Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Záloha dat z výpočetní techniky pro účely forenzního zkoumání

Bakalářská práce

Patrik Brejcha

Školitel: Ing. Jaroslav Kothánek, Ph.D.

České Budějovice 2012

Bibliografické údaje:

Brejcha P., 2012: Záloha dat z výpočetní techniky pro účely forenzního zkoumání

[Data back-up of memory storages for forensic analysis Bc.. Thesis, in Czech.] – Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace:

Úkolem této bakalářské práce je analýza náležitostí nutných při zálohování dat z paměťových médií pro užití v soudním řízení a vytvoření komplexního softwarového nástroje pro částečnou automatizaci tohoto procesu. Výsledkem bude program pracující pod OS Linux.

Abstract:

The aim of this Bachelor's dissertation is analysis of all the essentials required during data back-up of various memory storages for use in legal lawsuit and also developing of software for a partial automation of this process. The end result will be a program running under OS Linux.

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 výsledků 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, 25. listopadu 2012

Podpis

Poděkování

Děkuji vedoucímu práce Ing. Jaroslavu Kothánkovi, Ph.D. za rady a ochotu při řešení problémů. Zároveň děkuji své rodině za podporu během celé doby studia.

Obsah

1. Úvod a cíle práce

1.1 Úvod

O masovém rozšíření výpočetních technologií v moderní společnosti není pochyb. Podle údajů Českého statistického úřadu v roce 2012 dvě třetiny domácností vlastnily osobní počítač [1] . V podnicích je pak toto číslo ještě vyšší kdy 95% českých podniků má připojení na internet a dá se tedy předpokládat přítomnost osobního počítače[2]. Od chytrých telefonů po tablety, až po osobní počítače, lidé využívají tato zařízení pro komunikaci, získávání informací i jako nástroj pro práci. Zároveň se však výpočetní technika přímo či nepřímo stává nástrojem k páchání trestné činnosti a orgánům činným v trestním řízení tak vzniká potřeba zajišťování důkazů mimo jiné i z osobních počítačů.

Tato bakalářská práce se zaměřuje na získávání stop z počítačových systémů a to jednak údajů o hardwaru jako jsou název výrobce, model, sériové číslo a dále na proces zálohování dat z pevných disků pro další zkoumání.

1.2 Cíle práce

Nejprve shrneme všechny nutné postupy při zajišťování stop z osobních počítačů pro užití v trestním řízení. Dále zmapujeme existující technologie a softwarové prostředky, které se zde uplatňují. Chronologicky popíšeme postup práce znalce při zajišťování stop a nakonec navrhneme komplexní řešení pro usnadnění a automatizaci tohoto procesu ve formě pracovního postupu a softwarové aplikace. Konečným produktem mé práce bude terminálový program pro OS linux napsaný v jazyce C++ umožňující automatizované zálohovaní dat z paměťových medií do obrazů a sběr veškerých dostupných informací o přítomném hardware.

Přístup k internetu – portál Českého statistického úřadu

[1] http://www.czso.cz/csu/tz.nsf/i/dve_tretiny_ceskych_domacnosti_maji_pocitac20121204

[2] http://www.czso.cz/csu/redakce.nsf/i/2_internet_a_jeho_vyuziti

2. Zajišťování stop

2.1 Postup práce

V následující kapitole popíšeme postup práce, kterou provádí vyškolený pracovník policie ČR či soudní znalec při zajišťování stop z osobního počítače. Uvedené kroky a jejich náplň se mohou lišit v závislosti na konkrétní povaze případu.

2.1.1 Vnější prohlídka

Primárním účelem vnější prohlídky je fotodokumentace předložené techniky a zároveň zajištění optických paměťových medií či externích disků.

Při samotné fotodokumentaci se pak provádí tyto úkony:

- V případě zkoumání v počítačové laboratoři kontrola pečetí a jejich neporušenosti
- Dokumentace case[3] počítače, případné viditelné poškození, zapojení kabeláže, výrobní štítky či jiná identifikace samotného casu
- Po otevření casu počítače dokumentujeme vlastní komponenty uvnitř a to hlavně jejich zapojení či viditelné poškození
- V závislosti na povaze případu demontujeme určité komponenty, téměř vždy pevné disky a zadokumentujeme jejich výrobní štítky, nastavení jumperů[4] a jakékoliv viditelné poškození

Fotodokumentace slouží jako podklad pro další šetření ale i jako ochrana vyšetřovatelů v případě pozdějších stížností majitele techniky týkajících se mechanického poškození součástí.

[3] Case- neboli počítačová skříň

[4] Jumper- je mechanická spojka vodičů, u pevných disků slouží k dodatečnému nastavení vlastností

2.1.2 Zajišťování dat

Zajištění dat provádíme pomocí forenzní duplikace dat[5] do obrazů disků, pokud to situace umožňuje, hlavními kritérii jsou zde velikost a počet zkoumaných médií a tedy časová náročnost tohoto procesu. Hlavním požadavkem je za žádnou cenu nepoškodit původní data na paměťových médiích.

