FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Forenzní analýza dat mobilní aplikace
Student:	Šárka Nádvorníková
Vedoucí:	Ing. Marián Svetlík
Studijní program:	Informatika
Obor / specializace:	Informační bezpečnost 2021
Katedra:	Katedra informační bezpečnosti
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Z pohledu relevance představují digitální stopy zajištěné z mobilních zařízení (typicky mobilní telefony s OS Android nebo iOS) v dnešní době jeden z nejdůležitějších a informačně nejbohatších zdrojů. Nad ostatními pak zvláště vynikají data jednotlivých aplikací. Ačkoliv pro zajišťování a analýzu těchto dat existuje celá řada komerčních nástrojů, specializují se tyto zejména na nejrozšířenější a nejpoužívanější aplikace. Řadu potenciálně forenzně využitelných mobilních aplikací je tak nutno analyzovat manuálně.

Pro vypracování ZP:
1. Seznamte se způsobem a strukturou dat ukládaných v OS Android a iOS
2. Analyzujte a popište běžné způsoby ukládání aplikačních dat v OS Android a iOS
3. Svá zjištění demonstrujte na známé/popsané aplikaci ve vámi zvoleném mobilním OS
4. Nové poznatky použijte při vlastní analýze vybrané mobilní aplikace

Elektronicky schválil/a prof. Ing. Róbert Lórencz, CSc. dne 26. ledna 2024 v Praze.

Bakalářská práce

FORENZNÍ ANALÝZA DAT MOBILNÍ APLIKACE

Šárka Nádvorníková

Fakulta informačních technologií
Katedra informační bezpečnosti
Vedoucí: Ing. Marián Svetlík
14. května 2024

Odkaz na tuto práci: Nádvorníková Šárka. *Forenzní analýza dat mobilní aplikace.* Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Seznam obrázků

Seznam výpisů kódu

Chtěla bych poděkovat především vedoucímu práce, Ing. Mariánu Svetlíkovi, za rady a pomoc při zpracovávání bakalářské práce. Dále bych chtěla poděkovat rodině a přátelům za podporu, kterou mi v průběhu studia poskytovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 14. května 2024

Abstrakt

Obsahem této práce je rešerše metod forenzní analýzy mobilních zařízení, konkrétně mobilních telefonů s operačními systémy Android a iOS. Představuje architekturu mobilních operačních systémů, způsoby ukládání aplikačních dat, které operační systémy nabízí, a pojmy relevantní pro analýzu úložiště mobilních telefonů (šifrování paměti, zálohování, druhy eskalace privilegií). Věnuje se analýze paměti mobilních telefonů z hlediska míst, kde se ukládají aplikační data. Výsledkem je popis standardní adresářové struktury, která je dostupná aplikacím pro uložení jejich dat.

Další část práce se zabývá metodami forenzní analýzy a zaměřuje se převážně na způsoby získání dat z mobilních zařízení, jelikož tato část je nejvíce specifická právě pro toto odvětví digitální forenzní analýzy. Tyto způsoby se dělí podle náročnosti provedení a efektu na zařízení, ze kterého se data získávají. Samotná analýza získaných dat spočívá převážně v identifikaci zájmových souborů a prozkoumání jejich obsahu.

Předchozí zjištění jsou demonstrovaná na aplikaci WhatsApp. Tato aplikace byla zvolena, jelikož se z hlediska hodnoty informací jedná o velmi bohatý zdroj a mnoho prací se jí již zabývalo, tudíž je velmi dobře popsaná. Tato část je pak použita jako základ pro provedení vlastní analýzy jiné aplikace.

Pro vlastní analýzu byla zvolena aplikace Vinted. Výsledkem její analýzy je popis nalezených informací a skript, který vytvoří z nejdůležitějších dat PDF dokument.

Klíčová slova forenzní analýza mobilních telefonů, mobilní aplikace, aplikační data, Andriod, iOS, WhatsApp, Vinted

Abstract

This thesis contains research about methods of mobile forensics, specifically dealing with mobile phones running Android and iOS operating systems. It describes the architecture of said operating systems and the methods they offer for managing application data. It also presents some relevant topics such as file encryption, backup systems, and privilege escalation. Next, it provides an analysis of mobile phone storage, methods of accessing it, and locations where application data is stored. The result of this is a description of the standard file system structure that is accessible to every application for storing its data.

The next part of this thesis is dedicated to methods of mobile forensic analysis, specifically the methods of acquiring data from the devices, since this part is the most distinct from classic digital forensics. The methods are divided into groups according to the difficulty and the effect they have on the device. The stage of analyzing the data then consists of identifying relevant files and exploring their contents.

Previous findings are then demonstrated using the WhatsApp application. This application was chosen because its data is a rich source of information and is also well-described in many other papers. This part serves as a basis for the next chapter, which involves manual analysis of a different application.

For that, the chosen application is Vinted. Results of the analysis are a description of found data and a script that creates a PDF document with the most relevant information.

Keywords mobile forensics, mobile applications, app data, Android, iOS, WhatsApp, Vinted

Seznam zkratek

ADB	Android Debug Bridge
AES	Advanced encryption standard
AOSP	Android Open Source Project
APFS	Apple File System
API	Application programming interface
BSD	Berkeley Software Distribution
HFS+	Hierarchical File System
ISP	In-system programming
JSON	JavaScript Object Notation
JTAG	Joint Test Action Group
MTP	Media Transfer Protocol
NDK	Native development kit
OS	Operační systém
PDF	Portable Document Format
PIN	Personal identification number
PTP	Picture Transfer Protocol
SDK	Software development kit
SQL	Structured Query Language
UID	Uživatelský identifikátor
URL	Uniform Resource Locator
UUID	Universally unique identificator
WAL	Write-ahead log
XML	Extensible Markup Language
XTS	XEX Tweakable Block Ciphertext Stealing

Úvod

Mobilní operační systémy se vyvíjejí velmi rychle. Nové verze vycházejí každý rok, bezpečnostní aktualizace stávajících verzí i několikrát do roka. V posledních letech je kladen čím dál tím větší důraz na zabezpečení mobilních telefonů, obzvláště na zabezpečení uživatelských dat. To však ztěžuje jejich analýzu.

Tato bakalářská práce se zabývá forenzní analýzou mobilních telefonů, konkrétně pak dat mobilních aplikací. Prvním cílem práce je zanalyzovat úložiště mobilních zařízení a zjistit, kam a v jakém formátu se ukládají aplikační data. Dalším cílem je popsat metody forenzní analýzy mobilních zařízení, způsoby extrakce dat a jejich interpretace, a demonstrovat tyto poznatky na aplikaci, která je již dobře zdokumentovaná – konkrétně se jedná o aplikaci WhatsApp. Z těchto cílů pak vychází poslední cíl, a sice provést na základě poznatků z předchozích částí vlastní analýzu mobilní aplikace Vinted.

Práce začíná kapitolou „Teoretický úvod“, ve které jsou popsané mobilní operační systémy Android a iOS, jejich architektura a prostředky, které nabízejí pro ukládání aplikačních dat. Dále tato kapitola obsahuje popis některých technologií, které jsou relevantní pro analýzu dat. Konkrétně se jedná o popis způsobů šifrování paměti, způsoby zálohování dat a druhy eskalace privilegií na jednotlivých platformách. Kapitola „Analýza úložiště mobilních zařízení“ představuje způsoby komunikace mezi mobilními telefony a připojenými zařízeními (nejčastěji počítač) a definuje standardní adresářovou strukturu, která je dostupná pro každou aplikaci a kde je tedy možné nalézt její data. Forenzní analýza mobilních zařízení je specifická v technikách, které se používají pro získání dat. Tyto techniky jsou popsané v kapitole „Metody forenzní analýzy mobilních zařízení“. Tato kapitola taktéž obsahuje způsoby identifikace zájmových souborů. V kapitole „Příklad analýzy aplikace – WhatsApp“ jsou poznatky z předchozích kapitol demonstrovány na konkrétním příkladě – datech aplikace WhatsApp. V poslední kapitole s názvem „Analýza aplikace Vinted“ je popsán postup analýzy aplikace Vinted a jsou zde shrnuty výsledky, které z analýzy vyplynuly.

Kapitola 1

Teoretický úvod a způsoby ukládání aplikačních dat

Tato kapitola obsahuje základní informace o operačních systémech Android a iOS a některých klíčových technologiích, které tyto operační systémy využívají. Zmíněné jsou mimo jiné také způsoby šifrování paměti, způsoby zálohování dat a metody eskalace privilegií. Dále se věnuje způsobům ukládání aplikačních dat, která operační systémy nabízí.
Pro ukládání aplikačních dat se dá využít množství vestavěných řešení. Liší se v druhu dat, pro která je dané úložiště vhodné, v druhu přístupu, který k danému zdroji mají jiné aplikace, a v místě uložení – zda se jedná o paměť zařízení či vzdálené úložiště.

V následujících sekcích jsou představeny technologie používané v mobilních zařízeních, které je dobré znát před pokračováním ve čtení této práce. Konkrétně se jedná např. o představení mobilních operačních systémů, jaké nástroje nabízí pro uložení aplikačních dat a jak jsou tato data chráněna.

1.1 Mobilní operační systémy

Mobilních operačních systémů existuje celá řada. Největší zastoupení mají dnes operační systémy Android a iOS, kdy Android představuje zhruba 70 % a iOS zhruba 29 % mobilních telefonů na trhu [1]. Mezi ostatní mobilní operační systémy patří především odnože OS Android, které již nesplňují kritéria kompatibility a nemůžou se tudíž řadit pod Android.

1.1.1 Android

Android je operační systém pro mobilní zařízení vyvíjený pod vedením společnosti Google. Jedná se o produkt projektu Android Open Source Project (AOSP), na kterém se podílí skupina firem sdružená pod názvem Open Handset Alliance. Spolupracují na jedné implementaci, kterou si každý může upravit pro potřeby svého zařízení.

Udržují též tzv. Android Compatibility program, jehož cílem je definovat požadavky, které musí zařízení splňovat, aby bylo označené za kompatibilní. [2] Kompatibilní zařízení je takové zařízení, které je schopné podporovat jakoukoliv aplikaci třetí strany, která byla vytvořena s pomocí Android SDK (Software Development Kit) a NDK (Native Development Kit) [3].

Vedení projektu (Google) si zakládá na tom, aby se jednalo o ucelený produkt. Cílem je, aby výrobci zařízení naimportovali Android, ne implementovali specifikace. [2]

Dle [4] se operační systém skládá z následujících vrstev:

- **Aplikace**: Aplikace vyvinuté pomocí API
  - **Klasické aplikace**: vytvořené pouze s pomocí Android API; dostupné ke stažení např. pomocí Google Play Store; v některých případech mohou být předinstalované
  - **Privilegované aplikace**: vytvořené pomocí Android a System API, musí být předinstalované jako privilegované aplikace
  - **Aplikace od výrobce zařízení**: kombinace Android API, System API a přímého přístupu k Android framework; předinstalované a aktualizované při aktualizaci celého OS
- **API**: API dostupné pro vývoj aplikací
  - **System API**: dostupné pouze pro partnery
  - **Android API**: dostupné pro všechny vývojáře
- **Android framework**: skupina předkompilovaného Java kódu, na kterém staví aplikace; části dostupné přes API; běží uvnitř procesu aplikace
- **Systémové služby**: modulární komponenty; funkcionality dostupné skrze API komunikují se systémovými službami, aby získaly přístup k hardware
- **Android runtime**: Java runtime environment
- **Hardware abstraction layer**: vrstva abstrakce nad hardware se standardizovaným rozhraním; umožňuje implementaci funkcionalit bez zásahu do vyšších vrstev
- **Nativní démoni a knihovny**: interagují přímo s kernelem, nespoléhají se na uživatelské prostředí
- **Kernel**: centrální část systému, komunikuje s hardware

1.1.2 iOS

Jedná se o operační systém vyvíjený firmou Apple pro jejich mobilní zařízení (např. iPhone, iPad). Je zjednodušenou verzí operačního systému pro laptopy a desktopové počítače MacOS. Jde o proprietární software, tudíž konkrétní detaily jsou z velké části neznámé. [5]

Jako i předchozí operační systém, iOS je založen na vrstvách. Nejnižší vrstvy zpřístupňují základní služby a technologie, další pak na těchto službách staví (nebo je doplňují). [6]

Vrstvy systému jsou následující[1]:

- **Cocoa (Aplikační)**: vytváření uživatelského rozhraní, správa chování aplikací
- **Media**: přehrávání, nahrávání a úprava audiovizuálních médií, rendering 2D a 3D grafiky
- **Core Services**: základní funkcionality – od automatického počítání referencí po manipulaci s textovými řetězci a formátování dat
- **Core OS**: definuje programové rozhraní pro hardware a síťové technologie
- **Kernel a drivery zařízení**: Mach kernel, drivery zařízení, funkce BSD (Berkeley Software Distribution) knihovny a jiné nízkoúrovňové komponenty; podpora pro souborový systém, síťové technologie, bezpečnost atd.

[1] jedná se o popis archytektury OS X dle [6], ne přímo iOS, ale iOS je od OS X odvozen

1.2 Aplikační sandbox

Aplikační sandbox je označení pro separaci aplikace od okolního prostředí (souborový a operační systém, ostatní aplikace). Provedení se liší napříč platformami, ale princip zůstává stejný.

1.2.1 Android

Android využívá ochranu založenou na Linuxových uživatelích pro identifikaci a izolaci aplikací. Každá aplikace obdrží unikátní uživatelský identifikátor (UID) a běží v samostatném procesu.

UID je využito k nastavení aplikačního sandboxu na úrovni kernelu. Ve výchozím nastavení spolu aplikace nemohou komunikovat a mají omezený přístup k operačnímu systému. Pokud se aplikace pokusí o nějakou nepovolenou akci (např. přístup k datům jiné aplikace nebo započatí hovoru bez řádných oprávnění), nepodaří se jí to, jelikož nemá potřebná práva výchozího uživatele.

Veškerý software běžící nad kernelem (OS knihovny, aplikační frameworky a veškeré aplikace) běží v rámci aplikačního sandboxu, není tudíž nutné požadovat specifické metody vývoje pro udržení bezpečnosti.

Pro prolomení aplikačního sandboxu na korektně nakonfigurovaném systému by bylo potřeba kompromitovat bezpečnost samotného linuxového kernelu. Jednotlivá opatření vynucující aplikační sandbox nejsou ale bez zranitelností, tudíž je důležité dodržovat principy defense-in-depth[2] pro zamezení kompromitace operačního systému kvůli jedné zranitelnosti. [7]

1.2.2 iOS

Aplikační sandbox poskytuje ochranu systémových zdrojů a uživatelských dat. Každá aplikace, která je distribuovaná přes App Store, musí podporovat aplikační sandbox. [8]

Při povolení aplikačního sandboxu aplikace ztratí všechna práva krom nutného minima pro běh. Další práva musí být následně přidělena jedno po druhém přes tzv. entitlement. Jedná se o páry klíč-hodnota, které identifikují specifické schopnosti (např. schopnost vytvořit síťové spojení). Přes entitlement se také povoluje sandbox. [9]

Při prvním použití aplikace využívající aplikační sandbox se vytvoří kontejnerový adresář, ke kterému má aplikace bezvýhradný přístup. Pro přístup ke zdrojům mimo tento adresář musí mít ale aplikace přidělený daný entitlement. [10]

1.3 Souborové systémy

Souborový systém určuje strukturu ukládaných dat a umožňuje jejich logické dělení do adresářů. Je nutné si definovat, jaký souborový systém daný operační systém používá nebo podporuje, jelikož mezi různými souborovými systémy existují nemalé rozdíly.

