

# Dispositivos *Android*

Filipe Henriques

*Mestrado de Cibersegurança e Informática Forense*  
*Instituto Politécnico de Leiria*  
Leiria, Portugal  
2180066@my.ipleiria.pt

Jéssica Pedrosa

*Mestrado de Cibersegurança e Informática Forense*  
*Instituto Politécnico de Leiria*  
Leiria, Portugal  
2180067@my.ipleiria.pt

Patrícia Silva

*Mestrado de Cibersegurança e Informática Forense*  
*Instituto Politécnico de Leiria*  
Leiria, Portugal  
2180068@my.ipleiria.pt

Tiago Martins

*Mestrado de Cibersegurança e Informática Forense*  
*Instituto Politécnico de Leiria*  
Leiria, Portugal  
2182716@my.ipleiria.pt

**Resumo**—No âmbito da unidade curricular de Análise Forense Digital II do mestrado de Cibersegurança e Informática Forense, foi elaborado um cenário prático onde se efetuou uma análise forense a um dispositivo *Android*. Esta análise foi realizada num *Android* de versão 9 com uma ROM customizada. Serão exploradas diferentes formas de extração de dados, utilizando para tal, diferentes ferramentas. De modo a obter um cenário mais realista, esta análise foca-se principalmente nas aplicações de troca de mensagens instantâneas, como o *WhatsApp*.

**Index Terms**—*Android*, Extração de dados, Análise forense, *WhatsApp*, *Facebook Messenger*

## I. INTRODUÇÃO

No âmbito da unidade curricular de Análise Forense Digital II do mestrado de Cibersegurança e Informática Forense, da Escola Superior de Tecnologia e Gestão, do Instituto Politécnico de Leiria, foi elaborado o presente artigo sob o tema “Dispositivos *Android*”. Este artigo consiste na exploração de certos aspetos da análise forense a dispositivos móveis *Android*, aplicado a um caso prático.

Será explorada a versão 9 do *Android* que, até à data, é a mais recente, sendo feito um estudo à hierarquia de ficheiros do sistema *Android*, assim como aos tipos de sistemas de ficheiros utilizados, e uma exemplificação destes tópicos aplicados num dispositivo *Android*. Serão também explorados os diferentes métodos de aquisição de dados, as diferenças entre cada um destes e será realizada aquisições a um equipamento *Android*, recorrendo a alguns métodos. Uma vez que o cenário selecionado para a fase prático-laboratorial do trabalho está relacionado com o *WhatsApp* (uma aplicação de mensagens instantâneas e chamadas de voz) e com o *Facebook Messenger*, estes serão previamente alvo de uma análise, de modo a se compreender alguns aspetos bem como o funcionamento dos mesmos. Deste modo, o trabalho passará por entender o funcionamento interno do *Android*, perceber o conceito de análise forense em dispositivos móveis e ainda o funcionamento do *WhatsApp* e de outras aplicações de mensagens instantâneas. Para a análise serão eliminadas algumas

mensagens tanto no *WhatsApp* como no *Facebook Messenger*, serão efetuadas as aquisições e serão analisadas as imagens de forma a determinar se é possível ter acesso às mensagens das aplicações, incluindo as mensagens previamente eliminadas. Essa análise será efetuada recorrendo ao *Autopsy* e, se possível, ao *XAMN*.

Este relatório está dividido em quatro capítulos, sendo que o capítulo dois consiste no estado da arte, onde serão introduzidos os diversos conceitos explorados neste artigo. O capítulo três será o cenário prático-laboratorial e por fim, o quarto capítulo é constituído pela conclusão do trabalho.

## II. ESTADO DA ARTE

### A. Breve História do *Android*

A *Android Inc.* foi fundada por 4 pessoas, nomeadamente, Rich Miner, Nick Sears, Cohris White, e Andy Rubin em 2003 e em 2005 a *Google* adquiriu a companhia. [7] Mais tarde a *Google* juntou-se a várias empresas, tendo formando essas empresas a *Open Handset Alliance* (OHA), que desenvolveu o sistema operativo *Android*. [30] A *Google* lançou pela primeira vez, em 2008, o primeiro dispositivo com o sistema *Android*. [7]

Até ao início de 2011, a *Google* já tinha lançado pelo menos 7 versões *Android*, sendo que todas elas tiveram o nome baseado em doces, uma prática que se mantém até hoje, com exceção da primeira versão que foi lançada. [7]

O *Android* é um dos sistemas operativos mais usados em *smartphones*, tanto em Portugal, como no mundo. A Figura 1 - *Quantidade de sistemas operativos em dispositivos móveis pelo mundo* representa a quantidade de sistemas operativos nos *smartphones*, em todo o mundo.

A distribuição da quantidade de dispositivos que utilizam as várias versões *Android* está representada na Figura 2. Os dados foram recolhidos durante 7 dias em 2018 e todas as versões com menos de 0.1% não irão aparecer.

Como se pode perceber, além do sistema *Android* ser um dos mais utilizados em *smartphones*, as versões que nos

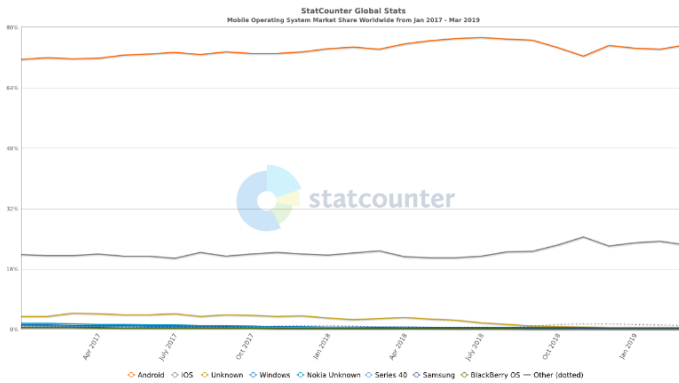


Figura 1. Quantidade de sistemas operativos em dispositivos m3veis pelo mundo [17]

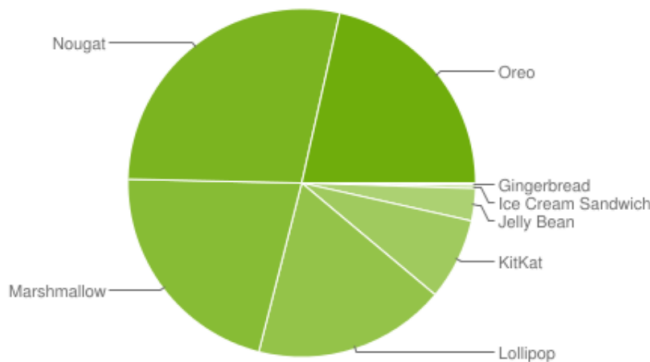


Figura 2. Distribuição das diferentes versões Android nos dispositivos [2]

dias de hoje são mais populares, são as versões *Nougat*, *Marshmallow*, *Oreo* e *Lollipop*.

### B. Estrutura Interna do Android

1) *Partições do Android*: O sistema *Android* está organizado em várias partições. Destas partições, apenas algumas são normalmente acedidas pelo utilizador, enquanto as outras são partições utilizadas pelo sistema.

A estrutura, que normalmente se encontra, consiste nas seguintes partições: [12]

- *boot* – Esta partição contém a informação e os ficheiros necessários para que o dispositivo arranque. Nela está contido o *kernel* e o *RAM disk*.
- *system* – Aqui encontram-se vários ficheiros e directorias referentes ao sistema operativo, como serviços e programas.
- *recovery* – Esta partição foi desenvolvida com o propósito de permitir a recuperação do sistema no caso deste ficar corrompido, tendo um conjunto de ferramentas para o mesmo.
- *data* – Esta partição contém ficheiros relativos à *ROM*, assim como outros dados de configurações do sistema e das aplicações.
- *cache* – Esta partição é utilizada para guardar informação frequentemente acedida pelo sistema, aplicações e

outros programas, assim como alguns *logs* para rapidez de acesso, sendo assim uma partição muito valiosa em termos de informação no ponto de vista forense.

- *misc* – Nesta partição estão guardadas várias definições diferentes tais como de *hardware*, *USB* entre outras, assim como informações sobre estados do dispositivo.
- *sdcard* – Nesta partição encontram-se os ficheiros dos utilizadores como as fotografias, os vídeos, as músicas, os contactos, os *SMS*, os ficheiros gerados pelas aplicações transferidas, entre outros ficheiros.

Dentro da partição *sdcard*, existem algumas pastas que são comuns às diversas versões do *Android*, assim como as pastas onde a informação pessoal do utilizador está contida. As pastas são as seguintes:

- *Android* – Dentro desta pasta existem mais duas pastas, a *data* e a *obb*. Nestas pastas é onde estão instaladas as aplicações transferidas entre outros dados gerados pelas mesmas.
- *DCIM* – Aqui é onde todas as fotografias tiradas a partir de uma aplicação com a câmara são guardadas, assim como os vídeos.
- *Pictures* – Esta pasta tem imagens que tenham sido guardadas a partir de aplicações como o *Facebook* e outras.
- *Download* – Nesta pasta são guardados todos os ficheiros relativos a *downloads*, seja o *download* a partir do *browser* ou outra aplicação que também tenha esta funcionalidade.

2) *Sistema de Ficheiros Android*: Sendo o *Android* baseado em *Linux*, os tipos de sistemas de ficheiros suportados pelo *Android* serão muito semelhantes aos suportados pelo *Linux*. No entanto, existem alguns tipos específicos ao *Android*. Os tipos de sistemas de ficheiros suportados por um dispositivo podem ser obtidos através da uma *ADB shell* com o comando "*\$ cat /proc/filesystems*". Na Figura 3 é possível observar os tipos suportados pelo dispositivo *Xiaomi Mi 5*.

Os tipos de sistema de ficheiros utilizados estão dependentes do dispositivo, mas no caso do dispositivo que será utilizado para este projeto, o *Xiaomi Mi 5*, é possível observar os tipos de sistema de ficheiros utilizados nas diferentes partições através da *ADB shell*, com o comando "*\$ mount*". Neste caso, existiam muitos *mountpoints* mas, serão apenas apresentados os que foram considerados fundamentais para o funcionamento básico do *Android*. Na Tabela I estão representados alguns dos *mountpoints* e respetivos sistemas de ficheiros e dispositivos. De notar que, a Tabela I foi construída através de informações retiradas do *smarthphone Xiaomi Mi 5*.