Existuje více způsobů jak tuto duplikaci provést. My zde budeme popisovat zálohu za použití zkoumané techniky a nabootování[6] operačního systému znalce např. z live cd.

Potřebné vybavení:

- Cílový disk – Pevný disk dostatečné velikosti určený pro uložení obrazů zkoumaných disků
- Forenzní operační systém – Operační systém zajišťující integritu připojených disků. V praxi používáme linuxové live distribuce zajišťující nepřipojení pevných disků při startu. Tento operační systém můžeme spouštět buď z optického média, anebo ze samotného cílového disku

Vlastní úkony při zajišťování dat:

- Kontrolovaný boot - Odpojíme veškeré pevné disky, zapneme počítač a pokusíme se získat přistup do BIOSU[7], zde zadokumentujeme veškeré nastavení a změníme boot sekvenci tak aby se zavedl náš forenzní operační systém
- Druhý kontrolovaný boot – Zapojíme cílový disk případně live CD s naším operačním systémem a přesvědčíme se, že počítač bootuje tento systém. V této fázi můžeme provést zajištění hardwarové konfigurace počítače pomocí softwarových prostředků
- Připojíme zkoumaná média a nabootujeme forenzní operační systém. Zajistíme informace o pevných discích jako je velikost, geometrie, přítomné oddíly jejich systémy souborů atd. Poté přikročíme k samotné forenzní duplikaci dat

[5] forenzní duplikace dat – záloha paměťových medií od začátku, až po poslední stopu do obrazu disku kdy zajistíme i data smazaných souborů, odstraněných oddílů atd.

[6] Nabootování - je proces zavádění operačního systému při zapnutí počítače.

[7] Bios - (anglicky Basic Input-Output System) Používá se hlavně při startu počítače pro načtení a konfiguraci připojených hardwarových zařízení a následnému spuštění operačního systému.

2.2 Forenzní zabezpečení získaných důkazů

Důkazy zajištěné postupy popsanými v předchozí kapitole je třeba zabezpečit proti změně v mezidobí od získání těchto důkazů do jejich použití v trestním řízení. Digitální data jsou svou povahou velice křehká a neodborným zacházením může dojít k jejich znehodnocení. V praxi se přistupuje k zajištění shromážděných důkazů pomocí hashovací funkce[8].

Ze zajištěných dat (logů obsahujících hardwarovou konfiguraci počítače i samotných obrazů disků) a výstupů hashovacích funkcí se vytiskne zpráva, kterou neprodleně po zajištění podepíše, jak odborný technik, který zajištění prováděl, tak i další nezúčastněná osoba.

Zde je ukázka obsahu souboru logs.md5 který je vytvořen naší aplikací jakožto zabezpečení logových souborů pomocí md5 hashovací funkce.

```
50775a2303ce579eabea77b45e8bec15  dmidecode.log
711ef3dcfb5033e9943efba511ffa039  fdisk_sdb.log
e023794149d3a3e8b4f82907d59b6787  hardware.log
97c3dbb3590a9e099e0fe5cbde910007  hdparm_sdb.log
47042e22a0a6975526a92949464cd448  ifconfig.log
1a74b0817848baa0c668017cccf1453d  parted_sdb.log
a83ec0cb1358ba1d8f70a84df013d2bd  sdb_info.log
c060f38b673180b689dcda5375897251  hardware.html
```

[8] Hashovací funkce - je matematický algoritmus pro převod vstupních dat do (relativně) malého řetězce znaků.

3. Užitečné nástroje

V této kapitole popíšeme hlavně linuxové distribuce, které jsou vhodné pro použití při procesu forenzní zálohy dat. Využít lze v podstatě jakoukoliv linuxovou live distribuci která nepřipojuje pevné disky při zavedení systému, nicméně existuje celá řada distribucí specializujících se na forenzní zkoumání.

Dále zde zmíníme hardwarové řešení ochrany pevných disků proti zápisu.

3.1 Systém ochrany proti zápisu

Chceme-li se ujistit, že za žádných okolností nedojde k poškození originálních dat, je vhodné použít systém pro ochranu proti zápisu. Některé pevné disky umožnují režim jen pro čtení nastavením jumperu, ale je to spíše výjimkou. V praxi se používají zvláštní řadiče pevných disků, ať už jako přídavné moduly pro slot PCI nebo jako externí zařízení, které se umístí mezi pevné disky a původní řadič počítače.

Takový řadič zachytí veškeré příkazy operačního systému a zajistí absolutní ochranu integrity dat. Některé řadiče pouze zablokují příkazy k zápisu a systém tak dostane zprávu o chybě. Komplexnější řadiče pak obsahují vlastní paměť, do které zapisují přepisovací příkazy a vytvářejí tak pro operační systém iluzi přepisovatelného média.