1.3.1 Android

Android podporuje většinu souborových systémů, které podporuje Linuxový kernel, jmenovitě jsou to např. exfat, ext4, fuse. Souborový systém musí splňovat určité požadavky, například podporu pro file-based šifrování přes `fscrypt` a autentizaci přes `fsverify`, aby byl vhodný pro použití. Požadavky jsou dány potřebou kompatibility s mechanismy jako proces aktualizací nebo ochrana soukromí. [11]

[2] princip zabezpečení systému; použití několika vrstev ochrany, aby byl systém stále chráněný i pokud některá selže

1.3.2 iOS

Primárním souborovým systémem pro iOS je Apple file system (APFS), kterému předcházel souborový systém (HFS+). Pro udržení jednoduchosti systému uživatel nemá přímý přístup k souborovému systému. [12]

Některé funkcionality APFS zahrnují klonování souborů, sdílení místa mezi svazky v rámci jednoho oddílu a podpora řídkých souborů. [13]

1.4 SQLite

SQLite je knihovna, která implementuje bezserverový transakční SQL databázový stroj. Čte a zapisuje přímo do souborů na disku, nemá žádný serverový proces. Kompletní SQL databáze s vícero tabulkami, triggery a pohledy je obsažena v jediném souboru. [14]

Mimo hlavní soubor databáze se ještě používá pomocný soubor, který drží dodatečné informace při transakci. Tímto souborem může být tzv. *rollback journal*, nebo soubor write-ahead log. Pokud je databáze v módu write-ahead log (WAL), používá se soubor write-ahead log, nelze tyto dva způsoby zaměňovat. Rollback journal soubor obsahuje informace, které umožní navrácení databáze do konzistentního stavu pokud systém či aplikace v průběhu transakce spadne. [15] Jedná se vlastně o kopii původního databázového souboru. Úpravy se pak provádí v hlavním souboru databáze a v případě spadnutí systémů či akci rollback se zkopírovaná data nahrají zpátky do hlavního souboru. Při použití WAL se tento proces obrátí, a sice hlavní soubor zůstane nezměněn a úpravy se zapisují do WAL souboru. Změny se projeví ve chvíli, kdy je do tohoto souboru zapsán speciální indikátor. [16]

Formát souborů je multiplatformní – dá se zkopírovat z 32-bit na 64-bit systémy i z big-endian na little-endian – což dělá z SQLite populární volbu pro formát aplikačních souborů. Spíše než náhradu databáze jako např. Oracle se tak jedná o náhradu funkcí jako `fopen()`.

Zdrojový kód je veřejně dostupný, celý projekt je open-source, tudíž není potřeba žádná licence a každý vývojář si může kód upravit podle svých potřeb. [14]

Jedná se o jednu z nejhojněji využívaných databází pro mobilní operační systémy. Apple tuto knihovnu používá pro většinu svých nativních aplikacích na všech svých zařízeních a iTunes používá SQLite i na zařízeních mimo Apple [17]. Android má pro SQLite nativní podporu, aplikace mají přístup k rozhraní skrze namespace *android.database.sqlite* [18].

1.5 Aplikační data

Mobilní aplikace často pracují s daty, která si potřebují nějakým způsobem ukládat – ať už se jedná o nějaký druh nastavení, nebo o dočasné soubory pro urychlení načítání a zmenšení množství opakovaných dotazů na servery. Tato data je potřeba nějakým způsobem uchovávat. Tato část se zabývá způsoby ukládání aplikačních dat, která jednotlivé operační systémy nabízí.

1.5.1 Android

Pro uložení aplikačních dat poskytuje Android několik možností:

- Úložiště specifické pro aplikaci, kam se ukládají data, která jsou k dispozici pouze dané aplikaci. Jedná se o dedikované adresáře v interní či externí paměti.
- Sdílené úložiště pro data, která aplikace sdílí s jinými aplikacemi (např. média).
- Primitivní soukromá data uložená v párech klíč-hodnota.
- Strukturovaná data v primitivní databázi.

K danému úložišti je možno se dostat dedikovanými metodami a aplikace pro některé z nich potřebuje specifická přístupová práva. Rozdíl je taktéž v tom, kdo všechno má k datům přístup (jiné aplikace) a zda se soubory smažou při odinstalování aplikace.

Android disponuje dvěma druhy fyzického úložiště: interním a externím. Interní paměť má zpravidla menší kapacitu, ale na rozdíl od externí je vždy k dispozici (externí úložiště může být odebráno). Aplikace jako takové jsou primárně ukládané v interní paměti, ale jde nastavit, aby se aplikace nainstalovala do externí paměti (např. když je aplikace příliš velká).

Na starších verzí Androidu potřebovala aplikace přístupová práva `READ_EXTERNAL_STORAGE` nebo `WRITE_EXTERNAL_STORAGE` (podle toho, co za akci chtěla provádět) pro přístup k jakémukoli souboru v externím úložišti mimo svůj domovský adresář. Na novějších verzích je větší důraz na účel souboru než na jeho umístění. Aplikace tak mají přístup jen k částem souborového systému, které opravdu využívají.

Pro aplikace na OS Android verze 10 a vyšší poskytuje android tzv. scoped storage. Tyto aplikace mají přístup pouze ke svému domovskému adresáři a ke specifickým typům médií, které si aplikace vytvořila. [19]

1.5.1.1 Úložiště specifické pro aplikaci

Toto úložiště je použito pro data, která nemají být dostupná pro jiné aplikace. Obsahuje adresáře pro uložení persistentních dat i místo pro dočasná data.

Může se jednat o interní i externí úložiště. Rozdíl je v tom, že v externím úložišti mají aplikace s potřebnými právy možnost přistoupit k souborům patřícím jiným aplikacím, zatímco v interním úložišti nikoliv. Při odinstalaci aplikace dojde k vymazání veškerých dat, která se v tomto úložišti nachází. [20]

1.5.1.2 Sdílené úložiště

Sdílené úložiště je vhodné použít ve chvíli, kdy se data mají sdílet s ostatními aplikacemi, nebo pokud mají data přetrvat v paměti i poté, co je aplikace odinstalována. Systém zpřístupňuje veřejné adresáře pro uložení fotografií, hudby a jiných typů dokumentů. [21]

1.5.1.3 SharedPreferences

Pro uložení malého množství dat jako páry klíč-hodnota lze použít SharedPreferences API. Jedná se o objekt, který míří na soubor, ve kterém se tato data ukládají, a který poskytuje jednoduché metody pro čtení a zápis. Každý soubor je spravovaný frameworkem a může být privátní či sdílený.

Moderní náhradou za toto řešení je DataStore, který používá stejnou strukturu pro uložení dat (páry), ale překonává množství nevýhod, které SharedPreferences má. [22]

1.5.1.4 Lokální databáze

Použít lokální databázi je vhodné pokud aplikace zpracovává netriviální množství strukturovaných dat. Nejběžnější použití je pro cachování dat, aby pak ve chvíli, kdy je zařízení odpojeno od sítě, mohl mít uživatel stále přístup k danému obsahu.

Android nabízí knihovnu Room, která poskytuje vrstvu abstrakce nad SQLite. Knihovna poskytuje několik benefitů oproti přímému použití SQLite API jako je např. kontrola SQL dotazů při kompilaci nebo anotace eliminující repetitivní boilerplate kód. [23]

1.5.2 iOS

Operační systém iOS nabízí pro ukládání aplikačních dat frameworky, které poskytují vrstvu abstrakce nad různými druhy úložiště. Významnou součástí je také možnost ukládání dat do cloudového úložiště.

1.5.2.1 Core Data

Základem je framework Core Data, skrze který je možné ukládat persistentní data pro použití bez připojení k síti nebo cachovat dočasná data. Pro synchronizaci dat napříč zařízeními Core data automaticky převádí datové schéma do CloudKit kontejneru. [24]

Core Data ve výchozím nastavení data nešifruje, ale lze přidat dodatečnou vrstvu, která šifrování zajišťuje.[3] [25]

Framework zprostředkovává tři (resp. čtyři) druhy úložiště: SQLite, Binární a In-Memory. XML úložiště není pro iOS dostupné. Taktéž je možné vytvořit si vlastní typ úložiště: atomické či inkrementální. Příkladem atomického úložiště je např. binární nebo XML úložiště. [26]

Přestože je SQLite podporovaným typem úložiště, jedná se o privátní formát. To znamená, že není možné vytvořit SQLite databázi s pomocí nativního API a použít ji přímo v Core Data. Taktéž se nedoporučuje manipulovat s existujícím Core Data SQLite úložištěm přes nativní API. Pro použití již existující databáze je potřeba ji importovat do Core Data. [27]

1.5.2.2 SwiftData

Nadstavbou nad Core Data je framework SwiftData. Kombinuje technologii perzistence z Core Data a funkce Swiftu pro přidání persistence do aplikace za pomoci minimálního množství kódu a bez externích závislostí. Velkou částí SwiftData je použití maker, které zjednodušují nastavování úložiště. Mimo ukládání lokálně vytvořeného obsahu se SwiftData hodí také na cachování dat získaných ze serveru pro podporu běhu aplikace bez přístupu k síti. SwiftData podporuje i synchronizaci dat přes více zařízení. [28]

1.5.2.3 CloudKit

Jedná se o framework poskytující rozhraní pro přesun dat mezi aplikací a iCloud[4]. Data aplikace jsou tak dostupná na vícero zařízeních.

Nejedná se o náhradu existujících datových objektů, spíše jde o doplňkovou službu, která zařizuje přesun dat mezi aplikací a iCloud serverem. Framework poskytuje minimální podporu pro cachování dat, tudíž spoléhá na připojení k síti a volitelně na validní iCloud účet. Aplikace může ukldádat data do veřejného prostoru, který je dostupný pro všechny uživatele, ale pokud požaduje ukládání dat specifických pro daného uživatele, potřebuje k tomu validní iCloud účet. [29]

1.6 Šifrování paměti

Výrobci mobilních operačních systémů nabízejí pro ochranu uživatelských dat různé technologie šifrování. Apple i tvůrci OS Android využívají pro šifrování paměti symetrickou kryptografii. Hlavním rozdílem v metodách ochrany je to, zda se používá jeden klíč pro šifrování celého disku, nebo zda jsou zašifrovány jednotlivé soubory samostatně. [30, 31]

[3] https://github.com/project-imas/encrypted-core-data

[4] Cloudová služba od Apple, více viz. https://support.apple.com/cs-cz/guide/icloud/welcome/1.0/icloud

1.6.1 Android

Android využívá dva druhy šifrování paměti:

File-based umožňuje používat pro jednotlivé soubory jiné šifrovací klíče. Soubory tak mohou být dešifrovány jednotlivě.

Full-disk využívá jeden šifrovací klíč pro ochranu celého oddílu s uživatelskými daty. Tento šifrovací klíč je chráněn přístupovými údaji uživatele. [30]

1.6.2 iOS

Apple používá pro svá zařízení technologii s názvem Data protection. Tato technologie využívá vestavěný kryptografický hardware k vytváření a správě klíčů. Při vytvoření souboru se zároveň vytvoří 256bitový klíč, s pomocí kterého je soubor zašifrován. Šifrovací algoritmus se liší podle čipu v zařízení, použit je algoritmus AES-256 XTS nebo AES-128 XTS[5]. [31]

1.7 Zálohování dat

Zálohy zařízení jsou vhodné pro přenesení dat ze starého mobilního telefonu do nového, dají se z nich ale také data získat, pokud je mobilní telefon nedostupný či nepoužitelný (např. zničený).

1.7.1 Android

Zálohovat data z mobilního zařízení s OS Android jde vícero způsoby. Příkladem je manuální zkopírování souborů do počítače, ale tímto způsobem lze zálohovat pouze soubory jako jsou fotky, dokumenty, videa atd. Existují také aplikace, které provedou zálohování automaticky.

Vestavěným řešením je zálohování dat pomocí účtu Google, který je na daném zařízení přihlášený. Zálohovat se dají fotografie a videa, ale i jiná data jako jsou kontakty, nastavení zařízení a aplikace a jejich data. Data jsou šifrovaná při přenosu i při uložení. K šifrování se používá heslo k účtu Google, případně přístupové údaje k zařízení (PIN, heslo, atd.). [32]

1.7.2 iOS

Apple nabízí pro zálohování svých mobilních zařízení dvě metody: pomocí iCloud a pomocí počítače.

Zálohovat data do iCloud je možné přes připojení k internetu, není k tomu potřeba mobilní zařízení nikam připojovat fyzicky. Pořízená záloha neobsahuje data, která jsou již s iCloud synchronizovaná (např. kontakty, kalendář, fotografie uložené v iCloud Photo). Záloha je vždy zašifrovaná. [33]

Záloha na počítači obsahuje téměř všechna data a nastavení ze zařízení [33]. Způsob pořízení zálohy se liší podle operačního systému počítače (macOS či Windows). Na macOS se zálohy pořizují pomocí aplikace Finder[6], na Windows pomocí aplikace Apple Devices či iTunes[7].

[5] operační mód šifry AES, více viz. `https://xilinx.github.io/Vitis_Libraries/security/2019.2/guide_L1/internals/xts.html`

[6] více viz. `https://support.apple.com/cs-cz/HT211229`

[7] více viz. `https://support.apple.com/cs-cz/108967`

1.8 Rooting a Jailbreak

Oba tyto názvy odkazují na způsob eskalace privilegií na mobilních zařízeních (*rooting* pro Android, *jailbreak* pro iOS). Existuje vícero způsobů, jak tohoto dosáhnout, závisejících na verzi daného operačního systému, jelikož některé metody byly díky aktualizacím znemožněné.

Tyto praktiky nejsou ze strany výrobců operačních systémů podporované a neexistuje k nim žádná oficiální dokumentace. Jedná se o výtvor komunity uživatelů a všechny informace jsou zjištěné pokusy na daných zařízeních.

1.8.1 Android

„Rooting“ je vpodstatě zpětné přidání klasické funkcionality Linuxu, která byla odebrána, a sice možnost přepnout se na privilegovaného uživatele (*root*). Díky tomu je pak uživatel schopen provádět operace, které by mu se standardními přístupovými právy nebyly umožněné (např. odinstalovat aplikace, které jinak odinstalovat nelze). [34]

1.8.2 iOS

Pojmem „jailbreak“ označuje softwarový nástroj, jehož použití umožní na mobilním zařízení s operačním systémem iOS provádět akce, které dříve nebyly dostupné. V zásadě se odeberou restrikce, které byly na zařízení implementované z výroby, a je tak možnost si na zařízení přidat neoficiální funkcionality, či si zařízení přizpůsobit vlastním potřebám (např. změna animace scrollování stránek). [35]

Kapitola 2

Analýza úložiště mobilních zařízení

Tato kapitola se věnuje analýze úložiště mobilních zařízení. Zahrnuje popis způsobů získání přístupu k paměti zařízení a protokoly, přes které zařízení komunikuje s počítačem. Dále se zde nachází popis struktury paměti mobilních zařízení s OS Android a iOS.
Aplikace na obou operačních systémech dostanou přidělený vlastní domovský adresář, do kterého mohou ukládat svá data. Tyto adresáře mají standardní názvy, aplikace si v nich případně mohou vytvářet podadresáře dle vlastních potřeb.
Některé informace byly dohledané v dokumentaci od tvůrců operačních systémů, jiné z již provedených výzkumů. Některé ukázky jsou pak pořízené ze zařízení patřícímu autorce práce.