O *ext4* (*Extended File System 4*) é vulgarmente utilizado em ambientes *Linux* como o tipo principal de sistema de ficheiros. [12] O *sdcardfs* é um sistema de ficheiros do estilo *FUSE* (*Filesystem in Userspace*) que basicamente funciona como um sistema de ficheiros virtual, onde os dados não são guardados em si, mas sim noutros lugares do sistema de armazenamento, servindo então como uma camada tradutora. [40] O *tmpfs*, como se pode deduzir do nome, é um tipo

```

~ ➤ $ adb shell
gemini:/ $ cat /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    tmpfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    debugfs
nodev    tracefs
nodev    sockfs
nodev    pipefs
nodev    ramfs
nodev    configfs
nodev    devpts
nodev    ext3
nodev    ext2
nodev    ext4
nodev    vfat
nodev    msdos
nodev    sdfat
nodev    exfat
nodev    sdcardfs
nodev    cifs
nodev    fuseblk
nodev    fuse
nodev    fusectl
nodev    f2fs
nodev    pstore
nodev    selinuxfs
nodev    functionfs
gemini:/ $ _

```

Figura 3. Tipo de ficheiros suportados pelo Xiaomi Mi 5

Tabela I

Mountpoints E RESPECTIVOS SISTEMAS DE FICHEIROS E DISPOSITIVOS

Dispositivo	Mountpoint	Sistema De Ficheiros
rootfs	/	rootfs
tmpfs	/dev	tmpfs
proc	/proc	proc
sysfs	/sys	sysfs
selinuxfs	/sys/fs/selinux	selinuxfs
tmpfs	/mnt	tmpfs
/dev/block/sda12	/persist	ext4
/dev/block/sde39	/system	ext4
/dev/block/sde38	/vendor	ext4
tmpfs	/system/etc	tmpfs
/dev/block/sda13	/cache	ext4
/dev/block/sda14	/data	ext4
tmpfs	/sbin	tmpfs
tmpfs	/storage	tmpfs
/data/media	/storage/emulated	sdcardfs
/data/media	/mnt/runtime/default/read	sdcardfs
/data/media	/mnt/runtime/default/write	sdcardfs
/data/media	/mnt/runtime/default/emulated	sdcardfs

de sistema de ficheiros para armazenamento temporário. [51] O *rootfs* é uma instância especial do *ramdisk*. [46] O *sysfs* é um pseudo sistema de ficheiros que exporta informação de vários subsistemas do *kernel*, dispositivos de *hardware* e *drivers* associadas a módulos do *kernel* carregados para o dispositivo. [50] O *selinuxfs* faz parte do *SELinux Module* (*SecurityEnhanced Linux*) que é uma das funcionalidades de segurança que podem ser utilizadas no *Linux*. [47] O *proc* ou *procfs* tal como o *sysfs*, é um sistema de ficheiros especial que apresenta informações acerca de processos, entre outras. [49]

3) *Hierarquia do Sistema de Ficheiros Android*: Neste dispositivo em concreto, o *Xiaomi Mi 5*, através da *ADB shell* com o comando "ls -lah" com o utilizador *root*, obteve-se a seguinte lista que é a hierarquia de ficheiros do dispositivo:

- .
- ..
- acct
- bin -> /system/bin
- bt\_firmware -> /vendor/bt\_firmware
- bugreports -> /data/user\_
- de/0/com.android.shell/files/bugreports
- cache
- charger -> /sbin/charger
- config
- d -> /sys/kernel/debug
- data
- default.prop -> system/etc/prop.default
- dev
- dsp -> /vendor/dsp
- etc -> /system/etc
- firmware -> /vendor/firmware\_mnt
- init
- init.envron.rc
- init.rc
- init.usb.configfs.rc
- init.usb.rc
- init.zygote32.rc
- init.zygote64\_32.rc
- mnt
- odm
- oem
- persist
- plat\_file\_contexts
- plat\_hwservice\_contexts
- plat\_property\_contexts
- plat\_seapp\_contexts
- plat\_service\_contexts
- proc
- product -> /system/product
- res
- root
- sbin
- sdcard -> /storage/self/primary
- sepolicy
- storage
- sys

- system
- ueventd.rc
- vendor
- vendor\_file\_contexts
- vendor\_hwservice\_contexts
- vendor\_property\_contexts
- vendor\_seapp\_contexts
- vendor\_service\_contexts
- vndservice\_contexts

De referir que os diretórios com o prefixo *init*, retêm informação acerca do *ramdisk*.

### C. Versão Utilizada para o Trabalho

Para este trabalho será feita a análise à versão 9.0 do *Android*, também conhecido como *Android Pie*. Tal como em grande parte das atualizações das versões do *Android*, esta também apresentou alterações ao nível da interface do utilizador. Esta versão apresenta o protocolo de segurança *DNS over TLS (DoT)*, que permite encriptar os pedidos *DNS*, o que permite evitar ataques *eavesdropping* e *spoofing*. [3] Também apresenta a *Android Dashboard*, que informa o utilizador acerca do tempo que o mesmo passa ao telemóvel. Para além das referidas, possui também outras funcionalidades novas e suporta o *Vulkan 1.1*, que é uma *API* de computação e gráficos. [21] [48]

Algumas características de segurança interessantes desta versão passam por os pedidos das aplicações, por omissão, serem feitos em *HTTPS*; os *backups* desta versão serem encriptados; as aplicações em *background* não terem acesso ao microfone e à câmara do dispositivo, mas também existem novas medidas de segurança relacionadas com os outros sensores do telemóvel. [31]

### D. Importância da Análise Forense a Dispositivos Móveis

Com o desenvolvimento tecnológico a que se assiste atualmente, é notório que a sociedade tem presente na sua vida a mais variada forma de tecnologia. Nos últimos anos, conceitos como *Internet of Things*, *Cloud Computing*, *Big Data* bem como os *smartphones* são tendências tecnológicas crescentes. Relativamente aos *smartphones*, é significativamente notório que estes, nos últimos anos, são cada vez mais uma presença assídua no dia-a-dia por diversas razões, estando de certo modo presentes em bastantes atividades do quotidiano. No gráfico apresentado (Figura 4 - *Utilização de telemóveis em comparação com outros dispositivos*) é possível verificar a diferença de percentagens de utilização entre *smartphones*, *desktops* e *tablets*. Conclui-se, por análise do gráfico, que de facto, é crescente a utilização dos *smartphones*, acabando mesmo por existir um destaque significativo em termos de utilização comparativamente com os restantes.

Como resultado desta utilização tão frequente, nos dias de hoje, os *smartphones* apresentam uma quantidade de informação considerável, diversificada e valiosa - acerca não só do dispositivo em si, mas também do utilizador do mesmo. Isto leva a que, atualmente, os *smartphones* sejam um elemento relevante a ser alvo de análise no âmbito de investigações

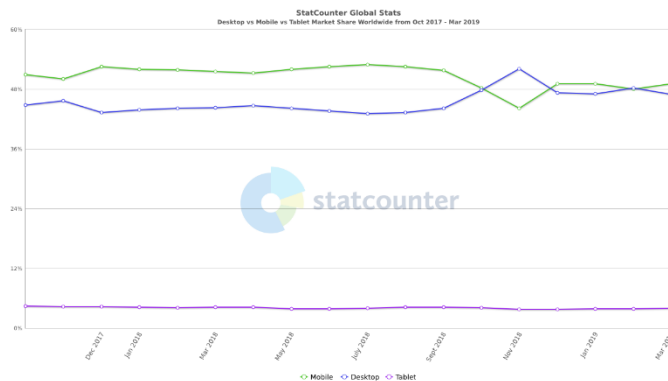


Figura 4. Utilização de telemóveis em comparação com outros dispositivos [16]

digitais, designando-se o referido por análise digital a dispositivos móveis. Segundo a *National Institute of Standards and Technology (NIST)*, análise forense a dispositivos móveis “(...) é a ciência da recuperação de provas digitais de um dispositivo móvel em condições forenses, utilizando métodos aceites” [6].

O processo de análise forense a dispositivos móveis apresenta um conjunto de fases. A primeira etapa, a apreensão do dispositivo, consiste principalmente em se proceder ao isolamento do dispositivo em relação à rede. De seguida, ocorre a aquisição que consiste na identificação e extração de dados e, por fim, ocorre a fase de examinação e análise dos mesmos. De uma forma geral, com a aquisição de dados presentes em *smartphones*, poderá ser possível ter acesso a fotografias, conversas recentes, dados de localização, histórico de chamadas (recebidas, efetuadas e não atendidas), histórico de navegação na Internet, *cookies*, listas de tarefas, dados apagados, informações acerca da conexão *WiFi*, entre outros. [33]

Tal como já foi mencionado anteriormente, um dos primeiros passos a se concretizar no processo de análise forense a dispositivos móveis é a realização do isolamento do dispositivo em relação à rede de modo a evitar-se alterações nos dados existentes no dispositivo bem como evitar *remote locks* ou *wipes*. O processo de isolamento pode ser efetuado de dois modos distintos. A primeira forma, que é mais objetiva e simples, passa por colocar-se o *smartphone* em modo avião. Já outras formas alternativas consistem em remover o cartão *SIM* e desligar a conexão *WiFi*, utilizar uma *faraday cage* e ainda utilizar os *signal jammers*. [15]

Relativamente à aquisição de dados em *smartphones*, há três métodos, nomeadamente o método físico, o lógico e ainda o manual. Na análise forense a dispositivos móveis, o método físico consiste em se criar uma cópia exata bit-a-bit do sistema de ficheiros, adquirindo-se informações do dispositivo através do acesso direto à memória *flash* – memória não volátil que é utilizada, por exemplo, em cartões de memória e *pens USB*. Os dados extraídos com este método normalmente estão na forma de dados brutos o que faz com que não sejam legíveis

ao olho humano de imediato. É de referir que para a utilização deste método de extração, geralmente recorre-se ao método *JTAG (Joint Action Test Group)* que permite efetuar uma aquisição física de forma não invasiva. [8] Um dos grandes benefícios deste tipo de aquisição passa precisamente pelo facto de esta permitir adquirir todos os dados presentes no dispositivo – incluindo tanto os que tenham sido apagados previamente bem como a informação presente em espaço não alocado. Já o método de aquisição lógica passa por sincronizar o conteúdo de um *smartphone* com um computador, através das *API* do fabricante do equipamento. Através deste método serão extraídos os ficheiros existentes num armazenamento lógico, como por exemplo, de uma partição do sistema de ficheiros. Neste método é relevante que o investigador tenha em atenção se, por alguma razão, o *smartphone* é modificado durante o processo de aquisição. Este tipo de aquisição é geralmente efetuado através da *ADB shell*. É de mencionar que a quantidade de dados recolhidos está diretamente dependente do *smartphone* ter ou não permissões *root*. [1] Poder-se-á obter dados como mensagens de texto, histórico de chamadas, lista das aplicações instaladas (com versão), detalhes de localização (dados do *GPS*) ou imagens; é de ter em conta que os dados existentes em espaço não alocado não são recuperados. [12] [45]

Quando de facto não há outra forma possível de adquirir informação do *smartphone* ou quando a informação que se pretende obter/verificar é reduzida, poder-se-á recorrer ao método de aquisição manual. Este consiste em visualizar o conteúdo do *smartphone* em causa e, geralmente, implica o uso dos botões, do teclado ou do *touchscreen*, sendo que as ações executadas podem ser gravadas com uma câmara externa. [6] Não se poderá ainda ignorar o facto de este ser um método de aquisição muito dispendioso em termos de tempo e que o mais pequeno erro poderá levar à perda de dados cruciais à investigação.

