Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

Automatizace forenzního zkoumání - forenzní audit pro účely zjištění porušování autorských práv

Bakalářská práce

Martin Dobiáš

Vedoucí práce: Ing. Jaroslav Kothánek, Ph.D.

České Budějovice 2015

Bibliografické údaje

Dobiáš M., 2015: Automatizace forenzního zkoumání - forenzní audit pro účely zjištění porušování autorských práv.

[Automation of Forensic Examining - Forensic Audit for the Purpose of Copyright Infringement Detection. Bc. Thesis, in Czech], Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce se zabývá forenzní analýzou souborů z obrazu pevného disku a registrů operačního systému Windows. Cílem je zhotovení nástroje k detekci instalovaných programů a potencionálně nelegálního softwaru.

Abstract

This bachelor thesis deals with the forensic analysis of files from a hard drive image and Windows registers. The aim is to provide a tool for the detection of installed programs and potentially illegal software.

Klíčová slova

bitová kopie, disk, obraz, image, software, analýza, autorská práva, libguestfs, hivex, pefile, forenzní zkoumání, audit

Keywords

image, disk, hard drive, software, copyrights, libguestfs, hivex, pefile, examining, forensic analysis, audit

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 24. 4. 2015

……..

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Jaroslavu Kothánkovi, Ph.D., především za trpělivost, ochotu a vynaložený čas.

OBSAH

Úvod ..... - 7 -

Motivace ..... - 8 -

Cíle práce ..... - 9 -

1 Forenzní analýza výpočetní techniky - pravidla, problematika a postupy při zajišťování a vyhodnocování dat ..... - 10 -

1.1 Digitální forenzní analýza ..... - 10 -

1.2 Postupy a pravidla při zajišťování dat ..... - 10 -

2 Autorská práva ..... - 12 -

3 Metodika zkoumání ..... - 13 -

3.1 Bitová kopie disku ..... - 13 -

3.2 Získávání informací ..... - 13 -

4 Návrh aplikace ..... - 14 -

4.1 Požadavky na aplikaci ..... - 14 -

4.2 UseCase diagram ..... - 14 -

4.3 Použité technologie ..... - 15 -

4.4 Návrh grafického prostředí ..... - 16 -

4.5 QThread ..... - 17 -

4.6 Knihovna libguestfs ..... - 18 -

4.7 Knihovna Pefile ..... - 18 -

5 Aplikace ..... - 19 -

5.1 Načtení bitové kopie ..... - 19 -

5.2 Analýza ..... - 19 -

5.2.1 Prohlížení registrů Windows ..... - 19 -

5.2.2 Prohlížení souborového systému ..... - 21 -

5.3 Výsledek analýzy ..... - 21 -

5.4 Export výsledků ..... - 25 -

6 Testování aplikace ..... - 26 -
7 Distribuce programu ..... - 27 -
8 Závěr ..... - 28 -
Seznam obrázků ..... - 30 -
Seznam tabulek ..... - 30 -
Příloha 1 - Programátorská dokumentace ..... - 31 -
Příloha 2 - Uživatelská příručka ..... - 35 -

ÚVOD

Nástupem éry výpočetní technologie a jejím postupným rozšiřováním po celém světě se stal počítač nedílnou součástí života každého z nás. Využíváme ho každý den, ať už ve formě chytrého telefonu, chytré televize, tabletu, laptopu, stolního počítače nebo dalších zařízení. Pro fungování každého přístroje je ale nutný software.

Software je každý program a operační systém počítače. Na rozdíl od věcí (počítače, telefonu) nám software nepatří a jeho použití je chráněno licenčním ujednáním a autorským zákonem. Porušením těchto ujednání a ustanovení se dopouštíme takzvaného softwarového pirátství a bereme na sebe odpovědnost za své protiprávní jednání.

V současné době je problematika počítačového pirátství bohužel velmi rozšířená. Mezi nejběžnější formy pirátství patří používání programů na více počítačích pouze s jednou licencí, kopírování programů pro známé, kopírování programů na internetová úložiště, nelegální prodej nebo využívání licencí určených pro soukromé účely při podnikání. Příkladem mohou být kancelářské balíčky (jako je *Microsoft Office*) nebo různé hry a antiviry. Vývojáři softwaru bojují proti těmto praktikám nejčastěji nutností zadat instalační klíč nebo vyžadují přístup k internetu pro ověření pravosti, ale i na tyto kontrolní mechanizmy piráti vyzráli například takzvanými „cracky“.

Pro potrestání viníků je zejména v trestním řízení potřebné získat důkazní materiál z informační techniky osob podezřelých z protiprávního jednání. K tomu policie nebo soudní znalci využívají digitální forenzní analýzu. S její pomocí je možné nalézt podezřelé soubory na disku a vytěžit registry operačního systému, v kterých nalezneme veškerý nainstalovaný software i s použitými licenčními čísly. Toto zkoumání je ale velmi zdlouhavé, z čehož vychází požadavek na automatizaci prostřednictvím snadno použitelné aplikace.

MOTIVACE

V České republice, ale i ve světě, je softwarové pirátství běžně rozšířený jev. Existuje mnoho internetových úložišť, které nabízejí nepřeberné množství filmů, her, programů a to zcela zdarma nebo za minimální poplatek. Z tohoto důvodu přicházejí nejenom tvůrci softwaru podstatnou měrou o zisky ze svých děl a nemohou tak dále rozšiřovat a zdokonalovat svůj produkt.

V současné době sice jsou dostupné programy detekující instalovaný software nebo instalační klíče, ale většinou je potřeba tyto programy instalovat nebo spouštět na aktivním operačním systému, který je potřeba analyzovat. Analyzovat aktivní operační systém není legislativně správná možnost a někdy ani není možná, například z důvodu zabezpečení přístupu hesly. Programy pracující na OS Linux nabízejí většinou velmi omezené možnosti analýzy nebo automatizace analýzy.

CÍLE PRÁCE

Cílem práce je vytvoření aplikace pro analýzu výpočetní techniky, pracující v operačním systému Linux. Tato aplikace načte bitovou kopii disku s OS Windows a vytěží veškeré dostupné skutečnosti dokumentující instalovaný software, softwarové prostředky, existenci cracků a další skutečnosti svědčící o porušování autorských práv. Výsledek analýzy bude dostupný v běžných formátech například typu *html*, *pdf*, *doc*. Vytvořená aplikace bude implementována do jednoúčelové live distribuce Linuxu. Její funkčnost bude ověřena na několika vzorcích výpočetní techniky. Na závěr bude zhodnocen přínos aplikace ve vztahu k řešení informační kriminality a ve vztahu k porušování autorských práv.

1 FORENZNÍ ANALÝZA VÝPOČETNÍ TECHNIKY - PRAVIDLA, PROBLEMATIKA A POSTUPY PŘI ZAJIŠŤOVÁNÍ A VYHODNOCOVÁNÍ DAT

1.1 DIGITÁLNÍ FORENZNÍ ANALÝZA

Digitální forenzní analýza (DFA) je forenzní věda, která je definována jako užití určitých vědecky odvozených postupů nebo osvědčených metod, které slouží ke sběru a izolování digitálních dat, jejich objektivnímu posouzení, identifikaci viníků a zdokumentování digitálních důkazů. [1]

DFA spadá do široké skupiny forenzních věd, kam patří i daktyloskopie, forenzní antropologie, forenzní balistika, forenzní entomologie, forenzní chemie, soudní lékařství, forenzní psychologie, forenzní genetika, písmoznalectví a další.

