SLEZSKÁ
UNIVERZITA
FILOZOFICKO-
PŘÍRODOVĚDECKÁ
FAKULTA V OPAVĚ

Šárka Vavrečková

Skripta do předmětu

Analýza dat
Bezpečnost a analýza dat

Úvod do digitální forenzní analýzy
a metod ochrany dat

Ústav informatiky
Filozoficko-přírodovědecká fakulta v Opavě
Slezská univerzita v Opavě

Opava
1. března 2024

Anotace: Tato skripta jsou určena pro studenty předmětu *Analýza dat* a *Bezpečnost a analýza dat*. U čtenáře se předpokládá alespoň základní znalost problematiky operačních systémů, hardwaru a počítačových sítí. Inspirací při sestavování skript byla především digitální forenzní analýza, nicméně především z časových důvodů se zaměřujeme spíše na bezpečnost, možnosti získávání a základní analýzy dat, obranu proti neoprávněnému získávání data apod.

Analýza dat, Bezpečnost a analýza dat

RNDr. Šárka Vavrečková, Ph.D.

Dostupné na: http://vavreckova.zam.slu.cz/analyzadat.html

Ústav informatiky
Filozoficko-přírodovědecká fakulta v Opavě
Slezská univerzita v Opavě
Bezručovo nám. 13, Opava

Sázeno v systému LaTeX

Předmluva

Co najdeme v těchto skriptech

Rychlý náhled: Tato skripta jsou určena pro studenty informatických oborů na Ústavu informatiky Slezské univerzity v Opavě. Obsahují látku vyučovanou v předmětu *Analýza dat*, po změně názvu *Bezpečnost a analýza dat*, v němž se zabýváme především úvodem do forenzní analýzy, obecně bezpečností dat, šifrováním, možnostmi získávání a analýzy dat a jejich zabezpečením před neoprávněným získáním třetí osobou.

Na rozdíl od jiných tento předmět není zakončen klasickými písemkami, ale k zápočtu je třeba odevzdat seminární práci. Proto skripta neberte jako souhrn znalostí, které je nutno si „nacpat do hlavy“ – mnohá témata jsou ve skriptech probírána až příliš do hloubky (více než bude probíráno na cvičeních) především proto, že každý student má trochu jiné zájmy a každého určité téma zaujme jinak. Tedy se neděste obsahu a jeho podrobnosti, účelem opravdu není děsit, ale motivovat.

Zmíněná seminární práce by měla být v dostatečném rozsahu, na schválené téma související s předmětem, měla by být vlastním dílem a neporušovat autorský zákon.

Probíraná témata předpokládají základní znalosti z oblasti operačních systémů, hardwaru a počítačových sítí, proto doporučuji předem absolvovat předměty Operační systémy (alespoň první část), Architektura počítačových systémů a případně Počítačová síť a internet. Nicméně není to podmínkou.

Značení

Ve skriptech je používána vyznačující symbolika, jejímž účelem je usnadnit orientaci a upozornit na místa s vysvětlením pojmů, postupy a odkazy na další informace.

Ve skriptech se používají následující barevné ikony:

na začátku každé kapitoly uvozuje *Rychlý náhled* (skript, kapitoly), ve kterém se dozvíme, o čem daná kapitola bude,

taktéž na začátku kapitoly určuje *Klíčová slova* kapitoly,

je ikona pro *Cíle studia*, dozvíme se zde, co nového se v dané kapitole naučíme,

tato ikona modré barvy určuje nový *pojem* a značení, používané symboly, nástroje, apod.,

ikona vyznačuje konkrétní *postupy* a návody pro různé činnosti či ukázky využití určitých nástrojů, příkazů a programů,

žlutou ikonou jsou označeny odkazy, na kterých lze získat *další informace* o tématu; nejčastěji zde najdeme webové odkazy na stránky, kde se dané tématice jejich autoři věnují podrobněji,

části textu označené fialovou ikonou jsou *nepovinné úseky*; jejich účelem je dobrovolné rozšíření znalostí studentů o pokročilá témata, na která obvykle při výuce nezbývá moc času.

Pokud je vysvětlení nového pojmu, postupu či souhrn odkazů na informace rozsáhlejší, je místo jediné ikony použito ohraničené prostředí.

Definice

Definice stanovují význam určitého (definovaného) pojmu. V těchto skriptech do definičního prostředí uzavíráme rozsáhlejší vysvětlení důležitých pojmů.

Postup

Delší a složitější postupy jsou uzavřeny v tomto prostředí, aby bylo možné na první pohled určit, kde popis postupu končí, a jak je tedy dlouhý.

Příklad

Takto vypadá prostředí s příkladem nebo popisem určité konkrétní situace, na které je vysvětlován určitý postup nebo je zdůvodňováno tvrzení. Neřešená varianta (Úkol) má zelenou ikonu .

Poznámka:

Důležité poznámky a upozornění či dodatečné informace zvýrazňujeme červenou ikonou, obvykle jsou uzavřeny v tomto prostředí, aby bylo zřejmé, kde text poznámky začíná a kde končí.

Další informace:

V tomto prostředí najdete souhrn odkazů na další související informace nebo odkaz, na kterém lze stáhnout probíraný program či dokumentaci.

Za některými kapitolami je seznam zdrojů informací a doporučených zdrojů pro případné další (samo)studium nebo semestrální práci.

Souhrn kapitoly

Za každou kapitolou najdete stručný souhrn informací o tom, co nového daná kapitola přináší. Účelem je připomenout si nejdůležitější části kapitoly a průběžně si udržovat přehled o svých nově nabytých či zopakovaných znalostech.