Paměť mobilního zařízení obsahuje mnohem více dat než jen data ukládaná aplikacemi. Zájmová data je tedy nejprve potřeba v paměti nalézt. K tomu se dají využít nějaké jmenné konvence, které operační systémy pro úložiště dedikované aplikacím využívají, a zmenšit tak prostor hledání.

2.1 Možnosti přístupu k paměti zařízení

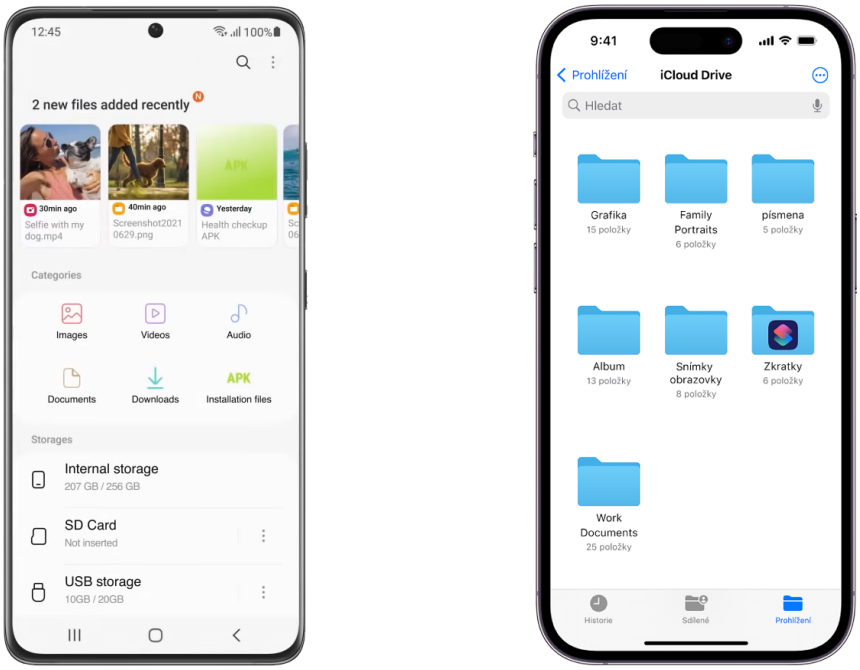
Jako první je třeba se k datům nějakým způsobem dostat. Toho lze dosáhnout vícero způsoby s různou mírou úspěchu. Tato kapitola se ještě nezabývá metodami používanými v rámci forenzní analýzy mobilních zařízení, spíše jde o způsoby nahlédnutí do paměti, které jsou dostupné běžnému uživateli bez pokročilých znalostí.

2.1.1 Přímo ze zařízení

Na zařízení s OS Android je uživateli část souborového systému zpřístupněná pomocí aplikace podobné správci souborů na počítači (např. na zařízení od společnosti Samsung se tato aplikace jmenuje *Moje Soubory*). Rozdíl oproti správci souborů na počítači je ale v tom, že přístupná je jen velmi malá část souborového systému a jedná se především o adresáře s fotkami, videi, staženými soubory atd.

Pro uživatele zařízení s operačním systémem iOS není souborový systém ze zařízení přístupný (viz. 1.3.2). Existuje aplikace *Soubory*[1], ale ta nemá přístup k souborovému systému zařízení tak, jako např. správce souborů, je spíše podobná službám typu *Google Drive* nebo *OneDrive*, pouze poskytuje místo pro uložení souborů, které jsou vytvořené či stažené uživatelem.

[1] více viz. `https://support.apple.com/en-us/102570`



(a) *Moje Soubory* na Samsung Galaxy [36]

(b) *Soubory* na iPhone [37]

Obrázek 2.1 Aplikace pro prohlížení souborů

2.1.2 Připojení přes USB

Jakmile je zařízení připojené pomocí USB konektoru k počítači, lze jej nalézt ve správci souborů. Ke komunikaci s ním je využívaný nějaký komunikační protokol, který zpřístupňuje jen některé adresáře a data. To, co je vidět ve správci souborů, tudíž nereflektuje reálný souborový systém.

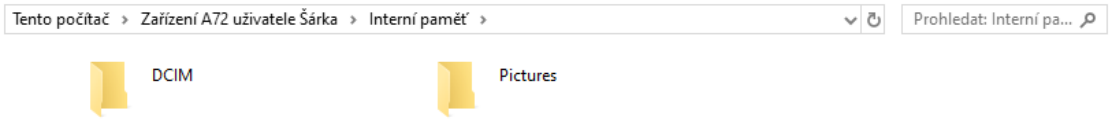
2.1.2.1 Protokoly pro přenos dat

Těchto protokolů existuje celá řada. Mezi nejrozšířenější patří protokoly PTP a MTP. Zařízení s operačním systémem Android podporují oba tyto protokoly [38], zatímco iOS podporuje jen PTP [39].

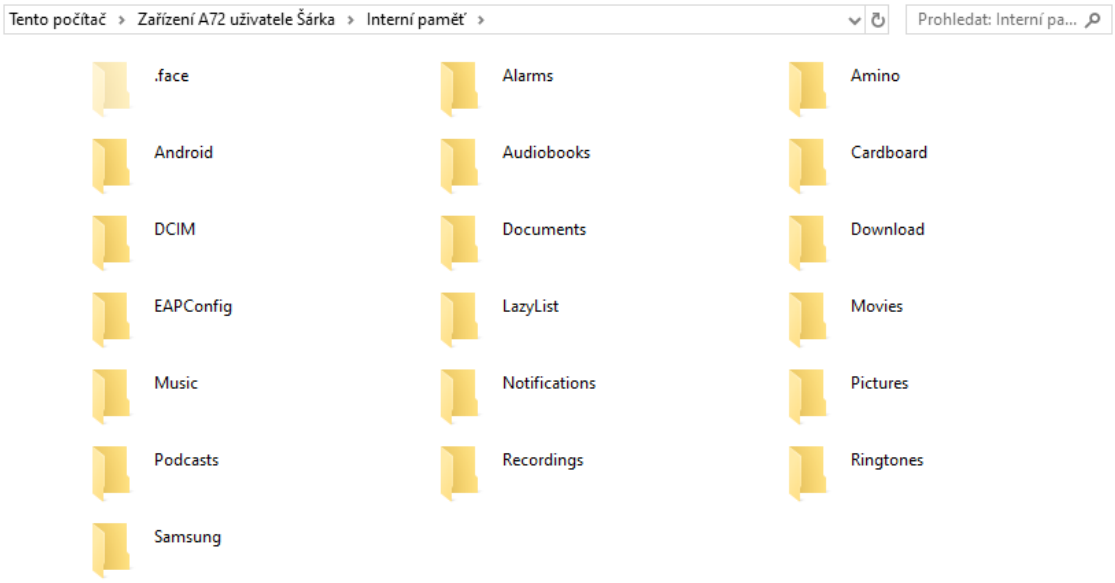
Picture Transfer Protocol (PTP) je standard pro komunikaci se zařízeními jako jsou digitální fotoaparáty. Definuje standardní operace, odpovědi, požadované chování atd., které zajišťují přenos a vytváření objektů jako jsou digitální fotografie a jiná média. [40]

Media Transfer Protocol (MTP) je navržen pro přenos dat a ovládání zařízení jako jsou digitální fotoaparáty, přenosné přehrávače hudby nebo mobilní telefony. Jedná se o rozšíření protokolu PTP. Účelem tohoto protokolu je zajištění komunikace se zařízeními, které mají velkou kapacitu úložiště a je možné je dočasně připojit k jinému zařízeni (např. PC). Skrze tento protokol je taktéž možné procházet obsah na daném zařízení. Další funkcí protokolu je možnost vyvolat příkazy na připojeném zařízení, např. čtení a úprava vlastností zařízení či vzdálené spuštění různých funkcionalit. [41]

Hlavní rozdíl v těchto protokolech je to, jak se mobilní zařízení jeví připojenému počítači a co za soubory zpřístupňuje. Při použití PTP na zařízení s OS Android (možnost „Přenášení obrázků“ v nastavení konfigurace USB) se zařízení jeví jako digitální fotoaparát a v počítači se zobrazí pouze adresáře, které jsou primárně určené pro média pořízená fotoaparátem zařízení,



Obrázek 2.2 Ukázka dostupných adresářu při „Přenášení obrázků“ na zařízení s OS Android

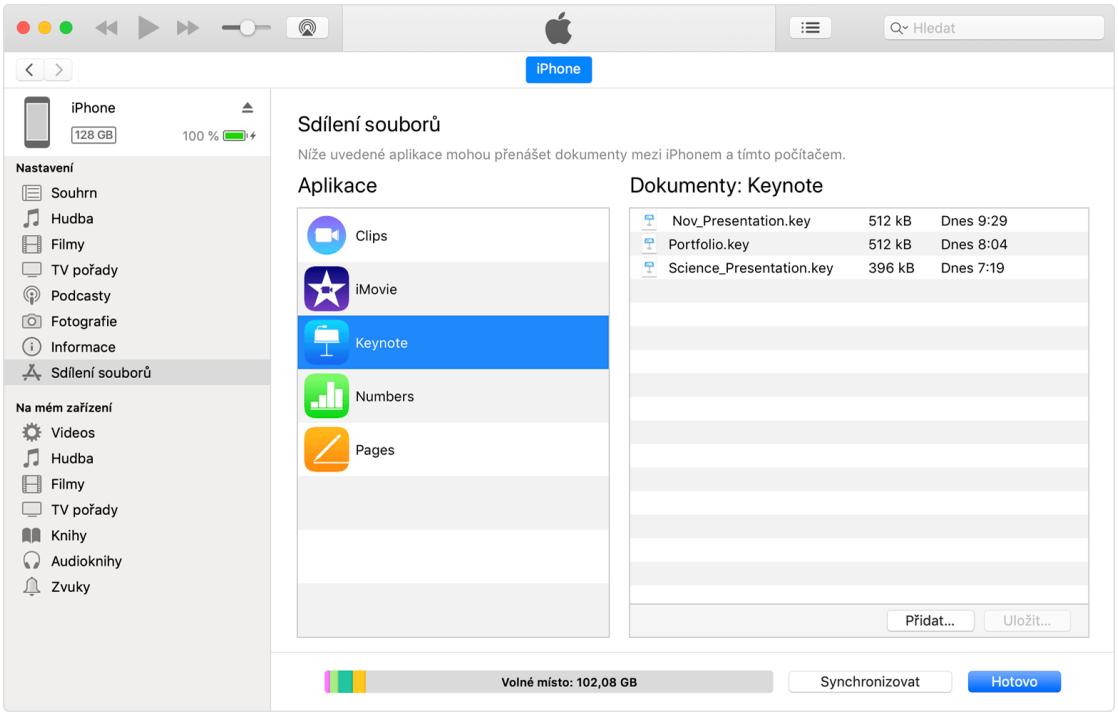


Obrázek 2.3 Ukázka dostupných adresářu při „Přenášení souborů“ na zařízení s OS Android

tj. `/DCIM` a `/Pictures` (viz. obrázek 2.2). Při komunikaci přes MTP (možnost „Přenášení souborů“ v nastavení konfigurace USB) je počítači zpřístupněno více adresářů, které obsahují více různých druhů souborů. Jsou mezi nimi i fotografie, ale zároveň různé jiné dokumenty, videa, stažené soubory atd. (viz. obrázek 2.3). Mimo jiné je zde i složka `Android`, která obsahuje adresáře jednotlivých aplikací. Tento adresář však zpravidla není přesný obraz toho, co všechno se v paměti reálně nachází.

2.1.3 Synchronizace souborů s iPhone

Pokud aplikace podporuje sdílení souborů, je možné přenášet soubory aplikací mezi zařízením iPhone a počítačem. Zařízení se k počítači připojuje pomocí USB kabelu či Wi-Fi. Přesný způsob provedení přenosu se liší podle toho, zda se jedná o počítač s OS Windows či MacOS. Na Windows se přenos souborů uskutečňuje pomocí aplikace *iTunes*, na MacOS pomocí aplikace *Finder*. [42, 43] Na obrázku 2.4 je příklad aplikací a souborů, které je možné přenést, viditelných v aplikaci iTunes.



Obrázek 2.4 Příklad aplikací podporujících sdílení souborů a souborů, které je možné přenést ze zařízení [44]

2.1.4 Android Debug Bridge

Android Debug Bridge (ADB) je terminálový nástroj, který dokáže komunikovat se zařízením a je skrze něj možné instalovat či ladit aplikace. Lze taktéž použít pro prohlížení souborového systému zařízení. Na zařízení je před propojením potřeba povolit debug operace (proces se liší podle modelu a výrobce). [45]

Součástí tohoto nástroje jsou tři procesy: **client** běžící na počítači, který posílá příkazy, **daemon** běžící na mobilním zařízení, který spouští příkazy, a **server** běžící na počítači, který zprostředkovává komunikaci mezi klientem a démonem. [45]

Pro komunikaci přes ADB je potřeba k sobě počítač s mobilním telefonem připojit. Toto připojení je možné navázat přes Wi-Fi (od Android 10) nebo USB. Následně je možné z počítače posílat příkazy. Je možné použít příkazy specifické pro ADB, nebo příkazy shellu. Je taktéž možné na zařízení spustit interaktivní shell. Android poskytuje většinu běžných Unixových příkazů. [45]

Souborový systém je možné si zobrazit přes příkaz `adb shell ls [adresář]` či jen příkaz `ls` v módu interaktivního shellu [45]. Tento souborový systém na rozdíl od adresářů zpřístupněných protokoly MTP či PTP reflektuje realitu toho, jak to v paměti zařízení vypadá.

Pro použití tohoto způsobu zpřístupnění paměti je již potřeba trochu více znalostí, poskytuje ale nejvíce informací ze všech ostatních možností pro OS Android.

2.2 Android

Následující sekce se zabývá popisem reálné struktury paměti na zařízení s OS Android. Některé informace byly získány z oficiální dokumentace, jiné z manuálu pro testování mobilních aplikacích.

