security-hacks/0596006438/ch01s06.html

- Propostas de Soluções para Aumento de Segurança
  - https://owasp.org/www-community/Web_Application_Firewall
  - https://www.softwaretestinghelp.com/best-proxy-server/
  - https://brightdata.com/proxy-types/super-proxy?gspk=dmlqYXlrdW1hcnNoaW5kZTYwMzc&gsxid=nOpSIeKZRP4i&utm_source=affiliates&utm_campaign=dmlqYXlrdW1hcnNoaW5kZTYwMzc