Por muito que já sejam efetuadas atualmente análises forenses a dispositivos, esta é uma área que ainda enfrenta alguns desafios como por exemplo a necessidade de possuir o material/*hardware* (como conectores) adequado para cada *smartphone*. [44]

#### E. Processo Geral de Instalação de uma ROM Customizada

Para dispositivos relativamente recentes, o processo inicia-se por conseguir obter permissões *root* no dispositivo. Dependendo de o dispositivo ser suportado ou não (<https://twrp.me/Devices/>), o próximo passo passa por instalar a aplicação do *TWRP* (<https://twrp.me/app/>), que até ao momento, encontra-se na *Google Play Store*. A partir da aplicação, é possível instalar uma *recovery* customizada, que irá permitir a instalação da *ROM* customizada. Dependendo da versão *Android* do dispositivo, poderá ser necessário instalar a última versão da *ROM* do fabricante antes de instalar a *ROM* customizada pretendida. Para *ROM* customizadas, a mais popular e com atualizações frequentes é a *Lineage OS*. Para instalar uma *ROM* baseada em *Lineage OS*, no caso de o dispositivo ser suportado, basta ir ao *site* oficial do *Lineage* na parte de *down-*

*loads* (<https://download.lineageos.org>), selecionar o dispositivo e fazer *download* da última versão. Uma particularidade das *ROM* baseadas em *Lineage OS*, é que não vêm com os serviços da *Google Play Store* nem com aplicações da *Google*. Para contornar isto, é possível instalar um pacote com os serviços da *Google Play* e outras aplicações da *Google*, que não estejam disponíveis na *Google Play Store*. Existem várias implementações deste método, no entanto, a mais popular é o projeto *OpenGapps* (<https://opengapps.org/>). No *site* do *OpenGapps* seleciona-se a arquitetura do processador, seguido da versão *Android* da *ROM* a instalar, e o pacote em si. Existem vários pacotes desde o “*pico*” que apenas contém o mínimo para os serviços da *Google Play* ficarem funcionais, até ao “*Aroma*”, que é um instalador gráfico que permite selecionar entre todas as aplicações disponíveis pelo projeto *OpenGapps*. Poderá também ser instalado um *root manager* de modo a ter permissões *root* no *smartphone* para assim desbloquear funcionalidades adicionais. Um dos *root managers* mais populares é o *Magisk Manager* (<https://magiskmanager.com>), e é relativamente suportado pela maior parte dos dispositivos/versões *Android* recentes. O *Magisk Manager* consiste numa aplicação normal *Android* onde se pode gerir o acesso a permissões *root* a determinadas aplicações que as requeiram, assim como outras funcionalidades adicionais. No entanto, para o *Magisk Manager* funcionar, é necessário ter o *Magisk Framework* instalado, em que o processo da instalação deste é semelhante à de instalação de uma *ROM* customizada. [38]

#### F. Aplicações de Mensagens Instantâneas

1) *WhatsApp*: O *WhatsApp* é uma aplicação *freeware* de partilha de mensagens entre diversos tipos de plataformas, e tem como metodologia de comunicação, *VoIP*. Esta aplicação foi criada pela empresa *WhatsApp Inc.* fundada por antigos trabalhadores do *Yahoo*, Jan Koum e Brian Acton em 2009, mais recentemente em 2014, o *WhatsApp Inc.* foi adquirida pelo *Facebook*. [22] [37] O *WhatsApp* para além de disponibilizar a funcionalidade de *SMS* básica, esta também apresenta funcionalidades para elaborar conversas em grupo, manter estas conversas sincronizadas entre diversos dispositivos, efetuar chamadas de voz ou de vídeo, partilhar ficheiros multimédia como imagens, vídeos, gravações de voz, ou documentos com tamanho não superior a 100 *MB*. E, em cima de todas estas funcionalidades, o *WhatsApp* fornece segurança na forma de encriptação de ponto-a-ponto assim que é instalada a aplicação. Todas estas funcionalidades tiram partido da Internet sempre que possível, pelo que tornam estas funcionalidades gratuitas, caso um utilizador quera usar a aplicação sem Internet este poderá ter de pagar taxas pelo serviço, isto dependendo da operadora. [23] [27]

Esta aplicação usa uma versão customizada do *Extensible Messaging and Presence Protocol*, *XMPP*, que é uma norma aberta ao público. É através deste protocolo que o *WhatsApp* efetua as comunicações *VoIP*. [29] Todas as chaves são geradas no lado do servidor por forma a evitar a criação de par de chaves inseguras. Estas chaves são por sua vez usadas no sistema



de encriptação ponto-a-ponto, este sistema é essencialmente o protocolo *Signal* criado pelo *Open Whisper Systems*. [26]

Os servidores do *WhatsApp* evitam guardar dados relativos aos conteúdos das conversas na aplicação cliente, desta forma ele segue um mecanismo de “*store and forward*” para a troca de mensagens entre utilizadores. A utilidade deste mecanismo é a de possibilitar conversas contínuas e de forma privada entre dispositivos, mesmo se estes não estiverem disponíveis, por exemplo, quando um utilizador envia uma mensagem, esta segue primeiro para os servidores do *WhatsApp* onde fica guardada temporariamente, assim que é possível enviar a mensagem para o destinatário o servidor envia e depois elimina essa mensagem do servidor. Caso o servidor não consiga enviar a mensagem dentro de 30 dias, o servidor eliminará a mensagem. Desta forma, todos os dados do *WhatsApp* são apenas armazenados de forma permanente no dispositivo cliente, mais concretamente na pasta “*Android/data/data/com.whatsapp/*” do armazenamento interno *Android*. Por estes motivos quando um utilizador necessitar de restaurar o histórico do seu *chat*, o *WhatsApp* oferece opções para exportar e importar esses dados. [24] [41] Em termos de conteúdo multimédia, conteúdo que tenha um tamanho máximo de 100 MB será enviado de forma normal, caso o tamanho ultrapasse o valor máximo, o *WhatsApp* oferece uma lista de outras aplicações recomendadas, como o *Dropbox* ou o *Google Drive*, pela qual será enviado o conteúdo multimédia. Tudo isto significa que o conteúdo da aplicação relativo às conversas na aplicação será guardado no dispositivo cliente. [25] Mas, do ponto de vista forense, o *WhatsApp* e outras as aplicações de partilha de mensagens como o *Skype*, o *Viber* e o *Tango* produzem artefactos como a data de instalação, os dados de tráfego, os dados de conteúdo, os dados do perfil de utilizador, os dados de autenticação de utilizador, os ficheiros partilhados, a base de dados de contactos, e os dados de localização, que têm valor numa análise forense digital. [6]

2) *Facebook Messenger*: O *Facebook Messenger*, ou simplesmente *Messenger*, é uma aplicação *freeware* de mensagens e foi desenvolvida pelo *Facebook* em 2008. Tal como o *WhatsApp*, o *Messenger* utilizava também uma versão personalizada do protocolo *XMPP* para as suas comunicações, mas atualmente este usa na sua versão móvel o protocolo *MQTT*, e *HTTP* na versão para *browser*. [34] [32] Atualmente esta aplicação apresenta duas versões, o *Messenger* e o *Messenger Lite*, sendo que a principal diferença entre elas está no seu tamanho e no número de funcionalidades que cada uma oferece. A versão *Messenger* pode ser considerada como a versão principal por esta apresentar todas as funcionalidades existentes, nomeadamente: partilhar mensagens de texto, vídeos, fotos, mensagens de voz, *stickers*, e *GIFs*; efetuar chamadas de voz e vídeo; aplicar filtros, máscaras e efeitos nas chamadas de vídeo; entrar em contacto com pessoas com ou sem conta do *Facebook*; visualizar quando pessoas estão *online*, e ver quando uma mensagem foi lida; criar *stories* e ver outras; personalizar os *chats*; participar em jogos com amigos; efetuar pagamentos ao adicionar um cartão de débito ou conta de *Paypal*; e partilhar a localização do dispositivo cliente. [35]

[19] Existe também uma funcionalidade opcional para efetuar mensagens em modo confidencial, “*Secret Conversations*”. Aqui os utilizadores podem estabelecer uma comunicação encriptada de ponto-a-ponto utilizando o protocolo *Signal* para tal. [18] Ao contrário do *WhatsApp*, o *Facebook Messenger* guarda o histórico das conversas tanto nos servidores do *Facebook* como nos dispositivos-cliente. Segundo a política de dados do *Facebook*, e na qual esta aplica-se tanto ao *Facebook* como ao *Instagram* e ao *Messenger*, todos os dados recolhidos nestas aplicações são armazenados nos servidores e tratados de acordo com o Regulamento Geral da Proteção de Dados. [13] Todos os dados nesta aplicação podem ser encontrados na pasta “*Android/data/data/com.facebook.orca*” nos dispositivos *Android*.

3) *WhatsApp versus Facebook Messenger*: Em termos de popularidade mundial, o *WhatsApp* apresenta ser a aplicação no topo destacando-se com um número de utilizadores de cerca de 1500 milhões, estando depois em segundo lugar o *Facebook Messenger* com o número perto de 1300 milhões (ver Figura 5).