Obsah

Předmluva **iii**

1 Úvod do digitální forenzní analýzy **1**
1.1 Co je to digitální forenzní analýza . . . 1
1.2 Kyberkriminalita . . . 4
1.2.1 Co je to kyberkriminalita . . . 4
1.2.2 Osoby útočící nebo zkoumající . . . 5
1.2.3 Osoby reagující . . . 7
1.2.4 Dopady podcenění kyberkriminality . . . 8
1.3 Průběh forenzního zkoumání . . . 10
1.3.1 Co se zkoumá . . . 10
1.3.2 Jak se zkoumá . . . 10
1.3.3 Metodiky . . . 12
1.4 Bezpečnostní týmy a národní podpora bezpečnosti . . . 15
1.5 Legislativa, standardy, normy . . . 19
Literatura . . . 23

2 Základní průzkum systému **25**
2.1 Příprava a možnosti . . . 25
2.1.1 Nastínění základního postupu . . . 25
2.1.2 Jak získat ISO obraz paměťového média . . . 27
2.2 Postupy ve Windows . . . 28
2.2.1 Zobrazení seznamu připojovaných zařízení . . . 28
2.2.2 Změna způsobu připojení disku . . . 31
2.2.3 Vytvoření obrazu disku . . . 34
2.2.4 Připojení ISO obrazu disku . . . 36
2.2.5 Záchrana poškozených dat . . . 37
2.2.6 Přístup k různým souborovým systémům . . . 38
2.3 Postupy v Linuxu . . . 39
2.3.1 Jak Linux eviduje zařízení . . . 40

2.3.2 Připojení disku . . . 40
2.3.3 Vytvoření obrazu disku . . . 41
2.3.4 Připojení ISO obrazu disku . . . 45
2.3.5 Záchrana poškozených dat . . . 46
2.4 Hotová řešení . . . 46
2.4.1 Forenzní distribuce . . . 46
2.4.2 Forenzní prostředí . . . 48
2.5 Lokální a vzdálená metoda dead analýzy . . . 50
2.6 Ověření integrity . . . 51
2.7 Překážky . . . 54
2.7.1 Skryté oblasti na disku . . . 54
2.7.2 Removable Media Bit a výměnná média . . . 56
2.8 Datové sklady a BigData . . . 58
2.8.1 Skladování velkých objemů dat . . . 58
2.8.2 Jak na Big Data . . . 58
2.8.3 Hadoop . . . 59

3 Ochrana dat **64**
3.1 Obchod s daty . . . 64
3.1.1 Kdo a proč chce uživatelská data . . . 64
3.1.2 Jak se s osobními daty obchoduje . . . 66
3.1.3 PRISM alias Velký Bratr . . . 68
3.2 Smluvní podmínky – s čím souhlasíme . . . 70
3.3 Sociální sítě a vyhledávače . . . 72
3.3.1 Facebook . . . 72
3.3.2 Google . . . 75
3.3.3 Jiné sociální sítě . . . 75
3.3.4 Nástroje k řízení sociálních sítí . . . 75
3.4 Webové prohlížeče . . . 76
3.4.1 Cookies . . . 76
3.4.2 Nástupce cookies: FLoC . . . 80
3.4.3 Sledování a propojení . . . 81
3.4.4 Anonymní surfování a anonymní režimy . . . 84
3.5 E-mail – ověření odcizení identity . . . 88
3.6 Ztracené nebo odcizené zařízení . . . 89
3.6.1 Eliminace následků . . . 89
3.6.2 Prevence . . . 90
3.6.3 Webkamera . . . 92
3.7 Operační systémy coby sběrači dat . . . 94
3.7.1 Windows 10 . . . 94

4 Úvod do kryptografie a ověřování dat **97**

4.1 Co je to kryptografie . . . 97
4.2 Šifrování – o co jde . . . 98
4.3 Symetrická kryptografie . . . 99
4.3.1 Proudové šifry . . . 99
4.3.2 Módy blokových šifer pro zajištění důvěrnosti dat . . . 100
4.3.3 Módy pro zajištění integrity dat . . . 104
4.4 Příklady symetrických kryptovacích algoritmů . . . 106
4.4.1 Historické algoritmy . . . 106
4.4.2 Používané proudové šifry . . . 108
4.4.3 DES a 3DES . . . 108
4.4.4 AES – Rijndael . . . 109
4.4.5 Další symetrické blokové šifry . . . 110
4.4.6 Hash funkce pro zajištění integrity dat . . . 111
4.5 Asymetrická kryptografie . . . 113
4.5.1 Pár klíčů pro asymetrickou kryptografii . . . 113
4.5.2 Algoritmus . . . 113
4.5.3 Hybridní šifrování . . . 115
4.6 Příklady asymetrických kryptovacích algoritmů . . . 115
4.6.1 Diffie-Hellman . . . 115
4.6.2 RSA . . . 116
4.6.3 Elgamal . . . 118
4.6.4 DSA . . . 118
4.6.5 ECC – eliptické křivky . . . 118
Literatura . . . 119

5 Kryptografie v praxi **121**
5.1 Digitální podpisy a certifikáty . . . 121
5.1.1 Digitálně podepsaná zpráva . . . 121
5.1.2 Certifikáty . . . 123
5.1.3 Certifikační autorita . . . 125
5.1.4 PKI . . . 125
5.1.5 Jak získat certifikát v ČR . . . 126
5.2 Šifrování dat na paměťových médiích . . . 129
5.2.1 Šifrovací nástroje vestavěné ve Windows . . . 129
5.2.2 Šifrování disků v Linuxu . . . 131
5.2.3 Nástroje třetích stran . . . 132
5.3 Autentizace . . . 133
5.3.1 Autentizace na počítači . . . 133
5.3.2 Reset hesla do Windows pomocí SystemRescueCD . . . 137
5.3.3 Dvoufaktorová autentizace . . . 138
5.4 Bezpečná komunikace se serverem . . . 139
5.4.1 SSL a TLS . . . 139

