Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Ústav Aplikované Informatiky

Analýza historie komunikace softwarového prostředku Facebook

Analysis of communication history of Facebook software

Bakalářská práce

Vladimír Cimbůrek

Školitel: Ing. Jaroslav Kothánek, Ph. D.

2012

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce.

Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne

Cimbůrek V. (2012) Analýza historie komunikace softwarového prostředku Facebook. [Analysis of communication history of Facebook software. Bc. Thesis, in Czech] - 26 p. Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic

Anotace

Tato bakalářská práce se zabývá zkoumáním komunikační historie softwarového prostředku Facebook(jinak také sociální sítě). Práce se zaměřuje na zkoumání datových fondů na úrovni lokálních PC, vyhledání potřebných dat a jejich vyhodnocení, následované realizací aplikace, která slouží k automatickému vyhledání a přehlednému zobrazení těchto dat.

Abstract

This thesis deals with examining the communication history of Facebook software product (social site). Thesis aims at examining of local data piles, finding the correct data and their evaluation, followed by creation of application that is used for automatic search and proper display of that data.

Poděkování

Rád bych poděkoval vedoucímu této práce Ing. Jaroslavu Kothánkovi, Ph. D., za ochotu konzultace a odbornou pomoc při realizace práce.

Obsah

1 Úvod a cíle práce

1.1 Úvod

Tato práce se zabývá problematikou získání chatové komunikace ze softwarového prostředku Facebook (jinak také sociální síť). Z důvodů velkého rozšíření tohoto prostředku mezi populaci je toto téma velice aktuální. Jen v české republice ho používá téměř 3 800 200 uživatelů ve věkovém rozpětí od 13 do 100 let. Největší zastoupení má věková skupina mezi 25 až 34 let, celých 1 078 020 uživatelů. Genderové[1] rozložení je pak 51% žen a 49% mužů. Celkově využívá Facebook přibližně 973 214 440 uživatelů. [0] Toto číslo se samozřejmě neustále mění, ale trend je spíše stoupající.

Tato práce má za úkol seznámit se se službou Facebook a vyhledat na úrovni lokálního počítače záznamy o komunikační („chatové“[2]) historii a záznamy o příspěvcích na Facebook „zdi“[3]. Cílem tohoto hledání je nalézt důkazy o jakékoliv nelegální činnosti, které přes tento sw. prostředek mohla probíhat (např. vydírání, šikanování, obtěžování, dětská pornografie). Aplikace, která vznikne jako výsledek této práce by měla sloužit hlavně policejním orgánům a forenzním znalcům, kteří se zabývají vyšetřováním kriminální činnosti na internetu.

1.2 Cíle

1. Nalézt potřebná data na lokálních discích zkoumaného počítače
2. Analyzovat tato data pro potřeby vývoje následné aplikace
3. Navrhnout prostředek na automatické vyhledání a zpracování
4. Realizovat softwarovou aplikaci pro automatické vyhledání a přehledné zobrazení s možností následného exportu

[1] Genderové – dle pohlaví
[2] chat - je krátká komunikace nebo rozhovor dvou nebo více lidí prostřednictvím komunikační sítě. Uskutečňuje se vždy v reálném čase.
[3] Zeď – v originále Wall, je místo, kam její vlastník a jeho přátelé můžou posílat krátké zprávy, obrázky, videa. Zobrazování těchto příspěvků lze modifikovat.

2 Facebook

2.1 Co je to sociální síť

Sociální síť (v originále social network) neboli také komunitní síť, komunita, je skupina lidí, kteří spolu udržují kontakt. V dnešním internetovém světě se ale spíše jedná o službu, která svým registrovaným uživatelům umožňuje komunikovat s ostatními uživateli, sdílet informace či nápady, díla (ať už se jedná o umělecká či jakákoliv jiná) a mít svůj více či méně veřejný profil. Tyto služby můžou využívat jak soukromé, tak firemní subjekty. Někdy jsou za sociální sítě považována i různá diskuzní fóra. Tato práce se zabývá nejrozšířenější sociální sítí, sítí Facebook.

2.2 Historie sítě Facebook

Internetová sociální síť Facebook je nejrozšířenější sociální sítí na světě a je druhou nejnavštěvovanější internetovou stránkou, hned po vyhledávači Google.[1] Facebook byl vytvořen Harvardským studentem druhého ročníku počítačových věd Markem Zuckerbergem, společně s jeho spolužáky Eduardo Saverinem, Dustinem Moskovitzem a Chrisem Hughesem. Původní název nebyl Facebook, ale Facemash. Původním smyslem stránky Facemash byla oblíbená hra na Harvardské univerzitě „Hot or Not“[4]. Zuckerberg se naboural do počítačové sítě Harvardské univerzity a stáhl odtud fotografie, které měli studenti na kolejních průkazkách. Facemash byl spuštěn 28. Října 2003, ale byl zrušen pár dní poté, díky představitelům výše zmíněné univerzity. Zuckerberg byl dokonce obviněn z narušení bezpečnosti, porušení kopírovacích práv a z narušení soukromí ostatních studentů krádeží jejich fotek. Obvinění byla však později stažena. 4. Února 2004 Zuckerberg otevřel svojí síť znovu, tentokrát pod názvem Thefacebook. Později roku 2004 se stal ředitelem společnosti investor Sean Parker a přejmenoval společnost pouze na Facebook poté, co zakoupil doménu facebook.com. [1]

[4] Žhavý nebo Ne – porovnávání dvou studentů/studentek podle přitažlivosti.