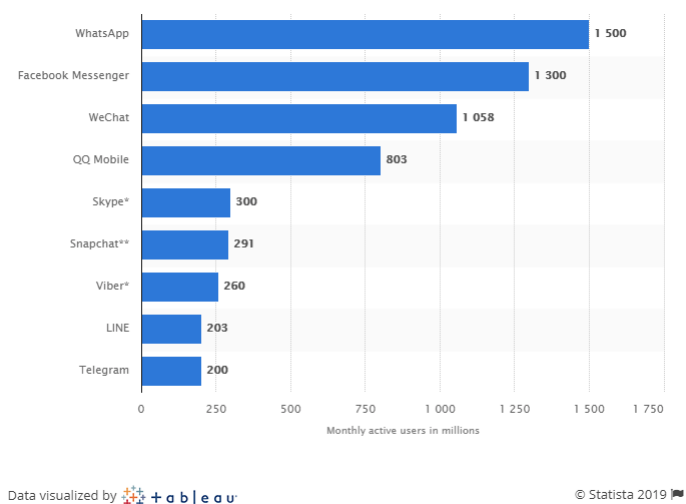


Figura 5. Ranking das aplicações de mensagens mais utilizadas [28]

Em termos de funcionalidades o *Messenger* apresenta mais funcionalidades ao comparar com o *WhatsApp* e apesar disto não é o *Messenger* que apresenta ter mais utilizadores. Existem vários motivos que elevaram a popularidade do *WhatsApp* acima do *Messenger*, nomeadamente: o suporte do *WhatsApp* para várias plataformas; o facto de as chamadas de voz utilizando sinais fracos de *WiFi* funcionavam melhor no *WhatsApp* do que no *Messenger*; e outros aspetos que o *Messenger* não tinha mas, que o *WhatsApp* tinha anteriormente, como o facto de não ser necessário a criação de uma conta de utilizador. [39] Mas, o que diferencia o *WhatsApp* do *Messenger* pode ser resumido em dois motivos: o facto do *WhatsApp* mostrar ser uma aplicação mais amigável a utilizadores novos, por exemplo para usar o *WhatsApp* não é necessário criar uma conta de utilizador; e a segurança e confidencialidade nas comunicações é estabelecido com

a encriptação de ponto-a-ponto de forma não opcional. O facto da privacidade dos dados no *WhatsApp* também é um fator importante, ao contrário do *Messenger*, o *WhatsApp* não guarda nenhum histórico das conversas de forma permanente. [20]

### III. CENÁRIO PRÁTICO-LABORATORIAL

#### A. Descrição do dispositivo e cenário

No contexto deste trabalho, foi utilizado um equipamento com as seguintes características:

- Marca: Xiaomi
- Modelo: Mi 5
- Nome de código: gemin
- SoC: Snapdragon 820
- Arquitetura: arm64
- RAM: 3GB

Este dispositivo já tinha permissões *root* e a *recovery TWRP* instalada, pelo que apenas se procedeu à instalação da última versão do *Lineage OS* disponível sendo essa a 16.0, baseada no *Android 9 (Pie)*. Também se procedeu à instalação do pacote *Aroma* do *OpenGapps* e à última versão da *Magisk Framework* e *Magisk Manager*.

Para além destas configurações, o telemóvel também terá as aplicações *WhatsApp* e *Facebook Messenger*, de forma a que possam ser trocadas e apagadas mensagens. Isto porque, para o cenário do trabalho será alvo a obtenção das mensagens trocadas com as aplicações mencionadas e tentar-se-á aceder também a mensagens previamente apagadas. É de mencionar que as mensagens serão apagadas normalmente (o utilizador apaga a mensagem da sua conversa), serão apagadas com recurso à eliminação da mensagem para todos os utilizadores envolvidos na conversa e será apagada uma conversação. Relativamente ao caso do *Facebook Messenger*, também será analisada a nova funcionalidade do mesmo, as *secret conversations*, de forma a determinar se é possível obter mensagens trocadas com esta funcionalidade e, caso elas sejam apagadas, se é possível obtê-las.

Relativamente ao processo de análise do dispositivo escolhido para este processo, decidiu-se efetuar a aquisição lógica. Esta escolha deveu-se ao facto de o seu processo ser mais automatizado e permitir obter mais informação comparado com a manual, além de que não requer equipamento especializado como a aquisição física. Tendo isto em conta, assim como a pesquisa efetuada relativa a métodos de aquisição lógica, optou-se por utilizar três métodos diferentes. Um destes métodos requer a utilização de equipamento e *software* próprio do *XRY* para efetuar a extração. Já nos restantes métodos, estes requerem uma ligação através da ferramenta *ADB*, sendo que esta faz parte do próprio *Android SDK*. Um dos métodos recorre ao comando "*adb backup -apk -shared -all -f <path\_to\_file>/backup.ab*" (subsecção III-B), enquanto que o outro subsecção III-C) requer a instalação de uma aplicação

específica denominada de *busybox*. Cada um destes processos tem os seus prós e contras, como se verá.

Também se tentará efetuar uma aquisição física com recurso ao *XRY*, como se verá na subsecção III-D.

Após as aquisições serão analisadas as imagens utilizando o *Autopsy* e o *XAMN*, de forma a determinar se é possível ou não aceder às mensagens das aplicações, incluindo as mensagens previamente apagadas.

É importante mencionar que as análises e aquisições terão em consideração a existência ou não de privilégios *root*.

#### B. Aquisição utilizando o Android SDK

O método relativo ao *Android SDK* pode ser executado num *smartphone Android* com o *Android Debugging Bridge* (versão 1.0.40) ativo, sem ser necessário ter privilégios *root*. Para se poder explorar todos os dados guardados neste *backup*, é necessário alterar o *header* do ficheiro para se conseguir extrair o mesmo. Assim, o comando "*( printf '\x1f\x8b\x08\x00\x00\x00\x00\x00'; tail -c +25 backup.ab ) | tar xfvz*" irá alterar o *header* do ficheiro de forma a que a ferramenta *Tar* consiga extrair o conteúdo do mesmo. [43] No entanto, o *backup* não possui informação sensível de acesso direto, nomeadamente, informação relativa a aplicações, como no caso do *WhatsApp* que apresenta as suas bases de dados encriptadas. Através deste comando foi possível recuperar as bases de dados encriptadas do *WhatsApp*, que estão na pasta */share/0/WhatsApp/Databases*.

#### C. Aquisição utilizando o busybox

Recorrendo à aplicação *busybox*, na versão 1.30.1, já é necessário ter privilégios *root*, sendo por isso mais difícil a sua utilização na maioria dos dispositivos. A *busybox* é uma *suite* de aplicações *Unix* adaptadas ao *Android*. O processo começava por instalar o *APK* da aplicação *busybox*, via *ADB*, com o comando "*adb -d install BusyBox.apk*". Após instalado, iniciava-se uma *ADB shell*, muda-se de utilizador para *root*, através do comando "*su*", e executa-se o comando "*mount*", de modo a listar as partições e respetivos *mountpoints*. Com esta informação é possível identificar qual o *id* do dispositivo onde se encontra a informação a recolher. Neste caso, recolheu-se toda a informação contida na partição */data*, incluindo subpartições, e o *id* associado a essa partição era */dev/block/bootdevice/by-name/userdata*. De seguida, estabeleceu-se uma ligação *TCP* via *ADB* entre o dispositivo e o computador, através do comando "*adb forward tcp:8888 tcp:8888*". Após a execução do comando, recorreu-se ao utilitário *dd* de modo a criar uma imagem desta partição e utilizou-se o *netcat* presente no *busybox* de modo a passar o ficheiro de imagem à medida que este era criado. Estes comandos foram, nomeadamente, o comando "*dd if=/dev/block/bootdevice/by-name/userdata | busybox nc -l -p 8888*" no dispositivo e o "*nc 127.0.0.1 8888 > android\_data.dd*" no computador. [14]

#### D. Aquisição utilizando o XRY

Ao correr o *software* de aquisição *XRY*, foi apresentada a opção de efetuar uma aquisição física, no entanto, para

se proceder a esta aquisição era necessário utilizar um cabo específico para dispositivos com o *SOC Snapdragon 820*. De referir que este cabo não constava no *kit* fornecido pela empresa do *software*. Após uma breve pesquisa, verificou-se que a construção deste cabo parecia ser um processo relativamente simples e por isso procedeu-se à criação do mesmo seguindo o esquema presente na Figura 6.

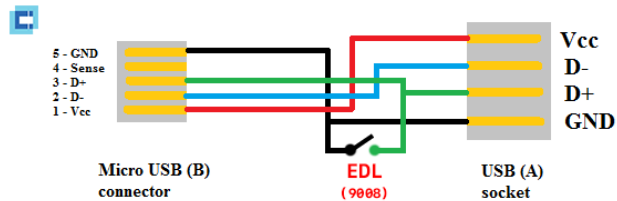


Figura 6. Esquema relativo às ligações de um cabo *EDL* [5]

Assim, para a construção do cabo foi necessário um cabo *usb type C* para *type A*, um *pushbottom switch* e um ferro de soldar.

Este cabo permite colocar o dispositivo num modo especial denominado *Emergency Download*, também conhecido por *EDL*. Este modo é normalmente utilizado para repor o dispositivo no caso de uma *ROM* ter sido mal instalada, fazendo com que o dispositivo não funcione corretamente ou até mesmo que deixe de funcionar. Neste caso, o modo *EDL* vai permitir aceder à memória interna sem a necessidade de permissões adicionais, podendo assim ser feita uma aquisição completa à memória. O cabo foi testado com um voltímetro para testar a continuidade, sendo que o resultado foi o esperado, permitindo concluir que o cabo tinha sido bem construído. No entanto, apesar do cabo ter sido bem construído, quando foi posto em prática, o dispositivo não entrou no modo *EDL*. Após uma breve pesquisa, encontrou-se uma maneira alternativa de induzir o dispositivo neste modo, recorrendo ao modo *Download Mode*. Com o dispositivo desligado, carregou-se simultaneamente nos botões de ligar e de reduzir o volume, colocando assim o dispositivo neste modo. Após colocar o dispositivo no *Download Mode*, ligou-se o mesmo ao computador e executou-se comandos *fastboot*. Introduziu-se o comando "*oem edl*" e o dispositivo entrou em modo *EDL*. Contudo, mesmo tendo entrado neste modo, o *XRY* foi incapaz de efetuar uma aquisição física.