5.4.2 Webový server a webový prohlížeč . . . 141
5.4.3 Vzdálená konfigurace serveru . . . 144
5.5 E-mail . . . 147
5.5.1 S/MIME . . . 147
5.5.2 PGP, GPG . . . 149
Literatura . . . 150

6 Bezpečnost síťové komunikace 151
6.1 E-mail . . . 151
6.1.1 Cesta e-mailu . . . 151
6.1.2 Z čeho se e-mail skládá . . . 152
6.1.3 Analýza e-mailu . . . 153
6.1.4 Konverzace s SMTP serverem . . . 157
6.2 Údaje o doméně a jejím vlastníkovi . . . 158
6.3 Struktura sítě . . . 159
6.4 Vyhledávání bezpečnostních problémů v síti . . . 160
6.4.1 Databáze zranitelností . . . 160
6.4.2 OVAL . . . 163
6.4.3 Nástroje pro skenování sítě . . . 164
6.5 Monitoring sítě . . . 167

Přílohy 169

A Nástroje a příklady 170
A.1 Autopsy . . . 170

Kapitola 1

Úvod do digitální forenzní analýzy

Předem je třeba upozornit, že v tomto předmětu se sice (kromě jiného) zabýváme digitální forenzní analýzou, ale probíraná látka ani zdaleka nepokrývá celou problematiku. Kompletní studium digitální forenzní analýzy by bylo na celý studijní obor.

Rychlý náhled: V této kapitole si projdeme základní pojmy z oblasti kybernetické bezpečnosti, zejména s ohledem na digitální forenzní analýzu. Určíme si, co je to kyberkriminalita, kterých osob se týká na obou stranách „fronty“, jak to může dopadnout, když kyberkriminalitu podceňujeme, jak obvykle probíhá forenzní zkoumání, jak je to v naší zemi s bezpečnostními týmy, a na konci stručně nahlédneme do legislativních souvislostí.

Klíčová slova: Digitální forenzní analýza, digitální forenzní audit, digitální prostředky, kyberprostor, kyberkriminalita, hacker, white hat, blue hat, black hat, cracker, phreaker, script kiddie, insider, soudní znalec, policejní expert, auditor, ICT odborník, bezpečnostní incident, digitální stopa, digitální důkaz, live analysis, dead analysis, integrita, metodika, bezpečnostní týmy, CERT, CSIRT, NÚKIB, Zákon o kybernetické bezpečnosti, GDPR, certifikace.

Cíle studia: Cílem kapitoly je získat základní přehled o pojmech, které se používají v oblasti kyberbezpečnosti a o bezpečnostní situaci v kyberprostoru. Student by po prostudování měl získat obecnou představu o potřebnosti a činnostech v oblasti digitální forenzní analýzy. Účelem není našprtat si pojmy zpaměti, účelem je jim porozumět.

1.1 Co je to digitální forenzní analýza

Pojem „forenzní“ se do našich končin dostal z angličtiny (forensic), a do angličtiny se dostal z latiny. Pojem „forum“ v latině znamená dvůr, náměstí, jednoduše místo, kde se scházejí lidé a rokují o nejrůznějších problémech, přeneseně místo, kde se veřejně rozhodují nejasnosti a spory.

Anglické „forensic“ se do češtiny obvykle překládá jako „soudní“, „forenzní analýza“ je (zatím neformálně) vědeckými argumenty podložená analýza daného objektu (předmětu), subjektu (osoby) včetně případného duševního stavu v daném čase, místa, situace apod. pro účely soudního či jiného rozhodovacího řízení. Na konkrétním účelu také záleží, jak moc striktní podmínky musí

tato analýza splňovat a jak musí být rozsáhlá. Dále probírané metody můžeme použít jednoduše ke zjištění způsobu bezpečnostního narušení firemní sítě, nebo může jít o práci soudního znalce pro trestněprávní soudní řízení.

Tato analýza tedy obvykle souvisí s určitým vědním oborem, s určitou profesí. Například:

- s medicínou, chemií a farmacií souvisí forenzní medicína, forenzní patologie, forenzní antropologie, forenzní toxikologie, analýza DNA,
- s psychologií souvisí forenzní psychologie,
- dokonce s několika vědními disciplínami okrajově souvisí forenzní daktyloskopie a další forenzní vědy, jejichž účelem je identifikace útočníka nebo oběti,
- rovněž s několika obory (ale především s ICT – informačními a komunikačními technologiemi, také například s psychologií, fyzikou apod.) souvisí digitální forenzní analýza.

Obecným účelem forenzních věd je rekonstrukce a zhodnocení uplynulého stavu či děje (kdo, co, kdy, kde, jak, případně proč).

☎ Pojem *digitální forenzní analýza* byl poprvé oficiálně definován roku 2001 ve zdroji [12], třebaže se příslušné metody používaly již dlouho předtím. Než si uvedeme přesnou definici, seznámíme se s dalšími potřebnými pojmy.

☞ Předně se tato analýza úzce dotýká určitých konkrétních *prostředků*. Tyto prostředky mohou být buď terčem útoku, nebo mohou být k útoku použity, anebo mohou naopak pomoci útok odhalit či zdokumentovat.

Do první skupiny (terč útoku) řadíme například server, na který zaútočili hackeři, bankomat, ze kterého byly odcizeny peníze, odcizený či škodlivým softwarem napadený počítač, notebook, tablet, mobilní telefon, GPS navigaci, USB flash disk, apod. Nemusí jít nutně o fyzická zařízení, lze sem zařadit například fotografie odcizené z účtu na sociální síti.

Do druhé skupiny řadíme počítač útočníka, síťový prvek, přes který byl veden útok na firemní infrastrukturu (může jít i o nelegálně připojený špatně zabezpečený Wi-fi router), apod.