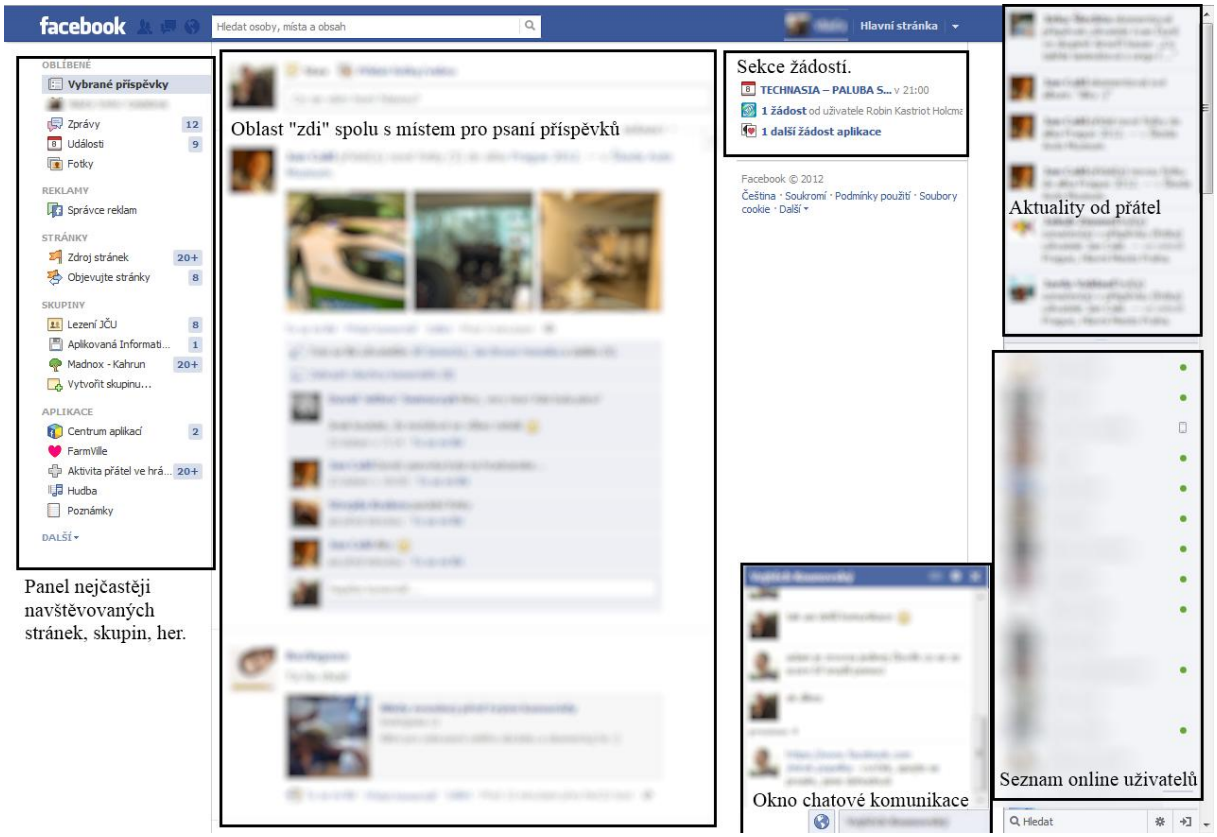
2.3 Portál Facebook.com

K využívání portálu Facebook uživatel potřebuje internetové připojení a prohlížeč. Při registraci uvede své osobní údaje a hlavně věk, z důvodů věkové hranice, která je licencí stanovena na 13 let (přesto však mnoho uživatelů internetu tvrdí, že je tato hranice často nedodržena[2]). Celý portál pracuje na OpenGraph, což je platforma, která byla vytvořena speciálně pro Facebook. Její systém je takový, že jako hlavní jednotka funguje uživatel, který přistupuje k ostatním elementům pomocí připojení. Vše je tak vázáno na vše a díky tomu lze realizovat komunikaci velice rychle. Open Graph, jak je realizovaný na Facebooku, dovoluje každé webové stránce fungovat stejně jako jakýkoliv jiný Facebook objekt.[3]

2.3.1 Základní funkce

Portál Facebook.com umožňuje využívat těchto služeb:

- Chatová komunikace mezi dvěma uživateli
- Mailová komunikace s jedním či více uživateli, se skupinou
- Video hovory
- Vytváření skupin a stránek (zájmové, fanouškovské)
- Nahrávání fotografií, videí, textu a odkazů
- Psaní příspěvků na „Zeď“
- Aplikace třetích stran
- Hry
- Vytváření událostí (srazy)
- Přístup z mobilního zařízení



Obrázek 1: Rozložení hlavní stránky facebook.com

2.3.2 Chatová komunikace a příspěvky na „zdi“

Hlavním cílem této práce je zkoumání chatové historie, proto je vhodné popsat tuto komunikaci blíže. Komunikace jako taková probíhá na dvou vrstvách. Jedna z nich je tzv. Instant Messaging[5]. Komunikace pak probíhá mezi dvěma uživateli okamžitě a nachází se v chatovém okně (viz *Obrázek 1.*). Zároveň jsou však jednotlivé zprávy posílány i prostřednictvím emailu. Každý uživatel má svojí emailovou adresu přidělenou Facebookem ve tvaru **<uzivatelskejmeno@facebook.com>.** Společnost Facebook toto zavedla ze dvou důvodů. První, mnohem snadnější uchovávání historie (dříve se chatová historie vůbec neukládala), druhý, sloučení zpráv a chatu. Před 15. Listopadem 2010 byly tyto funkce rozdělené, existoval systém zpráv a systém chatu.[4]

[5] IM – okamžité zprávy – komunikace v reálném čase

Každý uživatel v komunikaci je reprezentován svým jménem, které zadal při registraci. Dle uživatelova jména je možno zpřístupnit jeho profil, který má u společnosti Facebook. Uživatelův profil pak obsahuje informace spíše osobního charakteru, jako například:

- Jméno a příjmení
- Datum narození (viditelnost lze nastavit)
- Emailová adresa (viditelnost lze nastavit)
- Adresa původu a stávajícího bydliště (viditelno lze nastavit)
- Telefonní číslo (viditelnost lze nastavit)
- Údaje o zaměstnání (viditelnost lze nastavit)
- A další informace např. o zájmech, preferované hudbě atd.

Pro zpřístupnění profilu stačí do internetového prohlížeče napsat adresu:

www.facebook.com/uzivatelskejmeno

nebo

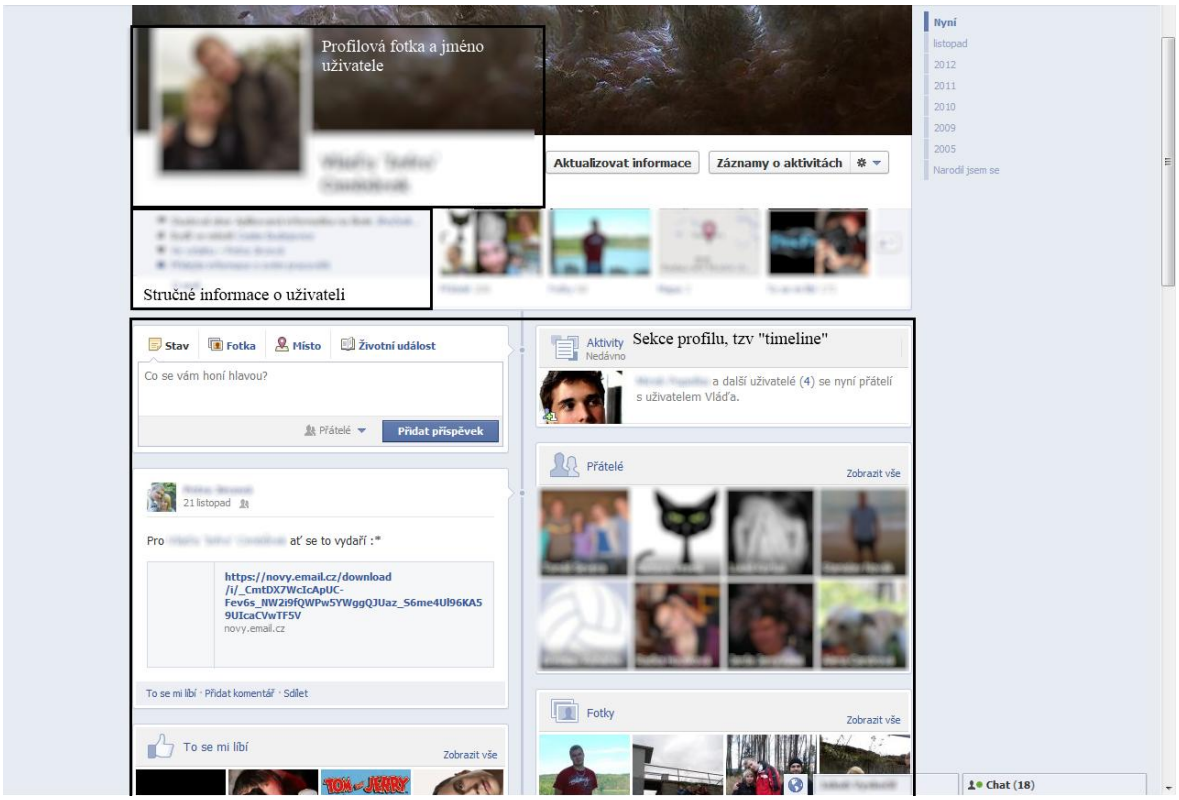
www.facebook.com/nick

Dvě verze jedné adresy existují z toho důvodu, že uživatelé Facebooku mají možnost nastavit si unikátní „nick“[6], dle kterého je mohou jejich přátelé a známí rychleji dohledat. Každý uživatel kdo nemá nick, je identifikován podle **uživatelského jména.** V případě duplicity jmen je pak uživatelské jméno doplněno číslovkou a vypadá takto:

jmeno.prijmeni.#

kde # je právě přidaná číslovka. Příklad uživatelského profilu je na Obrázku 2

[6] Nick – zkrácenina nickname – v překladu přezdívka



Obrázek 2: Příklad profilové stránky uživatele

Další způsob, jak je uživatel identifikován, je ID. ID je buď desetimístné, nebo patnáctimístné číslo, které jedinečně identifikuje uživatele na síti Facebook. Tento ID kód se nachází u každé zprávy a to jak ID kód odesílatele tak příjemce. ID kód identifikuje uživatele na hlubší vrstvě než jméno a to na vrstvě sociálního grafu, který je základním stavebním kamenem sítě Facebook. Při zadání jednoduchého http dotazu (zadáním adresy do adresního řádku internetového prohlížeče), lze zjistit některá veřejně dostupná data o uživatelích.

graph.facebook.com/<ID>

Výsledkem zadání této adresy s příslušným ID kódem do adresního řádku prohlížeče je výpis zobrazený na Obrázku 3.

Obrázek 3: Výpis prohlížeče po zadání adresy s ID.