Obrázek 1: Příklad externího řadiče pevných disků chránícího proti zápisu

3.2 Forenzní systém – linuxové distribuce

Základním požadavkem na forenzní systém je nedestruktivní chování k jakýmkoliv potencionálním důkazům. Tím se v první řádě rozumí pevné disky připojené k systému. V podstatě každá live distribuce Linuxu tento požadavek splňuje díky tomu, že pevné disky nejsou při startu připojeny k systému.

Na tomto místě uvedeme Linuxové distribuce, které lze s úspěchem použít pro proces forenzní zálohy a které jsme testovali s výslednou aplikací.

3.2.1 Deft linux [9]

Komplexní distribuce integrující oblíbený soubor forenzních nástrojů DART (Digital Advanced Response Toolkit). Hodí se k široké škále úkonů od analýzy bezpečnostních průniků, až po forenzní zkoumání. Nabízí možnost jednoduše připojit pevné disky v režimu jen pro čtení.

3.2.2 CAINE Computer Aided Investigative Enviroment [10]

Distribuce určená pro vyšetřovatele, která nabízí uživatelsky velmi přívětivý nástroj pro připojování pevných disků, kdy ve formě GUI aplikace lze nastavovat, které disky jsou připojeny v módu jen pro čtení.

3.2.3 Back|track 5 [11]

Distribuce určená pro bezpečnostní analytiky zaměřující se na principy ofenzivní bezpečnosti vhodná jednak pro audity, ale i jako nástroj k analýze bezpečnostních průniků. Od verze 4 nabízí forenzní mód, kdy nejsou automaticky připojeny žádné pevné disky.

V této distribuci jsme prováděli vývoj aplikace i většinu testování.

Domovské stránky projektů zmíněných distribucí

[9] http://www.deftlinux.net/

[10] http://www.caine-live.net/

[11] http://www.backtrack-linux.org/

4. Implementace

V následující kapitole popíšeme softwarovou aplikaci, která vznikla jako hlavní produkt této bakalářské práce. Jedná se o komplexní řešení sběru informací o hardwaru počítače a zajištění dat z paměťových médií v souladu s forenzními zásadami. Důraz je kladen na jednoduchost ovládání a vysokou míru automatizace celého procesu za účelem minimalizování rizika lidské chyby.

Požadavky na aplikaci:

- Funkční pod OS linux v terminálovém módu
- Maximální možná míra automatizace a jednoduchosti ovládání
- Shromáždění informací o zkoumané technice, tzn. údaje o výrobci, modelu a sériová čísla veškerého hardware
- Shromáždění informací o všech pevných discích jejich rozdělení a všech identifikačních informací
- Provedení přípravy cílového disku (nalezení vhodného oddílu, vytvoření oddílu nového, vytvoření souborového systému[12], připojení k systému)
- Provedení forenzní zálohy pevných disků na cílový disk a zabezpečení pomocí hashovací funkce
- Poskytnutí konečné zprávy se základními informacemi o nalezeném hardwaru a dále seznam vytvořených obrazů a jejich hash otisků vhodnou k tisku a podepsání
- Zároveň poskytnutí funkčnosti pro pozdější obnovu a kontrolu integrity dat

[12] Souborový systém – je způsob organizace souborů a adresářů, tak aby k nim mohl přistupovat operační systém

4.1 Existující řešení

V současné době nebyl nalezen nástroj, který by byl schopný provést komplexní zajištění jak informací o systému tak forenzní duplikace dat. Zároveň požadavek na co největší míru automatizace a jednoduchost vylučuje komerční i opensource[13] gui nástroje, protože ty se většinou naopak snaží nabídnout co nejvíce možností.

Ve znalecké praxi je stále obvyklé manuální používání příkazů na bázi dd. A shromažďování informací o hardwaru prostřednictvím fotodokumentace.

Za zmínku stojí Linuxová distribuce italského původu Forensic Hard Copy[14]. Ta se ovšem zaměřuje pouze na samotné pořízení forenzních záloh a nevytváří závěrečnou zprávu.

Závěrem je, že vývoj aplikace splňující dané požadavky je relevantní hlavně z hlediska vysoké míry automatizace a výstupu v podobě konečné zprávy v českém jazyce.

4.2 Použité technologie

Aplikace bude vyvíjena v jazyce C++ formou zdrojových souborů vyjadřujících logické celky v běhu programu obsahujících funkce řešící dílčí úkony. Vlastní vývoj bude probíhat za pomoci vývojového prostředí NetBeans IDE[15] 7.2.1 s použitím kompilátoru g++. Po dobu vývoje byla použita linuxová distribuce Back|track 5.