Do třetí skupiny můžeme zařadit například videokameru, která nahrála průběh trestného činu nebo pomohla trestnému činu zabránit (například bylo možné včas odhalit pokus o atentát či teroristický čin), alarm, který informoval bezpečnostní službu o vloupání, nebo IDS/IPS zařízení, které zaznamenalo útok na síť či na něj přímo reagovalo změnou nastavení politik v síti.

Souhrnně budeme všechny tyto prostředky označovat jako *digitální prostředky (zdroje)* a budeme sem řadit nejen počítače a servery, ale obecně jakákoliv digitální zařízení, která mohou být k uvedeným účelům použita, a dále elektronický obsah (například výše zmíněné odcizené fotografie, elektronické certifikáty, nahrávky s dětskou pornografií, nasdílené cracknuté soubory s videoobsahem).

☞ Dalším typickým znakem digitální forenzní analýzy jsou *specifické metody*, které sice vycházejí z metod ICT (informačních a komunikačních technologií), ale musejí splňovat určité vlastnosti, aby získané závěry bylo možno použít při dokazování. Tyto závěry mají být nenapadnutelné a v případě nutnosti použitelné i před soudem. Uvedené metody musejí být vědecky podloženy – míněno především v rámci věd souvisejících s informatikou a komunikačními technologiemi. Pojem vědeckosti metod bude upřesněn dále.

Ve výše zmíněném zdroji je digitální forenzní věda definována následovně:

> „*Digital Forensic Science: The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.*“[12, str. 16]

Po (trochu volnějším) překladu do češtiny můžeme napsat následující:

Definice (Digitální forenzní věda[12] a digitální forenzní analýza[9])

Digitální forenzní věda určuje vědecky podložené metody za účelem shromažďování, uchovávání, ověřování, identifikace, analýzy, interpretace, dokumentování a prezentace digitálních stop a získaných z digitálních zdrojů, a to buď za účelem podpory rekonstrukce událostí potenciálně souvisejících s kriminální činností nebo za účelem predikce neautorizovaných činností potenciálně vedoucích k narušení bezpečnosti organizace.

Analytický proces, ve kterém používáme výhradně metody založené na digitální forenzní vědě, nazýváme *digitální forenzní analýzou*. Jedná se o vyšetřování, jehož účelem je objektivně zdokumentovat zúčastněné osoby, důvody, průběh a důsledky bezpečnostního incidentu nebo porušení práva státu či pravidel dané organizace.

Pokud zjištěné skutečnosti mají či mohou být použity v soudním řízení, se používá pojem *forenzní* (pozor – třebaže na začátku celé analýzy to tak nemusí vypadat, může být dodatečně určeno, že zjištěné skutečnosti budou poskytnuty pro účely oficiálního vyšetřování). Jinak jde prostě o digitální analýzu (dat).

Co se týče využití v organizaci (ať už malé firmě, velké komerční společnosti, státní organizaci či úřadě), pak obvykle platí, že digitální (forenzní) analýza se typicky provádí při podezření na problém (například podezření na narušení sítě) nebo může jít o preventivní činnost (sledování provozu na síti), kdežto *digitální forenzní audit* (důkladnější nezávislá obdoba digitální analýzy) se může provádět víceméně pravidelně a je zaměřen především pro „odchytávání“ dlouhodobějších problémů v oblasti bezpečnosti.

Definice (Digitální forenzní audit[9])

Digitální forenzní audit je hloubková digitální forenzní analýza prováděná třetí stranou – nezávislou osobou (auditorem).

Existují firmy, které jsou zaměřeny na provádění forenzního auditu. Forenzní audit je obvykle objednán osobou z managementu organizace, jsou domluveny podmínky (co vše bude kontrolováno, případně jakým způsobem), účelem je odhalit slabá místa v zabezpečení organizace. Osoby provádějící audit bývají kvalifikované zejména v oblasti ICT, kybernetické bezpečnosti, ale také psychologie (používají metody sociálního inženýrství). Důležité jsou také *důsledky auditu* – posílení zabezpečení organizace, změny ve struktuře počítačové sítě, změny v konfiguraci, přenastavení bezpečnostních politik, školení zaměstnanců, apod.

Digital (forensic) investigation – digitální (forenzní) vyšetřování je pojem označující celý proces vyšetřování, jehož nejdůležitější součástí je právě digitální (forenzní) analýza.

1.2 Kyberkriminalita

1.2.1 Co je to kyberkriminalita

Kybernetika je věda zastřešující informatiku, výpočetní techniku, komunikační technologie a další související technologické obory.

Kyberprostor je prostor vytvořený a udržovaný prostředky kybernetiky, často chápaný jako nefyzický (virtuální), třebaže stojí na fyzických základech (síťové infrastruktuře). Pod tímto pojmem obvykle rozumíme Internet, sociální sítě, virtuální světy, prostředí počítačové infrastruktury, telekomunikační sítě, prostředí v elektronických zařízeních apod.

Podle Zákona o kybernetické bezpečnosti[16] je kyberprostor „digitální prostředí umožňující vznik, zpracování a výměnu informací, tvořené informačními systémy, a službami a sítěmi elektronických komunikací“.

Definice (Kyberkriminalita)
Kyberkriminalita (kybernetická kriminalita) je kriminalita zaměřená proti kybernetickým prostředkům nebo takovými prostředky páchaná, anebo kybernetické prostředky (např. počítačová síť nebo sociální média) tvoří prostředí, v němž probíhá.

Také lze říci, že kyberkriminalita je kriminalita probíhající v kyberprostoru.

Můžeme se setkat i s jinými názvy – například *počítačová kriminalita* (ale tento pojem není přesný, trestný čin se dá páchat i jinými technickými prostředky než počítači), nebo *high-tech kriminalita* (tj. použití sofistikované techniky pro páchání trestných činů).