Je využívaná zejména policejními složkami nebo soudními znalci pro potřeby zkoumání různého digitálního materiálu (výpočetní techniky, hard disků, flash disků, paměťových karet, telefonních zařízení a v podstatě všech zařízení s pamětí). Výsledky analýz mohou být použity jako důkazní materiál u soudu, který řeší porušování licenčních ujednání softwaru. Mohou však posloužit i v privátní sféře – například k odhalení porušení interních pravidel firmy.

V této bakalářské práci bude podrobněji vysvětlena forenzní analýza obrazu pevného disku s operačním systémem Microsoft Windows.

1.2 POSTUPY A PRAVIDLA PŘI ZAJIŠŤOVÁNÍ DAT

V České republice není jasně určeno, jaký postup se má pro DFA zvolit. Každý soudní znalec si volí dle svého uvážení nejlepší způsob, kterým analýzu provede. Jediné co je třeba dodržet, aby důkazy získané touto analýzou byly přijaté soudem za platné, je nutnost dodržet následující pravidla. [2]

- INTEGRITA – Důkazní materiál nesmíme DFA porušit ani pozměnit. Důkaz by nebyl u soudu uznán. Pracovat proto můžeme pouze s daty v režimu pro čtení (read-only).
- LEGALITA – Informace musíme získávat pouze legálním softwarem a legální cestou. Nesmíme porušovat další zákony (telekomunikační zákon, zákon na ochranu osobních údajů, atd.)
- OPAKOVATELNOST – Všechny metody a postupy užité v DFA musí být popsány. To zaručí možnost přezkoumání analýzy jiným expertem, který by měl dojít ke stejným výsledkům.
- NEPODJATOST – Nezávislost subjektu provádějící DFA na zkoumaném předmětu.
- DOKUMENTACE – Všechen důkazní materiál musí být řádně zdokumentován. To je důležité například pro revizní zkoumání.

2 AUTORSKÁ PRÁVA

Český právní řád definuje autorská práva zákonem č. 121/2000 Sb, autorským zákonem. [3] Autorský zákon upravuje právní vztahy mezi tvůrci a uživateli softwaru. Stanovuje přitom také pojem „užívání díla“, který obsahuje právo na rozmnožování, rozšiřování, pronájem, půjčování, vystavování a sdělování díla. Tato privilegia má pouze autor díla, který je ale může uživateli propůjčit prostřednictvím licenčního ujednání.

Licenční ujednání je smlouva mezi autorem a uživatelem díla, kde autor díla poskytuje oprávnění užívat dílo jiné osobě (uživateli programu). Porušení autorských práv nastane tehdy, když je dílo užíváno bez souhlasu autora. V tomto případě má autor mimo jiné právo na náhradu ušlého zisku a vydání bezdůvodného obohacení ve smyslu § 40 autorského zákona [4]. Dále se neoprávněný uživatel autorského díla dopouští přestupku podle § 105a autorského zákona [5] (u podnikatelů správního deliktu podle § 105b autorského zákona [6]).

V konečném důsledku se může neoprávněný uživatel dopustit též trestného činu podle § 270 zákona č. 40/2009 [7], trestního zákoníku, pokud zasáhne do práv nikoli nepatrně, za což trestní zákoník považuje situace, kdy škoda dosahuje nejméně 5 000 Kč.

3 METODIKA ZKOUMÁNÍ

3.1 BITOVÁ KOPIE DISKU

Zkoumána bude bitová kopie neboli obraz disku. Tento obraz je identickou kopií disku, která obsahuje všechna použitá uživatelská i systémová nastavení, všechny programy a soubory. Bitovou kopii můžeme vytvořit například v Linuxu příkazem „dd“ nebo „dc3dd“. Obraz disku musíme připojit v režimu off-line, tzn. pouze pro čtení. Připojením obrazu s možností zápisu bychom mohli porušit jednu z hlavních zásad forenzní analýzy (viz bod 1.2 – integrita dat).

3.2 ZÍSKÁVÁNÍ INFORMACÍ

Pro potřeby této práce musíme nahlédnout do registrů operačního systému Windows. Registry jsou speciální databází operačního systému, kde se nacházejí informace o profilech uživatelských účtů, systémovém hardwaru a instalovaných programech. Prohledáním registrů získáme názvy a informace o softwaru, který byl instalován. Informace o MS Windows nebo MS Office obsahují i licenční klíče, které jsou v nečitelné podobě a musí se dekódovat.

Jako další krok prozkoumáme obraz disku a zjišťujeme přítomnost cracků a dalších podezřelých souborů. Toto provedeme za pomoci vyhledávání předdefinovaných slov: *crack*, *hack*, *aktivátor* a dalších. Podobně zjistíme programy, které nemají uložené informace v registrech Windows. V tomto případě budeme vyhledávat soubory s příponou *exe*.

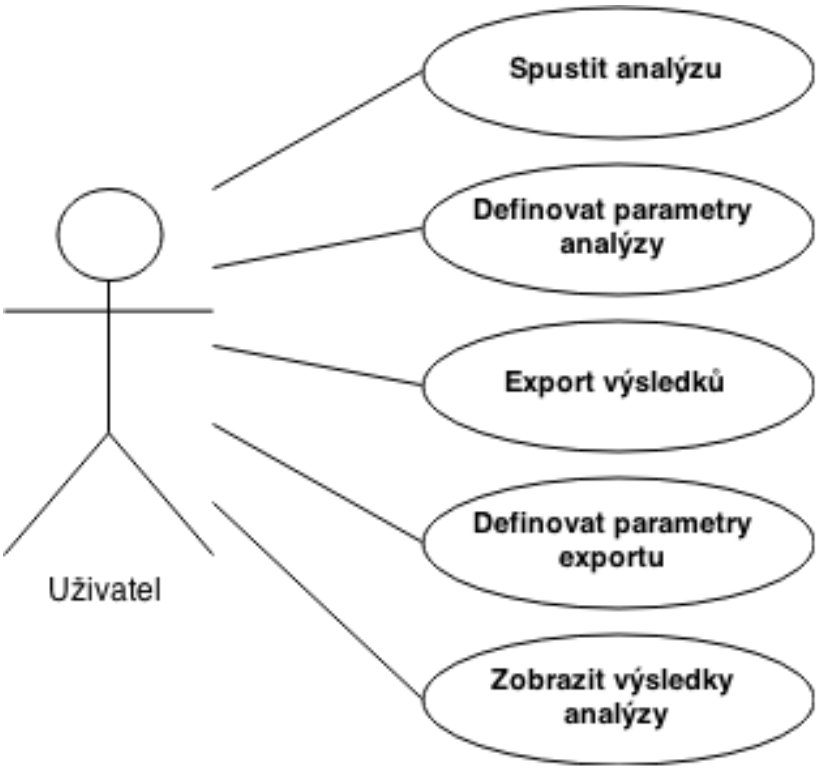
4 NÁVRH APLIKACE

4.1 POŽADAVKY NA APLIKACI

- Aplikace bude umístěna na jednoúčelové live distribuci Linuxu
- Bude schopna číst bitové kopie (obrazy) disků
- Vytěží informace o instalovaném softwaru a o souborech porušujících autorská práva
- Tyto informace bude schopna exportovat

4.2 USECASE DIAGRAM

UseCase diagram vychází ze zadání cílů této práce. Jedná se o zmapování požadavků uživatele na budoucí program.