Za předpokladu vlastnictví unikátního access tokenu[7] a zadání dotazu na **graph.facebook.com** by bylo možné získat více dat, ale z důvodu neukládání tohoto tokenu se tato metoda vylučuje. Chatová komunikace je k nalezení na lokálních discích.

Poslední neméně zajímavou složkou Facebooku jsou příspěvky na „zdi“ (z forenzního hlediska). Sem může vlastník účtu psát své poznatky, postřehy, libovolné informace stejně jako vkládat jakýkoliv audiovizuální obsah, který splňuje podmínky[8]. Stejnou možnost mají přátelé uživatele a i ostatní uživatelé na síti Facebook (toto jde upravit, aby příspěvky viděli například jen přátelé).

[7] Access token – přístupový klíč, unikátní řetězec.
[8] Obsah nesmí být urážlivý, explicitní, porušovat autorská práva

3 Analýza lokálních fondů

3.1 Identifikace dat na úrovni lokálního PC

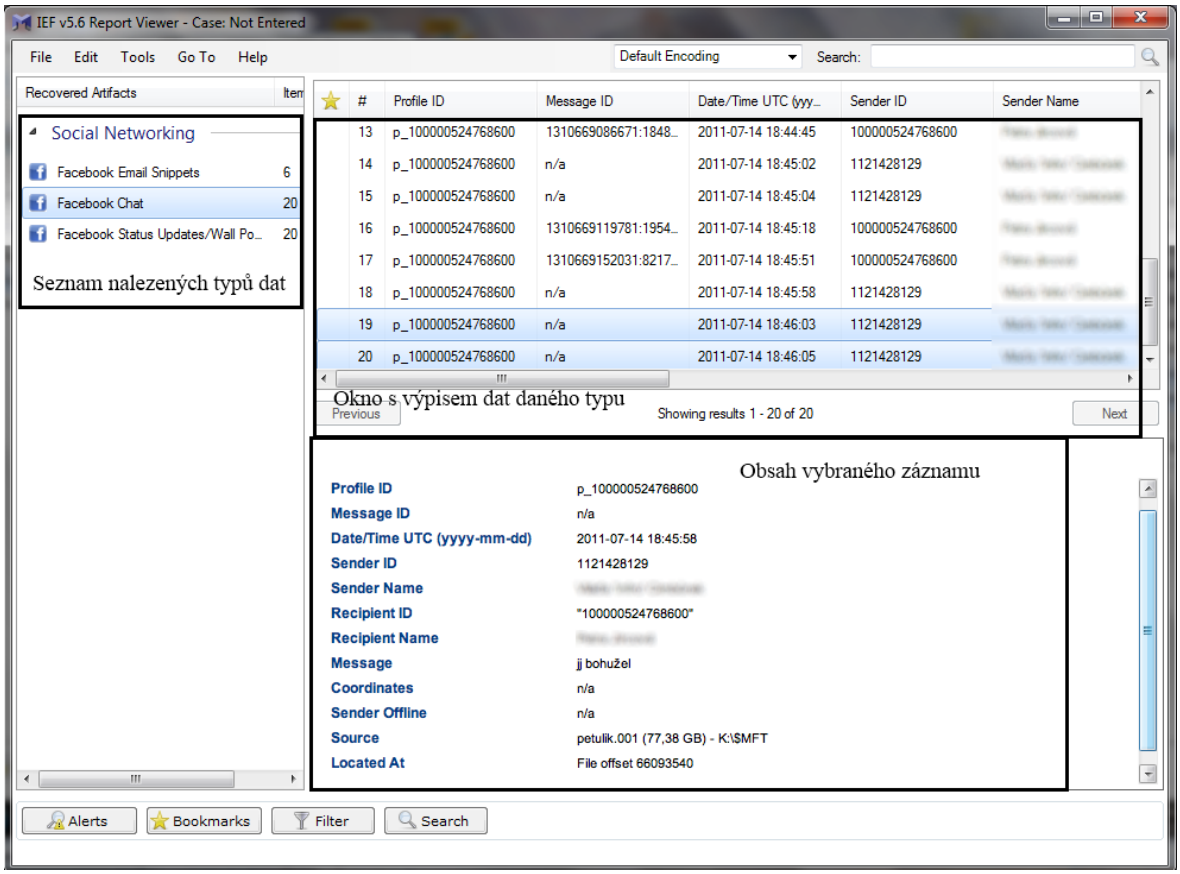
3.1.1 Použité prostředky

K identifikaci, nalezení a analýze dat byl využit PC s operačním systémem Microsoft Windows 7 a dále notebook s operačním systémem Microsoft Windows XP Home Edition. První stroj sloužil jako nástroj pro průzkum a zpracování, druhý byl použit jako cíl těchto analýz. První věc, kterou bylo nutno udělat, byla úplná bitová kopie systémového disku. Toto bylo provedeno softwarovým nástrojem **FTK Imager[5]** od společnosti **AccesData Group, LLC.** Jedná se o forenzní nástroj, který umožňuje vytvářet bitové kopie disků a jejich následné procházení stejně jako procházení lokálních disků. Stejný nástroj byl později použit k analýze nalezených dat. Tento proces je popsán v pozdějším odstavci.

3.1.2 Nalezení a identifikace dat

Druhým úkolem této práce bylo nalezení a identifikování dat, které obsahují záznamy o chatové komunikaci (nebo o příspěvcích na „zdi“). K tomuto účelu byl využit komerční softwarový prostředek **Internet Evidence Finder[6]** od společnosti **Magnet Forensics Inc. (dříve JAD Software).** Tento program je dostupný k vyzkoušení v Trial verzi, která je schopna dohledat záznamy o chatové historii na lokálním PC či na obrazu disku. Omezení Trial verze je takové, že se zobrazí pouze dvacet výsledků, pro identifikování dat a nalezení jejich umístění to však stačí. Výsledky se také uloží do souboru, který je poté možno otevřít v **Report Viewer**u[9] programu **Internet Evidence Finder.** Příklad výsledků je na *Obrázku 3.*

[9] Report Viewer – prohlížeč reportů – utilita programu Internet Evidence Finder



Obrázek 4: Okno prohlížeče záznamů s nalezenými daty.

Jak je z obrázku patrné, Internet Evidence Finder, díky svému komerčnímu původu je schopen nalézt veškerá potřebná data a poté zobrazit jejich obsah. Pro nás jsou důležité dva fakty. To, že se na disku nějaká hledaná data nacházejí a **kde** se nacházejí. Tento údaj je k nalezení v okně pod názvem Source. V případě, který je zobrazen na *Obrázku 3* se data nachází v souboru **$MFT**[10]**.** Při bližším prozkoumání byla nalezena tři umístění, kde se data nacházejí. Jednalo se o již zmíněný soubor **$MFT,** dále se jednalo o soubor **pagefile.sys**[11] a soubor **hyberfile.sys**[12]. Více o těchto souborech je v dalším oddíle.

[10] Master File Table – soubor kde se nachází záznamy o každém souboru či složce na disku.
[11] Pagefile.sys – stránkovací soubor systému Windows
[12] Hyberfile.sys – úložiště pro obsah paměti v případě hibernace

3.1.3 Umístění hledaných dat

Díky programu Internet Evidence Finder byla lokalizována tři umístění, kde se nachází hledaná data. Jedná se o:

- **$MFT**

  Jedná se o skrytý systémový soubor, takzvaný **Master File Table.** V principu se jedná o tabulku relační databáze, která obsahuje různé údaje o všech souborech a složkách na disku. Jedná se o „startovní bod“ kam systém přistupuje, když hledá nějaké soubory. Dá se o něm mluvit jako o „obsahové tabulce“. Vzdáleně se podobá **File Allocation Table**[13] systému FAT, ale MFT je mnohem více než jen seznam volných a zaplněných clusterů. Kdykoliv se vytvoří nový soubor či složka, v MFT se vytvoří nový záznam. Bohužel zde realizovaný aplikační prostředek nemá možnost přistoupit k tomuto souboru z důvodů omezení použitého programovacího jazyka.

- **Pagefile.sys (nebo také stránkovací soubor)**

  Jedná se o skrytý systémový soubor, kam si operační systém ukládá data, když mu již nestačí operační paměť RAM. Pakliže k tomu dojde a RAM je plná, systém začne ukládat „stránky“[14] na disk právě do souboru pagefile.sys. Ten se nachází v kořenovém adresáři systémového disku (tedy ve většině případů na C:). Velikost stránkovacího souboru bývá většinou stejná jako operační paměť. Doporučená velikost je pak 1.5x velikost operační paměti.