Kyberkriminalita má oproti jiným druhům kriminálních činů jedno specifikum – digitální stopy, které při vyšetřování kyberkriminality hledáme, jsou hůře dohledatelné, nestálé a náchylné k poškození, kompromitování, v čase rychle mizí (dají se snadno zahladit, zničit), popřípadě vůbec nevzniknou (například pokud v síti není řádně vyřešeno logování). Chytit podezřelého při činu je náročnější, protože obvykle na místě činu nebývá osobně.

Odhalení podezřelého je složitější (také co se týče přiřazení virtuální identity ke skutečné – fyzické – identitě podezřelého), je náročnější mu dokázat trestný čin, také prevence je problematická. Taktéž identitu případných svědků je náročnější prokázat. V tomto směru je nejhorší situace taková, kdy podezřelý či svědkové se fyzicky nacházejí v jiném státě než kde je kyberkriminalita páchána. Specifickou vlastností kyberprostoru jako celku je globálnost (zejména díky Internetu).

V tabulce 1.1 vidíme srovnání typických parametrů zločinu provedeného ve fyzickém světě (ozbrojená bankovní loupež) se srovnatelným kybernetickým zločinem (také ozbrojená loupež, ale na dálku s využitím kybernetických prostředků). Uvedené hodnoty se vztahují k americkým zákonům, vycházejí ze statistik FBI. Jak vidíme, přesun do kyberprostoru se zločincům rozhodně vyplatí ve všech směrech. Jedinou nevýhodou je nutnost určitého hardwarového, softwarového a znalostního vybavení.

Parametr	**Průměrné ozbrojené přepadení**	**Průměrný kybernetický útok**
Riziko	Pachatel může být zraněn nebo zabit	Bez rizika fyzického zranění
Zisk	Prům. 5–6 tis. USD	Prům. 50–500 tis. USD
Pravděpodobnost dopadení	50–60 %	cca 10 %
Pravděpodobnost odsouzení	cca 95 %	15 % dojde k soudnímu řízení, z toho 50 % skutečně odsouzeno
Trest	Prům. 5–6 let, pokud pachatel někoho nezranil	Prům. 2–4 roky

Tabulka 1.1: Porovnání následků klasického a kybernetického trestného činu[8]

Nutnost určitého hardwarového, softwarového a znalostního vybavení je samozřejmě nutná i na „bílé“ straně, tedy u orgánů činných v trestním řízení, přičemž u nich musíme přidat ještě jednu nevýhodu – jako první většinou táhne kyberzločinec (má „v šachu bílé figury“) a na jeho straně je moment překvapení a prvotní volba nástrojů.

1.2.2 Osoby útočící nebo zkoumající

Kdo vlastně páchá zločiny v kybernetickém prostoru? Projděme si příslušné názvosloví.

Pod pojmem *hacker* byl v 50. letech chápán technicky velmi zdatný člověk, který dokázal často až nečekaně úspěšně řešit různé technicky náročné situace (například zlepšit výkon svého zařízení, obejít omezení daná jeho technickým stavem, atd.). V té době to byl termín spíše neutrální až pozitivní. V současné době tímto pojmem také rozumíme odborně fundovaného člověka (alespoň většinou) přístupného nejrůznějším výzvám ve svém oboru, nicméně pozitivní význam se postupně ztrácí. Rozlišujeme několik druhů hackerů:

White Hat – „hodný“ (etický) hacker, obvykle člověk, který provádí bezpečnostní audity, tedy zkoumá zabezpečení daného systému na zakázku a získané informace (o bezpečnostních problémech) poskytuje správci infrastruktury,

Blue Hat – obdoba etického hackera, zaměřuje se na testování softwaru a obecně systémů před jejich uvedením na trh; v současnosti jsou tito lidé buď přímo zaměstnáváni softwarovými firmami, nebo pracují nezávisle (také *Grey Hat*) a za úplatu (podle závažnosti problému) vyhledávají bezpečnostní mezery v softwaru,

Black Hat – správnější název je cracker (viz dále).

Cracker je osoba využívající své technické a jiné znalosti a dovednosti k nelegálním účelům. Crackeři pronikají do systémů, ve kterých nemají co dělat, prolamují bezpečnostní ochranu chráněného softwaru a dat (hudba, video apod.), vytvářejí či používají software umožňující takovou činnost, případně software vytvářející a řídící sítě napadených počítačů, které zneužívají k rozesílání spamu nebo k provádění útoků, atd.

Úmysly crackerů obvykle nebývají zrovna čisté, tito lidé využívají své schopnosti pouze pro vlastní obohacení, politické či náboženské cíle (v negativním smyslu slova) nebo jednoduše z potřeby někomu uškodit, z nenávisti. V současné době se crackeři sdružují do skupin, také jsou velmi pravděpodobně najímáni vládami či státními organizacemi některých států a provádějí politicky či nábožensky motivované útoky.

Hacktivista je hackerský aktivista. Od běžného crackera se liší hlavně svou motivací (ochrana životního prostředí, práv menšin apod.), například se rozhodne útočit na servery či účty na sociálních sítích společnosti těžící ropu ve vrtech pod mořem, aby upozornil na neekologické praktiky této společnosti.

Phreaker je osoba zneužívající telekomunikační systémy (vlastně tato činnost byla jednou z oblíbených činností některých původních hackerů v 50. letech). Phreaker se dokáže nenápadně napojit na telekomunikační linku a vést hovory či přistupovat na internet zcela zdarma.

Script kiddie je začátečník bez rozsáhlejších technických znalostí. Tito lidé typicky využívají hackerské sety naprogramované někým jiným (v šedé zóně Internetu se podobné produkty dají snadno sehnat).