[13] Open-source – počítačový software s otevřeným zdrojovým kódem

[14] Forensic Hard Copy – webový portál projektu - http://fhclive.sourceforge.net/ENGLISH.html

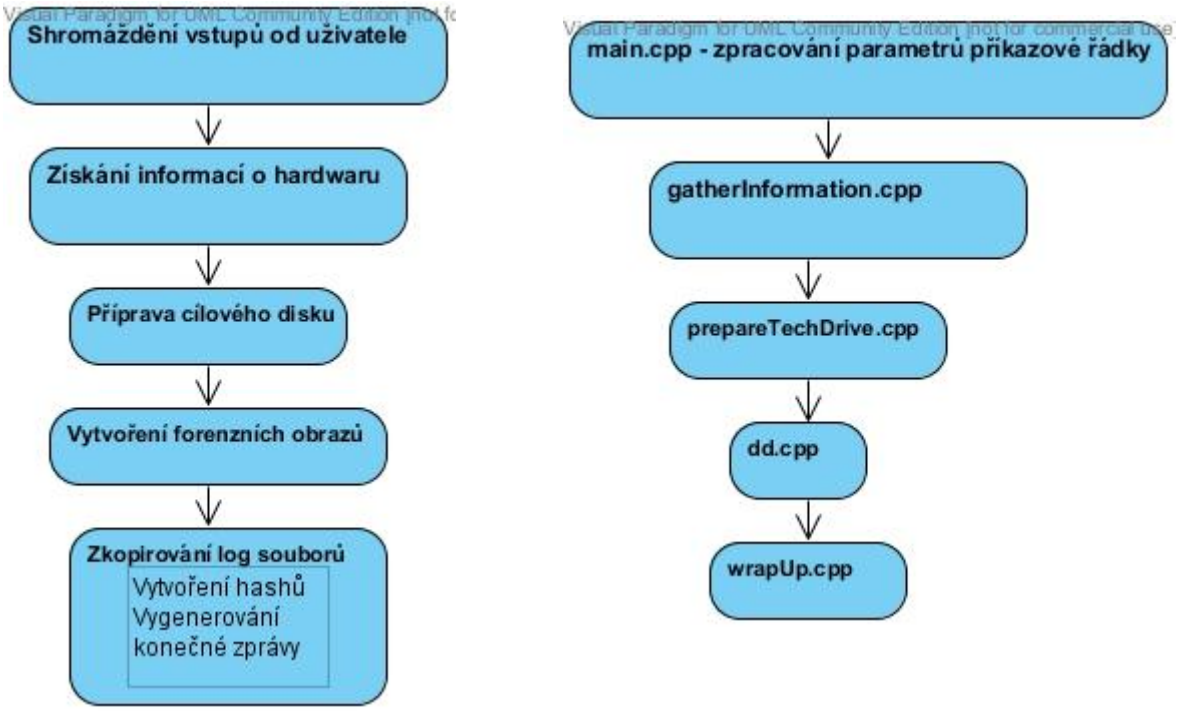
[15] Netbeans - vývojové prostředí – webový portál projektu - netbeans.org

4.3 Model běhu programu

Program bude poskytovat dvě základní funkcionality:

- Část pro forenzní zálohu dat, kdy na základě definování cílového disku program automaticky provede shromáždění informací a vytvoření obrazů a jejich uložení na cílový disk i s vytvořenou konečnou zprávou
- Část pro obnovu dat z již vytvořených případů a kontrolu integrity dat jak samotných obrazů tak i logových souborů obsahujících informace o hardwaru zkoumané techniky

Požadavky na funkčnost aplikace zároveň diktují i model běhu a vzhledem k lineárnosti procesu zajišťování stop jsme podle tohoto modelu postupovali při návrhu zdrojových souborů aplikace.



Obrázek 2:Diagram sekvenční logiky zajišťování stop a zdrojové soubory vzniklé na základě těchto dílčích úkonů

Z tohoto diagramu tedy vznikly základní zdrojové soubory, které řeší funkcionalitu daných kroků. Nakonec byl ještě vytvořen soubor restore.cpp sloužící pro část obnovy obrazů a kontrolu integrity dat pomocí hash funkcí.

4.4 Rozbor funkcionality programu

V této kapitole popíšeme jednotlivé zdrojové soubory a funkčnost, kterou realizují. Včetně některých zásadních funkcí.

V běhu programu je nejzásadnější volání funkce system( ) ze standartní knihovny přístupné v stdlib.h. Pomocí této funkce spouštíme linuxové příkazy z C++ zdrojových kódů.

4.4.1 main.cpp – zpracování parametrů

Hlavní zdrojový soubor obsahující funkci main. Zde se volají další funkce u ostatních zdrojových souborů a provádí se zpracování parametrů[16], se kterými byl program spuštěn. Aplikace přijímá tyto parametry:

- -t jmenovka | kde jmenovka je zkratka cílového disku (např. "-t sda")
- -nobackup disk1,disk2 | kde disk je jmenovka disků které nechceme zálohovat (např. "-nobackup sda,sdc")
- -name název_případu
- -op oddíl1,oddíl2 | kde oddíl= zkratka oddílu k zálohování (např. "-op sda1,sda3") použitím tohoto parametru se již nebude zálohovat celý disk ale jen vybrané oddíly
- -number evidenční_číslo_případu
- -gzip | zkomprimuje všechny výsledné obrazy
- -split MB | kde MB je číselná velikost v MB (rozdělí výsledné obrazy po MB megabytech
- -restore | namísto zálohování dat spustí část programu určenou pro obnovu dat ze zajištěných případů

[16] Parametr programu – doplňující informace předávána programu v příkazové řádce, většinou uvozená znakem pomlčky

4.4.2 gatherInformation.cpp – shromažďování informací

Obsahuje veškerý zdrojový kód pro shromáždění informací o hardware přítomném v počítači a rozdělení všech pevných disků. Tyto informace jsou ukládány ve formě logových souborů do dočasného úložiště (defaultně složka /tmp/). Dále se zde kontroluje správnost parametrů týkajících se disků přítomných v počítači.