2.2.1 Oddíly

Dle [46] zařízení s OS Android obsahují následující standardní diskové oddíly:

`boot`	Tento oddíl obsahuje kernel image a je vytvořený přes *mkbootimg*. Obsahuje také obecný ramdisk pro verze Android nižší než 13.
`init_boot`	Obsahuje obecný ramdisk pro zařízení s verzí Androidu 13 a vyšší.
`system`	Obsahuje Android framework.
`odm`	Obsahuje original design manufacturer (ODM) přizpůsobení pro system-on-chip (SOC) balíčky. Jedná se o volitelný oddíl.
`odm_dlkm`	Oddíl dedikovaný pro ODM kernel moduly – díky oddělení od oddílu odm je možné kernel moduly aktualizovat separátně.
`recovery`	Obsahuje recovery image, který je spouštěný při procesu OT. Zařízení podporující plynulé aktualizace mohou tento image ukládat v oddílu boot nebo init_boot.
`cache`	Zde se ukládají dočasná data. Jedná se o volitelný oddíl, pokud zařízení podporuje plynulé aktualizace. Nemusí být povolen zápis z bootloaderu, ale musí být možnost data smazat. Velikost závisí na druhu zařízení a velikosti oddílu *userdata*, typicky 50–100 MB.
`misc`	Využívaný oddílem recovery. 4 KB nebo větší.
`userdata`	Obsahuje uživatelem nainstalované aplikace a data včetně dat přizpůsobení.
`metadata`	Zde se ukládá šifrovací klíč pro šifrování metadat (pokud zařízení metadata šifruje). Tento oddíl není šifrovaný, při továrním nastavení se vymaže a přístup je striktně limitovaný.
`vendor`	Obsahuje jakýkoli spustitelný soubor, který není distribuovaný v rámci AOSP. Pokud zařízení neobsahuje proprietární informace, tento oddíl může být vypuštěn.
`vendor_dlkm`	Dedikovaný oddíl pro kernel moduly dodavatele – díky oddělení od oddílu vendor je možné kernel moduly aktualizovat separátně.
`radio`	Obsahuje radio image. Je potřeba pouze pro zařízení obsahující rádio se specifickým software v dedikovaném oddíle.
`tos`	Obsahuje binární obraz Trusty OS. Používá se pouze v případě, že zařízení tento OS obsahuje.
`pvmfw`	Zde je uložen Protected Virtual Machine Firmware, což je první kód, který běží v chráněném virtuálním stroji.

2.2.2 Adresáře

Jak je již zmíněno v teoretickém úvodu, Android podporuje dva druhy úložiště: interní a externí. Externí úložiště, jako je např. SD karta, jsou součástí souborového systému a jsou reprezentována cestou jako je např. `/sdcard`. [19]

Pojmenování adresářů se na různých zařízeních s OS Android mírně liší. V interním úložišti se jedná o adresář `/data/data/[název balíčku]` nebo `/data/user/0/[název balíčku]`. V externím úložišti je to pak adresář s názvem `/sdcard/Android/data/[název balíčku]` nebo `/storage/emulated/0/Android/data/[název balíčku]`. [47]

V adresáři v interním úložišti jsou pak běžné tyto podadresáře:

- `/cache`, kde jsou uložená cachovaná data,
- `/code_cache`, kde je uložený cachovaný kód,
- `/lib`, kde jsou uložené nativní knihovny napsané v jazyce C/C++,
- `/shared_prefs`, kde je uložen soubor ve formátu XML obsahující hodnoty uložené pomocí SharedPreferences (viz. 1.5.1.3),
- `/files`, kde jsou uložené různé soubory vytvořené aplikací,
- `/databases`, kde jsou uložené soubory SQLite databází pro danou aplikaci. [47]

2.3 iOS

Následující sekce obsahuje informace zveřejněné v dokumentaci od společnosti Apple. Informace o oddílech, do kterých je paměť jejich zařízení rozdělena, nebyly nalezené. Adresáře, kam aplikace standardně ukládají svá data, jsou však v dokumentaci popsané. Některé informace jsou získané z manuálu pro testování mobilních aplikací.

2.3.1 Adresáře

Domovským adresářem každé aplikace je adresář jejího aplikačního sandboxu. Tento adresář je identifikován absolutní cestou `/private/var/mobile/Containers/Data/Application/[UUID]`. UUID je zkratka pro Universal Unique Identifier, což je 128bitové číslo, které je každé aplikaci přiděleno při instalaci. [48]

Vlastní aplikace je uložena v adresáři `/var/containers/Bundle/Application/[UUID]` [48]. V tomto adresáři je podadresář s názvem `AppName.app`, který obsahuje aplikaci a všechny její zdroje. Při instalaci aplikace je podepsaný a pokud se v něm něco změní, aplikace nepůjde spustit [12].

Tyto dva adresáře nemají stejné názvy, jedná se o rozdílná UUID. Zjištění těchto identifikátorů lze např. pomocí nástrojů třetích stran jako je `ipainstaller` nebo Objection přímo na zařízení (pokud je proveden jailbreak). [48]

Adresáře běžně používané aplikacemi:

- `/Documents`: Obsahuje obsah vytvořený uživatelem.
- `/Documents/Inbox`: Soubory, které aplikace otevřela na popud jiné entity (např. e-mailové přílohy).
- `/Library`: Jakékoli soubory, které nemají být viditelné uživateli a nejsou uživatelskými daty. Některé standardní podadresáře jsou `/Application Support` nebo `/Caches`.
- `/tmp`: Dočasné soubory, které není třeba přechovávat po vypnutí aplikace. [12]

Kapitola 3

Metody forenzní analýzy mobilních zařízení

Tato kapitola se zabývá popisem metod použitelných při forenzní analýze mobilních zařízení. Zabývá se popisem způsobů získání dat z mobilních telefonů a způsobům jejich identifikace. Metody extrakce se liší v množství a formátu získaných dat. Může se jednat o binární kopii obsahu paměťového čipu či souboru informací získaných pomocí dotazování se API. Kdy jakou metodu použít záleží na tom, zda jsou proveditelné a zda je jejich výsledek postačující.

Tato práce se zaměřuje na dva hlavní kroky forenzní analýzy mobilních zařízení: získání dat a jejich analýza. Nejsou to samozřejmě jediné kroky, které celý proces obnáší, ale jsou to ty nejdůležitější pro tuto práci. Hlavní rozdíl při získávání dat z mobilních zařízení na rozdíl od získávání dat např. z pevných disků je v tom, jaké metody je možno použít. Architektura paměti mobilního zařízení se totiž od pevného disku poměrně zásadně liší, obzvláště v možnostech přístupu k datům. Některé způsoby přístupu byly popsány v předchozí kapitole, následující text pojednává o způsobech, jak tato data ze zařízení získat. Samotná analýza se pak příliš neliší od analýzy jiných dat.

3.1 Získání dat

Prvním krokem při analýze mobilních zařízení je získání dat. K tomu existuje několik metod, které se liší ve složitosti, invazivnosti a požadavcích na schopnosti a vybavení technika provádějícího analýzu.

Díky bezpečnostním opatřením, kterými většina dnešních mobilních zařízení disponuje (např. šifrování), je mnoho tradičních technik neúčinných, nebo je jejich provedení značně ztížené.

3.1.1 Manuální extrakce

Manuální extrakce zahrnuje zobrazení informací uložených na zařízení přímo na obrazovce zařízení. Nejsou k tomu potřeba žádné další pomůcky krom nějaké metody zachycení zobrazených dat (např. fotoaparát). Touto metodou lze získat pouze data, která jsou dostupná uživateli. Pokud je zařízení nějak poškozené (např. nefunkční display), stává se tato metoda složitou až nepoužitelnou. [49]

Pro použití této metody musí být známé přístupové údaje k zařízení, případně další přístupové údaje k jednotlivým aplikacím. [50]

3.1.2 Logická extrakce

Jednou z definic logické extrakce dat je zachycení kopie objektů logického úložiště (např. adresářů a souborů), které jsou uložené v logickém úložišti (např. oddíl souborového systému) [49]. Nejedná se tedy o získání celé paměti, ale dat, která se v ní nacházejí. Při logické extrakci se nezískají smazané soubory nebo data z nealokovaného prostoru paměti.

Jedná se o poměrně jednoduchou a rychlou techniku. K získání dat je potřeba připojit mobilní zařízení k pracovní stanici pomocí kabelu (USB, RS-232 či jiný podporovaný) nebo bezdrátového připojení (Wi-Fi, Bluetooth, …). Data jsou pak získána pomocí dotazování se operačního systému přes API. [51, 49]

Pokud jsou známé potřebné autentizační údaje (např. PIN, heslo, gesto, …) a je možné na zařízení povolit debug operace, lze logickou extrakci na dnešních zařízeních provést. Jelikož získání přístupových údajů k danému zařízení nemusí být vždy možné, objevují se různé způsoby, jak autentizaci obejít. Tyto metody zahrnují vymazání dat týkajících se zamykacího mechanismu ze zařízení, modifikace procesu zapnutí zařízení, aby se přeskočila fáze odemykání atd. [50]

3.1.3 Fyzická extrakce

Fyzická extrakce je definovaná jako získání kopie nebo image fyzické paměti (např. paměťový čip) [49]. Jedná se tedy o data přesně tak, jak jsou uložena na čipu. Touto metodou se lze dostat i ke smazaným souborům a nealokovanému prostoru v paměti. Výsledkem fyzické extrakce je binární kopie paměti zařízení.

Techniky fyzické extrakce se dělí podle toho, zda je potřeba při jejich použití mobilní zařízení rozebírat či nikoliv.

3.1.3.1 Neinvazivní techniky

Při použití neinvazivní techniky zůstane zařízení neporušené, nerozebrané. Extrakce obnáší připojení zařízení přes kabel k pracovní stanici (počítač) a použití nějakého software, který je schopný ze zařízení přečíst data. Ve většině případů nebude výsledkem použití těchto technik přesná binární kopie paměti, ale získaná data budou ve stejném formátu, jako jsou uložena v zařízení. [52]

Extrakce paměti zařízení může obnášet nahrání modifikovaného boot loaderu[1] do paměti zařízení, připojení k pracovní stanici a flasher boxu[2] nebo přepnutí zařízení do diagnostického módu [49], není to ale vždy nezbytně nutné [52]. Zařízení však musí být ve stavu zvýšených privilegií (viz. 1.8), aby mohl zvolený software extrakci provést [52].

3.1.3.2 Invazivní techniky

Jako invazivní techniky se označují ty, při jejichž použití je zařízení potřeba částečně či úplně rozebrat. Patří sem techniky, při kterých stačí zařízení rozebrat natolik, aby se zpřístupnilo nějaké žádoucí rozhraní, i techniky, při kterých se zařízení rozebere tak, že jej nelze už dále používat. [52]

Jedním z rozhraní je *Joint Test Action Group (JTAG)*. Jedná se o standard definující testovací rozhraní pro různé čipy používané v mobilních zařízeních (procesor, paměť, …). Pokud dané zařízení toto rozhraní podporuje, dá se použít k přečtení dat na paměťovém čipu. [49]

Další možností je použití metody *In-System Programming*. Tato metoda obnáší přímé připojení k paměťovému čipu zařízení a stažení celého jeho obsahu. Funguje na čipech typu Embedded Multimedia Card a Embedded Multi-Chip Package , které jsou v dnešních mobilních telefonech standardem. Tato metoda umožňuje obejití zamykacího mechanismu zařízení a je použitelná

[1] Image, který je zodpovědný za spuštění kernelu, viz. `https://source.android.com/docs/core/architecture/bootloader`

[2] Zařízení původně určené k servisu mobilních zařízení, je schopné komunikovat s mobilním zařízením pomocí diagnostických protokolů a zpřístupnit tak paměť zařízení [49]

i v případě, že zařízení není JTAG kompatibilní. Navíc je tato metoda rychlejší než JTAG při kopírování dat. Jedná se o nedestruktivní alternativu Chip-Off. [53]

Chip-Off je označení pro metody, při kterých se data získají přímo z paměťového čipu, který je vyjmut za zařízení. Výsledkem je binární obraz paměti. Tato metoda se nejvíce podobá klasickému klonování počítačového hard disku. [49]

Micro Read je nejkomplikovanější a časově nejnáročnější metodou. Zahrnuje použití elektronového mikroskopu pro přečtení stavů logických hradel na paměťovém čipu. [49] Dnešní čipy jsou však tak drobné, že je tato metoda téměř nepoužitelná [50].

S dnešními zařízeními je problém v tom, že i když mnoho z nich podporuje výše zmíněná rozhraní, jsou často zablokovaná či zamčená z výroby. Pokud s podaří k těmto rozhraním připojit a získat obraz paměti, nastává další problém ve formě zašifrovaných dat. [50]

3.2 Analýza dat

Jakmile jsou data k dispozici, je třeba je nějakým způsobem interpretovat. Formátů souborů pro uložení dat existuje celá řada, je tedy potřeba nějakým způsobem určit, o který se jedná.

Pro analýzu, stejně jako pro získání dat, existuje množství software, které tuto práci usnadňují. Mezi některé populární patří XRY Logical[3], Celebrite UFED[4], MOBILEdit Forensics[5] nebo Autopsy[6].

3.2.1 Identifikace formátu souboru

Název souboru sestává ze dvou částí – jména souboru před tečkou a přípony za tečkou. Přípona je zkratkou formátu, ve kterém je soubor uložen. Některé běžné přípony jsou např. `png` nebo `jpg` pro fotografie, `txt` pro obyčejné textové soubory a `mp3` pro hudbu.

Přípon a odpovídajících formátů existují tisíce, tudíž je nemalá šance, že se při analýze získaných dat objeví soubor s neznámou příponou. Pro tento případ je možné využít nějakou databázi souborových přípon (např. `https://fileinfo.com/` nebo `https://soubory.info/`), kde je možné danou příponu vyhledat a zjistit, o jaký soubor se jedná.

Soubor ale příponu mít vůbec nemusí, nebo může mít příponu jinou, než jaká odpovídá jeho reálnému formátu, přesto se stále jedná o platný soubor. V tu chvíli je potřeba k identifikaci použít jiné indikátory.

Většina souborů začíná nějakou specifickou sekvencí bitů, díky které je lze identifikovat. I pro tato tzv. „magic numbers“ existuje databáze (např. `https://www.garykessler.net/library/file_sigs.html`), ve které lze tyto sekvence vyhledat a identifikovat tak daný soubor.

3.2.2 Identifikace relevantních dat

Zda jsou data relevantní se odvíjí od účelu analýzy, zpravidla podle druhu vyšetřované aktivity. Nástroje pro analýzu většinou mají nějakou funkci pro třídění nalezených souborů (např. tag v Autopsy[7]).