Obrázek 1 - UseCase diagram

4.3 POUŽITÉ TECHNOLOGIE

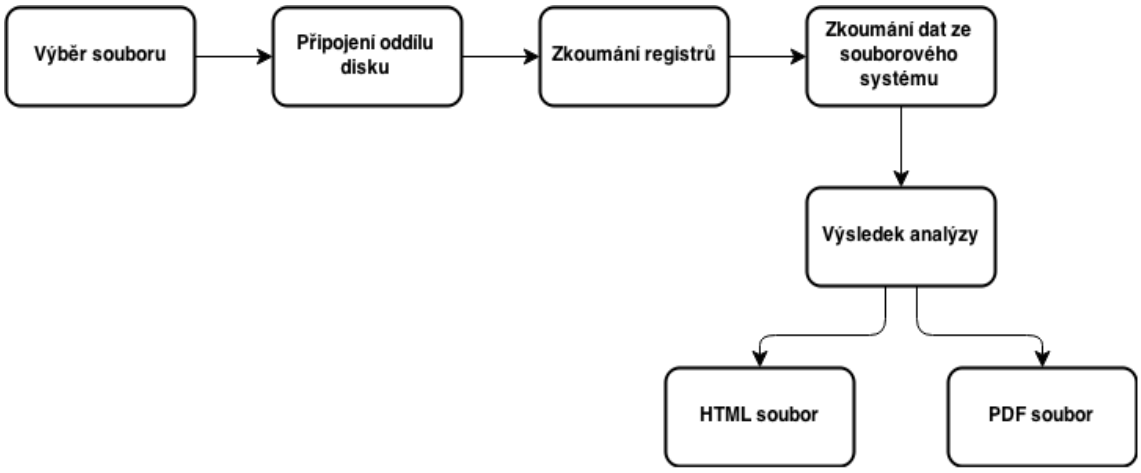
Aplikace je vytvořena pod operačním systémem Ubuntu 13.10 a naprogramována v jazyce Python 2.7.

Jako vývojové prostředí byl použit nástroj Monkey Studio IDE.

Pro vytvoření uživatelského rozhraní byl použit balíček nástrojů Qt4, který je použit prostřednictvím PyQt4 ve výše zmíněném vývojovém prostředí.

Pro práci s bitovými kopiemi pevných disků byla použita knihovna libguestfs a její podknihovna hivex, která je využita k prohlížení registrů systému Windows.

Pro analýzu *exe* souborů byla použita knihovna Pefile.



Obrázek 2 - Fáze programu

4.4 NÁVRH GRAFICKÉHO PROSTŘEDÍ

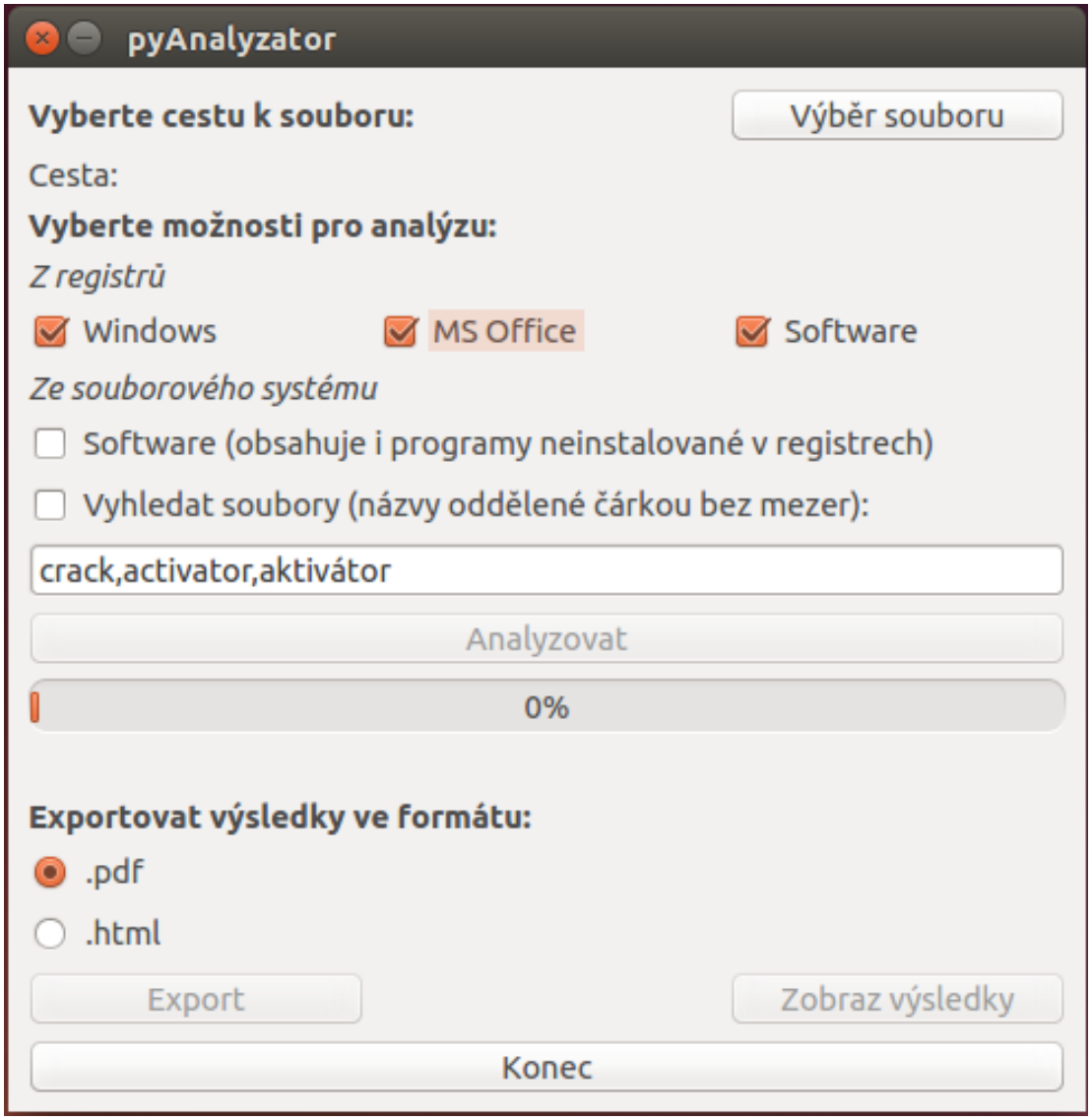
Pro vytvoření grafického prostředí byl použit zmíněný balíček Qt4 v programu Monkey Studio IDE. Okno programu má pevně danou velikost a pro jednoduchost ovládání obsahuje tři sekce.

Do první sekce patří výběr souboru, který je umístěn v horní části okna. Po stisku tlačítka „Výběr souboru“ je zobrazeno okno, ve kterém vybereme umístění souboru obrazu disku. Tento soubor musí být ve formátu *img*. Po výběru souboru se pro kontrolu zobrazí vybraná cesta a zaktivní druhá sekce.