Insider je jinak běžný, ale ve skutečnosti nespokojený zaměstnanec určité firmy (na druhou stranu „outsider“ je „člověk zvenčí“), případně může jít o obchodního partnera autorizovaného pro přístup do firemní sítě. V literatuře také můžeme najít konkrétnější název „malicious insider“. Hodně záleží na skutečných možnostech dotyčného insidera – člověk s přístupem do databáze zaměstnanců nebo do systému řízení dopravy může napáchat mnohem více škody než člověk pracující u pásu.

Do skupiny insiderů můžeme zařadit i takové zaměstnance, kteří neškodí úmyslně, ale důsledky jejich činnosti z neznalosti nebo neschopnosti mohou být stejné.

Příklad

Vezměme si zaměstnance, který v zaměstnání používá více zařízení, která mají (třeba i kromě jiného) bezdrátové síťové rozhraní. Tento zaměstnanec se rozhodne zjednodušit si život a z domu si donese vlastní pěkný levný Wi-fi router nebo access point. Připojí ho do firemní sítě bez toho, aby kohokoliv informoval nebo požádal o dovolení (taková zařízení jsou pak označována jako „rogue access point“ – nelegální, škodící AP, případně „neautorizovaný“).

Tento zaměstnanec si neuvědomil, že tím, co udělal, vytvořil ve firemní síti zadní vrátka – pokud toto jeho zařízení není dostatečně zabezpečeno a může se k němu asociovat (připojit) kdokoliv, kdo je v dosahu (třeba na ulici za oknem), pak se kdokoliv může dostat i dál. Zaměstnanec totiž svůj router (access point) určitě připojil k rozhraní určenému pro připojení důvěryhodných zařízení. Počítačová síť je natolik zabezpečená, jak je zabezpečen její nejslabší článek.

Jako insidera můžeme brát i takového zaměstnance, který v pracovní době místo plnění pracovních povinností brouzdá po síti, hraje hry, čte články na zpravodajských webech, vkládá komentáře na Facebook či Twitter apod. – pokud to ovšem není náplň jeho pracovní činnosti. Proč? Protože zneužívá firemní síť a jemu svěřená zařízení k soukromým účelům a neplní řádně své pracovní povinnosti. Svému zaměstnavateli generuje zbytečné personální náklady.

„Škodící“ zaměstnanec či partner je schopen napáchat hodně škody, protože má nejen motivaci k nekalé činnosti, ale také má možnosti a vyšší přístupová práva než hacker zvenčí. Nespokojení zaměstnanci mohou například vynášet tajné a citlivé informace z firmy, zveřejňovat údaje zjednodušující někomu jinému průnik do informačního systému firmy, případně jsou schopni vytvořit „zadní vrátka“ k firemnímu systému či alespoň dovnitř sítě.

Poznámka:

Škodlivý vliv insiderů byl probírán v mnoha studiích, můžeme se podívat například na [7]. V této studii se například píše, že 87 % insiderů si vystačilo s takovými metodami, které běžně používali ve svém zaměstnání, přičemž v 70 % proběhl incident v normální pracovní době. U 85 % insiderů o jejich plánech věděl (alespoň částečně) někdo další – rodina, přátelé, kolegové, atd., ve většině případů šlo o osoby do činu zapojené nebo z něj mající prospěch. Přibližně v třetině případů bylo možné insidera včas zachytit a zabránit mu ve škodním jednání. V 61 % případů byli insideři zjištěni osobami, které toto zjišťování nemají v popisu práce.

Je to pouze jedna z mnoha studií, navíc cílená na bankovní a finanční sektor v USA. V jiných sektorech (a oblastech světa) by výsledky byly odlišné, nicméně dostatečně ilustrují nebezpečí podcenění vlivu nespokojených či odcházejících zaměstnanců. Obranou je správná motivace, v případě propuštění okamžité zamezení přístupu do systémů či na fyzická místa, kde by bylo možné napáchat škody (propouštěný zaměstnanec může být převeden na jinou práci nebo jít na nucenou dovolenou), důsledné dodržování bezpečnostních politik, monitorování.

1.2.3 Osoby reagující

Podíváme se na opačnou stranu barikády. Kdo proti kyberkriminalitě bojuje? Především se jedná o osoby fundované v oblasti ICT, případně jiných oborech.

Soudní znalec je osoba splňující podmínky dané zákonem (viz [18]) – bezúhonnost, fundovanost v daném oboru, apod., seznam registrovaných soudních znalců pro různé forenzní vědní obory je veden na krajských soudech. Typicky se práce soudních znalců používá v soudním řízení, ale není to nutná podmínka (například je možné, že vyšetřovaný případ před soud ani nedorazí).

Policejní expert je (obvykle) zaměstnanec Policie ČR s potřebnými technickými znalostmi proškolený k forenznímu ohledání místa činu se zaměřením na digitální stopy, případně k následnému zkoumání těchto stop mimo místo činu.

Auditor provádí bezpečnostní audit v organizacích, se kterými má pro tento účel sjednanou smlouvu. Měl by být špičkovým odborníkem v oboru podobně jako soudní znalec. Firmy, které smluvně provádějí bezpečnostní audit, by samy měly občas procházet důkladným bezpečnostním auditem.

Odborník v oboru ICT může provádět některé činnosti, které mají charakter digitální analýzy, ale pokud se výsledky analýzy mají dotáhnout až před soud, měl by spolupracovat soudní znalec a je třeba zachovávat všechny potřebné podmínky. Pokud jde jen například o prověření toku dat na síti pro účely správného nastavení bezpečnostních politik či zjištění příčin některých problémů, nemusí se jednat o soudního znalce.

Velké společnosti nebo organizace státní správy by měly mít vlastní *Computer Response Team* nebo *Security Team*, tedy tým odborníků (kromě jiného v oblasti ICT) určený k zásahům v případě zjištění jakéhokoliv bezpečnostního problému.