Pro shromáždění informací se využívají tyto linuxové příkazy:

- Lshw – kompletní výpis přítomného hardwaru a všech informací, poskytuje výstup ve formátu html
- Dmidecode – podobný jako lshw, ale je schopný získat některá sériová čísla která lshw nevyhledal
- Ifconfig – Informace o sítových zařízeních, používáme hlavně pro zjištění ip adresy v případě přítomnosti dhcp serveru[17]
- Příkazy fdisk a parted pro získávání informací o pevných discích, jejich oddílech a souborových systémech
- Příkazy hdparm a udevadm pro získávání sériových čísel pevných disků

4.4.3 prepareTechDrive.cpp – příprava cílového disku

Zdrojový soubor zodpovědný za přípravu cílového disku. Nejdříve se zde vypočítá velikost všech pevných disků a oddílů označených k forenzní záloze. Pak jsou vyhledány všechny oddíly na cílovém disku, které mají dostatek volného místa pro uskutečnění zálohy. V případě, že je nalezeno více vhodných oddílů vybere ten s největším volným místem.

Když na cílovém disku není nalezen vhodný existující oddíl, je vytvořen oddíl nový, pokud to velikost disku dovoluje. Dalším krokem je vytvoření systému souborů na nově vzniklém oddílu.

Dále se pak vytvoří složka pro veškerý výstup programu ve formátu název_případu_současné_datum_čas. Nakonec se oddíl připojí k operačnímu systému.

[17] DHCP – síťový protokol pro automatické nastavování ip adres síťových klientů

4.4.4 dd.cpp – vlastní forenzní záloha

Zde se provádí vlastní forenzní duplikace vybraných paměťových medií. V současné době je zde použit příkaz „dd“ s parametry noerror a sync. Jádro příkazu lze měnit nastavením proměnné ddcore a daly by se tak zde snadno definovat i jiné verze příkazů pro vytváření diskových obrazů.

Prvním krokem v tomto souboru je zjištění dostupnosti programu Pipeviewer[18], který využíváme pro zobrazení procentuálního ukazatele v průběhu zálohy.

Pokud tento nástroj není nainstalovaný v systému ani se nenalézá ve složce s naším programem ve formě binárního souboru, přikročíme ke způsobu zálohy, kdy vlastní příkaz spustíme na pozadí. V programu pak realizujeme smyčku zobrazující procentuální hodnotu na základě velikosti zálohovaného média a výstupního souboru obrazu.

Zároveň se samotnou forenzní duplikací se provádí i výpočet hashovací funkce md5. Toho se dosahuje využitím nástroje „tee“, který rozděluje výstup z programu dd na dvě větve- výstupní soubor obrazu a jako vstup příkazu md5sum.

4.4.5 wrapUp.cpp – kopírovaní logových souborů a vytvoření závěrečné zprávy

Tento soubor se stará o zkopírování všech logových souborů z dočasného úložiště do výstupní složky, vytvoření konečné zprávy a finální úpravy.

Provádí se zde tyto úkony:

- Zkopírování veškerých logových souborů do výstupní složky
- Zkopírování md5 hashů obrazů disků vytvořených během zálohy
- Vypočítání md5 hashů logových souborů
- Vytvoření koneční zprávy obsahující základní informace o proběhlé záloze
- Smazání obsahu dočasného úložiště
- Odpojení cílového disku od systému

[18] Pipeviewer – volně šiřitelný terminálový nástroj sledující tok dat skrz pipeline

Ukázka konečné zprávy:

Zpráva o záloze vytvořená dne 12.11.2012 v 12:41:11 hodin
Název připadu: Brejcha
Evidenční číslo: 890312/89
Informace o počítači:
Procesor:
Výrobce: Intel
Model: Intel(R) Core(TM) i5-2380P CPU @ 3.10GHz
Kmitočet: 3100 MHz
Počet jader: 4

Základní deska:
Výrobce: MSI
Název produktu: Z68A-G43 (G3) (MS-7750)
Verze: 1.0
Sériové číslo: To be filled by O.E.M.