Apesar disto, efetuou-se uma aquisição lógica com o *XRY* de modo a explorar a ferramenta *XAMN* e compará-la ao *Autopsy*. É importante realçar que a aquisição lógica foi interrompida devido ao tempo que a mesma estava a demorar.

#### E. Um pouco de análise utilizando o *Autopsy* e a *ADB Shell*

A imagem criada com o *busybox* foi aberta com o *Autopsy* e foram corridos os módulos de forma independente, de modo a que a análise corresse de maneira estável, sem que ocorram erros.

Os dados relativos à versão do *Autopsy* utilizada são os seguintes:

- *Product Version: Autopsy 4.9.0 (RELEASE)*
- *Sleuth Kit Version: 4.6.3*
- *Netbeans RCP Build: 201609300101*
- *Java: 1.8.0\_181; Java HotSpot(TM) 64-Bit Server VM 25.181-b13*

1) *Resultados no Autopsy:* Graças à existência dos privilégios *root*, foi extraída toda a informação contida na partição */data*, partição esta que inclui muitas outras partições como o */sdcard*. Dentro da partição */data*, foi possível encontrar a pasta referente aos dados sensíveis de muitas aplicações, sendo o caminho */data/data*. Nesta referida pasta, para o caso do *WhatsApp*, foi possível aceder à chave privada e às bases de dados descriptadas, facilitando assim o processo de análise. Existem várias bases de dados necessárias para o bom funcionamento do *WhatsApp*, sendo algumas delas a base de dados relativa a mensagens, conversas, entre outras informações e a base de dados com informação a respeito das definições.

Também foi observado que tipo de informação o *Autopsy* permitia obter relativamente à imagem em causa. De seguida serão apresentados os resultados que aparecem na secção resultados e vistas do *Autopsy*.

Conseguiu-se obter todos os 57 contactos que o *smartphone* possuía e a informação associada a cada contacto como o nome, números de telefone/telemóvel e *e-mails* caso existam.

Relativamente a imagens, foi possível detetar 118 fotos com alguma informação *exif*. Destas 118 fotos pode-se concluir que, em relação ao modelo da câmara, 2 foram tiradas com o *Canon EOS 600D*, 4 tiradas com o *MI 5 N6P* e 111 tiradas com o *MI 5 P3XL*. Também se observou que grande parte das fotos possui a localização. A última foto, apesar de não possuir modelo ou fabricante da câmara, possui a localização. É de referir que, apesar de só terem sido detetadas 118 fotos com dados *exif*, foram adquiridas mais imagens, sendo que várias delas dizem respeito a imagens de *sites* visitados.

Foram detetados 9 vídeos, sendo que 4 deles foram retirados pelo *smartphone*, 1 era proveniente da *cache* de um *browser* e 4 tinham como fonte o *WhatsApp*.

Existem cerca de 165 áudios, sendo que a maioria era relativa a *cache* de aplicações e também apareceram aqui os 9 vídeos, pois os mesmos estavam no formato *mp4* e *3gp*. Foram somente encontradas 14 músicas com o formato *opus*.

Foram encontrados 6 documentos *pdf*, sendo que alguns deles tinham como fonte a *cache* de aplicações. No entanto, vários deles foram transferidos. Desses documentos, alguns eram documentos do *office*, nomeadamente *docx* e *pptx* de âmbito escolar.

Em relação às mensagens, é possível aceder aos *SMS* trocados, tanto os recebidos como os enviados. Para estes casos, consegue-se facilmente detectar quem envia e quem recebe (número do remetente ou nome), a mensagem em si, se a mesma foi lida e a sua data. Para este cenário não foi detectado qualquer *MMS*. Também, através da secção das comunicações, consegue-se obter algumas estatísticas associadas aos *SMS* trocados, nomeadamente, a quantidade de *SMS* trocados para



os diferentes números. Para além disso, pode-se ainda ver, associado ao número, as trocas de mensagens que houve.

Em relação aos marcadores, foram encontrados 216, todos eles do *browser Google*, sendo que estes marcadores incluem os *sites* e as pastas. É possível saber os *urls* e o domínio, a data de criação do marcador e o nome dado. No entanto, confirmouse que estes resultados não apresentam os marcadores que existem dentro das pastas que aparecem.

É também possível aceder ao histórico e aos *downloads* efectuados pelo *browser*. No caso dos *downloads*, é possível saber o *url* onde o mesmo foi efectuado, o domínio, o caminho onde foi armazenado o ficheiro bem como o nome do ficheiro e a data. Já em relação ao histórico, é possível saber o *url*, o domínio, a data de acesso e o nome da página. Também é possível visualizar as pesquisas efectuadas.

Relativamente à procura de informação na imagem da aquisição é possível também, retirar várias conclusões.

Relativamente à pasta *data* é possível observar algumas aplicações instaladas no *smartphone*, como por exemplo, aplicações nativas do sistema operativo *Android*, *Kanji Tree*, *Facebook*, *Discord* e *WhatsApp*.

Pode-se observar a base de dados *callog.db* para ter acesso às chamadas efectuadas. Esta base de dados encontra-se em *"/data/data/com.android.providers.contacts/databases"*. Pode-se observar várias informações como a duração da chamada, a data e se a mesma foi eliminada.

Também é possível encontrar o ficheiro dos marcadores, dando para analisar a hierarquia dos mesmos e todas as outras informações anteriormente referidas, sendo que também se consegue ter acesso aos marcadores que existem dentro das pastas. Estas e outras informações relativas ao *browser* podem ser encontradas em *"/data/data/com.android.chrome/app\_chrome/Default"*.

Em relação aos perfis é possível, utilizando a *ADB Shell*, correr o comando *"adb shell pm list users"*, o que permite obter a lista de perfis do *smartphone*. Para este caso, o resultado do comando foi *"UserInfo0:Tiago:13 running"*. Também no *Autopsy* é possível confirmar este resultado, visualizando o conteúdo da pasta */data/user*, podendo-se observar que só existe o perfil com o *userId* 0. [11]

2) *Mais resultados com a ADB Shell*: Utilizando a *ADB shell* é possível ter acesso a várias informações. Um exemplo é o comando *"adb shell getprop"* que permite ter acesso a várias propriedades do dispositivo, permitindo saber, por exemplo, a versão do sistema operativo, o *IMEI* e a versão da *API*. [42] Já o comando *"adb shell pm list packages"* permite observar todas as *packages* associadas ao dispositivo. Também dá para filtrar as *packages* por utilizador, opção *"-user userId"*. Para este caso, sendo que só existe o utilizador com *id* 0, o resultado é o mesmo. A partir da lista obtida pode-se obter a lista de aplicações instaladas no dispositivo. O comando *"adb shell dumpsys meminfo"* permite visualizar processos que estejam a correr. Podem ser processos persistentes, processos visíveis, processos perceptíveis e em *cache*. Isto pode ser útil para tentar identificar *malware* e processos escondidos.

## F. Um pouco do XAMN

O *XAMN* é uma ferramenta de análise que permite analisar aquisições de forma a tentar encontrar evidências de algum incidente. As informações associadas à versão desta ferramenta são as seguintes:

- *XAMN* versão 4.1.0 (64 bit)
- *XryCore* 1.0.101
- *Build* 20.52.7

A ferramenta apresenta uma componente visual simples, bem organizada e intuitiva. É possível, de forma fácil, observar as estatísticas do que a ferramenta conseguiu adquirir na aquisição efectuada com o *XRY Extract*. Assim, para este caso, é possível visualizar pela Figura 7 que só se encontrou 4 aplicações no dispositivo e encontrou-se vários documentos, imagens e itens que não conseguiu reconhecer. Também nesta secção é possível obter um breve resumo sobre a aquisição, sobre o estado do *smartphone* e os registos de *logs*. Assim, pode-se obter rapidamente uma breve análise em relação ao dispositivo sobre investigação, como se pode ver na Figura 8.

Statistics

Category name	Number of Items	Deleted Items
Device / Event Log	0	
Device / Installed Apps	4	
Messages / Emails	3	
Web / Cookies	3	
Files / Pictures	73242	
Files / Audio	278	
Files / Videos	31	
Files / Documents	76445	
Files / Archives	408	
Files / Databases	5531	
Files / Application Binaries	1531	
Files / Unrecognized	16330	

Figura 7. Estatísticas do XAMN

Attribute	Data
Device Family	Phone
Device Name	Android Generic
SIM Status	LOADED
Network Code (from IMSI)	26801
Manufacturer	Xiaomi/Xiaomi
Model	MI 5
Revision	9/PQ3A.190505.002
Device Timezone	Europe/Lisbon
Serial Number	3f106f8b
Device Clock	04/06/2019 10:17:34 UTC
PC Clock	04/06/2019 11:17:33 UTC, GMT Daylight Time
Device Status	Bootmode = unknown
Baseband Version	msm

Figura 8. Informação geral do dispositivo

É importante referir que quase toda a informação adquirida por esta ferramenta é tratada como artefacto. Assim, uma base de dados ou o registo de aplicações instaladas são artefactos.

Existem 4 vistas rápidas, que basicamente estão pré-configuradas com alguns filtros que permitem focar em alguns tipos de informação. Por exemplo, a vista do último mês permite, de forma rápida, obter uma *timeline* dos eventos do último mês. Claro que a ferramenta permite filtrar por outros

períodos temporais, como o último dia ou semana. Também é possível escolher um período de tempo em específico.

Existe uma vista própria focada nas mensagens, chamadas e contactos, mas para este caso não é apresentada qualquer informação, possivelmente devido à aquisição ter sido interrompida antes de obter estes dados.

As vistas permitem filtrar os vários artefactos, por exemplo, por categorias. Também se pode visualizar os artefactos pela hierarquia de pastas, o que permite saber a organização interna do dispositivo.

Relativamente a uma investigação, a ferramenta possui funcionalidades semelhantes ao *Autopsy*, como a utilização de *tags*, criação de notas, entre outros. No entanto, para observar o conteúdo dos artefactos, por exemplo de um arquivo *xml* é necessário abrir o artefacto com um programa que esteja no computador. Esta é uma desvantagem, pois o *Autopsy* permite a visualização de alguns tipos de ficheiros sem a necessidade de utilização de outros programas no computador.