1.2.4 Dopady podcenění kyberkriminality

Kyberkriminalita má negativní dopady různého druhu (taky záleží na tom, kdo konkrétně byl postižen). Mnoho lidí si myslí, že jde především o materiální a finanční ztráty, ale vážnou škodou může být ztráta důvěry u zákazníků a obecně obchodních partnerů – poškození pověsti, ztráta pracovního času odborníků, kteří se musejí věnovat vyšetřování bezpečnostního incidentu (z toho také vyplývají finanční ztráty), místo aby se věnovali něčemu „užitečnějšímu“ – náklady na vyšetření incidentu, a v případě útoku na stát nebo pro stát a občany důležitou infrastrukturu může jít i o ohrožení národní bezpečnosti. Prozrazením obchodního či výrobního tajemství (tj. intelektuálních hodnot, know-how) může firma nejen přijít o velkou část zisku, ale i zkrachovat. Dotyčného zaměstnance také označujeme jako (škodlivého) insidera.

Příklad

Představme si následující situaci: zaměstnanec technologické (nebo jiné, to teď není důležité) společnosti ztratí firemní notebook (nechá ho ve špatně zajištěném autě, někdo mu ho odcizí na ulici, během dovolené ho nechá doma a je vykraden, atd.). Situace se může dále vyvíjet různě:

- Jaká data v notebooku jsou? Mohou být pro někoho (třeba konkurenci) přínosná? Mohou být zpeněžena? Mohou původního vlastníka nějak ohrozit?
- Komu konkrétně se notebook dostane do rukou? Bude to člověk, který chce prostě „jen notebook“, nebo se zařízení dostane do rukou člověku, kterého zajímají i data a ví, jak se k nim dostat? Bude dotyčný chtít data zpeněžit či jinak zneužít? Ví, jak najít zájemce? Nebyl náhodou notebook získán „na zakázku“ (konkurenta, apod.)?
- Jsou data v notebooku šifrována? Jakým způsobem (jak silný je šifrovací klíč, jaký algoritmus je použit, atd.)? Co konkrétně je chráněno – není možné se k datům dostat vyjmutím disku z notebooku a připojením do jiného počítače?
- Je notebook zabezpečen nějakou anti-theft technologií? Pokud ano, je tato technologie natolik dobrá, že ji nelze obejít? Případně – je „novým majitelem“ odhalitelná? Je možné tuto technologii použít nejen k snadnějšímu vypátrání notebooku, ale také k preventivnímu znepřístupnění dat – jakým způsobem? Jak rychle a jakým způsobem je možné to vše provést? (O anti-theft technologiích se budeme bavit později.)
- Jsou data z notebooku někde zálohovaná?

Nejhorší situace by asi byla taková, kdy na odcizeném notebooku jsou data nejen pro danou společnost důležitá, ale také mohou být velmi cenná pro konkurenci, přičemž zabezpečení notebooku rozhodně není ukázkové. Celková škoda se pak vyčísluje velmi těžce.

Další informace:

Podobné situace bohužel ve světě nastávají a bylo by vhodné se jim vyhnout. Na Internetu můžeme najít různé studie typu „The cost of a lost laptop“ (cena ztraceného notebooku), například na [14]

(autorem studie z roku 2009 je Ponemon Institute, je sponzorovaná společností Intel), a hodně dalších článků a studií se na ni odvolává.

V této studii najdeme informaci, že průměrná cena ztraceného notebooku je $50 000 (slovy: padesát tisíc dolarů), tedy přesně $49 246 – přibližně 80 % z toho je právě cena ztracených dat. Čím dřív je ztráta zjištěna a čím dřív společnost reaguje, tím menší jsou škody (při zjištění tentýž den klesne ztráta na cca $9 000).

Pokusům o průnik do firemní sítě se dá do určité míry zabránit tím, že firma má dostačující hardwarové a softwarové vybavení, správně nakonfigurované, existují správně formulované firemní bezpečnostní předpisy, o techniku se starají fundovaní zaměstnanci (opravdu starají, monitorují, průběžně aktualizují, mění konfiguraci podle momentální bezpečnostní situace apod.), zaměstnanci jsou proškoleni a mají správně přidělena přístupová oprávnění (každý má přístup pouze tam, kam potřebuje), jsou patřičně motivovaní k věrnosti firmě a problematika insiderů je brána vážně.

Cokoliv z toho, co je jmenováno v předchozím odstavci, může být problém, pokud to není dodrženo. O většině těchto problémů se pojednává v předchozím nebo následujícím textu, nyní se zaměříme na technické vybavení.

V této době je ve většině společností a firem zapotřebí určité technické vybavení určené k bezpečnostním účelům – firewally, IDS/IPS, antivirové programy, ale také taková (jinak běžná) síťová zařízení, která kromě své běžné funkce plní i určitou bezpečnostní funkci (za což se připlácí). Ve firmách (no, vpodstatě všude) o nákupu těchto zařízení rozhodují lidé z managementu.

Předně se tito lidé potřebují dozvědět, že je nutno takové zařízení koupit, a dále musejí být přesvědčeni, že je tento nákup opravdu nutný. První podmínka není až tak nesplnitelná, u druhé je splnění mnohem těžší. Náklady na bezpečnostní zařízení (nebo navýšení nákladů o bezpečnostní funkce) totiž firmě negenerují žádný přímý zisk, a může být velmi těžné technicky málo vzdělaného manažera přesvědčit. Je třeba nejen žádost formulovat s ohledem na nižší ICT gramotnost manažera, ale také vhodně zdůraznit možné negativní důsledky na hospodářské výsledky v případě podfinancování oblasti bezpečnosti.