[3] více viz. `https://www.msab.com/product/mobile-data-extraction/`
[4] více viz. `https://cellebrite.com/en/ufed/`
[5] více viz. `https://www.mobiledit.com/mobiledit-forensic`
[6] open-source software určený pro analýzu obrazů disků, více viz. `https://www.sleuthkit.org/autopsy/`
[7] více viz. `https://sleuthkit.org/autopsy/docs/user-docs/4.21.0/tagging_page.html`

Následující seznam pokrývá některé druhy souborů, které by mohly obsahovat obecně relevantní data (informace k sestavení seznamu získané z [52]):

- **Soubory SQLite databází**: většinou soubory s příponou `db` nebo `sqlite`. Stav SQLite databáze je většinou uložen v jediném souboru, při transakci jsou ale doplňující informace uchovávány v dalším souboru, kterému se říká *rollback journal*, či v souboru write-ahed log. [15]
- **Property lists**: soubory s příponou `plist`, ve kterých se nachází informace o konfiguracích a nastaveních systému i aplikací, historie aplikací a dočasné informace. Data uvnitř těchto souborů mohou být uložena v XML nebo binárním formátu.
- **Soubory s logy**: soubory s příponou `txt` nebo `log`, obsahují informace o běhu aplikace.
- **Soubory ve formátu JSON (JavaScript Object Notation)**: soubory s příponou `json`, obsahují různá data ve formátu klíč–hodnota.
- **Soubory ve formátu XML (Extensible Markup Language)**: soubory s příponou `xml` obsahující nastavení a preference (příklad viz. 1.5.1.3)

Kapitola 4

Příklad analýzy aplikace – WhatsApp

Tato kapitola se zabývá popisem aplikace WhatsApp a dat, která ukládá do paměti mobilního telefonu. Tato data převážně zahrnují databáze se zprávami v rámci konverzací mezi uživateli, databáze kontaktů, multimediální soubory přijaté či odeslané v rámci konverzací atd. Analýza těchto dat spočívá převážně v nalezení relevantních informací, což v rámci této kapitoly byly obsahy odeslaných a přijatých zpráv.

WhatsApp je jednou z nejpoužívanějších aplikací pro komunikaci na světě [54]. Díky tomu existuje spousta zdrojů, ze kterých čerpat informace potřebné k provedení analýzy. V následujících sekcích jsou popsané postupy, jak se k datům této aplikace dostat, kde je hledat a jak je interpretovat. Existuje mnoho nástrojů určených speciálně pro analýzu dat aplikace WhatsApp, kvůli jejich konkrétnímu zaměření se jimi však tento text zabývat nebude. Tato kapitola by měla sloužit jako položení základu pro provedení vlastní analýzy jiné aplikace, která je popsaná v další kapitole.

4.1 O aplikaci

WhatsApp používá více než dvě miliardy uživatelů po celém světě. Jedná se o aplikaci určenou ke komunikaci ve formě textových zpráv s možností posílání různých médií (fotografie, videa, dokumenty atd.). Zároveň je možné v rámci aplikace uskutečňovat hlasové hovory i videohovory. [54]

4.1.1 Koncové šifrování

Od roku 2016 používá WhatsApp koncové šifrování pro všechny konverzace. Protokol pro šifrování je založený na protokolu *Signal* vytvořeném skupinou Open Whisper Systems.

Koncové šifrování v rámci aplikace WhatsApp je definované jako komunikace, která zůstává zašifrovaná od chvíle, co opustí zařízení odesílatele až do doby, než je přijata zařízením příjemce. Znamená to tedy, že nikdo, krom těchto dvou stran – ani WhatsApp, ani Facebook – nemůže mít přístup k obsahu dané komunikace. [55]

4.1.2 Zálohování

WhatsApp nabízí zálohování historie konverzací na Google Disk, iCloud, případně přímo do paměti telefonu. Při zálohování do cloudu mohou být zálohy taktéž chráněny koncovým šifrováním.

Na zařízení iPhone se zapnutým zálohováním celého zařízení jsou historie konverzací ze zálohy vyloučeny, pokud je povoleno koncové šifrování záloh. [56]

Na mobilním zařízení s OS Android probíhá zálohování do paměti telefonu automaticky každý den. Tyto zálohy jsou určené k tomu, aby v případě poškození aplikace při aktualizaci bylo možné konverzace obnovit. Manuální obnova konverzací z této zálohy je možná pouze na starších telefonech s OS Android verze 9 a starší. [57]

Zálohování do cloudu (ať se jedná o Google či iCloud) je možné nastavit tak, aby se provádělo automaticky v daných časových intervalech. Je taktéž možné provést manuální zálohu konkrétní konverzace. [58, 59]

4.2 Získání dat

Pro získání dat ze zařízení s OS Android je potřeba mít privilegia uživatele root a extrahovat paměť zařízení. Na zařízení s iOS jsou data součástí zálohy zařízení (iTunes), pokud není nastavené koncové šifrování záloh, tudíž nemusí být vůbec nutné extrahovat paměť zařízení. [60]

4.2.1 Data v zařízení s OS Android

Na zařízení s OS Android se soubory aplikace ukládají v adresáři `/data/data/com.whatsapp`. Hlavními zájmovými soubory v tomto adresáři, resp. podadresáři `/databases`, jsou ty obsahující SQLite databáze `wa.db` a `msgstore.db`.

Soubor `wa.db` obsahuje SQLite databázi kontaktů uživatele uložených v aplikaci. Záznamy obsahují telefonní číslo, jméno zobrazené v aplikaci a další informace poskytnuté při registraci do aplikace. Na obrázku 4.1b jsou vypsané všechny tabulky, které tato databáze obsahuje. Za vyzdvihnutí stojí tabulka `wa_contacts`, která obsahuje právě informace o kontaktech uživatele.

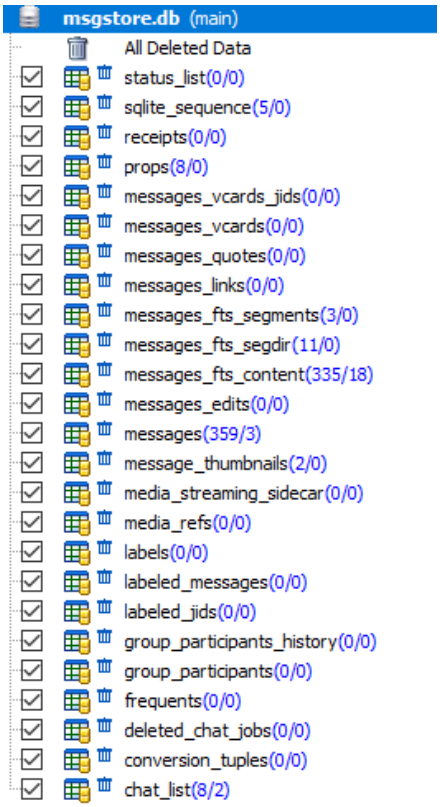
Soubor `msgstore.db` obsahuje databázi odeslaných zpráv. Informace ke zprávám zahrnují telefonní číslo příjemce, obsah zprávy, status (odesláno, přečteno, …), časovou značku atd. Na obrázku 4.1a je ukázka tabulek nacházejících se v této databázi. Nejdůležitějšími tabulkami jsou `message_fts_content`, která obsahuje text zpráv a `messages`, která obsahuje ostatní informace ke zprávě. Dalšími zájmovými tabulkami jsou `messages_thumbnails` obsahující informace o poslaných obrázcích a `chat_list` obsahující informace o konverzacích.

Mimo soubory databází obsahuje domovský adresář aplikace ještě několik dalších souborů:

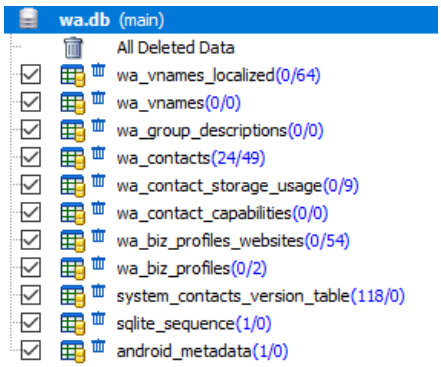
- Soubor `msgstore.db.cryptXX`, kde „XX“ reprezentuje číslo od 0 do 12 označující verzi algoritmu použitou pro šifrování. Tento soubor obsahuje zašifrované zálohy zpráv (viz. 4.1.2). Nachází se v externím úložišti v adresáři `/WhatsApp/Databases`.
- Soubor `key` v podadresáři `/files` v interním úložišti obsahující kryptografický klíč pro šifrování a dešifrování záloh.
- Soubor `com.whatsapp_preferences.xml` obsahující informace o profilu uživatele. Nachází se v podadresáři `/shared_prefs`.
- Soubor `axolotl.db`, který obsahuje databázi s kryptografickými klíči a dalšími daty potřebnými pro identifikaci vlastníka účtu v aplikaci.
- Soubor `chatsettings.db` obsahující databázi konfigurací aplikace

V podadresáři `/files/Logs` se pak nachází soubor s logy (`whatsapp.log`) a jeho zálohy (`whatsapp-yyy-mm-dd.1.log.gz`).

Obrázek 4.2 zobrazuje přehled popsané adresářové struktury. Obsahuje navíc výše nepopsané podadresáře v externím úložišti určené pro uložení přijatých médií (fotky, videa, audio soubory, hlasové zprávy atd.). Jedná se o podadresáře v adresáři `/Media`. [60]

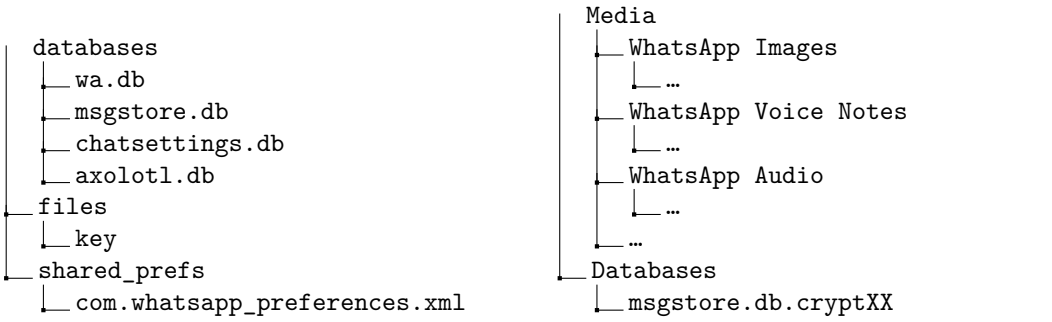


(a) *msgstore.db*; dostupné z: `https://website.cdn.group-ib.com/wp-content/uploads/figure5-min.png` [60]



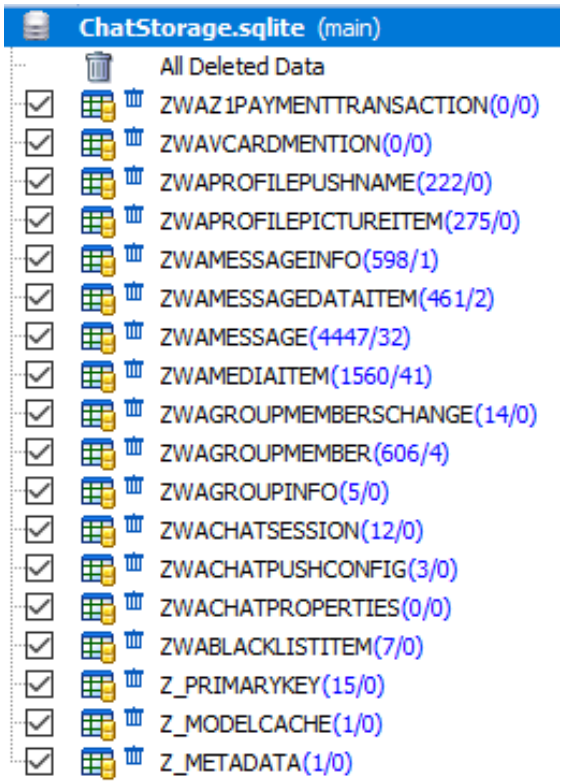
(b) *wa.db*; dostupné z: `https://website.cdn.group-ib.com/wp-content/uploads/figure3-min.png` [60]

Obrázek 4.1 Tabulky v databázích aplikace WhatsApp na zařízení s OS Android



(a) V interním úložišti (*/data/data/com.whatsapp*) **(b)** V externím úložišti (*/data/sdcard/WhatsApp*)

Obrázek 4.2 Přehled zájmových souborů aplikace WhatsApp v paměti mobilního telefonu s OS Android



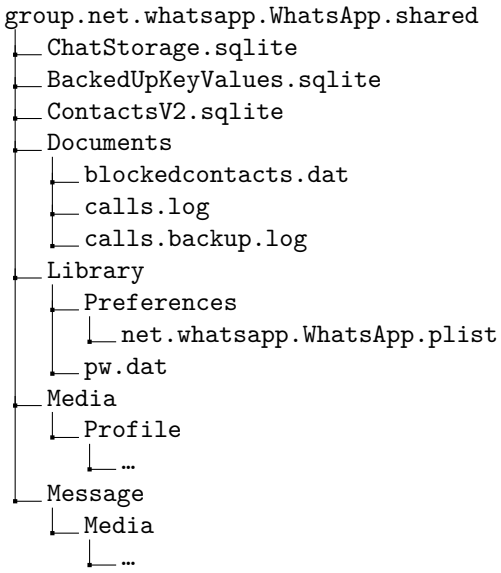
Obrázek 4.3 Tabulky v databázi *ChatStorage.sqlite* aplikace WhatsApp na zařízení iPhone [60]

Dostupné z: `https://website.cdn.group-ib.com/wp-content/uploads/figure13-min.png`

4.2.2 Data v iPhone

Adresář `/private/var/mobile/Applications/group.net.whatsapp.WhatsApp.shared/` je domovským adresářem aplikace v iTunes záloze telefonu iPhone. V tomto adresáři se nachází následující zájmové soubory:

- `ChatStorage.sqlite`: databáze obsahující tabulky s informacemi o konverzacích, zprávách, médiích atd. Výpis všech tabulek v této databázi je na obrázku 4.3. V tabulce *ZWAMESSAGE* jsou obsažené informace o zprávách (obsah zprávy, časová značka, zda se jedná o přijatou či odeslanou zprávu atd.). V tabulce *ZWAMEDIAITEM* jsou pak informace o odeslaných a přijatých médiích (šířka a výška v pixelech, velikost souboru, název souboru, zašifrované URL souboru, atd.).
- `BackedUpKeyValue.sqlite`: obsahuje kryptografické klíče a jiná data pro identifikaci vlastníka účtu v aplikaci.
- `ContactsV2.sqlite`: obsahuje informace o kontaktech uživatele. Informace zahrnují jméno, telefonní číslo, status (např. „Hey there! I am using WhatsApp.“) atd.
- `/Documents/blockedcontacts.dat`: obsahuje informace o blokovaných kontaktech.
- `net.whatsapp.WhatsApp.plist` nebo `group.net.whatsapp.WhatsApp.plist`: Obsahuje informace o účtu v aplikaci. Je uložen v podadresáři `/Library/Preferences`.
- `/Library/pw.dat`: obsahuje zašifrované heslo.



Obrázek 4.4 Přehled zájmových souborů aplikace WhatsApp v paměti telefonu iPhone

Další zájmové lokace představují adresáře s médii. Profilové obrázky účtu uživatele a jeho kontaktů, stejně tak jako náhledové obrázky soukromých a skupinových konverzací jsou uložené v adresáři `/Media/Profile`. Multimediální soubory a jejich náhledy jsou uložené v adresáři `/Message/Media`. Soubor s logy běhu aplikace (`calls.log`) a jeho záloha (`calls.backup.log`) jsou uložené v adresáři `/Documents`.

Na obrázku 4.4 je přehled zájmových souborů a jejich umístění v domovském adresáři. [60] Jelikož iPhone narozdíl od mobilních telefonů s OS Android nepodporuje SD karty, žádná data nemohou být v externím úložišti uložena.