V druhé sekci zvolíme, které informace chceme z vybraného obrazu získat. Na výběr máme pět možností (voleb analýz). První tři (Windows, MS Office, Software) jsou předem vybrané a získávají informace z registrů Windows. Zbylé dvě možnosti (Ostatní software, Vyhledat soubory) procházejí souborový systém. První z nich vyhledá všechny *exe* soubory na disku. Druhá zjišťuje shodu mezi uživatelem zadaným seznamem (umístěným pod poslední možností) a názvy souborů nebo složek na daném obrazu. Tyto analýzy jsou ovšem časově náročnější, a proto nejsou předem zvolené. Poté co zvolíme požadované vlastnosti, klikneme na tlačítko „Analyzovat“. Pod tímto tlačítkem je graficky zobrazen průběh a aktuálně zpracovávaná činnost analýzy.

Třetí část se zaktivní po ukončení analýzy. Stiskem tlačítka „Zobraz výsledky“ se zobrazí výsledek v okně prohlížeče. Export výsledků je umožněn ve formátech *pdf* a *html*. Vybereme požadovaný formát a stiskneme tlačítko „Export“. Následně je zobrazeno okno, ve kterém je možné určit cílové umístění a název souboru.

Ve spodní části je umístěno tlačítko „Konec“ pro ukončení programu.



Obrázek 3 - Grafické prostředí aplikace (GUI)

4.5 QTHREAD

Jedná se o součást balíčku PyQT4, která umožní rozdělit program do více vláken. Analyzátor je rozdělen celkově do tří vláken. První vlákno má na starosti GUI pro uživatele a export výsledků. Druhé vlákno je určeno pro načtení a analyzování obrazu disku. Třetí vlákno generuje pohybující se text. Druhé a třetí vlákno komunikují pomocí signálů s hlavním oknem a zobrazují informace probíhající analýzy. Bez tohoto rozdělení by měl uživatel po celou dobu analýzy „zamrzlé“ okno aplikace, které by nebylo možné ovládat.

4.6 KNIHOVNA LIBGUESTFS

Pomocí této knihovny můžeme pracovat s bitovými kopiemi pevných disků. Zajišťuje především načtení a připojení obrazu, možnost procházet souborový systém, prohlížet a otevírat soubory a složky. Zahrnuje také velké množství funkcí, které jsou dostupné z příkazového řádku.

Její součástí je knihovna hivex, která otevírá soubory registrů a prohlíží je ve stromové struktuře, podobně jako soubory a složky v souborovém systému.

Autorem této knihovny je Richard W. M. Jones.

4.7 KNIHOVNA PEFILE

Tato knihovna nám umožňuje přistoupit pomocí jazyka Python k obsahu souboru PE (Portable Executable). Do této skupiny patří například soubory s příponou *exe* nebo *dll*. Knihovna má tu výhodu, že na vstupu nemusí být uvedena cesta k souboru, ale místo ní může být použit buffer s načteným obsahem. Tato vlastnost nám umožní použít knihovnu Pefile na soubory připojené virtuálně knihovnou libguestfs, bez nutnosti připojovat obraz disku lokálně.

Knihovna je napsaná v jazyce Python a autorem je Ero Carrera.

5 APLIKACE

5.1 NAČTENÍ BITOVÉ KOPIE

Nejdříve uživatel vybere cestu k samotné bitové kopii. Zvolený soubor program otestuje na přítomnost diskových oddílů s operačním systémem Windows. Tyto oddíly se připojí v režimu pro čtení postupně, přičemž je každý oddíl analyzován samostatně. Cesta k souboru se od uživatele získá metodou buttonLoad_Click(). Bitová kopie je testována a připojována pomocí již zmíněné knihovny libguestfs a v programu její použití nalezneme v metodě Analyze().

5.2 ANALÝZA

Analýzu bitové kopie disku můžeme rozdělit celkem na dvě části a postup bude probíhat podle napsané metodiky z bodu 3. Před samotnou analýzou se prověří, jestli složka, ve které se nachází zkoumaný obraz disku, neobsahuje informace o jeho vytvoření. Typicky je to soubor stejného názvu jako bitová kopie, pouze s doplněnou příponou *info* nebo *nfo*. Tento soubor je načten a bude začleněn do výsledného *html* souboru. V průběhu analýzy se nalezené informace ze všech sekcí programu předem připravují a stylují pro výsledný *html* dokument. Pomocí signálů mezi vlákny aplikace se do hlavního okna promítá posun progressbaru a pod ním se zobrazuje aktuálně prováděná činnost.

5.2.1 PROHLÍŽENÍ REGISTRŮ WINDOWS

V této části je využita knihovna hivex, součást knihovny libguestfs, která nejprve načte soubor s registry, které jsou použity pro analýzu. Registry najdeme v souboru SOFTWARE a v souborech *ntuser.dat*. Cesta k prvnímu souboru je *<u>/Windows/System32/config/SOFTWARE</u>*. Soubory *ntuser.dat* jsou umístěny na jednom ze dvou cest v závislosti na operačním systému. V OS Windows Vista a novějších OS je cesta k těmto souborům *<u>/Users/username/ntuser.dat</u>*.

Pro kompatibilitu s dnes ještě stále používaným Windows XP je použito umístění */Documents and Settings/username/NTUSER.dat*. V obou případech je v cestě uveden název složky „username“, která značí jméno uživatele.

Po načtení registru Software se zavolá metoda aWindows(), která najde základní informace o instalovaném operačním systému. Registr má tyto informace uloženy v *Microsoft/Windows NT/CurrentVersion*. Součástí těchto informací je i položka Digital Product Id obsahující instalační klíč OS. Hodnota instalačního klíče je v položce zakódována a aplikace musí zavolat metodu DecodeProductKey(), která tuto hodnotu dekóduje a vrací zpět do metody aWindows().

Další následuje metoda aOffice(), která zjistí, zda registry obsahují záznamy o instalacích Microsoft Office a zapíše informace o všech instalacích. V registrech jsou tyto záznamy umístěny v *Microsoft/Office/xx.x/Registration*, kde „xx.x“ značí verzi těchto produktů (např. 10.0, 12.0,...). V programu je tato situace řešena vyhledáním všech podadresářů adresáře *Office* a v těchto adresářích je vyhledáván adresář *Registration*. Podobně jako metoda aWindows(), i tato metoda posílá položku Digital Product Id na dekódování, kde se vrací instalační klíč MS Office.

Třetí a poslední funkce, která pracuje s registry, je metoda aSoftware(). Zatímco předchozí dvě metody aWindows() a aOffice() využívaly pouze soubor registrů Software, tato metoda analyzuje i všechny registry s uživatelským nastavením *ntuser.dat*. V těchto registrech jsou informace o programech, které se instalovaly pouze pro konkrétní účty na operačním systému. Tyto soubory registrů program hledá v adresářích „Users“ a „Documents and Settings“. Adresáře se prohledají a v případě nalezení registru se tento soubor načte podknihovnou hivex a zavolá se metoda aSoftware(). Tato metoda tedy prochází tři umístění v registrech, v závislosti na registru, který se má zkoumat. Pro soubor ntuser.dat je to cesta *Software/Microsoft/Windows/CurrentVersion/Uninstall*. V souboru Software je analyzováno umístění *Microsoft/Windows/CurrentVersion/Uninstall*, ve kterém jsou uloženy informace k instalovaným programům 32bitového nebo 64bitového systému. Umístění *Wow6432Node/Microsoft/Windows/CurrentVersion/Uninstall* se analyzuje v případě 64bitového systému, kde se na toto umístění ukládají informace o 32bitových programech. Tímto způsobem je zajištěna analýza veškerého softwaru z registrů, přičemž nezáleží na bitové verzi operačního systému.