Operační paměti:
Pamět ram:
Velikost:: 4096 MB
Rychlost: 1333 MHz (0.8 ns)
Výrobce: Kingston
Serial Number: 0502D0AA

Nalezené síťové interfaci:
eth0 MAC: 8c:89:a5:83:7f:be
IP adresa:192.168.10.104 Maska:255.255.255.0

Nalezeny byly tyto pevné disky:
Disk /dev/sda: 500.1 GB, 500107862016 bytes
Model: WDC_WD5000AZRX-00A8LB0
Sériové číslo: WD-WMC1U3378774

Disk /dev/sdb: 60.0 GB, 60022480896 bytes
Model: OCZ-VERTEX3
Sériové číslo: OCZ-9Q57JA0IAFK5O2HU

Vytvořeny byly tyto zálohy paměťových médií:

disk sda
b6b4988654ee0ad6ae8d4bcfcce2d17f sda.dd
disk sdb

2e533db5bf2b798592cd4854fb39f04d sdb.dd
Podpis:

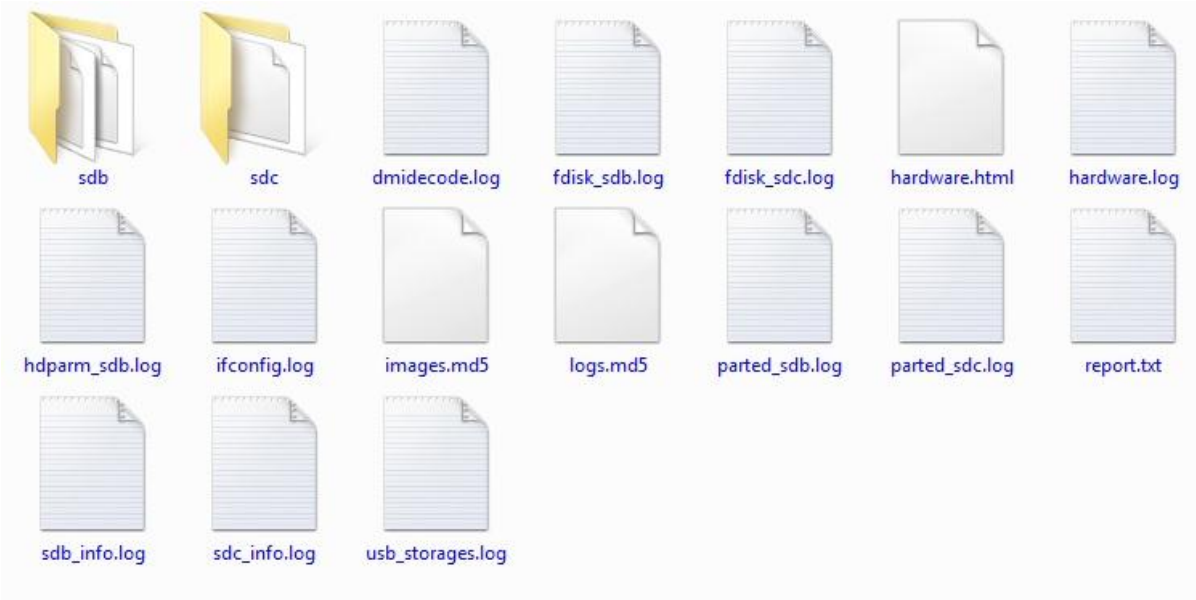
4.4.6 restore.cpp – Obnovování dat a kontrola integrity

Zdrojový soubor obsahující veškerou logiku pro realizování obnovy obrazů pevných disků a kontroly integrity dat dříve zajištěných případů. Hlavní funkce restore se zavolá v případě, že byla aplikace spuštěna s parametrem – restore.

Nejprve uživatele požádá o vybrání disku se zajištěnými případy. Zde prohledá veškeré oddíly. Pokud jsou nalezeny zajištěné případy na více oddílech, nechá uživatele vybrat. Následuje výběr konkrétního případu, s nímž je možné provádět další dílčí úkony. V této fázi je použito interaktivní rozhraní, kdy uživatel stiskem kláves 1-5 provádí požadované kroky.

4.4.7 myheader.cpp – pomocné funkce

Zdrojový soubor, který je importován ve všech ostatních souborech obsahující rozličné pomocné funkce přes jednoduché přetypovávání proměnných, operace s vektory, až po funkce na zjištění velikosti disků, diskových oddílů či souborů. Nachází se zde celkem 29 funkcí.



Obrázek 3 Ukázka struktury souborů konkrétního zajištěného případu

5. Využití programu ze strany uživatele

V této kapitole popíšeme způsoby využití aplikace samotnými uživateli. Program je primárně určen pro zajišťování stop z osobních počítačů ať už desktopového typu nebo počítačů přenosných. Předpokládá se zde využití Linuxové live distribuce vhodného typu na optickém médiu.

5.1 Spuštění programu s parametry

Používání aplikace s vložením parametrů v příkazové řádce je určeno spíše zkušenějším uživatelům či těm, kteří naší aplikaci používají opakovaně. Vložené parametry umožňují vyčleňovat vybrané disky ze zálohy a ušetřit tak čas např. pokud nás zajímají jen potenciální důkazy určité povahy. Stejně tak použití příkazů split a gzip je žádoucí možností kdy zpřehledníme pozdější forenzní analýzu na jednotlivých fragmentech obrazu či ušetříme paměť na cílovém disku.