4.3 Analýza dat aplikace

Pro analýzu dat z aplikace WhatsApp existuje řada specializovaných nástrojů, dají se ale použít i ty obecné. Postup analýzy spočívá v prozkoumání nalezených souborů a získání relevantních informací. Asi nejdůležitější informací, kterou lze získat z dat ukládaných aplikací WhatsApp, je obsah zprávy. Tyto informace jsou obsažené v databázových tabulkách popsaných v předchozí sekci. Případně je možné získat přímo exportovanou konverzaci a provést analýzu tohoto souboru, ale tu je potřeba provést manuálně ze zařízení, k čemuž je potřeba mít funkční odemčené zařízení, což často není dostupné. Tato část je zaměřená na analýzu databázových tabulek.

4.3.1 Android

Na Androidu se tato data nacházejí v databázi `msgstore.db`, konkrétně se jedná o tabulky `message_fts_content` a `messages`.

V tabulce `message_fts_content` se nachází pouze text zprávy a jeho identifikátor. Žádná další data jako kdo zprávu odeslal, kdy atd. zde nejsou. Pro tyto informace je potřeba se podívat do tabulky `messages`. V této tabulce jsou kromě textů zpráv právě i informace o odesílateli, času odeslání, zda byla zpráva přečtena, o jaký typ zprávy se jedná (text, obrázek, video, …) atd. Na obrázku 4.5 je ukázka z této tabulky. Některé důležité sloupce této tabulky jsou pak vypsané níže. Nejedná se o všechny sloupce, které tabulka obsahuje, ale převážně o ty, jejichž obsah je známý a relevantní z hlediska zaměření analýzy. [60]

key_remote_jid	key_from_me	key_id	status	needs_push	data	timestamp	media_url
9[redacted]@s.whatsapp.net	0	3A803799543CC370399C	0	0	Plz call my person	1511770782000	
8[redacted]3@s.whatsapp.net	0	A5B5DC9927BFE2099AF41C64AF4A7F	0	0	hi	1511772438000	
8[redacted]3@s.whatsapp.net	0	3CF979D759CA3C6B210A352496ADC2	0	0	do you have BTC and BCH today?	1511772467000	
9[redacted]@s.whatsapp.net	0	call:15[redacted]	6	0		1511772745000	call_screen_presented
9[redacted]@s.whatsapp.net	0	8F9E8C26E3FAF517D717A20B401AE9	6	0		1511772745000	
9[redacted]@s.whatsapp.net	0	3AEC4DDBE88E62C56B78	0	0	60k	1511772762000	
7[redacted]@s.whatsapp.net	0	C1ECEF8A7189D81B8A7CA26E7E0644	0	0	??	1511773070000	
9[redacted]@s.whatsapp.net	0	3AE96FBFC0DA4EE264DC	0	0	Can you deal today?	1511773141000	
9[redacted]@s.whatsapp.net	0	call:15[redacted]	6	0		1511773697000	call_screen_presented
9[redacted]@s.whatsapp.net	0	3A8A511901CE30D53248	0	0	Money is ready	1511773705000	

Obrázek 4.5 Ukázka z tabulky *messages* v databázi *msgstore.db* na mobilním telefonu s OS Android [60]

Dostupné z: `https://website.cdn.group-ib.com/wp-content/uploads/figure8-min.png`

_id	identifikátor přidělený databází
key_remote_jid	WhatsApp ID účtu, se kterým probíhá konverzace, v rámci které byla zpráva odeslána
key_from_me	má hodnotu 0 pokud se jedná o příchozí zprávu, 1 pokud o odchozí
key_id	identifikátor zprávy
status	zda byla zpráva přečtena (0 = přijata, 4 = čeká na serveru, 5 = přijata v cíli, 6 = kontrolní zpráva, 13 = zpráva zobrazená příjemcem – přečtená)
data	pokud je typ zprávy „text“, v tomto poli je uložen text zprávy
timestamp	čas odeslání ve formátu Unix Epoch time
media_*	vícero sloupců, všechny s předponou *media*; zahrnují informace o velikosti, druhu atd. poslaného média; pokud se jadná o textovou zprávu, jsou převážně prázdná
media_wa _type	typ zprávy (0 = text, 1 = obrázek, 2 = audio, 3 = video, 4 = vizitka kontaktu, 5 = geolokace)
recieved_timestamp	čas přijetí zprávy ve formátu Unix Epoch time
send_timestamp	vždy hodnota -1

4.3.2 iOS

Na zařízení s operačním systémem iOS se tabulka s informacemi o zprávách nachází v databázi `ChatStorage.sqlite` pod názvem `ZWAMESSAGE`. V této tabulce jsou uloženy informace o odesílateli, textu zprávy, času odeslání atd. Na obrázku 4.6 je ukázka z této tabulky. Některé sloupce jsou popsané níže, bohužel účel většiny z nich není znám, tudíž je jich zde popsaný jen zlomek. Jsou to ale ty nejdůležitější z hlediska zaměření této analýzy. [60]

ZFROMJID	ZMEDIASECTIONID	ZPHASH	ZPUSHNAME	ZSTANZAID	ZTEXT
91[redacted]@g.us			Ad[redacted]jee	89538636C6AB441DA893BA0B20E10ABD	His comebacks are epic.
91[redacted]@g.us			Ka[redacted]????	D8C0A4958364E2DADF8DF3C69658C7C7	https://github.com/M4cs/BabySploit/blob/master/READM...
91[redacted]@g.us			NU[redacted]	C40CAD2AE00869588CE00FB137FFC029	I'm getting a free lancing project for Appsec How mu...
91[redacted]@g.us	2018-11		sa[redacted]	7AFCAB33225A7675D75AFDF0D04A9D23	
91[redacted]@g.us			Ad[redacted]jee	9E540381EB5FD84B2A358C1E42B29361	700 - 1k for each hour.
91[redacted]@g.us			NU[redacted]	D0ADB6EB2AF19972A3FFDB60F324CB0F	What i have decided is to cost around 35-40k
91[redacted]@g.us			NU[redacted]	539DCBA54649CB971E86C0504127699A	I need overall costing sir not per hour
91[redacted]@g.us			Su[redacted]	AC2FE5F587061A99054868DB26551D5C	Costing will depend on number of unique screens / forms,...
91[redacted]@g.us			Ad[redacted]jee	A323E65AC84AFC34C53EF7D37172293B	Yes. ??Go for it.
91[redacted]@g.us			NU[redacted]	D7FB4A9BC53CDFCD3A4BA198976D6D4B	Yup
91[redacted]@g.us			NU[redacted]	5FB2694A2C166E074AE188BF6877E814	If i will cost him this then we can negotiate till 30-27k??...
91[redacted]@g.us			Ad[redacted]jee	0CFBF9DAD89F44BAF739107C86ED830A	Is it an Indian client?
91[redacted]@g.us			Ad[redacted]jee	2A5D4F4DC22D5A636BEEB44D356C936D	Sounds a little expensive. But if the client is ready then s...
91[redacted]@g.us			Su[redacted]	B37EAF9B72C83568F81356793253E0CB	You can use that to create an estimate
91[redacted]@g.us			NU[redacted]	EA54310F2A1DE4F20CBB527C9DA64B24	Ok sir
91[redacted]@g.us			NU[redacted]	8F63B0C95B3D9224D68FCC96061241A0	Thank you sir
91[redacted]@g.us			Al[redacted]	D7A0E3384BA942C40048B22EE32A6ABE	effort estimate
91[redacted]@g.us			Al[redacted]	C6107D95635A77B2686804C0AFB0CB5C	well said bro
91[redacted]@g.us			D3[redacted]	CF613866E19C3B7A4D501F2D3B216E4A	Very good initiative Murty sir????
91[redacted]@g.us			Ad[redacted]jee	33C490A14CF7F446667AA4D6F7F0C3C9	Off topic : Anyone plays pubg here? ??
91[redacted]@g.us			Nik[redacted]	3AA30DB36249BCE81F5D	1
91[redacted]@g.us			D3[redacted]	BE47500D5ED89C28DEDD32D1CDB8649C	*Reply one to one*
91[redacted]@g.us			Ad[redacted]jee	4D0FD2BB9D2EC4E0D9ED00EB7DE089EC	??
91[redacted]@g.us			Ni[redacted]	21C43455DD36BDB1EC8601F042B1EB68	Yeah! Sometimes ??

Obrázek 4.6 Ukázka z tabulky *ZWMESSAGE* v databázi *ChatStorage.sqlite* na mobilním telefonu iPhone [60]

Dostupné z: `https://website.cdn.group-ib.com/wp-content/uploads/figure14-min.png`

Z_PK	identifikátor přidělený databází
ZISFROMME	má hodnotu 0 pokud se jedná o příchozí zprávu, 1 pokud o odchozí
ZMESSAGETYPE	typ zprávy
ZSENTDATE	čas kdy byla zpráva odeslaná ve formátu OS X Epoch Time
ZFROMJID	WhatsApp ID odesílatele
ZMEDIASECTIONID	měsíc a rok, kdy byl odeslaný mediální soubor
ZPUSHNAME	jméno kontaktu, který odeslal mediální soubor
ZSTANZID	identifikátor zprávy
ZTEXT	text zprávy
ZTOJID	WhatsApp ID příjemce

4.4 Výsledek

Nejkomplikovanějším krokem se zdá být získání dat a identifikace, o jaké informace se jedná a zda mají nějakou hodnotu. Identifikovat data v databázi lze podle pojmenování sloupců, ve kterých jsou uložená, společně s nějakým odhadem podle obsahu jednotlivých řádků. Databáze aplikace WhatsApp mají sloupce pojmenované poměrně dobře (alespoň ty s relevantním obsahem, jako je tělo zprávy, odesílatel a příjemce), tudíž jejich identifikace není složitá. To však nemusí být vždy pravidlem, příkladem je velké množství sloupců tabulky se zprávami v databázi `ChatStorage.db` na iPhone, jejichž obsah není identifikován.

Pro následující analýzu tedy bude zapotřebí nejprve nalézt soubory obsahující data, bude-li se jednat o SQLite databáze, identifikovat jejich tabulky a určit podle obsahu, zda se jedná o relevantní data. Bude-li se jednat o jiné formáty souborů, bude třeba dekódovat jejich obsah a převést data do čitelné podoby.

Kapitola 5
Analýza aplikace Vinted

V této kapitole je popsán postup manuální analýzy dat mobilní aplikace Vinted. Popis zahrnuje fázi získání dat, kde jsou popsané nalezené adresáře a jejich obsah, a fázi analýzy dat, kde jsou představené soubory se zajímavými daty a jejich obsah. Také je zde popsán skript pro automatickou analýzu těchto dat, který vznikl pro usnadnění procesu analýzy.

Pro vlastní analýzu byla zvolena aplikace Vinted. Jedná se o aplikaci pro obchodování se zbožím z druhé ruky, primárně s oblečením. Její funkce zahrnují prohlížení nabídek, komunikaci nakupujícího s prodejcem, integrovaný platební systém, vytvoření a sledování zásilky atd. [61] O tom, jaká data ukládá aplikace v mobilních zařízeních, nebyly nalezené žádné informace.

5.1 Získání dat

Data byla poskytnuta vedoucím práce. Jedná se o data z mobilního telefonu iPhone získaná z iTunes zálohy.

Poskytnutý adresář má název `9F089834-B142-4678-9179-B213A0BB7A3A`. Tato sekvence znaků označuje UUID aplikace, které je přiděleno každé aplikaci při instalaci (viz. 2.3.1). Uvnitř tohoto adresáře se nachází standardní podadresáře, jak jsou popsané v sekci 2.3.1, a dalších několik podadresářů. Všechny podadresáře a jejich obsah jsou popsány níže. Celý výpis adresářové struktury se nachází v příloze v souboru *data_dirtree.pdf*.

- `CloudKit`: Tento adresář slouží pro data používaná frameforkem CloudKit. Nenachází se zde žádné soubory, pouze několik prázdných podadresářů.
- `Documents`: Nachází se zde jeden podadresář, ve kterém je několik souborů pojmenovaných 36 znakovými řetězci s příponou `log`. Krom tohoto podadresáře obsahuje adresář ještě několik souborů bez přípony, jejichž názvy obsahují slovo „Events“.
- `Library`: V tomto adresáři se nachází nejvíce adresářů a souborů ze všech ostatních, nebudou zde tedy podrobně popsány všechny. Obsahuje též dva soubory s příponou `json`, jejichž název začíná řetězcem „com-facebook-sdk“, a jeden soubor s příponou `config`.
  - `Application Support`: V tomto adresáři se většinou nachází podpůrné soubory, které aplikace potřebuje pro svůj běh, tzn. konfigurace, šablony, datové soubory atd. [12]. V tomto případě obsahuje osm podadresářů, které obsahují převážně Property list soubory, SQLite databáze a JSON soubory. Za zmínku stojí adresář `com.braze.core.persistence`, v němž se nachází adresář `data`, v jehož podadresáři se nachází několik JSON souborů,

mimo jiné soubor s názvem `user.json`, a adresář `users`, který obsahuje další adresáře pojmenované 44znakovými řetězci, v nichž jsou další JSON soubory.

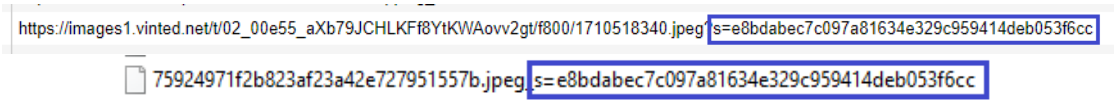
- `Caches`: V tomto adresáři se nacházejí dočasné soubory, převážně k různým nástrojům, které aplikace využívá. Tyto nástroje zahrnují InMobi[1], AddMob[2] nebo Crashlytics[3]. Celkem obsahuje 17 adresářů, kdy velká část z nich je prázdná. Zbytek obsahuje různorodé soubory v různých formátech, převažují SQLite databáze, obrázky ve formátu JPEG a soubory bez přípon. Zmínit si zaslouží dva adresáře, které nekorespondují k žádnému nástroji: `lt.manodrabuziai.fr` a `temp-image-cache`. První z nich obsahuje soubory bez přípon pojmenované řetězci náhodných znaků, druhý pak obrázky s příponou `jpeg`, jejichž název za příponou obsahuje ještě další řetězec znaků.
- `Cookies`: Obsahuje jeden soubor s příponou `binarycookies`.
- `HTTPStorage`: Obsahuje jeden podadresář, ve kterém jsou soubory SQLite databáze (přípona `sqlite`).
- `Preferences`: V tomto adresáři se nachází několik Property list souborů s příponou `plist`. Jedná se z většiny o data k nastavení různých nástrojů (Firebase, OneTrust nebo AppLovin).
- `Saved Application State`: Obsahuje několik vnořených podadresářů na jejichž konci se nachází soubor `data.data`.
- `SplashBoard`: Ve vnořených podadresářích obsahuje spoustu souborů s příponou `ktx`.
- `WebKit`: Tento adresář obsahuje data používaná frameworkem WebKit, který umožňuje integrovat do aplikace webovový obsah (HTML, CSS, Javascript) [62]. Obsahuje adresář `WebsiteData`, v jehož podadresářích se nacházejí převážně SQLite databáze, nebo jsou prázdné. V podadresáři `Default` se pak nachází soubor s názvem `salt` a v podadresářích tohoto adresáře je uložen soubor `origin`.