  Využití tohoto způsobu má ale jedno úskalí. Pakliže existuje mnoho procesů uložených v stránkovacím souboru a pak je systém rychle potřebuje, vše závisí na rychlosti čtení disku, která je vždy menší než čtení z operační paměti. Toto pak může systém zpomalovat.

  Pokud by byla analýza prováděna na lokálním disku při běhu systému, tento soubor by nebylo možno zpřístupnit z důvodu zamezení přístupu.

[13] File allocation table – alokační tabulka souborů
[14] Page – fragment virtuálního paměťového prostoru o velikosti 4 KB

- **Hyberfile.sys**

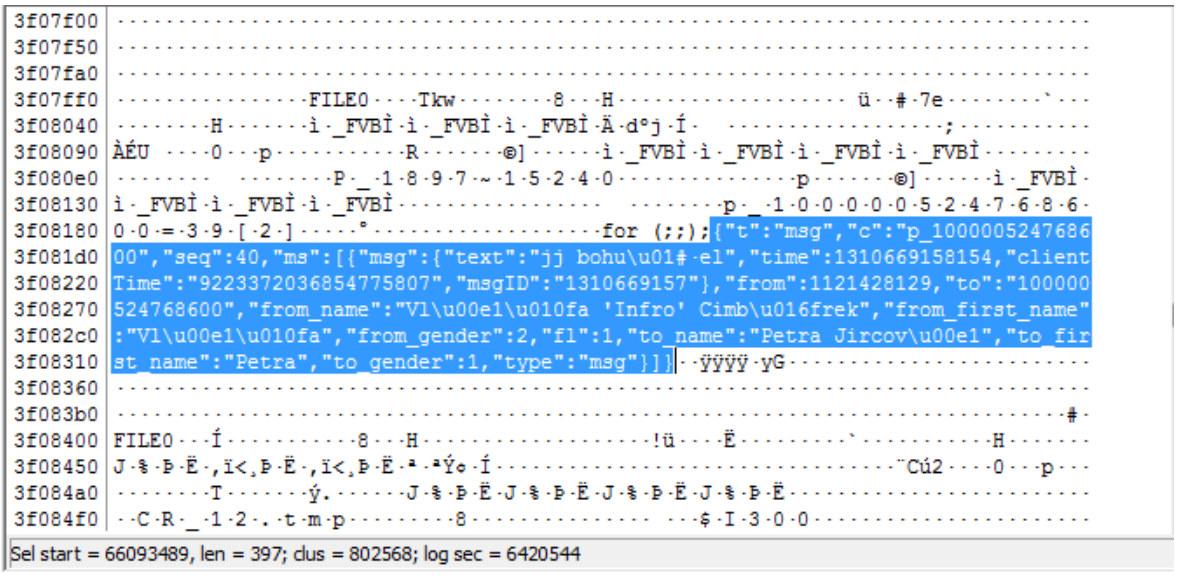
  Jedná se o skrytý systémový soubor, kam se ukládá obsah operační paměti při použití funkce systému Microsoft Windows hibernace. Tato funkce zkomprimuje obsah operační paměti a uloží ho do souboru hyberfile.sys, z kterého se pak při znovu zapnutí tentýž obsah znovu načte do operační paměti. Velikost hyberfile.sys je většinou 0.75x velikost operační paměti a nachází se taktéž v kořenovém adresáři systémového disku. Pokud by byla analýza prováděna na lokálním disku při běhu systému, tento soubor by nebylo možno zpřístupnit z důvodu zamezení přístupu.

- **Obraz operační paměti**

  Jedním z posledních umístění dat může být i operační paměť, ale aby bylo možno obraz zkoumat, byl by nutný přístup k zapnutému stroji a vytvořit obraz paměti, což umožňují pouze jiné forenzní nástroje.

3.2 Analýza nalezených dat

Po nalezení hledaných dat následovala etapa analyzování dat, aby existoval počáteční bod pro návrh aplikace, která bude zajišťovat hledání potřebných řetězců. Byl použit forenzní nástroj **FTK Imager** pro zpřístupnění jednotlivých umístění. Díky předchozí analýze programem Internet Evidence Finder bylo zjištěno, který řetězec může být použit. Výsledná data, která bude aplikace hledat jsou zobrazena na Obrázku 5.



Obrázek 5: Výpis programu FTK Imager s nalezeným řetězcem

Z obrázku je možno vidět, že nalezený řetězec obsahuje velké množství dat, zajímavé hlavně z forenzního hlediska. Vše začíná tagem[15] „msg“

[15] Tag – značka, element

Seznam údajů, které je potřeba nalézt

- „text“ : „“ – samotný řetězec, který je odeslanou či přijatou zprávou
- „time“ : „“ – čas kdy byla zpráva odeslána/přijata, je ve formátu
  který využívá operační systém UNIX, musí se převést
- „msgID“ : „“ – ID kód zprávy, dá se díky němu vidět zpráva na
  graph.facebook.com, ale je zapotřebí access token
- „from“ : „“ – ID uživatele, který zprávu odesílá
- „to“ : „“ – ID uživatele, který zprávu přijímá
- „from_name“ : „“ – jméno odesílatele
- „to_name“ : „“ – jméno příjemce
- „from_gender“ : „“ – pohlaví odesílatele
- „to_gender“ : „“ – pohlaví příjemce

A další údaje, jako například křestní jména a podobně. Se znalostí tvaru těchto řetězců je možno začít s návrhem a realizací aplikace.

4 Návrh a realizace aplikační části

4.1 Současné řešení

V tuto chvíli existuje pouze jedna vědecká práce zabývající se forenzním zkoumáním softwarového prostředku Facebook a ta se stala předlohou pro tuto bakalářskou práci. Jedná se o studii z roku 2011 od skupiny **Valkyrie-X Security Research Group.[7]**

Alternativou pro vyhledávání dat forenzní povahy může být použitý softwarový prostředek **Internet Evidence Finder** od společnosti **Magnet Forensic Inc.** Který je ale komerční a dostupný pouze k vyzkoušení v Trial verzi.

4.2 Návrh aplikace

Aplikace bude realizována v programovacím jazyce Java. Bude používat pouze dostupné prostředky v prostředí operačního systému Microsoft Windows, například nástroj **DiskPart,** který slouží k připojení obrazů disku ve formátu *.vhd. K vývoji bude použito vývojové prostředí NetBeans IDE 7.2.1. Požadavky na aplikaci vycházejí z provedených analýz.

4.2.1 Požadavky na aplikaci

Vytvořená aplikace musí být schopna splnit následující úkoly:

- Nalézt na úrovni lokálních disků soubory, které obsahují hledaná data
- Nalézt hledané řetězce a následně je zpracovat
- Přehledně vypsat nalezená data
- Umožnit uživateli data exportovat do souboru csv

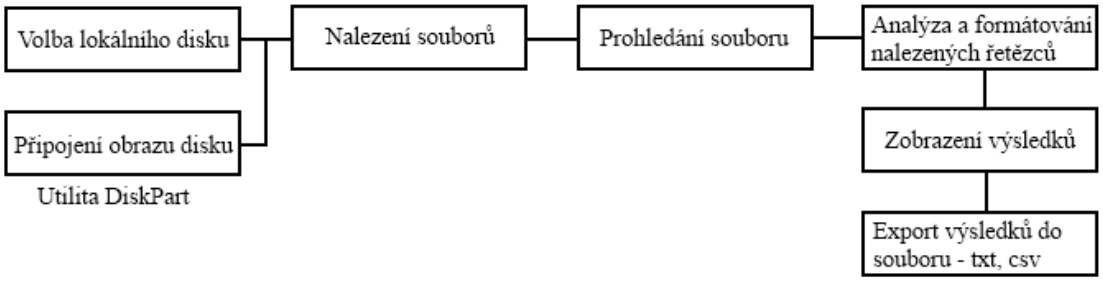
4.2.2 Metoda vývoje

Prvním krokem byla definice požadavků a cílů, které musí aplikace splňovat. Poté byly navrženy metody, které budou splňovat zadané cíle, a dle nich byl následně vytvořen návrh samotných tříd, které budou realizovat samotný běh aplikace.