Zároveň však nesprávnou specifikací parametru – t, který je určen k vybrání cílového disku může dojít k nenávratnému poškození originálních medií. Stejně tak vyloučení zálohy některých oddílů či disků muže bez odborného zvážení zapříčinit opomenutí některých důkazních materiálů.

Ukázka terminálovíého okna při běhu programu:

Získávám informace o připojených pevných discích

Disk /dev/sda: 500.1 GB, 500107862016 bytes
Disk /dev/sdc: 1000.2 GB, 1000204885504 bytes
Disk /dev/sdb: 8027 MB, 8027897856 bytes

---

sdc je označen jako cílový disk
sdb je disk připojený k usb rozhraní

Získávám informace o ostatním hardware počítače

Zahajuji přípravu cílového disku

Celková velikost k zálohování je 0 GB
Nebyl specifikován žádný oddíl na cílovém disku
Pokusím se vyhledat vhodný oddíl

nalezeno 756 GB volného místa na oddílu sdc1

Kopírovat se budou tyto úložiště:

Zahajuji fázi zálohování paměťových médií na zařízení "sdc" oddíl "sdc1"
Výstup do složky "/Zalohy/none_12_12_2012_19_8_50/"

Kopíruji logy do /fullbackupmounts/sdc1/Zalohy/none_12_12_2012_19_8_50/

Zapisuji kontrolní sumy

5.2 Spuštění programu bez parametrů

Spuštění programu bez parametrů je nejjednodušší způsob, jak mít jistotu korektní zálohy. Hned na začátku běhu programu je uživatel vyzván, aby od počítače odpojil veškeré pevné disky krom disku cílového. Dokud tak neučiní, smyčka s touto výzvou se opakuje. Poté co program detekuje jen jeden připojený pevný disk, vypíše uživateli jeho výrobce, model a sériové číslo. Následuje potvrzení uživatelem, a pokud v tomto bodě uživatel odpoví kladně, je vyzván k opětovnému připojení všech pevných disků, zadání názvu případu a evidenčního čísla.

Od tohoto okamžiku celý zbytek aplikace provede automaticky zálohu bez dalšího požadavku na vstup uživatele, včetně případného vytvoření odpovídajícího oddílu na cílovém disku nastavení souborového systému atd.

5.3 Obnova dat pomocí parametru – restore

Tento mód programu není určen pro spuštění na zkoumané technice. Proto je na to uživatel hned na začátku důrazně upozorněn. Po vybrání disku obsahujícího zajištěné zálohy a vybrání konkrétního případu je uživatel prezentován následujícími možnostmi:

```
Vybrali jste případ split_9_12_2012_8_28_32
_____________________________________________________
Co si přejete:
1) Vypočítat a porovnat md5 logových souborů
2) Vypočítat a porovnat md5 obrazů disků
3) Zobrazit konečnou zprávu případu
4) Obnovit data z obrazů
5) Návrat na výběr případů
ESC ) ukončí program
```

1. Vypočítá md5 hash otisk všech logových souborů v případu, uloží je do dočasného úložiště a porovná s původními otisky v souboru logs.md5. Na závěr uživatele informuje které logové soubory se liší, anebo vypíše zprávu „všechny logové soubory jsou korektní“.

2. Vypočítá hash otisk všech obrazů disků přítomných v případu. Při vybrání této možnosti je uživatel upozorněn, že tato operace může trvat dlouhou dobu a žádá se potvrzení zároveň se sdělením informace o tom, že kontrola md5 otisků se provádí i při zpětné obnově jednotlivých obrazů

3. Vytiskne konečnou zprávu případu do terminálového okna.

4. Obnoví data z obrazů. Při zvolení této možnosti je uživatel znovu důrazně upozorněn, že může dojít ke ztrátě dat. Po zvolení příslušného obrazu a odpovídajícího cíle proběhne vlastní obnova. Po jejím dokončení je provedeno porovnání hash otisků a uživatel je o výsledku informován.

5. Návrat na výběr případů, kdy je možné zvolit jiný případ a provádět s ním veškeré předešlé kroky.

6. Testování

6.1 Provedené zálohy

S aplikací bylo provedeno mnoho zajištění forenzních duplikátů. Od kapacitně malých usb pamětí, po ssd[19] disky až po terabytová úložiště.

Testování proběhlo na čtyřech různých počítačových sestavách a ve všech případech proběhla identifikace hardwaru i sběr sériových čísel a jiných identifikačních údajů úspěšně. Stejně tak porovnávání md5 otisků vytvořených obrazů bylo vždy identické.

Program při provádění forenzní zálohy zároveň vypočítává md5 otisk a důsledkem toho byly tyto zálohy o něco pomalejší (v řádu 1-3%) než samotné použití příkazu dd. Vzhledem k tomu, že každé individuální pořízení otisku vyžaduje přečtení všech dat z obrazu, což zabírá velké množství času je ale toto řešení výhodné.