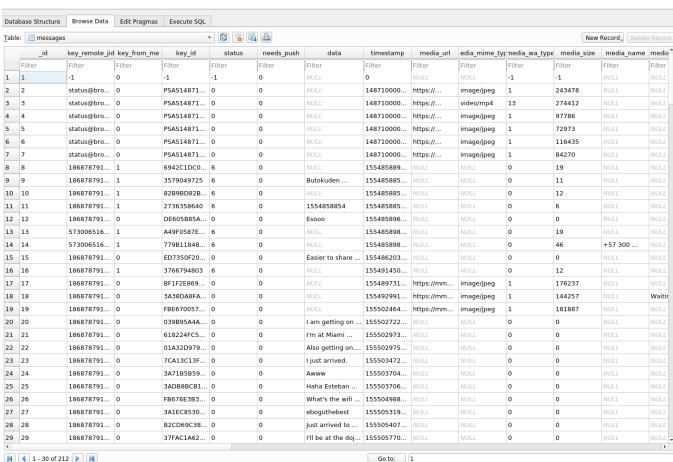
Em suma, o *XAMN* possui funcionalidades semelhantes ao *Autopsy*, no entanto, em termos de organização e ao lidar com os ficheiros, já o faz de forma diferente. Infelizmente só foi possível realizar uma comparação geral em termos de interface e funcionalidades, pois, como a aquisição lógica não foi concluída, a informação que aparece acaba por não ser relevante para o caso. Por exemplo, os contactos, as mensagens, as fotografias, os vídeos, as bases de dados do *WhatsApp* ou do *Messenger* não foram adquiridas.

### G. Obter as bases de dados descriptadas sem privilégios root

De modo a recuperar a chave num dispositivo sem privilégios *root*, utilizou-se uma ferramenta chamada "*WhatsDump*" (versão 0.2 Beta), encontrada no repositório <https://github.com/MarcoG3/WhatsDump> executando o comando "*python whatsdump.py -wa-phone +351<número\_telefone> -wa-verify sms*", onde o número de telemóvel é o número do dispositivo em causa. Esta ferramenta possui um processo de recuperação da chave do *WhatsApp*, sendo que esse processo passa por instalar o *WhatsApp* num emulador. Após a instalação, a partir do dispositivo original (através de uma ligação *ADB*), são copiadas as bases de dados guardadas de modo a se conseguir obter o *recovery token*. Depois, a ferramenta solicita o número de telefone e, quando este é inserido, é recebida uma mensagem no dispositivo original com um *security token*. Através destes *tokens* e do número de telefone, é gerada a chave e exportada para o computador, podendo assim proceder-se à descriptação das bases de dados. [36]

Para a descriptação das bases de dados, recorreu-se à ferramenta "*WhatsApp Crypt12 Database Decrypter*" encontrada no repositório <https://github.com/EliteAndroidApps/WhatsApp-Crypt12-Decrypter>. Foi executado o comando "*python decrypt12.py key msgstore.db.crypt12 msgstore.db*", sendo que após especificar a chave e a base de dados encriptada, é gerada a base de dados descriptada, podendo ser visualizada com

um *browser SQLite*, sendo que para este trabalho foi utilizada a aplicação *DB Browser for SQLite*. A Figura 9 apresenta parte da base de dados que tem informações relativas, por exemplo, a mensagens.



id	key_remote_jid	key_from_me	key_id	status	needs_push	data	timestamp	media_url	media_mime_type	media_wa_type	media_size	media_name	media
1	4	0	1	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
2	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
3	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
4	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
5	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
6	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
7	status@bns...	0	PSAS14871...	0	0	NULL	148710000	https://	image/jpeg	1	243478	NULL	NULL
8	186878791...	1	6842C1D0C...	6	0	NULL	155485889...	NULL	NULL	0	19	NULL	NULL
9	186878791...	1	3579049725	6	0	Bookkuden...	155485889...	NULL	NULL	0	11	NULL	NULL
10	186878791...	1	828808D2B...	6	0	NULL	155485889...	NULL	NULL	0	12	NULL	NULL
11	186878791...	1	2736538D0...	6	0	155485889...	155485889...	NULL	NULL	0	6	NULL	NULL
12	186878791...	0	D84058D4...	0	0	Exame	155485889...	NULL	NULL	0	0	NULL	NULL
13	573005316...	1	4A9F0587E...	6	0	NULL	155485889...	NULL	NULL	0	19	NULL	NULL
14	573005316...	1	770811848...	6	0	NULL	155485889...	NULL	NULL	0	46	+57 300...	NULL
15	186878791...	0	ED730F02...	0	0	Easier to share	155486203...	NULL	NULL	0	0	NULL	NULL
16	186878791...	1	3766794803	6	0	NULL	155491450...	NULL	NULL	0	12	NULL	NULL
17	186878791...	0	897325861...	0	0	NULL	155499731...	https://mem...	image/jpeg	1	176237	NULL	NULL
18	186878791...	0	3A3D0A8FA...	0	0	NULL	155492991...	https://mem...	image/jpeg	1	146257	NULL	WhatsApp
19	186878791...	0	F8E67D057...	0	0	NULL	155502464...	https://mem...	image/jpeg	1	181887	NULL	NULL
20	186878791...	0	038085AA...	0	0	I am getting on	155502722...	NULL	NULL	0	0	NULL	NULL
21	186878791...	0	648224FC...	0	0	I'm at Miami	155502973...	NULL	NULL	0	0	NULL	NULL
22	186878791...	0	01A32D97E...	0	0	Also getting on	155502975...	NULL	NULL	0	0	NULL	NULL
23	186878791...	0	7CA13C13F...	0	0	I just arrived	155503472...	NULL	NULL	0	0	NULL	NULL
24	186878791...	0	3A718B59...	0	0	Awake	155503704...	NULL	NULL	0	0	NULL	NULL
25	186878791...	0	3A8B8CB3...	0	0	Haha Esteen	155503706...	NULL	NULL	0	0	NULL	NULL
26	186878791...	0	F8076E383...	0	0	What's the wif	155504083...	NULL	NULL	0	0	NULL	NULL
27	186878791...	0	3A1E0520...	0	0	etiquetabue	155505219...	NULL	NULL	0	0	NULL	NULL
28	186878791...	0	82C056C8...	0	0	Just arrived to	155505407...	NULL	NULL	0	0	NULL	NULL
29	186878791...	0	37AC1A62...	0	0	I'll be at the do	155505770...	NULL	NULL	0	0	NULL	NULL

Figura 9. Mensagens do *WhatsApp* guardadas na base de dados descriptada.

Das bases de dados encontradas na pasta */data/data/com.whatsapp/databases*, a mais interessante do ponto de vista forense, aparenta ser a *msgstore.db*, visto que contém as mensagens, conversas e contactos. Esta base de dados contém múltiplas tabelas, sendo que as que contém mais informação são a *messages*, que contém as mensagens relativas a todas as conversas e onde consta a mensagem, o destinatário ou remetente; a *call\_log*, que contém o registo das chamadas efetuadas, para quem foi feita ou de quem foi recebida a chamada, onde indica se foi vídeo-chamada ou não e a duração da chamada; a *chat\_list*, onde é possível ver o nome da conversa assim como o número da pessoa envolvida na conversa; a *jid* onde é possível visualizar todos os números de pessoas que tenham participado em conversas com o utilizador. [4]

### H. Mensagens eliminadas no WhatsApp

De modo a analisar o comportamento de remoção de mensagens, procedeu-se à eliminação de uma conversa. Após analisar a base de dados do *WhatsApp*, mais precisamente o ficheiro *msgstore.db*, foram encontradas as mensagens que tinham sido supostamente eliminadas. Isto aparenta acontecer devido à natureza das bases de dados em formato *SQLite*, onde a eliminação de linhas faz parte de um serviço do *Android* associado à limpeza de bases de dados deste tipo, processo este chamado *Vacuuming*. É de referir que, apesar de neste caso terem aparecido as mensagens à mesma na tabela das mensagens, como se estas não tivessem sido apagadas, isto poderá não se verificar sempre. O facto de as mensagens não aparecerem não quer dizer que as bases de dados tenham sofrido o processo de "limpeza". Assim, de modo a se tentar recuperar tais mensagens apagadas, poderá ter que se recorrer a ferramentas especializadas em recuperar dados apagados de

bases de dados *SQLite*, tais como a *undark* [9] ou a *SQLite-Deleted-Records-Parser* [10].

Já para mensagens eliminadas por parte do remetente para todos os participantes da conversa, foi possível detectar que a mensagem era removida da base de dados, mas a entrada continuava lá com o campo da mensagem a *null*. Ao empregar as mesmas técnicas de recuperação de dados, não foi possível obter as mensagens, no entanto, isto poderá variar de caso para caso, sendo assim recomendável correr estas ferramentas.

Conclui-se então que poderá ser possível recuperar mensagens apagadas no *WhatsApp*, sendo que tudo dependerá se o serviço de "limpeza" tenha corrido ou não para aquela base de dados.

O *WhatsApp* faz também um registo de todas as ações executadas num ficheiro de *log*, sendo que este ficheiro encontra-se na pasta */data/data/com.whatsapp/Files* e apenas está acessível com privilégios *root*. Dentro deste *log* é possível encontrar a ação de apagar uma mensagem, como se pode observar na Figura 10, onde o texto a vermelho corresponde ao número de telemóvel associado à conversa, e a verde o texto relacionado com o serviço de remoção de mensagens.

```
[1022:WhatsApp Worker #17] msgstore/deletemsgs/service/jid
35192678XXXX@s.whatsapp.net
[1022:WhatsApp Worker #17] msgstore/lastmsgid/count 1
[1022:WhatsApp Worker #17] msgstore/deletemsgs/mark
jid:35192678XXXX@s.whatsapp.net lastDeletedMessageId:208
lastDeletedStarredMessageId:208
[1022:WhatsApp Worker #17] conversation-delete-service/start-service
[2:main] badger/getbadger -1
[1107:update_widget] widgetprovider/updatebadgecount:0
[1053:Notifications] messagenotification/updateOnly skip
[1107:update_widget] widgetprovider/updatebadgecount:0
[2:main] conversation-delete-service/onCreate
[2:main] conversation-delete-service/startcommand intent=Intent
{ act=action_delete cmp=com.whatsapp/.data.ConversationDeleteService (has
extras) }
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
conversation-delete-service/handleintent intent=Intent { act=action_delete
cmp=com.whatsapp/.data.ConversationDeleteService (has extras) }
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/countmessagesstodelete/count: 4
[1029:WhatsApp Worker #21] msgstore/countmessagesstodelete/count: 4
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deletemedia/batch/files/timer/stop: 0
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deletemedia/batch/files 35192678XXXX@s.whatsapp.net
deleteFiles:true timeSpent:0
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deleteMessageThumbnailsFor-jid/0
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deletemsgs/batches/timer/stop: 54
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deletemsgs/batches 35192678XXXX@s.whatsapp.net
haveMessagesToDelete:false timeSpent:54 currentMessages:4 totalMessages:4
[1108:IntentService[com.whatsapp.data.ConversationDeleteService]]
msgstore/deletemsgs/unmark jid:35192678XXXX@s.whatsapp.net
```

Figura 10. Linhas correspondentes ao serviço de remoção de mensagens do *WhatsApp*

Foi feita também um procura por ficheiros de áudio e de vídeo que pudessem corresponder às chamadas de áudio e de vídeo gravadas pelo *WhatsApp*, no entanto, não foram encontrados indícios da existência destes ficheiros.