4.3 Realizace

4.3.1 Návrh logického modelu

Navržený model zobrazuje jednotlivé kroky programu, které slouží jako předloha pro samotnou aplikaci. Podle jednotlivých kroků byly realizování třídy aplikace



Obrázek 6: Logický model aplikace

V prvním kroku je uživatel vyzván k zvolení úložiště, na kterém chce hledání provádět. Má možnost hledat data na lokálním disku nebo na bitové kopii disku. V případě obrazu disku má uživatel dvě možnosti. Buď si připojí disk sám, například pomocí externího programu pro připojování obrazů disků či pomocí forenzního nástroje jako je **například FTK Imager**, nebo, pokud je obraz disku ve formátu vhd[16], aplikace použije zadaná data (umístění souboru, písmeno ke kterému chce uživatel jednotku připojit) pro vygenerování skriptu pro nástroj **DiskPart,** který je součástí operačních systému Microsoft Windows XP a výše. Tento nástroj pak podle skriptu sám obraz disku připojí jako normální disk.

[16] VHD – virtual hard drive - formát obrazu disku

Následuje samotné prohledání zadaného média na výskyt souborů, o kterých bylo zjištěno, že obsahují hledaná data. K tomuto účelu není potřeba žádného jiného nástroje či knihovny, o všechno se starají metody, které obsahuje samotný programovací jazyk Java. Podrobnosti o metodách, které hledání umožňují, se nacházejí v detailnějším popisu jednotlivých tříd programu.

Po nalezení potřebného souboru následuje samotné prohledání souboru, které má za úkol nalezení řetězců, které obsahují informace, které aplikace hledá. Jak řetězec vypadá, bylo popsáno v kapitole 3.2 Analýza nalezených dat.

Při nalezení hledaného řetězce či hledaných řetězců aplikace zavolá metodu analýzy a formátování řetězců. Nalezený řetězec je nutno nejdříve analyzovat a naformátovat, aby bylo možno ho zobrazit. Toto se provádí pomocí páru klíč – hodnota. Jednotlivé klíče jsou také popsány v kapitole 3.2 Analýza nalezených dat. Je potřeba řetězec rozkouskovat podle určitého výrazu a spárovat pár klíč a hodnota. Tyto páry jsou pak uloženy do kolekce, kde s nimi můžeme pracovat dále, hlavně je zobrazit.

Po analýze a formátování následuje zobrazení výsledků. Aplikace nalezené výsledky zobrazí ve svém okně k náhledu uživateli. Zobrazená data si může sám uživatel zkopírovat či zvolit volbu exportu (uložení dat do souboru).

Pakliže si uživatel přeje uložit data do souboru, má na výběr ze dvou možností. Buď do textového souboru, nebo do souboru formátu CSV[17], pro pohodlnější zobrazení například v tabulkovém programu společnosti Microsoft, Microsoft Office Excel.

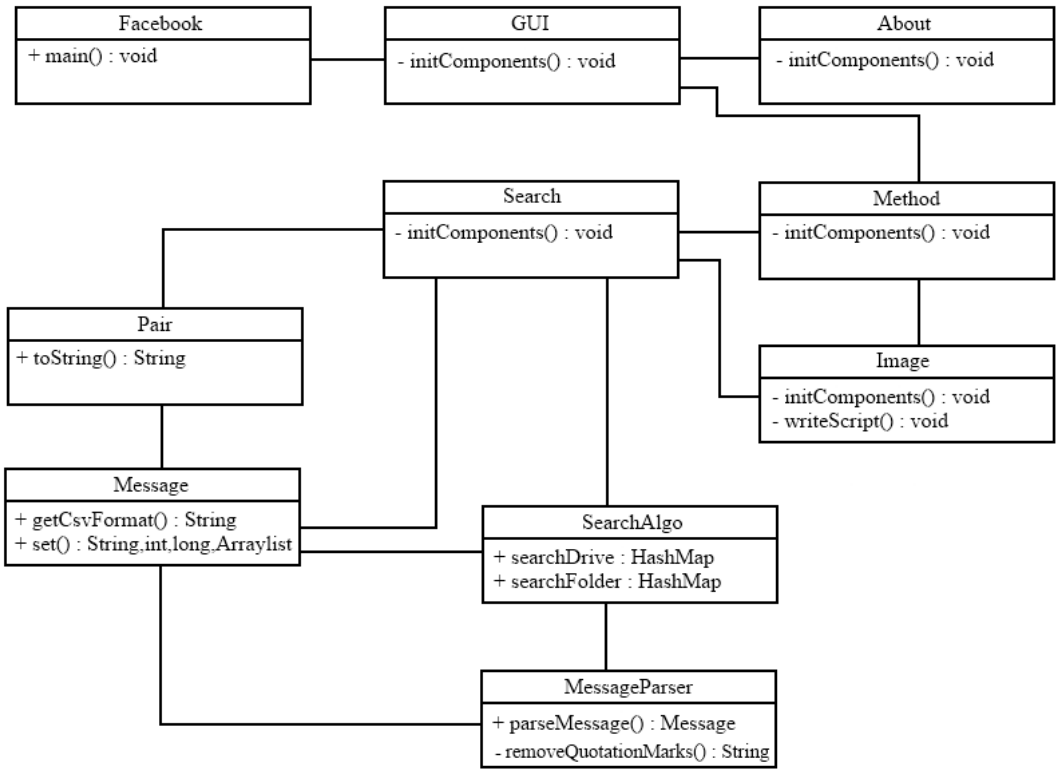
[17] CSV – formát souboru tabulkového programu společnosti Microsoft, Microsoft Office Excel

4.3.2 Programové třídy aplikace

Samotná aplikace bude obsahovat tři druhy programových tříd. Jednotlivé třídy realizují odlišné úkoly. Druh třídy, který se stará o zobrazování a vykreslení uživatelského rozhraní je typu grafická. Tyto třídy ulehčují práci uživateli a předávají si mezi sebou informace o volbách, které uživatel provedl. Další typ třídy je typ Objektu. Tyto třídy v sobě nesou obsah objektů, s kterými později pracujeme. Například třída **Message.java** obsahuje objekt, v kterém nalezneme celou jednu nalezenou zprávu. Třída **Pair.java** pak nese informace o obsahu této zprávy a jejích parametrech. Posledním typem tříd jsou třídy typu algoritmické. Tyto třídy v sobě nesou celé jádro aplikace a starají se o práci se soubory, načítání obsahů souborů, procházení disků, formátování nalezených řetězců a podobně. Bez těchto tříd by program vůbec nefungoval.

4.3.3 Realizace jednotlivých tříd

Po navržení logického modelu a jednotlivých postupů, byl vytvořen následující diagram programových tříd, tak jak jsou implementovány v samotné aplikaci. Diagram obsahuje základní metody uvnitř tříd včetně jejich návratových hodnot. Vazby v diagramu plně odpovídají vazbám jednotlivých tříd v samotné aplikaci.



Obrázek 7: Diagram programových tříd

- **Třída Facebook.java**
  - Toto je hlavní třída celé aplikace, takzvaná Main Class. Ta sama o sobě neobsahuje žádné metody, vytváří instanci třídy GUI, která se stará o původní grafické okno programu.