- `StoreKit`: Tento adresář patří datům frameworku StoreKit umožňujícímu např. nákupy v aplikaci nebo komunikaci s App Store [63]. Nachází se zde pouze jeden soubor s názvem `reciept`.
- `SystemData`: Tento ardesář je prázdný.
- `tmp`: V tomto podadresáři se nachází čtyři podadresáře:
  - `al`: obsahuje dva soubory s příponou `html`
  - `instrument`: obsahuje jeden JSON soubor, jehož název začíná řetězcem „crash_lib_data“
  - `models`: zde se nachází jeden soubor s příponou `weights`
  - `WebKit`: tento adresář obsahuje dva podadresáře, jeden z nich je prázdný, druhý obsahuje jeden soubor s příponou `plist`.

5.2 Analýza dat aplikace

Prozkoumání nalezených souborů probíhalo ručně – otevření souboru, jednalo-li se o čitelný formát, případně použití nějakého specializovanějšího nástroje. Pro SQLite databáze bylo použito rozšíření chromu SQLite Viewer[4]. JSON a textové soubory byly zobrazeny v běžném textovém editoru nebo v programu Visual Studio Code[5], který podporuje formátování JSON souborů.

[1] více viz. `https://www.inmobi.com/`
[2] více viz. `https://admob.google.com/home/`
[3] více viz. `https://firebase.google.com/products/crashlytics`
[4] více viz. `https://chromewebstore.google.com/detail/sqlite-viewer/golagekponhmgfoofmlepfobdmhpajia`
[5] více viz. `URL`



Obrázek 5.1 Ukázka shodného řetězce

Jako nejzajímavější se ukázal být obsah adresáře `Library/Caches/lt.manodrabuziai.fr`. V tomto adresáři se nachází soubor SQLite databáze s názvem `Cache.db`. Tato databáze obsahuje tři tabulky: *cfurl_cache_blob_data*, *cfurl_cache_reciever_data* a *cfurl_cache_response*. V poslední zmíněné se nachází sloupec *request_key*, v němž jsou uložena různá URL, většina z nich patřící Vinted API. Několik těchto URL odkazuje na obrázky. Tyto obrázky odpovídají několika souborům uloženým v adresáři `Library/Caches/temp-image-cache`. Neshodují se však v názvu, ale za příponou tohoto souboru je přidaný řetězec korespondující k hodnotě proměnné „s“ v URL. Příklad těchto souborů je na obrázku 5.1.

V podadresáři `fsCachedData` se pak nachází několik souborů bez přípony, všechny pojmenované řetězcem náhodných znaků (nejspíše se jedná o nějaký hash). Tyto soubory mají ale poměrně různé velikosti – v rozmezí jednotek až stovek kilobajtů – tudíž se dalo předpokládat, že se nejedná o stejný formát. Po otevření některých souborů bylo zjištěno, že některé jsou obrázky ve formátu JPEG, jiné pak textové soubory ve formátu JSON. V těchto textových souborech se nachází např. informace o účtu uživatele (viz. 5.1), konverzace (viz. 5.2) nebo prohlížené nabídky (viz. 5.3). Další textové soubory obsahují hodnocení uživatele nebo informace o konfiguraci aplikace.

Ostatní soubory obsahují obrázky prohlížených produktů. Názvy souborů se však neshodují s žádnými poli v textových souborech, tudíž nebylo zjištěno, které obrázky patří ke kterým záznamům v textových souborech. Soubory však obsahují URL obrázků, takže se dají zobrazit v prohlížeči. Tato URL jsou shodná s některými nalezenými v databázi zmíněné výše. Tyto obrázky pak byly dohledány v adresáři `temp-image-cache` stejně. Ne všechny se ale v tomto adresáři nachází.

Jelikož souborů se záznamem o uživateli bylo nalezeno více, bylo potřeba zjistit, jak určit, který uživatel je aktuálně přihlášený. Po porovnání obou nalezených souborů bylo zjištěno, že jeden z nich obsahuje výrazně více položek, než ten druhý. Jedná se např. o položky s klíči „birthday_permission“ nebo „real_name_permission“, jejichž přítomnost nasvědčuje tomu, že tento záznam je o uživateli, který je právě přihlášen, jelikož tyto informace by nebyly u prohlíženého profilu dostupné. Také obsahuje položku s klíčem „external_id“, jejíž hodnota odpovídá hodnotě „id“ v souboru `user.json` a názvu podadresáře v adresáři `users`, které byly zmíněné v předchozí sekci. Soubory v tomto podadresáři obsahují data k nástroji od společnosti Braze používanému pro online marketing[6].

5.2.1 Skript pro automatickou analýzu

Jelikož prohledávání adresářů a procházení jednotlivých souborů je poměrně zdlouhavé, vznikl v rámci této práce skript, který automaticky projde zájmové soubory a vygeneruje PDF (Portable Document Format) dokument obsahující nalezená data. Pro vytvoření byl zvolen jazyk Python, jelikož obsahuje knihovny pro práci s JSON soubory i pro vytváření PDF dokumentů. Jejich použití je přímočaré a jednoduché, což byl další důvod pro výběr tohoto jazyka.

Program přijímá na vstupu cestu k adresáři aplikace, v případě analyzovaných dat se jednalo o adresář `9F089834-B142-4678-9179-B213A0BB7A3A`. Nestačí předat pouze název adresáře, jelikož ten nemusí být v rámci celého souborového systému unikátní.

[6] více viz. `https://www.braze.com/`

```
{
  "user": {
      "id": 201859113,
      "anon_id": "8aee0deb-1290-4833-b3cd-add8e27e2d76",
      "login": "noviskarl",
      "real_name": "Karel Novák",
      "real_name_permission": 2,
      "birthday_permission": 1,
      "email": "macrons01_gene@icloud.com",
      "birthday": null,
      "gender": "",
      "currency": "CZK",
      "item_count": 0,
      "given_item_count": 0,
      "taken_item_count": 0,
      "favourite_topic_count": 0,
      "forum_msg_count": 0,
      "forum_topic_count": 0,
```


Výpis kódu 5.1 Začátek souboru s daty o uživateli


```
{
  "conversation": {
      "id": 11498613475,
      "read_by_current_user": true,
      "read_by_opposite_user": false,
      "localization": "manual",
      "translated": false,
      "allow_reply": true,
      "is_suspicious": false,
      "is_deletion_restricted": false,
      "subtitle": "Poslete prosim šířku v pase",
      "messages": [
          {
              "entity_type": "message",
              "entity": {
                  "body": "Poslete prosim šířku v pase",
                  "photos": [],
                  "user_id": 201859113,
                  "sent_via_mobile": true,
                  "id": 31092511209,
                  "reaction": null,
                  "is_hidden": false
              },
              "created_at_ts": "2024-04-28T18:01:22+02:00",
              "created_time_ago": "méně než jedna minuta"
          }
      ],
```


Výpis kódu 5.2 Začátek souboru s daty o konverzaci

```
{
  "item": {
      "id": 4226713752,
      "title": "Oblek Casino lesklý vel 188",
      "description": "Prodám použitý černý oblek zn. Casino. Stav horší. ",
      "status_id": 4,
      "disposal_conditions": 4,
      "catalog_id": 1789,
      "color1_id": 1,
      "package_size_id": 3,
      "is_hidden": 0,
      "is_reserved": 0,
      "is_closed": 0,
      "is_draft": false,
      "is_processing": false,
      "active_bid_count": 1,
      "item_closing_action": null,
      "currency": "CZK",
      "created_at_ts": "2024-03-15T16:59:00+01:00",
```


Výpis kódu 5.3 Začátek souboru s daty o nabízeném produktu

5.2.1.1 Načtení souborů

Python disponuje knihovnou `json`, která poskytuje rozhraní pro načítání JSON souborů a jejich převedení do objektů. [64]. Díky tomu načtení dat obnáší pouze použití dostupné funkce.

Nalezení požadovaného podadresáře je provedeno pomocí zřetězení názvů podadresářů za dodanou cestu pomocí knihovní funkce, aby byla zajištěna přenositelnost mezi platformami. Názvy jsou ale specifické pro iOS, skript tedy nelze použít pro data z telefonu s OS Android.

Soubory s textovými daty není možné od obrázků rozeznat předem, tudíž načítání probíhá tak, že se projdou všechny soubory v daném adresáři a zkusí se načíst. Pokud se operace nepodaří, přejde se na další soubor. Jelikož data mohou obsahovat unicode znaky (např. emotikony), bylo potřeba specifikovat, že se jedná o kódování „utf8“. Toho lze dosáhnout argumentem funkce, přes kterou se otevírá soubor. Ukázková funkce je na výpisu 5.4.

```
def getJsonData(files : list) -> dict:
    data = dict()
    for name in files:
        with open(name, encoding="utf8") as file:
            try:
                data[name] = json.load(file)
            except: pass
    return data
```

Výpis kódu 5.4 Funkce pro načítání souborů

5.2.1.2 Formátování

Prvotní plán vytvořit formátování nezávislé na obsahu souboru bohužel selhal v tom, že soubory obsahují velké množství polí, které nemají velkou vypovídající hodnotu. Výsledný dokument by tudíž nebyl o nic přehlednější, než původní JSON. Z originálních souborů jsou tedy vybrány jen nejdůležitější položky. Skript funguje pro následující objekty, které obsahují nejvíce zajímavých dat:

- conversation: s kým je konverzace vedena, o jaký produkt se jedná, jednotlivé zprávy a jejich odesílate
- conversations: texty zpráv a jejich odesílatelé; obsahuje pouze zprávy z jedné konverzace
- item: název produktu, jeho popis, kdo ho vytvořil a kdy a odkaz na produkt na webových stránkách
- items: zkrácený popis produktu; JSON objekty obsahují stejné informace jako předchozí položka, ale jelikož jich může být v jednom souboru velké množství, je zápis zkrácen
- user: uživatelské jméno, reálné jméno, email, datum narození, popis profilu a zda se jedná o aktuálně přihlášeného uživatele

Než jsou vybraná data převedena do PDF, je pro jejich zformátování použit jazyk HTML[7]. Data z jednotilvých souborů jsou podle obsahu seskupena do odstavců a tabulek. Pro upravení rozložení stránek jsou použity kaskádové styly, primárně jsou použity pro rozložení tabulek. Taktéž jsou využity tagy specifické pro knihovnu pro generování PDF dokumentu.

5.2.1.3 Vytvoření PDF dokumentu

Pro převedení HTML souboru do PDF dokumentu je použita knihovna `xhtml2pdf`. Tato knihovna může být použita v různých prostředích a poskytuje též nástroj v příkazové řádce. [65]

V tomto skriptu je pro vytvoření dokumentu použita funkce `CreatePDF()`. Tato funkce bere několik parametrů, zde bylo použito jen pár z nich, konkrétně *src* (soubor s HTML), *dest* (výsledný dokument) a *encoding* (kódování znaků ve zdroji).

Standardní písma, která knihovna podporuje, neobsahují české znaky. Bylo proto potřeba dodat vlastní písmo. Toho bylo dosaženo pomocí přidání odkazu na soubor s písmem v lokálním adresáři do souboru s kaskádovými styly. Zvoleno bylo písmo „Roboto“[8]. Ukázka stylů je ve výpisu 5.5

Zprovoznění použití vlastního písma obnáší několik opatření: přidání cesty funkci pro vytvoření dokumentu a použití příkazu `pisaFileObject.getNamedFile = lambda self: self.uri` před zavoláním funkce pro vytvoření dokumentu. První opatření je potřeba proto, aby se správně překládaly relativní cesty k souborům, druhé kvůli fungování skriptu na operačním systému Windows, jelikož byl problém s přístupem k dočasným souborům, které se v rámci použití písma vytváří [66].

5.3 Výsledky

Nalezení užitečných dat v rámci aplikace Vinted bylo poměrně zdlouhavé a složité. Aplikace neobsahuje SQLite databázi tak, jako aplikace WhatsApp, data jsou uložena v textových souborech ve formátu JSON. Tyto soubory jsou ukryty mezi obrázky v podadresáři adresáře `Library/Caches` a nebyly jednoduše identifikovatelné, jelikož na první pohled (podle názvu přípony) nebyly od souborů s obrázky rozeznatelné.

[7] Hypertext Matkup Language
[8] viz. `https://fonts.google.com/specimen/Roboto`

```
@font-face {
  font-family: font;
  src: url('templates/fonts/roboto/Roboto-Regular.ttf')
}
body {
  font-family: font
}
```

Výpis kódu 5.5 Ukázka kaskádových stylů pro přidání vlastního písma

Nalezená data obsahují poměrně hodně informací, nejzajímavějším nálezem byly zprávy mezi uživateli, které obsahovaly celou historii komunikace v jediném souboru, tudíž nebylo nutné dohledávát ostatní zprávy. Celkem bylo nalezeno 16 textových souborů s daty o uživatelích, produktech, konverzacích atd.

Vytvořený skript vybere ze všech dostupných souborů nejrelevantnější data a vytvoří z nich PDF dokument. Je však závislý na formátu dat uložených v souboru, nepokrývá tudíž všechny soubory, ve kterých aplikace ukládá svá data. Dostupná data také nemusela obsahovat všechny druhy JSON objektů, které aplikace používá a ukládá, skript je tudíž nepodporuje.

Skript pro vytvoření reportu a report vygenerovaný tímto skriptem se nachází v příloze, stejně tak jako analyzovaná data. Report je možné nalézt v souboru *report.pdf*.

Kapitola 6
Závěr

Cílem práce bylo zjistit, kam se v mobilních telefonech ukládají aplikační data, popsat metody forenzní analýzy dat mobilních aplikací, demonstrovat poznatky na popsané aplikaci a na základě předchozích zjištění provést manuální analýzu jiné mobilní aplikace.

Aplikace mají v mobilních telefonech přidělený adresář, který jim slouží pro ukládání dat. V tomto adresáři se nachází několik standardních adresářů, ale aplikace si do něj mohou přidávat další podle svých potřeb.

Metody forenzní analýzy jsou specifické ve způsobech získávání dat, samotná analýza se od analýzy dat z jiného zařízení příliš neliší.

Pro demonstraci byla zvolena aplikace WhatsApp. V práci jsou popsané databáze a jejich tabulky, které obsahují zajímavá data.

Pro vlastní analýzu byla vybraná aplikace Vinted. Výsledkem je popis zájmových souborů, kde v adresáři aplikace se nacházejí a co obsahují za data. Jednalo se především o soubory ve formátu JSON, které obsahovaly informace o uživatelích, produktech a odeslaných a přijatých zprávách. Pro jednoduché vytvoření přehledu nalezených dat vznikl skript, který ze zájmových souborů vezme nejdůležitější informace a vygeneruje PDF dokument. Pro vytvoření skriptu byl zvolen jazyk Python a knihovna xhtml2pdf. Zdrojový kód skriptu se nachází v příloze práce.

Jelikož byl dostupný pouze jeden vzorek dat ukládaných aplikací, skript nemusí nutně pokrývat všechny JSON objekty, ve kterých aplikace ukládá data, v budoucnu by tedy bylo možné skript rozšířit o podporu většího množství formátů.