## I. Análise ao Facebook Messenger

Também foi efetuada uma análise às bases de dados do *Facebook Messenger*, encontradas na pasta */data/data/com.facebook.orca/databases*. Estas bases de dados

também estavam em formato *SQLite* e foram analisadas com a mesma ferramenta utilizada para as do *WhatsApp*. Destas bases de dados, a que continha informação mais relevante era a *threads\_db2* onde estavam presentes todas as mensagens relativas às conversas do *Facebook Messenger*. Nesta base de dados, existem várias tabelas, mas as que aparentam ter informação mais relevante é uma tabela com as mensagens, uma com as reações às mensagens, uma com as conversas e uma com os participantes das conversas.

id	msg_id	thread_key	text	sender	not_forwardable	attachments	shares	sender_id	msg_type	affected_users
55	13427	mid...	ONE_TO_ON... alis, meto...	["user_key"...]	0	155976024...	NULL	NULL	0	NULL
56	13426	mid...	ONE_TO_ON... comprei por...	["user_key"...]	0	155976022...	NULL	NULL	0	NULL
57	13425	mid...	ONE_TO_ON... mas vou ...	["user_key"...]	0	155976020...	NULL	NULL	0	NULL
58	13424	mid...	ONE_TO_ON... se ganhar ...	["user_key"...]	0	155976020...	NULL	NULL	0	NULL
59	13423	mid...	ONE_TO_ON... isso	["user_key"...]	0	155976017...	NULL	NULL	0	NULL
60	13422	mid...	ONE_TO_ON... mas pronto...	["user_key"...]	0	155976017...	NULL	NULL	0	NULL
61	13421	mid...	ONE_TO_ON... Verde enia ...	["user_key"...]	0	155976017...	155976017...	NULL	0	NULL
62	13418	mid...	ONE_TO_ON... acho que vo...	["user_key"...]	0	155976015...	NULL	NULL	0	NULL
63	13417	mid...	ONE_TO_ON... hmmm	["user_key"...]	0	155976014...	NULL	NULL	0	NULL
64	13416	mid...	ONE_TO_ON... Put	["user_key"...]	0	155976013...	155976013...	NULL	0	NULL
65	13413	mid...	ONE_TO_ON... mas é pena...	["user_key"...]	0	155976012...	NULL	NULL	0	NULL
66	13412	mid...	GROUP... https://...	["user_key"...]	0	155976011...	NULL	NULL	0	NULL
67	13411	mid...	GROUP... N	["user_key"...]	0	155976008...	NULL	NULL	0	NULL
68	13410	mid...	GROUP... https://...	["user_key"...]	0	155976007...	NULL	NULL	0	NULL
69	13409	mid...	ONE_TO_ON... I have a few...	["user_key"...]	0	155976787...	NULL	NULL	0	NULL
70	13407	mid...	ONE_TO_ON... Irem	["user_key"...]	0	155976784...	NULL	NULL	0	NULL
71	13406	mid...	ONE_TO_ON... M3 Real GB...	["user_key"...]	0	155976780...	155976780...	NULL	0	NULL
72	13404	mid...	ONE_TO_ON... vou ver	["user_key"...]	0	155976777...	NULL	NULL	0	NULL
73	13402	mid...	ONE_TO_ON... Sorry mand...	["user_key"...]	0	155976761...	155976761...	NULL	0	NULL
74	13400	mid...	ONE_TO_ON... https://...	["user_key"...]	0	155976760...	155976760...	["fbid":null]	0	NULL
75	13393	mid...	ONE_TO_ON...	["user_key"...]	0	155976758...	155976758...	NULL	0	NULL
76	13394	mid...	ONE_TO_ON... N sei, este...	["user_key"...]	0	155976755...	155976755...	NULL	0	NULL
77	13391	mid...	ONE_TO_ON... pois mas ...	["user_key"...]	0	155976752...	NULL	NULL	0	NULL
78	13390	mid...	ONE_TO_ON... Mas tem qu...	["user_key"...]	0	155976752...	155976752...	NULL	0	NULL
79	13387	mid...	ONE_TO_ON... Tenho o ...	["user_key"...]	0	155976748...	155976748...	NULL	0	NULL
80	13384	mid...	ONE_TO_ON... Eu até to ...	["user_key"...]	0	155976745...	155976745...	NULL	0	NULL
81	13381	mid...	GRP TO ON... Só teendo	["user_key"...]	0	155976743...	155976743...	NULL	0	NULL

Figura 11. Mensagens do *Facebook* guardadas na base de dados.

Para o cenário do *Facebook Messenger*, foram trocadas mensagens em conversas de grupo, individuais e também se utilizou uma das novas funcionalidades, a *secret conversations*, como já foi referido anteriormente. As mensagens trocadas em grupo e individualmente foram facilmente identificáveis na base de dados, no entanto, não se conseguiu obter nenhuma informação referente às *secret conversations* em nenhuma base de dados. Quanto a conversas eliminadas, as mensagens referentes à conversa e a própria conversa não se encontravam na base de dados, não apresentando sequer as entradas.

Recorrendo às ferramentas de recuperação de *SQLite* utilizadas previamente para o *WhatsApp*, tentou-se recuperar as mensagens, sendo que também não foram encontradas nenhuma entradas correspondentes às linhas apagadas. Assim, uma outra forma de tentar obter a informação relativa às conversas era obtendo autorizações especiais, de forma a que o *Facebook Messenger* desse acesso às mesmas. Isto é possível pois o *Messenger* guarda a informação na *cloud*, ao contrário do *WhatsApp*.

## J. Conclusão

Com este trabalho foi possível ter uma primeira experiência no contexto da análise forense digital a um *smartphone* com o sistema operativo *Android*, versão 9. Assim, conseguiu-se perceber de que forma é que um sistema *Android* está organizado e conhecer algumas das pastas e partições que poderão guardar informação sensível, sendo que, no âmbito de uma investigação digital, esta informação tem o potencial de conter informação indicativa de algum crime.

Houve a tentativa de obter 4 aquisições, três lógicas com recurso a técnicas e *software* diferentes e uma física. A física

não foi possível de se efectuar, pois o *smartphone* não entrava no modo necessário para proceder à aquisição dos dados. Relativamente às lógicas, todas elas foram efectuadas, sendo uma através de um *backup* do *smartphone* (não possuindo privilégios *root*), a outra aquisição efectuada com recurso à *busybox* (possuindo privilégios *root*), e a última com recurso ao *software* e equipamento de extração de dados do *XRY*, mas esta extração foi cancelada devido ao tempo que estava a demorar.

Como o cenário proposto para este trabalho era com foco nas aplicações de mensagens instantâneas, nomeadamente o *WhatsApp* e o *Facebook Messenger*, foi possível compreender de que forma as mesmas funcionam e que artefactos elas criam que poderão ser importantes no contexto de uma investigação digital. Assim, foi possível localizar onde se encontram as bases de dados destas aplicações e, também, o que é necessário para aceder às mesmas.

Por exemplo, no caso do *WhatsApp* é possível aceder facilmente às bases de dados e, consequentemente, às mensagens trocadas, se se tiver privilégios *root*. Se esses privilégios não existirem, poder-se-á fazer uso de ferramentas de forma a conseguir obter acesso à chave de encriptação, para posteriormente decifrar as bases de dados, processo este realizado no presente trabalho. Assim, o *WhatsApp* possui bases de dados para o mesmo, em localizações diferentes, sendo que em uma localização elas se encontram descriptadas, localização essa que só fica acessível quando se tem privilégios *root*. A existência destas bases de dados serve para o acesso rápido aos dados que possuem. A outra localização possui as bases de dados cifradas, sendo necessária a chave de encriptação para se compreender o conteúdo das mesmas, sendo que neste caso, as bases de dados são utilizadas como *backup* do dados. Relativamente às chamadas de vídeo e voz, não se conseguiu obter qualquer informação das mesmas, com excepção de um registo sobre as chamadas, tanto de voz ou vídeo, efectuadas no dispositivo analisado.

Já no caso do *Facebook Messenger*, as bases de dados encontram-se na mesma localização que as bases de dados que não são cifradas no *WhatsApp*, sendo que é necessário privilégios *root* para aceder às mesmas. No entanto, sem privilégios não é possível aceder a essas bases de dados, sendo que o *Messenger* guarda essa informação na *cloud*. Assim, seria necessário ter autorizações especiais para se tentar obter a informação quando não se tem privilégios *root*. Relativamente às *secret conversations*, uma funcionalidade nova do *Messenger*, apesar de a mesma ter sido testada e se ter apagado mensagens nessa conversa, não foi possível encontrar nem obter qualquer informação sobre essas mensagens.

Para além destas descobertas, foi possível perceber que relativamente às mensagens previamente eliminadas, o acesso às mesmas irá depender de caso para caso. Como as bases de dados em questão eram base de dados *SQLite*, se o mecanismo de limpeza não tiver ocorrido poderá ser possível aceder facilmente às mensagens eliminadas, algo que ocorreu neste trabalho. Por outro lado, se não se encontrar as mensagens na base de dados, não implica que o mecanismo de limpeza

ocorreu e por isso devem ser utilizadas ferramentas que permitam recuperar dados eliminados nessas bases de dados *SQLite*. Este processo também foi executado no trabalho, no entanto, não foi possível efectivamente recuperar as mensagens apagadas, quando eliminadas no *Messenger*.