- **Třída GUI.java**
  - Tato třída se stará o hlavní a startovní grafické okno, které uživatel uvidí. Třída je rozšířením *javax.swing.JFrame,* a byla vytvořena pomocí Designeru programu NetBeans IDE. Veškerá další grafická rozhraní jsou také realizována pomocí Designeru, protože umožňuje rychlejší návrh formulářů (oken). Velikost okna nelze měnit. Okno obsahuje několik ovládacích prvků (tlačítek), kterými uživatel aplikaci ovládá. Tlačítka jsou objekty typu *javax.swing.JButton.* Tlačítka, která třída **GUI.java** obsahuje, jsou: Hledání chatové komunikace, O programu, Nápověda a Konec. Každé tlačítko obsahuje EventListener, pomocí kterého provádí naprogramované funkce. Většinou se jedná o vytvoření instance jiné třídy, pouze v případě tlačítka Ukončit pouze ukončí celý program. Tlačítko Hledání chatové komunikace vytváří novou instanci třídy **Method.java,** tlačítko O programu pak vytváří novou instanci třídy **About.java.** Všechna tlačítka volají po stisknutí a otevření nového okna metodu na zavření předchozího okna, tedy okna třídy **GUI.java.** Použitá metoda je *dispose(),* která nenásilně zavře okno typu *JFrame.*

- **Třída About.java**
  - Tato třída má za úkol jedinou věc. Zobrazení okna s informacemi o programu. Uživatel zde uvidí informace jako jméno autora, datum vydání a verze programu. Třída je opět rozšířením *javax.swing.JFrame,* protože znovu potřebujeme vytvořit nový formulář (JFrame, okno). Třída dále obsahuje dvě tlačítka typu *javax.swing.JButton,* a 6 popisků (*JLabel*) typu *javax.swing.JLabel.* Popisky slouží pro zobrazení textu. Nebylo zde využito prvku *JTextField* z *javax.swing.JTextField* a to z důvodu, že popisky byly tak krátké, že nebylo nutné používat tento větší prvek. Místo něj byly zvoleny právě popisky *JLabel.* Tlačítka se starají o ovládání okna.

- Tlačítka jsou: Konec a Zpět. Tlačítko konec, jak již jeho název napovídá, vypne celou aplikaci, zatímco tlačítko Zpět pouze zavře okno O programu a vytvoří novou instanci třídy **GUI.java.** Obě dvě tlačítka volají metodu dispose(), stejně jako u třídy GUI**.java**

- **Třída Method.java**
  - Tato třída dává uživateli na výběr, jaké umístění si uživatel přeje prohledat. Uživatel má na výběr ze dvou možností. Buď prohledat lokální disk, nebo prohledat obraz disku. **Třída Method.java** je rozšířením *javax.swing.JFrame.* V tomto okně nalezneme opět několik tlačítek, již popsaná tlačítka Zpět a Konec a dále tlačítka Hledat data na lokálním disku a Hledat data na obrazu disku. Tlačítka jsou typu *JButton* z *javax.swing.JButton.* Tlačítko Hledat data na lokálním disku rovnou vytvoří novou instanci třídy **Search.java**, přičemž posílá informace o cíli, kde se bude vyhledávat. V tomto případě se jedná o lokální disk, a protože u operačních systémů Microsoft Windows je většinou systémovým diskem disk s písmenem C, tak předáme tuto informaci. Pokud uživatel zvolí tlačítko Hledat data na obrazu disku, tak se vytvoří nová instance třídy **Image.java.** Samozřejmostí je volání metody *dispose()* pro zavření okna po zmáčknutí tlačítka.

- **Třída Image.java**
  - Tato třída se stará o připojování obrazů disků ve formátu **vhd.** Třída Image.java patří do grafického typu tříd, tedy je rozšířením *javax.swing.JFrame.* Obsahuje několik tlačítek *JButton* z *javax.swing.JButton.* Dále obsahuje roletu typu *JComboBox* z *javax.swing.JComboBox,* která obsahuje seznam písmen jednotek, kam by se měl obraz disku připojit. Po zvolení písmena jednotky se toto písmeno uloží do proměnné, aby bylo možno ho poslat do dalšího třídy pro zpracování. Tlačítko Připojit obraz otevírá okno pro výběr souboru *JFileChooser.* Zde uživatel zvolí svůj obraz disku ve formátu **vhd** a třída si zapíše jeho cestu do proměnné. Poslední komponentou je zaškrtávací tlačítko typu *JCheckBox*. Toto tlačítko uživatel zaškrtne, pouze pokud má již obraz disku připojen pomocí jiného nástroje, nebo se jedná o obraz jiného formátu, který je nutno připojit jinak.

- Po výběru obrazu třída **Image.java** volá metodu pro generaci skriptu pro nástroj **DiskPart** a následné zavolání dávkového souboru, který podle informací ve skriptu připojí disk na zvolené písmeno jednotky. Samotný skript obsahuje příkazy pro výběr souboru s obrazem disku, cestu k samotnému souboru a dále pak příkaz pro připojení. Disk se připojí na první volné písmeno, které operační systém má. Toto písmeno disku se pak odesílá do metody **Search.java**, kde slouží jako cíl pro prohledávání.

• Třída Search.java

- Tato třída zobrazuje nejdůležitější okno z celého programu, a to okno kde budou zobrazeny samotné výsledky vyhledání. Tato třída již má všechny potřebné údaje k hledání od ostatních tříd jako je **Method.java** a **Image.java**. Jedná se o poslední grafickou třídu, dále následují pouze třídy s algoritmy pro hledání a formátování, analýzu. Základem je stále JFrame, do kterého byl umístěn objekt typu JScrollPane, pro umožněné rolování výsledků a následně do *JScrollPane* byl umístěn objekt typu *JTextArea*, který již zobrazuje samotné výsledky. Výsledky jsou zobrazeny tak, že se do objektu JTextArea vkládají výsledné řetězce, již naformátované. Dále zde nalezneme ovládací prvky ve formě tlačítek. Nejdůležitější jsou dvě tlačítka a to Start a Uložit. Tlačítko start spouští vyhledávání, vytváří totiž novou instanci třídy **SearchAlgo.java**, která realizuje samotné vyhledávání potřebných řetězců v souborech a tlačítko Uložit uloží vyhledaná data do souboru ve formátu csv. Ukládání do formátu csv se provádí pomocí metody v třídě **Message.java**. Nejprve se uloží popisky jednotlivých hodnot pro větší přehlednost při dalším zkoumání uloženého výpisu. Poté se zavolá metoda, která vezme nalezené zprávy již ve správném formátu, rozdělí jednotlivé hodnoty středníky a ty poté uloží do souboru. Jednotlivé hodnoty jsou v korespondujících sloupcích seřazeny dle příslušných popisků. Výsledný soubor **output.csv** se pak nachází ve stejném adresáři kde je uložena aplikace.

- **Třída SearchAlgo.java**
  - Tato třída je základem celého programu. Stará se o procházení disku, jeho adresářů a souborů a zároveň hledá potřebné řetězce. Procházení disku se provádí rekurzivně, proto je možné projít celý strom a prohledat každý soubor na disku na výskyt hledaných dat. Při prohledávání souboru se postupuje následovně. Nejdříve se otevře samotný soubor, poté se načte první řádek souboru a otestuje se na výskyt hledaného řetězce, respektive jeho začátku. Pakliže řádek řetězec neobsahuje, algoritmus načte další řádek a pak znovu, dokud hledaný řetězec nenalezne. Pokud při hledání řetězce dosáhne konce souboru, tak čtení přeruší a pokračuje dalším souborem. Pokud ale hledaný začátek nalezne, prohledá zbytek řádku na výskyt ukončovacího řetězce. V případě nenalezení tohoto uzavíracího řetězce načítá další řádky, dokud ho nenalezne. Ve chvíli kdy algoritmus nalezne celý řetězec, tak ho uloží do kolekce, kde se uchová pro další zpracování. Toto se provede se všemi řetězci, které se nacházejí v souboru, a pokračuje se dalším souborem, dokud není projit celý disk. Následně se volá metoda ze třídy **MessageParser.java,** která se postará o naformátování nalezených řetězců.

- **Třída MessageParser.java**
  - Tato třída se stará o formátování nalezených řetězců do tvaru, ve kterém je možno nalezený řetězec zobrazit. To znamená, že se načte jeden řetězec, který obsahuje celou hledanou zprávu a rozdělí se dle daných klíčů. Klíče, dle kterých se řetězec dělí, byly blíže probrány v kapitole 3.2 Analýza nalezených dat. Základním oddělovacím prvkem byla zvolena následující dvojice znaků**[,"],** protože tento prvek byl při analýze nalezen, že se skutečně nachází mezi každou dvojicí klíč – hodnota. Po rozdělení řetězce zavolá metoda na odstranění uvozovek, toto je potřeba z důvodu lepšího zobrazení. Následuje spárování dvojice klíč a hodnota, poté následuje uložení těchto dvojic do objektu třídy Message. Celý algoritmus pak tento objekt vrátí metodě, která tento algoritmus volala, v tomto případě algoritmu ze třídy **SearchAlgo.java.**

- **Třída Message.java**
  - Tato třída v sobě má všechny parametry hledané zprávy. Obsahuje metody set, díky kterým lze po jejich zavolání nastavit hodnotu parametru (klíče) z libovolného místa v programu. Dále pro potřeby třídy **SearchAlgo.java** obsahuje i kolekci všech parametrů v jednom objektu pro snadnější přistupování. Třída obsahuje pouze jednu metodu, a to metodu, která se v okně Search stará o ukládání do souboru ve formátu csv. Při zavolání konstruktoru této třídy se do proměnných uloží popisky jednotlivých párů klíč – hodnota.