5.2.2 *PROHLÍŽENÍ SOUBOROVÉHO SYSTÉMU*

V této části program nejdříve použitím knihovny libguestfs zjistí názvy a cesty všech adresářů a souborů na zkoumaném disku, kde tyto cesty uloží do pole a poskytne je metodám aFiles() a aOtherSoftware(). Když tedy uživatel vybere analýzu souborového systému a zvolí obě možnosti současně, zajistí se tímto řešením jistá úspora času potřebná pro analýzu oproti dvěma samostatným prohlížením.

Metoda aFiles() načte seznam jmen, která uživatel zadal pro analýzu, upraví jej pro potřeby porovnání se seznamem všech souborů a výsledné shody zaznamená do připravovaného *html* souboru. Tento způsob prohledávání není závislý na velikosti písmen a vyhledá jak soubory, tak složky, které v názvu obsahují hledaný výraz. Diakritika je v tomto případě zohledněna.

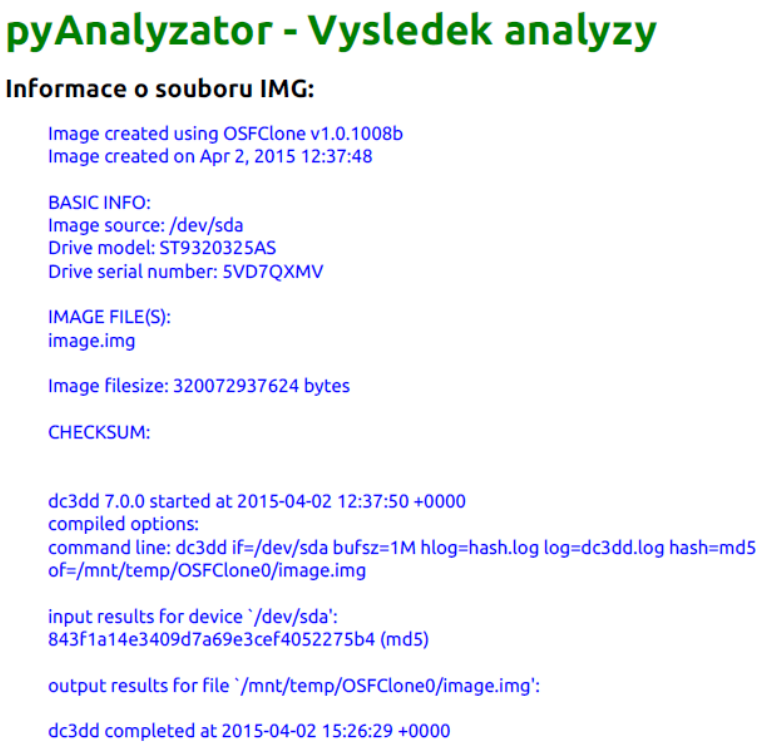
Druhá metoda aOtherSoftware() vyhledá všechny soubory s příponou *exe*, nacházející se na analyzovaném disku. Nalezené soubory analyzátor postupně pomocí knihovny libguestfs načítá do bufferu a poté předává knihovně Pefile, která umožní soubory prohlížet. V těchto souborech se analyzátor pokusí nalézt blok s názvem StringFileInfo, který obsahuje informace týkající se daného softwaru a tyto informace uloží do výstupního dokumentu.

5.3 VÝSLEDEK ANALÝZY

V průběhu celé analýzy jsou data ukládána a připravována pro výsledný dokument. Pro přehledné zobrazení s možností vyhledávání (v prohlížeči pomocí klávesové zkratky Ctrl+F) byl vybrán formát *html*, který je vytvořen modulem xml.etree.cElementTree. Tento modul je součástí Pythonu 2.7 a byl vybrán kvůli rychlejší tvorbě dokumentů a podstatně menší náročností na paměť oproti modulu xml.etree.ElementTree.

V samotném *html* dokumentu najdeme na začátku informace o bitové kopii, získané z již zmíněného souboru *nfo* nebo *info*. Pro odlišení této části, která má za účel

pouze identifikovat obraz disku, ke kterému zjištěné informace patří, má tento blok nastaven menší font písma a modrou barvu.



Obrázek 4 - Výsledek - hlavička

Na začátku další části máme zeleně zvýrazněn údaj o oddílu na disku, ke kterému výsledné informace patří. Pod tímto údajem jsou informace o operačním systému a o balíčcích MS Office. Tyto informace jsou zobrazeny v tabulkách, kde levý sloupec obsahuje její název a pravý sloupec odpovídající informaci. Titulek tabulky a produktová jména jsou pro přehlednost napsána tučně. Tabulky jsou pro přehlednost odsazeny od levého okraje a v případě kdy analyzátor nenajde informace k požadované sekci, je tato skutečnost v dokumentu zaznamenána. V bloku MS Office může být uvedeno více instalovaných verzí nebo také žádná. Při nulovém výskytu se tedy pod nadpisem „Informace o MS Office“ uvede nápis „nenalezeny“.

device: /dev/sda2

Informace o OS:

Product name: **Windows 7 Professional**
Service Pack: Service Pack 1
Organization:
Owner: Martina
Type: Client
Product ID: 55041-147-9227357-86583
Product key: BBBBB-BBBBB-BBBBB-BBBBB-BBBBB

Informace o MS Office:

Product name: **Microsoft Office Professional 2010**
Product ID: 82503-018-0000106-48248
Product key: H73H9-Y3976-PQBJ2-QQ97X-TFVP7

Obrázek 5 - Výsledek - OS a MS Office

Jako další v pořadí jsou uvedeny informace o softwaru, který byl získán z registrů operačního systému. Tabulky mají v tomto bloku pevně dány hodnoty, kterých mohou nabývat (Display name, Version, Publisher,...). Registry nemusí obsahovat všechny tyto položky, a proto nemusí být všechna políčka vyplněna. Nalezený software je abecedně seřazen a jméno softwaru je opět tučně zvýrazněno.