Também foi utilizado o *Autopsy* e o *XAMN* de forma a compreender que tipo de informação as ferramentas que permitem uma análise automatizada apresentavam. Assim, percebeu-se que ambas as ferramentas possibilitam encontrar facilmente diversa informação como os contactos, os *SMS* trocados, as chamadas realizadas, as fotografias, entre outros. Para além disso, as ferramentas também possuem recursos interessantes como a *timeline*, o que faz com que permitam uma análise mais intuitiva, rápida e interativa.

Considera-se que os objectivos deste trabalho foram cumpridos, sendo que o mesmo foi uma mais valia para a compreensão dos processos de aquisição e análise relativamente a dispositivos *Android*.

## REFERÊNCIAS

- [1] Android. Android debug bridge. <https://developer.android.com/studio/command-line/adb>. Visualizado: 2019.04.05.
- [2] Android. Distribution dashboard. [https://developer.android.com/about/dashboards?fbclid=IwAR3Ewcja2d80p77\\_okz\\_gD4G3OnTFkLG1kV39-Vq40JnC3wEhQZjYb997HIQ](https://developer.android.com/about/dashboards?fbclid=IwAR3Ewcja2d80p77_okz_gD4G3OnTFkLG1kV39-Vq40JnC3wEhQZjYb997HIQ).
- [3] Android. Dns-over-tls. <https://developers.google.com/speed/public-dns/docs/dns-over-tls>. Data original: 2019.01.09.
- [4] Cosimo Anglano. Forensic analysis of whatsapp messenger on android smartphones. <https://arxiv.org/pdf/1507.07739.pdf>. Visualizado: 2019.05.18.
- [5] Atom. Guides - how to make qualcomm edl cable (9008 in com 9008) xiaomi, lg etc. <http://forum.gsmdevelopers.com/hardware-cable-modification-tricks/10684-guides-qualcomm-edl-cable-9008-com-9008-xiaomi-lg-etc.html>. Visualizado: 2019.05.16.
- [6] Nurul Hidayah Ab; Glisson William Bradley; Choo Kim-Kwang Raymond Cahyani, Niken Dwi Wahyu e Rahman. *The Role of Mobile Forensics in Terrorism Investigations Involving the Use of Cloud Storage Service and Communication Apps*. Springer Science + Business Media New York, 2016.
- [7] John Callaham. The history of android os: its name, origin and more. <https://www.androidauthority.com/history-android-os-name-789433/>. Data original: 2018.07.03.
- [8] Corelis. What is jtag? <https://www.corelis.com/education/tutorials/jtag-tutorial/what-is-jtag/>. Visualizado: 2019.04.05.
- [9] Paul L. Daniels. Undark - a sqlite deleted and corrupted data recovery tool. <https://pldaniels.com/undark/>. Visualizado: 2019.05.18.
- [10] Mari DeGrazia. Sqlite-parser. <https://github.com/mdegrazia/SQLite-Deleted-Records-Parser>. Visualizado: 2019.05.18.
- [11] Android Developers. Work profiles. <https://developer.android.com/work/managed-profiles>. Visualizado: 2019.05.17.
- [12] Heather Mahalik e Satish Bommisetty e Rohit Tamma. *Practical Mobile Forensics*. Packt Publishing Ltd, 2016.
- [13] Facebook. Política de dados. <https://www.facebook.com/policy.php>. Visualizado: 2019.03.28.
- [14] Andrea Fortuna. Android forensics: imaging android filesystem using adb and dd. <https://www.andreafortuna.org/2018/12/03/android-forensics-imaging-android-file-system-using-adb-and-dd/>. Visualizado: 2019.05.14.
- [15] Baltazar Frade, Miguel e Rodrigues. Mobile devices (mcif - digital forensic analysis 2). Publicado: Leiria(Instituto Politécnico de Leiria) 2019.
- [16] StatCounter GlobalStats. Desktop vs mobile vs tablet market share worldwide - march 2019. <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201710-201903>. Visualizado: 2019.04.03.

- [17] StatCounter GlobalStats. Mobile operating system market share worldwide - march 2019. <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201701-201903>. Visualizado: 2019.04.03.
- [18] Andy Greenberg. You can all finally encrypt facebook messenger, so do it. <https://www.wired.com/2016/10/facebook-completely-encrypted-messenger-update-now/>. Visualizado: 2019.03.27.
- [19] Mark Hendrickson. Facebook chat launches, for some. [https://techcrunch.com/2008/04/06/facebook-chat-enters-pre-release-beta/?guccounter=1&guce\\\_referrer\\\_us=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce\\\_referrer\\\_cs=bBJKWAn7G4q-MztZfc6oYQ](https://techcrunch.com/2008/04/06/facebook-chat-enters-pre-release-beta/?guccounter=1&guce\referrer\_us=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce\_referrer\_cs=bBJKWAn7G4q-MztZfc6oYQ). Visualizado: 2019.03.27.
- [20] Simon Hill. The best text messaging apps for android and ios. <https://www.digitaltrends.com/mobile/best-text-messaging-apps/>. Visualizado: 2019.03.08.
- [21] The Khronos Group Inc. Vulkan. <https://www.khronos.org/vulkan/>.
- [22] WhatsApp Inc. About whatsapp. <https://www.whatsapp.com/about/>. Visualizado: 2019.03.06.
- [23] WhatsApp Inc. Making voice calls. <https://faq.whatsapp.com/en/android/28000016/?category=5245237>. Visualizado: 2019.03.06.
- [24] WhatsApp Inc. Saving your chat history. <https://faq.whatsapp.com/en/android/23756533/?category=5245251>. Visualizado: 2019.03.06.
- [25] WhatsApp Inc. Sending media, documents, location and contacts. <https://faq.whatsapp.com/en/android/23112542/?category=5245251>. Visualizado: 2019.03.06.
- [26] WhatsApp Inc. Whatsapp encryption overview. <https://www.whatsapp.com/security/>. Visualizado: 2019.03.06.
- [27] WhatsApp Inc. Whatsapp features. <https://www.whatsapp.com/features/>. Visualizado: 2019.03.06.
- [28] Mansoor Iqbal. Whatsapp revenue and usage statistics (2019). <http://www.businessofapps.com/data/whatsapp-statistics/>. Visualizado: 2019.03.08.
- [29] Aishwarya Jagtap. What is the protocol used by whatsapp? <https://www.quora.com/What-is-the-protocol-used-by-WhatsApp>. Visualizado: 2019.03.06.
- [30] Gary C. Kessler Jeff Lessard. Android forensics: Simplifying cell phone examinations. *Small Scale Digital Device Forensic Journal*, pages 1–12, 2010.
- [31] Jon Knight. 12 important privacy & security features google added to android 9.0 pie. <https://android.gadgethacks.com/news/12-important-privacy-security-features-google-added-android-9-0-pie-0184332/>. Visualizado: 2019.08.30.
- [32] knolleary. Mqtt used by facebook messenger. <https://mqtt.org/2011/08/mqtt-used-by-facebook-messenger>. Visualizado: 2019.03.27.
- [33] Dimitar Kostadinov. Introduction: Importance of mobile forensics. website accessed on: 2019.03.11. <https://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/the-mobile-forensics-process-steps-types/?fbclid=IwAR2JYKoO45b5gu7ocGVpAMNMsBuMX-\nfbIXnDSIJ2SKKRmrDVeWzgoNP6hY>.
- [34] Victor Loh. Which protocol does facebook use for its messages? <https://www.quora.com/Which-protocol-does-Facebook-use-for-its-messages>. Visualizado: 2019.03.23.
- [35] Facebook Messenger. Conversations come to life on messenger. <https://www.messenger.com/features>. Visualizado: 2019.03.27.
- [36] MGP25. Re-whatsapp. <https://github.com/mgp25/RE-WhatsApp>. Visualizado: 2019.05.18.
- [37] Parmy Olsen. Exclusive: The rags-to-riches tale of how jan koum built whatsapp into facebook's new \$19 billion baby. <https://www.forbes.com/sites/parmyolson/2014/02/19/exclusive-inside-story-how-jan-koum-built-whatsapp-into-facebooks-new-19-billion-baby/#61ff24032fa1>. Visualizado: 2019.03.06.
- [38] The LineageOS Project. Install lineageos on gemini. <https://wiki.lineageos.org/devices/gemini/install>. Visualizado: 2019.04.05.
- [39] Quora. Why are people choosing whatsapp messenger over facebook messenger? what can whatsapp offer that facebook messenger does not? <https://www.quora.com/Why-are-people-choosing-WhatsApp-Messenger-over-Facebook-Messenger-What-can-WhatsApp-offer-that-Facebook-Messenger-does-not>. Visualizado: 2019.03.29.
- [40] Mishaal Rahman. Diving into sdcardfs: How google's fuse replacement will reduce i/o overhead. <https://www.xda-developers.com/diving-into-sdcardfs-how-googles-fuse-replacement-will-reduce-i-o-overhead/>. Data original: 2017.01.17.
- [41] Gaurav Rathee. Explore whatsapp clock sign, single tick, double tick. <http://digitalperiod.com/explore-whatsapp-clock-sign-and-tick/>. Visualizado: 2019.03.06.
- [42] ADB Shell. adb shell getprop. <http://adbshell.com/commands/adb-shell-getprop>. Visualizado: 2019.05.17.
- [43] Stackoverflow. How to extract or unpack an .ab file (android backup file). <https://stackoverflow.com/questions/18533567/how-to-extract-or-unpack-an-ab-file-android-backup-file>. Visualizado: 2019.05.14.
- [44] T3K-Forensics. 10 challenges in mobile forensics. <http://www.t3k-forensics.com/allgemein-en/10-main-challenges-in-mobile-forensics2/>. Visualizado: 2019.04.05.
- [45] Donnie Tamma, Rohit; Tindall. *Learning Android Forensics*. Packt Publishing Ltd, 2015.
- [46] Debian Wiki. rootfs. <https://wiki.debian.org/rootfs>. Data original: 2010.12.27.
- [47] SELinux Project Wiki. Nb lsm. [http://selinuxproject.org/page/NB\\_LSM](http://selinuxproject.org/page/NB_LSM).
- [48] Wikipedia. Android version history. [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history). Visualizado: 2019.04.03.
- [49] Wikipedia. procs. <https://en.wikipedia.org/wiki/Procs>.
- [50] Wikipedia. sysfs. <https://en.wikipedia.org/wiki/Sysfs>.
- [51] Wikipedia. tmpfs. <https://en.wikipedia.org/wiki/Tmpfs>.