- **Třída Pair.java**
  - Tato třída obsahuje parametry pro pár klíč – hodnota. Stará se o spárování popisku a hodnoty, kterou získá z nalezeného řetězce. Obsahuje dvě metody get, díky kterým můžeme získat hodnotu z dvojice klíč – hodnota. Dále obsahuje takzvaný Override[18], díky kterému můžeme využít název a funkci jiné metody, ale s upraveným obsahem a tedy i funkčností.

4.4 Testování

Testování je nezbytnou součástí vývoje každé aplikace, proto i zde vytvářená aplikace podstupovala řadu testování, a to jak při vývoji, tak i po dokončení samotné aplikace.

4.4.1 Uživatelské testy

Nejúčinnější metodou jak otestovat aplikaci bylo její rozeslání úzkému okruhu uživatelů, kteří plnili funkci testerů. Vybrány byly celkem 3 osoby, které nezávisle na sobě testovali jednotlivé funkce programu. Jednalo se tedy o 3 nezávislé stroje a 3 nezávislé operační systémy. Jednotlivé stroje měly různé operační systémy, všechny od společnosti Microsoft a to: Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7. Dále bylo využito i virtuálního stroje s operačním systémem Microsoft Windows XP Professional. Toto byl přídavek k již prováděnému testování ze strany vývojáře, které bylo prováděno opakovaně v různých etapách vývoje. Nejdříve bylo testováno grafické rozhraní, postupně byly přidány testy připojování obrazů, funkčnosti vyhledávacích algoritmů a také funkčnosti formátovacího algoritmu.

[18] Override – přepsání

5 Návrhy pro budoucí rozšíření

Během vývoje aplikace a na základě poznatků od testerů a návrhů od vyučujících z Ústavu aplikované informatiky byly identifikovány oblasti či funkce, které by mohly být rozšířením této aplikace do budoucna.

- Podpora pro ostatní operační systémy
    - V tuto chvíli je možné aplikaci využívat pouze na platformě Microsoft Windows z důvodu použití nástroje, které tyto operační systémy obsahují a dále z důvodu samotné konstrukce programu. Ostatní operační systémy jako například Linux mají naprosto jinou adresářovou strukturu, proto by bylo nutné modifikovat vyhledávací algoritmy
- Podpora vyhledávání v historii externích programů
    - Jedná se například o ICQ, Skype, qip
- Možnost přistoupit k datům na profilu uživatele a to i k citlivým datům
    - Bylo by nutné získat unikátní Access Token, který umožňuje přístup i k ostatním datům, ne jen k těm veřejně dostupným
- Podpora pro připojování jiných formátů obrazů disků
    - Momentálně je zde omezení ze strany využitého nástroje pro připojování obrazů
- Podpora dalších formátů pro export
    - Například export do databáze
- Podpora vyhledávání v mobilních zařízeních

6 Závěr

6.1 Shrnutí

Tato bakalářská práce se úspěšně seznámila se softwarovým prostředkem Facebook, zdokumentovala jeho funkce a možnosti. Na základě studie vědecké práce skupiny byly navrženy možnosti získání dat z lokálních fondů. Díky znalosti těchto možností byla navržena aplikace pro automatické vyhledání a zobrazení hledaných dat. Tato aplikace byla řádně otestována a zdokumentována.

6.2 Konečné hodnocení

Bylo úspěšně dosaženo všech stanovených cílů této práce. Práce odpovídá zadání a požadavkům, které byly stanoveny. Aplikace vytvořená v rámci této práce, představuje přínos v ohledu zautomatizování procesu vyhledávání dat forenzní povahy z lokálních fondů. Na základě analýzy současných řešení lze prohlásit, že se jedná o jediný automatický prostředek s volnou licencí, který zkoumá lokální datové fondy do určené hloubky.

7 Seznam použité literatury

7.1 Internetové portály

[0] Social Bakers. *Social Bakers* [online]. 2012 [cit. 2012-12-12]. Dostupné z: http://www.socialbakers.com/countries/continents/

[1] Alexa. *Alexa* [online]. 2012 [cit. 2012-12-12]. Dostupné z: http://www.alexa.com/topsites/global

[2] About.com: Inventors. BELLIS, Mary. *About.com: Inventors* [online]. 2012 [cit. 2012-12-12]. Dostupné z: http://inventors.about.com/od/fstartinventions/a/Facebook.htm

[3] PECKHAM, Matt. Time.com: Techland. In: *Time.com: Techland* [online]. 2011 [cit. 2012-12-12]. Dostupné z: http://techland.time.com/2011/05/10/shocker-millions-of-facebook-users-underage/

[4] YAROW, Jay. Bussines Insider. In: *Bussines Insider* [online]. 2010 [cit. 2012-12-12]. Dostupné z: http://www.businessinsider.com/facebook-new-email-2010-11

[5] Facebook: Developers. *Facebook: Developers* [online]. 2012 [cit. 2012-12-12]. Dostupné z: https://developers.facebook.com/docs/concepts/opengraph/overview/

[6] FTK Imager 3.1.1.8 – Trial verze, AccesData Group, LLC**.** [cit. 2012-12-12] Dostupné z: http://www.accessdata.com/support/product-downloads

[7] Internet Evidence Finder 5.6.8 – Trial verze, Magnet Forensics Inc. [cit. 2012-12-12] Dostupné z : http://www.magnetforensics.com/products/internet-evidence-finder/

[8] WONG, Kelvin, Anthony C. T. LAI, Jason C. K. YEUNG, W. L. LEE a P. H. CHAN. *Facebook Forensics* [online]. 2011 [cit. 2012-12-12]. Výzkumná práce. Valkyrie-X Security Research Group. Dostupné z: https://www.fbiic.gov/public/2011/jul/facebook_forensics-finalized.pdf.

[9] ING. KOTHÁNEK JAROSLAV. Ph.D. *Znalecká a detektivní kancelář* [online][2010] [cit. 2012-12-12] Dostupné z : http://www.it-znalec.cz/

8 Přílohy

[0] CD obsahující tuto práci v elektronické podobě, zdrojové kódy, aplikaci a manuál

[1] Uživatelský manuál k aplikaci Facebook Forensic Tool

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Ústav Aplikované Informatiky

Příloha č. 1

Uživatelský manuál k aplikaci

Facebook Forensic Tool

Příloha k bakalářské práci

Vladimír Cimbůrek

Školitel: Ing. Jaroslav Kothánek, Ph. D.

2012

Obsah

1 Úvod

Tato aplikace je součástí a produktem bakalářské práce, která se zabývá zkoumáním komunikační historie softwarového prostředku Facebook. Aplikace slouží k prozkoumání a přehlednému zobrazení komunikačních zpráv výše zmíněného prostředku.

Aplikace je schopna:

- Prozkoumat lokální disky či obrazy disků pro výskyt komunikační historie
- Přečíst a zpracovat tuto komunikaci pro přehledné zobrazení
- Exportovat nalezená data do souboru ve formátu csv

2 O aplikaci

2.1 Doporučené systémové požadavky

Pro spuštění aplikace a její bezproblémový chod by měl uživatelův PC splňovat tyto požadavky:

- Operační systém Microsoft Windows XP a vyšší
- Procesor AMD Athlon 2400 MHz nebo Intel Pentium 2 2200 MHz
- Operační paměť RAM 1024 MB a více
- Místo na pevném disku 1024 MB a více
- Nainstalované rozhraní Java SE Runtime Enviroment (JRE) verze 7u9 a vyšší, dostupné ke stažení z: http://www.oracle.com/technetwork/java/javase/downloads/index.html

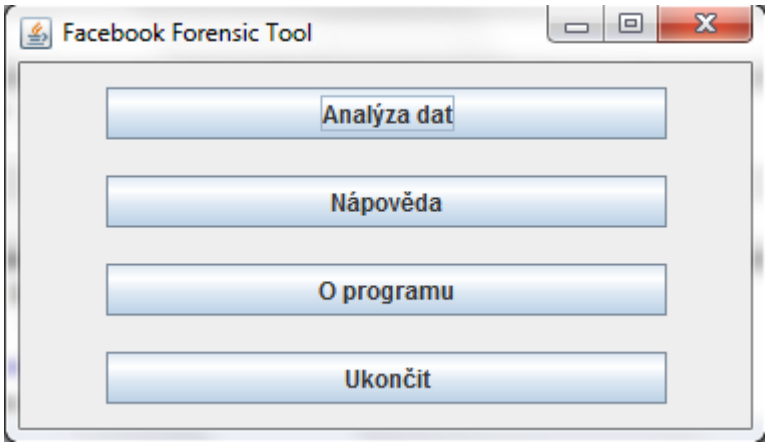
2.2 Spuštění aplikace

Aplikace se spouští pomocí souboru facebook.jar, který se nachází v kořenovém adresáři programu. Veškeré potřebné soubory jsou k nalezení také v kořenovém adresáři a je velice doporučeno je neměnit.