Bibliografie

1. *Mobile OS Market share worldwide 2009-2023* [online]. Statista, 2024 [cit. 2024-02-04]. Dostupné z: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/.
2. *AOSP overview | Android Open Source Project* [online]. Google LLC, 2024 [cit. 2024-04-02]. Dostupné z: https://source.android.com/docs/setup/about.
3. *Android Compatibility Program Overview | Android Open Source Project* [online]. Google LLC, 2024 [cit. 2024-02-04]. Dostupné z: https://source.android.com/docs/compatibility/overview.
4. *Architecture overview | Android Open Source Project* [online]. Google LLC, 2023 [cit. 2024-02-04]. Dostupné z: https://source.android.com/docs/core/architecture.
5. KENTON, Will. *Apple iOS* [online]. Investopedia, 2021 [cit. 2024-02-08]. Dostupné z: https://www.investopedia.com/terms/a/apple-ios.asp.
6. *About Developing for Mac* [online]. Apple Inc., 2015 [cit. 2024-02-08]. Dostupné z: https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/About/About.html#//apple_ref/doc/uid/TP40001067-CH204-TPXREF101.
7. *Application Snadbox | Android Open Source Project* [online]. Google LLC, 2024 [cit. 2024-02-21]. Dostupné z: https://source.android.com/docs/security/app-sandbox.
8. *App Sandbox | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-10]. Dostupné z: https://developer.apple.com/documentation/security/app_sandbox.
9. *App Sandbox in Depth* [online]. Apple Inc., 2016 [cit. 2024-02-10]. Dostupné z: https://web.archive.org/web/20221223145702/https://developer.apple.com/library/archive/documentation/Security/Conceptual/AppSandboxDesignGuide/AppSandboxInDepth/AppSandboxInDepth.html#//apple_ref/doc/uid/TP40011183-CH3-SW4.
10. *Migrating your app's files to its App Sandbox container | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-10]. Dostupné z: https://developer.apple.com/documentation/security/app_sandbox/migrating_your_app_s_files_to_its_app_sandbox_container.
11. *Android Kernel File System Support | Android Open Source Project* [online]. Google LLC, 2023 [cit. 2024-02-06]. Dostupné z: https://source.android.com/docs/core/architecture/android-kernel-file-system-support.

12. *File System Basics* [online]. Apple Inc., 2018 [cit. 2024-02-08]. Dostupné z: https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html#//apple_ref/doc/uid/TP40010672-CH2-SW12.
13. *About Apple File System | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-12]. Dostupné z: https://developer.apple.com/documentation/foundation/file_system/about_apple_file_system.
14. *About SQLite* [online]. Hipp, Wyrick & Company, Inc., 2023 [cit. 2024-02-06]. Dostupné z: https://www.sqlite.org/about.html.
15. *Database File Format* [online]. Hipp, Wyrick & Company, Inc., 2023 [cit. 2024-04-12]. Dostupné z: https://www.sqlite.org/fileformat.html.
16. *Write-Ahead Logging* [online]. Hipp, Wyrick & Company, Inc., 2018 [cit. 2024-04-12]. Dostupné z: https://www.sqlite.org/wal.html.
17. *Well-Known Users Of SQLite* [online]. Hipp, Wyrick & Company, Inc., 2023 [cit. 2024-02-06]. Dostupné z: https://www.sqlite.org/famous.html.
18. *SQLite Android Bindings* [online]. Hipp, Wyrick & Company, Inc., 2024 [cit. 2024-02-06]. Dostupné z: https://sqlite.org/android/doc/trunk/www/index.wiki.
19. *Data and file storage overview | Android Developers* [online]. Google LLC, 2024 [cit. 2024-02-08]. Dostupné z: https://developer.android.com/training/data-storage.
20. *Access app-specific files | Android Developers* [online]. Google LLC, 2024 [cit. 2024-02-15]. Dostupné z: https://developer.android.com/training/data-storage/app-specific.
21. *Overview of shared storage | Android Developers* [online]. Google LLC, 2023 [cit. 2024-02-15]. Dostupné z: https://developer.android.com/training/data-storage/shared.
22. *Save simple data with SharedPreferences | Android Developers* [online]. Google LLC, 2024 [cit. 2024-02-15]. Dostupné z: https://developer.android.com/training/data-storage/shared-preferences.
23. *Save data in a local database using Room | Android Developers* [online]. Google LLC, 2024 [cit. 2024-02-16]. Dostupné z: https://developer.android.com/training/data-storage/room.
24. *Core Data | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-27]. Dostupné z: https://developer.apple.com/documentation/coredata.
25. MUELLER, Bernhard; SCHLEIER, Sven; WILLEMSEN, Jeroen; HOLGUERA, Carlos. *Data Storage on iOS - OWASP MASTG* [online]. OWASP, 2024 [cit. 2024-03-12]. Dostupné z: https://mobile-security.gitbook.io/mobile-security-testing-guide/ios-testing-guide/0x06d-testing-data-storage.
26. *SwiftData | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-27]. Dostupné z: https://developer.apple.com/documentation/coredata/nspersistentstore.
27. *Persistent Store Types and Behaviors* [online]. Apple Inc., 2017 [cit. 2024-02-27]. Dostupné z: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/PersistentStoreFeatures.html.
28. *NSPersistentStore | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-02-27]. Dostupné z: https://developer.apple.com/documentation/swiftdata.
29. *CloudKit | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-03-27]. Dostupné z: https://developer.apple.com/documentation/cloudkit.
30. *Encryption | Android Open Source Project* [online]. Google LLC, 2024 [cit. 2024-04-04]. Dostupné z: https://source.android.com/docs/security/features/encryption.

31. *Data Protection overview* [online]. Apple Inc., 2024 [cit. 2024-04-04]. Dostupné z: https://support.apple.com/cs-cz/guide/security/secf6276da8a/web.
32. *Back up or restore data on your Android device - Android Help* [online]. Google LLC, 2024 [cit. 2024-03-25]. Dostupné z: https://support.google.com/android/answer/2819582?hl=en.
33. *About backups for iOS devices - Apple Support* [online]. Apple Inc., 2024 [cit. 2024-03-25]. Dostupné z: https://support.apple.com/en-us/108771.
34. HILDENBRANDL, Jerry. *Everything you need to know about rooting your Android phone | Android Central* [online]. Android Central, 2022 [cit. 2024-03-17]. Dostupné z: https://www.androidcentral.com/root.
35. *Jailbreak Noob Guide v1.0* [online]. r/jailbreak community, 2021 [cit. 2024-03-17]. Dostupné z: https://www.reddit.com/r/jailbreak/comments/ltfrpk/discussion_jailbreak_noob_guide_v10/.
36. *Samsung Support: How to manage files on your Galaxy device* [video]. Samsung, 2022 [cit. 2024-04-08]. Dostupné z: https://www.youtube.com/watch?v=0bjr02-MyXM.
37. *Úpravy souborů, složek a stažených položek v aplikaci Soubory na iPhonu* [online]. Apple Inc., © 2024 [cit. 2024-04-08]. Dostupné z: https://support.apple.com/cs-cz/guide/iphone/iphc61044c11/ios.
38. *Honeycomb | Android Developers* [online]. Google LLC, 2019 [cit. 2024-04-08]. Dostupné z: https://developer.android.com/about/versions/android-3.0-highlights#Digital.
39. *Import photos using Apple camera adapters - Apple Support (OM)* [online]. Apple Inc., 2024 [cit. 2024-04-08]. Dostupné z: https://support.apple.com/en-om/118280.
40. *PTP Standards* [online]. Society for Imaging Sciences a Technology, © 2023 [cit. 2024-04-04]. Dostupné z: https://www.imaging.org/IST/ist/standards/PTP_Standards.aspx.
41. *USB Media Transfer Protocol Specification*. Rev. 1.1. USB Implementers Forum, 2011. Dostupné také z: https://www.usb.org/document-library/media-transfer-protocol-v11-spec-and-mtp-v11-adopters-agreement.
42. *Synchronizace souborů z Macu s iPhonem, iPadem nebo iPodem touch* [online]. Apple Inc., © 2024 [cit. 2024-04-14]. Dostupné z: https://support.apple.com/cs-cz/guide/mac-help/mchl4bd77d3a/mac.
43. *Přenos souborů mezi PC a zařízeními prostřednictvím iTunes* [online]. Apple Inc., © 2024 [cit. 2024-04-14]. Dostupné z: https://support.apple.com/cs-cz/guide/itunes/itns32636/windows.
44. *Sdílení souborů mezi počítačem a iOS nebo iPadOS zařízením pomocí iTunes* [online]. Apple Inc., 2022 [cit. 2024-04-14]. Dostupné z: https://support.apple.com/cs-cz/HT201301.
45. *Android Debug Bridge (adb) | Android Studio | Android Developers* [online]. Google LLC, 2024 [cit. 2024-03-20]. Dostupné z: https://developer.android.com/tools/adb.
46. *Overview | Android Open Source Project* [online]. Google LLC, 2024 [cit. 2024-03-08]. Dostupné z: https://source.android.com/docs/core/architecture/partitions.
47. MUELLER, Bernhard; SCHLEIER, Sven; WILLEMSEN, Jeroen; HOLGUERA, Carlos. *Android: Accessing App Data Directories - OWASP Mobile Application Security* [online]. OWASP Foundation, 2023 [cit. 2024-03-23]. Dostupné z: https://mas.owasp.org/MASTG/techniques/android/MASTG-TECH-0008/.
48. MUELLER, Bernhard; SCHLEIER, Sven; WILLEMSEN, Jeroen; HOLGUERA, Carlos. *iOS: Accessing App Data Directories - OWASP Mobile Application Security* [online]. OWASP Foundation, 2023 [cit. 2024-03-23]. Dostupné z: https://mas.owasp.org/MASTG/techniques/ios/MASTG-TECH-0059/.

49. AYERS, Rick; BROTHERS, Sam; JANSEN, Wayne. *Guidelines on Mobile Device Forensics*. NIST Special Publication 800-101r1, 2014. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf.

50. FUKAMI, Aya; STOYKOVA, Radina; GERADTS, Zeno. A new model for forensic data extraction from encrypted mobile devices. *Forensic Science International: Digital Investigation*. 2021, roč. 38. Dostupné z DOI: 10.1016/j.fsidi.2021.301169.

51. *Logical Extraction Mobile Forensics* [online]. Cellebrite Digital Intelligence Glossary, 2022 [cit. 2024-03-13]. Dostupné z: https://cellebrite.com/en/glossary/logical-extraction-mobile-forensics/.

52. REIBER, Lee. *Mobile Forensics Investigation: A Guide to Evidence Collection, Analysis, and Presentation*. 2. vyd. McGraw-Hill Osborne Media, 2019. ISBN 978-1-26-013509-1.

53. *In-System Programming (ISP) For Mobile Device Forensics* [online]. Teel Technologies, © 2020 [cit. 2024-04-10]. Dostupné z: https://www.teeltech.com/mobile-device-forensics-training/in-system-programming-for-mobile-device-forensics/.

54. *O aplikaci WhatsApp* [online]. WhatsApp LLC, © 2024 [cit. 2024-04-16]. Dostupné z: https://www.whatsapp.com/about?lang=cs_CZ.

55. *WhatsApp Encryption Overview* [online]. 2023. [cit. 2024-04-17]. Bílá kniha. WhatsApp LLC. Dostupné z: https://scontent.fprg5-1.fna.fbcdn.net/v/t39.8562-6/384251896_820338303082371_8514785982310046047_n.pdf?_nc_cat=100&ccb=1-7&_nc_sid=e280be&_nc_ohc=f3xulTFbetUAb7DEV-0&_nc_oc=Adg7hFh3clNaP2h9-fUcfuZXNOOViGRUJmv-xmz6Z6WlBzy0W2jKu3GGNugU87FyoBPSmohaZ-1M227YW3oR0MZ5&_nc_ht=scontent.fprg5-1.fna&oh=00_AfBJTC8IDWBjFg2neu_XyUE6AkUqF7KU2iF2VwHqAAJnYQ&oe=6624C211.

56. *Koncové šifrování záloh* [online]. WhatsApp LLC, © 2024 [cit. 2024-04-17]. Dostupné z: https://faq.whatsapp.com/490592613091019?helpref=faq_content.

57. *Chaty uložené v databázi aplikace WhatsApp ve vašem telefonu* [online]. WhatsApp LLC, © 2024 [cit. 2024-04-17]. Dostupné z: https://faq.whatsapp.com/947033946530087/?helpref=hc_fnav&cms_platform=android.

58. *Jak zálohovat na účet Google* [online]. WhatsApp LLC, © 2024 [cit. 2024-04-17]. Dostupné z: https://faq.whatsapp.com/481135090640375/?helpref=hc_fnav&cms_platform=android.

59. *Jak zálohovat na iCloud* [online]. WhatsApp LLC, © 2024 [cit. 2024-04-17]. Dostupné z: https://faq.whatsapp.com/902477924463699/?helpref=hc_fnav&cms_platform=iphone.

60. MIKHAILOV, Igor. *WhatsApp in Plain Sight: Where and How You Can Collect Forensic Artifacts* [online]. Group-IB, 2019 [cit. 2024-04-20]. Dostupné z: https://www.group-ib.com/blog/whatsapp-forensic-artifacts/.

61. *Vinted – oblečení z druhé ruky* [online]. Google Play, 2024 [cit. 2024-04-23]. Dostupné z: https://play.google.com/store/apps/details?id=fr.vinted.

62. *WebKit | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-05-01]. Dostupné z: https://developer.apple.com/documentation/webkit.

63. *StoreKit | Apple Developer Documentation* [online]. Apple Inc., 2024 [cit. 2024-05-01]. Dostupné z: https://developer.apple.com/documentation/storekit.

64. *json — JSON encoder and decoder* [online]. Python Software Foundation, 2024 [cit. 2024-05-05]. Dostupné z: https://docs.python.org/3/library/json.html.

65. BREMBECK, Timo. *Welcome to xhtml2pdf's documentation!* [Online]. xhtml2pdf, 2023 [cit. 2024-05-05]. Dostupné z: https://xhtml2pdf.readthedocs.io/en/latest/index.html.

66. *TTFError at / Can't open file "C:\Users\-\AppData\Local\Temp\tmp52b4p0wi.ttf"* [online]. Stack Exchange Inc, 2023 [cit. 2024-05-06]. Dostupné z: https://stackoverflow.com/questions/72910094/ttferror-at-cant-open-file-c-users-appdata-local-temp-tmp52b4p0wi-ttf.

Obsah příloh

- `README.md` ..... stručný popis přiloženého média
- `script`
  - `README.md` ..... popis skriptu a jeho použití
  - `src` ..... adresář se zdrojovým kódem skriptu
  - `templates` ..... adresář s css souborem a adresářem s písmem
- `text`
  - `thesis` ..... adresář se zdrojovými soubory pro text práce ve formátu LaTeX
  - `thesis.pdf` ..... text práce ve formátu PDF
- `other`
  - `data_dirtree.pdf` ..... struktura adresáře s daty ve formátu PDF
  - `data.zip` ..... data použitá k analýze
  - `report.pdf` ..... dokument vygenerovaný skriptem