Informace o SW:

Display name: **Adobe Flash Player 17 ActiveX**
Version: 17.0.0.134
Publisher: Adobe Systems Incorporated
Install location:
Install date:
User:
Product ID:

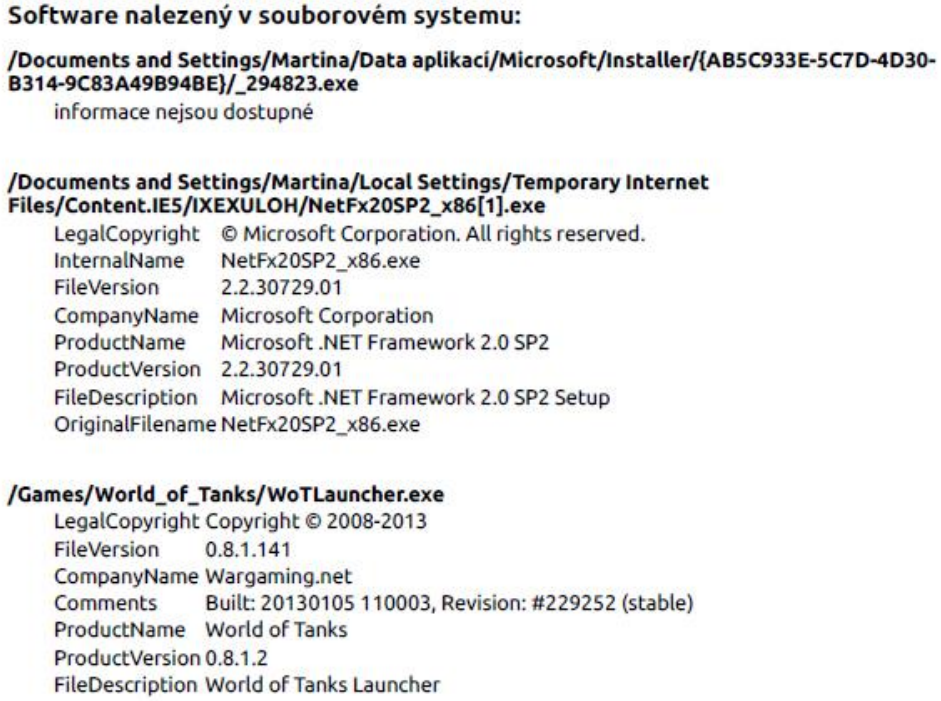
Display name: **Adobe Flash Player 17 NPAPI**
Version: 17.0.0.134
Publisher: Adobe Systems Incorporated
Install location:
Install date:
User:
Product ID:

Display name: **Adobe Reader X (10.1.7) - Czech**
Version: 10.1.7
Publisher: Adobe Systems Incorporated
Install location: C:\Program Files\Adobe\Reader 10.0\Reader\
Install date:
User:
Product ID:

Obrázek 6 - Výsledek - nalezený software z registrů

Tímto je ukončena sekce výsledků z analýzy registrů a dostávají se na řadu informace získané analýzou souborového systému.

Jako první je zde uvedena sekce nalezeného softwaru. V této sekci jsou uvedeny cesty k jednotlivým *exe* souborům z analyzovaného oddílu na disku a k nim informace nalezené prostřednictvým knihovny pefile. V případě, že analyzovaný *exe* soubor nemá o sobě uloženy žádné informace, je tato skutečnost u souboru zmíněna.



Obrázek 7 - Výsledek - nalezený software ze souborového systému

Poslední sekce zobrazuje soubory a složky, které se shodovaly s filtrem zadaným od uživatele. Tučně jsou zde zapsána hledaná slova a pod nimi jsou uvedeny cesty k těmto souborům nebo složkám z daného oddílu disku. V případě že se k danému slovu nenalezne shoda, je tato skutečnost opět uvedena.

Obrázek 8 - Výsledek - nalezené soubory

5.4 EXPORT VÝSLEDKŮ

Exportovat výsledky můžeme ve formátu *html* nebo *pdf*. V obou případech je uživatel dotázán, kam chce výsledný soubor exportovat a jak se má soubor jmenovat.

V případě exportu *html* je soubor po dokončení analýzy vytvořen automaticky, takže program pouze zkopíruje soubor pojmenovaný *output.html* z adresáře analyzátoru na uživatelem požadovanou destinaci, a to s názvem, který uživatel zadá. Kopírování souboru zajišťuje modul shutil.

K exportu souboru *pdf* je použit další modul balíčku PyQT4 a to QTextDocument. Tento modul načte obsah souboru *output.html*, rozdělí ho na stránky formátu A4 a uloží s názvem a destinací zadanou uživatelem.

O zdárně provedeném exportu je uživatel informován zprávou v informačním okně.

6 TESTOVÁNÍ APLIKACE

K otestování aplikace byly použity bitové kopie disků s operačními systémy Microsoft Windows XP, 7, 8, 10. Tyto kopie byly vytvořeny linuxovým příkazem dc3dd prostřednictvím liveCD OSFClone od společnosti PassMark.

Celkově byly vytvořeny a analyzovány obrazy z následující tabulky.

Tabulka 1 – Seznam bitových kopií

	Windows XP	Windows 7	Windows 8.1	Windows 10	Ubuntu
image.img	x	x			
image2.img		x			x
image3.img			x		
image4.img				x	

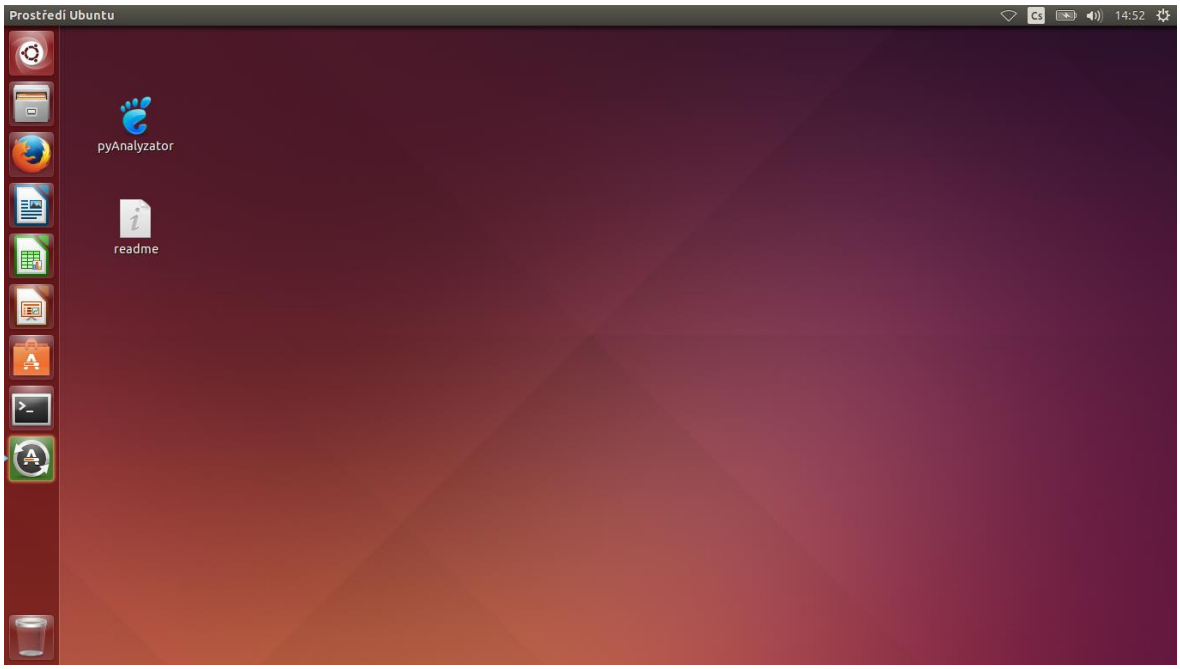
Legenda: řádky = bitové kopie; sloupce = operační systémy; „x“ označuje instalovaný OS na dané bitové kopii

Jak je z tabulky patrné, testování nezahrnovalo pouze schopnost vyhledat instalovaný software a další informace související s analýzou, ale testovala se i schopnost poradit si s bitovou kopií, na které se nacházel více než jeden operační systém. Obraz *image.img* obsahoval dvě instalace OS Windows a obraz image2.img obsahoval dualboot s linuxovou distribucí Ubuntu.