3 Ovládání aplikace

3.1 Hlavní okno aplikace

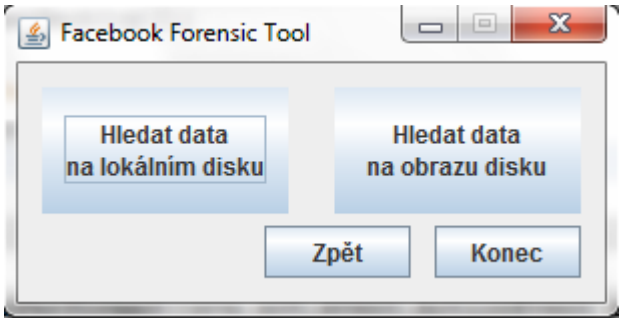
Při spuštění programu se uživateli otevře okno s hlavní nabídkou programu. První položkou je Analýza dat. Toto tlačítko uživateli umožní vstoupit do nabídky výběru umístění, které může procházet. Následující tlačítko otevírá tento manuál v souboru manual.txt. Manuál je pouze v textové podobě. Tlačítko O Programu pak zobrazí okno s informacemi o aplikaci.



Obrázek 1: Hlavní okno programu

3.2 Výběr umístění pro analýzu

Po přejití z hlavní nabídky do okna výběru umístění, uživatel má možnost vybrat si ze dvou typů umístění. Jedná se o lokální disk, nebo o obraz disku. Pakliže si uživatel vybere lokální disk, je mu rovnou zobrazeno okno vyhledávání, protože všechny potřebné informace již aplikace zná. Odlišná situace nastává, pokud zvolí, že chce prohledávat obraz disku. V tomto případě se otevírá nové okno pro obsluhu obrazů disků.



Obrázek 2: Okno výběru umístění pro analýzu

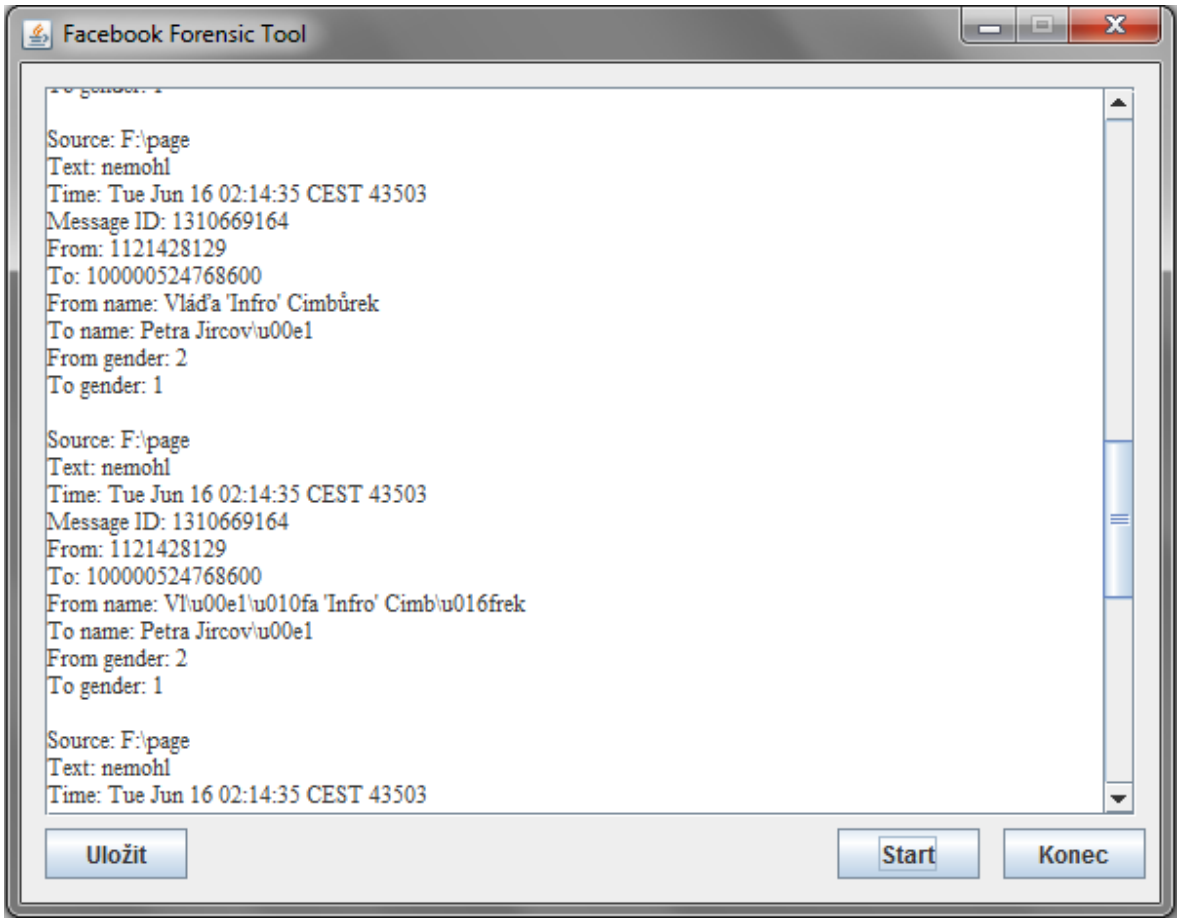
3.3 Práce s obrazem disku

Pokud si uživatel zvolí, že chce připojit obraz disku, zobrazí se okno z Obrázku 3. Zde má uživatel možnost vybrat svůj obraz disku, ten však musí být ve formátu vhd, jiný formát nástroj DiskPart nepřijme. Po zvolení obrazu disku pomocí grafického průzkumníka pokračuje uživatel do okna samotného hledání. Speciální situací je pak situace, kdy již má uživatel obraz připojen přes jinou aplikaci. Poté zaškrtne odpovídající políčko a vybere písmeno jednotky, ke kterému je obraz disku připojen, Analýza se pak provede na tomto disku.

Obrázek 3: Okno pro práci s obrazem disku

3.4 Vyhledávání

Samotný proces vyhledávání uživatel již nijak neovlivní, jediné co musí udělat je stlačením tlačítka start spustit proces hledání. Okno může nějakou dobu nereagovat, hlavně pokud je disk příliš velký. Toto je zcela v pořádku, aplikace prochází všechny soubory a složky, proto okno nereaguje. Pokud by došlo k výjimce, bylo by toto ohlášeno uživateli vyskakujícím oknem. Po skončení vyhledávání se uživateli zobrazí nalezené výsledky v hlavním okně a má následně možnost je uložit do souboru. Tento soubor se bude nacházet v kořenovém adresáři aplikace.



Obrázek 4: Okno vyhledávání a zobrazování výsledků

4 Export do souboru

Aplikace umožňuje uživateli exportovat nalezená data do souboru formátu csv. Soubor obsahuje popisky jednotlivých buněk a výsledky jsou do něj zapisovány, tak, že jedna řádka reprezentuje jednu nalezenou zprávu. Data jsou do souboru zapisována v následujícím formátu:

Hodnota prvního klíče; Hodnota druhého klíče; ....

Toto umožňuje zobrazení dat v tabulkovém editoru jako například Microsoft Office Excel.