Jako problematické se ukázalo použití příkazu gzip, kdy rychlost zálohy a obnovy obrazů rapidně poklesla. Toto ovšem není zapříčiněno samotnou aplikací, ale spočívá v procesorové náročnosti komprimace a dekomprimace dat typu bit-by-bit.

6.2 Kompatibilita s vybranými linuxovými distribucemi

Aplikace byla testována na těchto live CD linuxových distribucí:

- Deft Linux verze 7.2
- Caine verze 3.0
- Back|track verze 4 a 5

Deft Linux:

Aplikace fungovala korektně, nicméně při výpisu informací do terminálu byl problém s kódováním textu, kdy se místo písmen se znaménky háčku a čárky zobrazovaly náhodné znaky. Výsledná zpráva i logové soubory tyto problémy neobsahovaly.

Opravou tohoto problému jsme se nezabývali.

[19] Solid State Drive – pevné disky bez pohyblivých součástí

Cain:

Zde nastal zásadní problém s během aplikace ve fázi připojování cílového disku k systému. Většina linuxových distribucí ve výchozí formě příkazu mount bez dalších parametrů připojí disk pro čtení i zápis. Distribuce cain však chrání pevné disky proti zápisu tím, že každý příkaz mount vykonaný v terminálu vnímá jako by byl spuštěn s parametry pouze pro čtení.

Tento problém byl vyřešen přidáním parametru – rw k řádce s mount příkazem.

Back|track 4 a 5:

Ve verzi 5 byl prováděn vývoj a při závěrečné fázi testování tak nebyly nalezeny žádné problémy.

Verze 4 nebyla schopna připojovat k systému ntfs a ext2-4 souborové systémy bez specifikování parametru – t tak jako ostatní dvě distribuce.

Tento problém byl opraven přidáním parametru – t do všech řádek s příkazem mount v programu.

7. Závěr

Aplikace je funkční a splňuje všechny zadané požadavky. Míra automatizace a jednoduchost ovládání je v případě spuštění bez parametrů vysoká. Naopak za použití parametrů poskytuje dostatečné možnosti modifikace průběhu zálohy.

Přesto je ale aplikace schopná neodborným zacházením poškodit původní data. Tento nedostatek není možné v současném návrhu práce při použití live cd úplně odstranit. Nabízí se zde jen pevně vložit identifikaci cílového disku do zdrojového kódu.

7.1 Dosažené výsledky

Všechny testy proběhly úspěšně a efektivita aplikace byla uspokojivá. Nabízí se možnost použití ve forenzní praxi hlavně jako nástroj pro standardizaci a dobrou katalogizaci jednotlivých případů záloh.

U autora však panuje nespokojenost s nedosažením plné míry automatizace.

7.2 Návrhy na pokračování práce

- Testování paralelního vytváření forenzních záloh s cílem zjistit míru efektivity v závislosti na výkonu cílového disku při operaci zápisu
- Testování aplikace na široké škále linuxových distribucí a generalizace kódu pro maximální kompatibilitu
- Vytvoření jiného pracovního postupu s možností bootování např. z usb paměťového médi. Tím umožnění vytvoření konfiguračního souboru obsahujícího sériová čísla disků používaných jako cílových
- Možnost pomocí parametrů přímo měnit použitou technologii forenzní zálohy (dd , sdd, dcfldd a jiné)
- Možnost specifikovat jiné hashovací funkce než md5
- Vytvoření dalších zdrojových souborů pro umožnění zálohy po síti
- Vytvoření Gui pro aplikaci s důrazem na jednoduchost

8. Seznam použité literatury

8.1 Portály

Belkasoft [online].2012[cit.2012-11-27]. Dostupné z WWW: <http://www.belkasoft.com/>
IT znalec [online]. 2012 [cit. 2012-11-27]. Dostupné z WWW: <http://www.it-znalec.cz/>.
Sleuthkit [online]. 2012 [cit. 2012-11-27]. Dostupné z WWW: <http://www.sleuthkit.org/>.
Backtrack [online]. 2012 [cit. 2012-11-27]. Dostupné z WWW: <http://www.backtrack-linux.org/>.
Deft Linux [online]. 2012 [cit. 2012-11-27]. Dostupné z WWW: <http://www.deftlinux.net/ />.
Caine [online]. 2012 [cit. 2012-11-27]. Dostupné z WWW: <http://www.caine-live.net/ />.

8.2 Seminární práce

Formánek Martin, Vízner Martin. Metodika zajišťování důkazů při vyšetřování počítačové kriminality [cit. 2012-11-27]. ČVUT – FEL PRAHA 2007
Dostupné z WWW: service.felk.cvut.cz/anc/ofa/pub/doc/metodika.pdf

9. Přílohy

0. Cd obsahující zdrojové kódy, binární soubor programu pipeviwer, složku projektu vývojového prostředí netbeans, doxygen dokumentaci a elektronickou verzi této práce.