Analyzátor si poradil se všemi bitovými kopiemi, které mu v testu byly poskytnuty. Oddíly disků obsahující MS Windows byly připojeny a odpojeny bez porušení celistvosti uložených dat. K registrům a souborovému systému se podařilo přistoupit na všech testovaných vzorcích. Analyzátor správně identifikoval instalované operační systémy a balíčky Microsoft Office, kde v obou případech úspěšně dekódoval licenční klíče. Nalezené programy jak z registrů, tak hledané v souborovém systému obsahovaly veškeré dostupné informace. Výstup aplikace tedy obsahoval veškeré skutečnosti týkající se zvolených sekcí.

7 DISTRIBUCE PROGRAMU

Pro jednoduché použití byla vytvořena live distribuce Linuxu, kde byl jako základ zvolen operační systém Ubuntu ve verzi 14.04. Tato distribuce byla vybavena knihovnami potřebnými pro správnou funkčnost analyzátoru, konkrétně se jedná o balíčky „libguestfs0“, „libguestfs-tools“, „python-libguestf“, „python-pefile“ a o samotný analyzátor „pyAnalyzator“. Live distribuce byla vytvořena příkazem „sudo remastersys backup“, pomocí nástroje remastersys ve verzi 3.0.4-2.



Obrázek 9 - Live distribuce

Po vložení DVD do počítače a zdárném nabootování zvolíme možnost spuštění „Live“. Operační systém se načte a automaticky přihlásí předem vytvořeného uživatele „User“. Na ploše nalezneme ikonku „pyAnalyzator“ a soubor „readme“. Ikonka „pyAnalyzator“, která je znázorněna modrou stopou spouští vytvořený program. Soubor „readme“ obsahuje informace k vytvořené aplikaci a k systému Ubuntu. V tomto souboru je také uloženo uživatelské jméno a heslo pro přihlášení se jako root.

8 ZÁVĚR

V rámci této práce vznikla aplikace pyAnalyzator předinstalovaná na jednoúčelové live distribuci Linuxu. Tento program analyzuje bitové kopie disků a dokumentuje z nich informace o instalovaném softwaru jak z registrů OS Windows, tak ze souborového systému. Dále zjišťuje přítomnost souborů, které mohou porušovat autorská práva. Výsledek analýzy je uložen do přehledného *html* dokumentu, který je možný exportovat ve formátu *pdf*.

Aplikace by díky automatizaci postupů forenzní analýzy měla ulehčit práci soudním znalcům v odhalování instalací nelegálního softwaru a protiprávního jednání konaném na zkoumaném zařízení.

Cíle práce byly úspěšně splněny.

CITOVANÁ LITERATURA

[1] J. Kadlec, "forenzni analyza 1," 21 4 2005. [Online]. Available: http://www.root.cz/clanky/forenzni-analyza-1/. [Accessed 2015].

[2] I. M. Svetlík, "Digitální forenzní analýza a bezpečnost informací," 1 2010. [Online]. Available: http://www.rac.cz/RAC/homepage.nsf/CZ/Clanky/$FILE/DSM-Digit%C3%A1ln%C3%AD%20forenzn%C3%AD%20anal%C3%BDza-01-2010.pdf. [Accessed 23/4/2015].

[3] *Zákon č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským,* ve znění pozdějších předpisů.

[4] *§ 40 zákona č. 121/2000 Sb., Ochrana práva autorského (autorský zákon),* 2000.

[5] *§ 105a zákona č. 121/2000 Sb., Přestupky (autorský zákon),* 2000.

[6] *§ 105b zákona č. 121/2000 Sb., Správní delikty právnických a podnikávajících fyzických osob (autorský zákon),* 2000.

[7] *§ 270 zákona č. 40/2009 Sb., Porušení autorského práva, práv souvisejících s právem autorským a práv k databázi (trestní zákoník),* 2009.

SEZNAM OBRÁZKŮ

Obrázek 1 - UseCase diagram ..... - 14 -
Obrázek 2 - Fáze programu ..... - 15 -
Obrázek 3 - Grafické prostředí aplikace (GUI) ..... - 17 -
Obrázek 4 - Výsledek - hlavička ..... - 22 -
Obrázek 5 - Výsledek - OS a MS Office ..... - 23 -
Obrázek 6 - Výsledek - nalezený software z registrů ..... - 23 -
Obrázek 7 - Výsledek - nalezený software ze souborového systému ..... - 24 -
Obrázek 8 - Výsledek - nalezené soubory ..... - 25 -
Obrázek 9 - Live distribuce ..... - 27 -

SEZNAM TABULEK

Tabulka 1 – Seznam bitových kopií ..... - 26 -

PŘÍLOHA 1 - PROGRAMÁTORSKÁ DOKUMENTACE

Program pyAnalyzátor obsahuje soubory:

- ***main.py*** - slouží pouze pro start aplikace a zobrazení okna s PyQT4
- ***mainwindow.py*** - obsahuje celý zdrojový kód programu
- ***mainwindow.ui*** - v tomto souboru jsou uloženy informace o grafickém rozložení celé aplikace
- ***pyAnalyzator.xpyqt*** - obsahuje informace o projektu pyAnalyzator
- ***pyAnalyzator.png*** - ikonka aplikace

Soubor „mainwindow.py“

třída MainWindow (QMainWindow)

- ***progress (self,sig)***
  - posunuje progressbarem v okně aplikace na hodnotu „sig“ (typ int)
  - po přijetí hodnoty sig == 100 zpřístupní tlačítka „Export“ a „Zobraz výsledky“, ukončí činnost vlákna WorkThread2, vymaže text na prvku progress_label a nastaví text tlačítku button_Analyze na hodnotu „Analýza dokončena“
- ***progresslabel (self,sig)***
  - nastavuje text prvku progress_label na hodnotu přijímanou v proměnné sig (typ string)
- ***progress2 (self,sig)***
  - nastavuje text tlačítku „Analýza“ na hodnotu přijímanou v proměnné sig (typ string)
- ***buttonLoad_click (self)***
  - zobrazí dialogové okno pro výběr img souboru a zvolenou cestu k souboru uloží do globální proměnné „imgPath“
  - deaktivuje tlačítka „Export“ a „Zobraz výsledek“
  - aktivuje tlačítko buttonAnalyze a nastaví mu text „Analyzovat“

- ***buttonAnalyze_click (self)***
  - uloží bool hodnoty radio buttonů do globálních proměnných chbWindowsChecked, chbOfficeChecked, chbSoftwareChecked, chbOtherSoftwareChecked, chbSearchFilesChecked
  - uloží seznam názvů zadaný od uživatele do globální proměnné *wanted*
  - spustí vlákno *workThread* a *workThread2*
- ***buttonResults_click (self)***
  - spustí okno prohlížeče se souborem „output.html“, který je výsledkem analýzy nacházející se v adresáři programu
- ***buttonExport_click (self)***
  - otevře dialogové okno pro výběr cílové destinace a pro zadání názvu souboru
  - podle zvoleného radio buttonu vygeneruje požadovaný typ souboru a uloží ho na destinaci zvolenou od uživatele a s názvem zadaným od uživatele
  - po dokončení exportu zobrazí hlášku o úspěšném exportu

třída WorkThread (QtCore.QThread)

- ***run (self)***
  - zavolání metody Analyze()
- ***DecodeProductKey (self, digitalProductId)***
  - přijímá originální hodnotu klíče registrů DigitalProductId (type list)
  - vrací dešifrovaný instalační klíč (type list)
- ***aWindows (self, h, _os)***
  - „h“ označuje přístupný bod k souboru s registrem
  - „_os“ označuje nadřazenou značku v *html* dokumentu
  - tato metoda získává informace z registru „Software“ a získává hodnoty ProductName, CSDVersion, RegisteredOrganization, RegisteredOwner, InstallationType,ProductId a DigitalProductId
  - nalezené hodnoty ukládá do tabulky vnořené pod značku „_os“

- ***aOffice (self, h, _mso)***
  - „h“ označuje přístupný bod k souboru s registrem
  - „_mso“ označuje nadřazenou značku v *html* dokumentu
  - tato metoda získává informace z registru „Software“ a získává hodnoty ProductName, ProductId a DigitalProductId
  - nalezené hodnoty ukládá do tabulky vnořené pod značku „_mso“
- ***aSoftware (self, h, id)***
  - „h“ označuje přístupný bod k souboru s registrem
  - „id“ určuje, která cesta v registru se má zkoumat
    - 0 - pro software aktuální bitové verze OS
    - 1 - pro 32bit software na 64bit OS
    - 2 – pro registry ntuser.dat
  - tato metoda získává hodnoty: *DisplayName, DisplayVersion, Publisher, InstallLocation, InstallDate, User, ProductId*
  - metoda vrací seznam (typ list), záznamy jsou v pořadí, v jakém je metoda získává
- ***aFiles (self, _f, filelist)***
  - „_f“ označuje nadřazenou značku v *html* dokumentu
  - „filelist“ přijímá seznam ve formátu list, obsahující cesty k souborům a složkám
  - metoda hledá shodu mezi názvy v seznamu „filelist“ a „wanted“
  - nalezené soubory a složky ukládá do tabulky vnořené pod značku „_f“
- ***aOtherSoftware (self, _g, _e, filelist)***
  - „_g“ označuje přístupný bod k souborovému systému
  - „_e“ označuje nadřazenou značku v *html* dokumentu
  - „filelist“ přijímá seznam ve formátu list, obsahující cesty k souborům a složkám
  - metoda ze seznamu „filelist“ nalezne soubory s příponou „.exe“, které poté prohledává na přítomnost uložených informací o daném souboru
  - nalezené hodnoty ukládá do tabulky vnořené pod značku „_e“

- ***Analyze (self)***
  - tato metoda je zodpovědná za připojení a odpojení bitové kopie disku, prozkoumání oddílů disku, načtení registrů, vytvoření *html* dokumentu s výsledky aplikace a za zdárné provedení celé analýzy

třída WorkThread2 (QtCore.QThread)

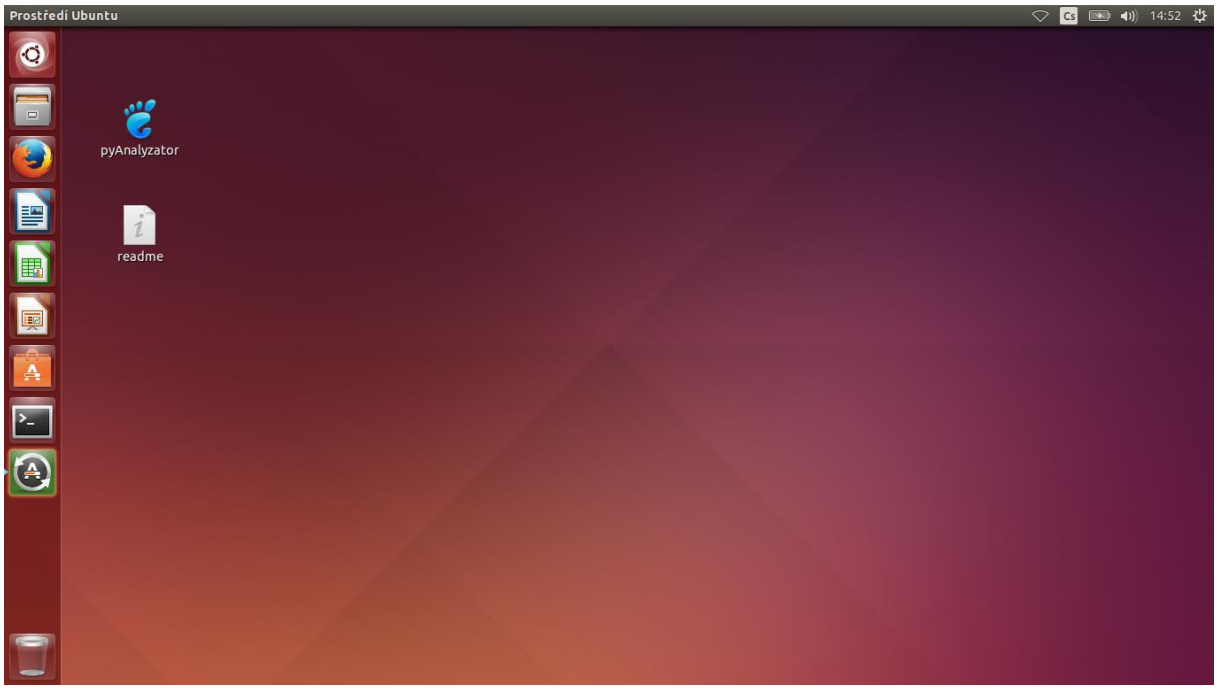
- ***run (self)***
  - generování pohybujícího se textu a odesílání do metody *progress2 (self, sig)* ve třídě *MainWindow*

PŘÍLOHA 2 - UŽIVATELSKÁ PŘÍRUČKA

Spuštění Live distribuce

Live distribuci spustíme z DVD podle následujících kroků:

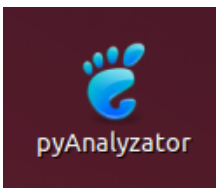
1. Soubor pyAnalyzator.img vypalte na disk DVD (pokud již nemáte)
2. Na PC v BIOSu nastavte boot z CD/DVD mechaniky
3. Vložte vypálené DVD do mechaniky
4. Zapněte PC a potvrďte boot z CD/DVD
5. Vyberte možnost spuštění: Live OS



Administrace Live distribuce **-** uživatelské jméno: „User“; heslo: „user“

Spuštění aplikace pyAnalyzator

Aplikaci spustíme kliknutím na modrou ikonku pyAnalyzator, kterou najdeme vlevo nahoře na ploše OS Linux.



Práce s aplikací

Po spuštění vyberte pomocí tlačítka „Výběr souboru“ soubor bitové kopie disku (*.img). Pomocí zaškrtávacích políček pod nadpisem „Možnosti pro analýzu“ vyberte, co chcete na daném obrazu analyzovat a stiskněte tlačítko „Analyzovat“. Ve stavovém řádku se zobrazuje průběh zpracování a právě analyzovaná sekce. Po dokončení analýzy je možné výsledek zobrazit v okně prohlížeče kliknutím na tlačítko „Zobraz výsledky“. Dále je možné výsledky exportovat ve formátu *pdf* nebo *html* výběrem požadovaného typu a stiskem tlačítka „Export“. Po dokončení analýzy můžeme program ukončit tlačítkem „Konec“ nebo zadat cestu k další bitové kopii a proces opakovat.