Ve zdroji [11] (zajímavě ilustrovaná studie z roku 2014 provedená společností McAfee ve spolupráci s Intelem) najdeme informaci, že celosvětové roční náklady na kyberzločin jsou víc než 400 miliard dolarů (v angličtině se pro označení miliardy používá pojem billion). Tento odhad může být dokonce podhodnocený, protože o některých činech se často nedozví nejen veřejnost, ale ani orgány činné v trestním řízení. Také může být u různých společností a států používána odlišná metodika na určení nákladů souvisejících/nesouvisejících s bezpečností a kyberkriminalitou. Ve studii jsou také informace o těchto nákladech ve vztahu k HDP různých států (včetně ČR).

Zajímavé jsou také přílohy – například v příloze B najdeme tabulku se změnami na trhu s různými bezpečnostními technologiemi v letech 2011–2013 (není tam napsáno, v jakých jednotkách jsou vypsané údaje, ale jde o miliony USD). Najdeme tam například informaci, že trh s firewally během těchto let rostl o 9.3 %, ale trh s NG firewally (pokročilejší filtrace, apod.) o 43 %. Trh s forenzními technologiemi narostl o 67 %.

Poznámka:

Čím dál víc se ve firmách prosazuje princip BYOD (Bring Your Own Device), kde je zaměstnancům umožněno používat v zaměstnání své vlastní zařízení. Zaměstnanci svá zařízení každodenně

přenášejí mezi domácností a pracovištěm a stoupá nebezpečí zneužití dat uložených na těchto zařízeních. Navíc v mnoha firmách ve skutečnosti nemají infrastrukturu ani bezpečnostní politiku na BYOD připravenou, často ani v případech, kdy si myslí, že připraveni jsou.

1.3 Průběh forenzního zkoumání

1.3.1 Co se zkoumá

Pokud nepočítáme bezpečnostní audit, kdy obvykle není předem zřejmé, na co se při zkoumání máme zaměřit, obvykle je vyšetřován bezpečnostní incident.

Bezpečnostní incident je taková událost, která má negativní vliv na plánovaný chod či funkci daného systému nebo takový vliv může mít do budoucna.

Bezpečnostním incidentem může být například vynesení informací o chystaném patentu nespokojeným zaměstnancem, použití USB flash disku infikovaného škodlivým softwarem na firemním počítači, neoprávněný přístup do informačního systému firmy, neoprávněné pozměnění konfigurace síťových prvků (případně pozměnění údajů v zónových souborech DNS serverů, promazání seznamu pravidel ve firewallu), vytvoření zadních vrátek ve firemní síti, ...

Digitální stopa a digitální důkaz (digital evidence – v angličtině se pojmy stopa a důkaz nerozlišují) je hodnota důležitá pro dané vyšetřování uložená v digitální podobě nebo v takové podobě přenášená. Existuje obecnější definice: digitální stopa je informace, kterou po sobě zanechává konkrétní uživatel dané digitální služby (například komentář na Facebooku).

Pod tímto pojmem chápeme například data uložená na pevném disku zkoumaného počítače, data na výměnných médiích, logy na síťových zařízeních ve firemní síti, data a konfiguraci v mobilním telefonu, záznam z bezpečnostní kamery, atd.

Digitální stopa může být *nestálá* – není uložena v žádné permanentní paměti a je nezbytné ji získávat z běžícího systému. Typicky se jedná třeba o obsah vyrovnávacích pamětí, operační paměti počítače, na síťových prvcích je obvykle momentální konfigurace také nestálá (pokud není pro uložení momentální konfigurace použita flash paměť, což obvykle není – byla by pomalejší, pak po restartu tyto informace ztratíme). Sběr nestálých dat je náročnější, protože se musí provádět „na místě“ a je složitější dokázat, že s daty nebylo neoprávněně manipulováno.

1.3.2 Jak se zkoumá

Již na začátku je dobré předpokládat, že zjištěné závěry by mohly být použity v soudním řízení (i když to tak za začátku vůbec nemusí vypadat). Původně se může jednat o bezpečnostní audit nebo o prověření podezření na činnost insidera. Zkoumání by měla provádět fundovaná osoba, protože při špatném postupu může dojít ke kompromitování digitálních stop, čímž přestanou být pro případné soudní řízení použitelnými.

Forenzní zkoumání by mělo splňovat tyto náležitosti:

Legalita. Vše, co je v rámci zkoumání prováděno a získáváno, musí být provedeno a získáno legální cestou. Například nelze použít výsledky nelegálního odposlechu nebo data z odcizeného USB flash disku.

Integrita. Cokoliv, co se děje v souvislosti se zkoumáním, musí být prováděno tak, aby bylo zřejmé, že zkoumání nebylo zmanipulováno (cíleně či náhodně ovlivněn a pozměněn výsledek), kdo, kdy, jak a proč zkoumání provedl.

Opakovatelnost. Celý postup musí být důkladně dokumentován a mohou být použity pouze takové postupy, které toto dokumentování umožňují; účelem je umožnit případné revizní zkoumání provedené stejným či ekvivalentním způsobem, ale jinou osobou, při revizním zkoumání by se mělo dojít ke stejným závěrům. Je zřejmé, že potenciálně destruktivní metody je možno použít pouze na kopii zkoumaných dat.

Nepodjatost. Zkoumající osoba by měla být zcela nezávislá na jakékoliv osobě či organizaci (subjektu), jehož se zkoumání jakkoliv dotýká.

Součástí dokumentace je nejen popis stavu dat a případně hardwaru a softwaru při jejich obdržení a informace o zkoumajícím, ale také například důkladná fotodokumentace.

Důležitou součástí zkoumání digitálních stop je sběr dat a jejich analýza. Rozlišujeme:

- *Live Analysis* – sběr dat z „živého“ systému, tedy za běhu tohoto systému; účelem je především získat nestálá data, která by byla po vypnutí systému ztracena, nebo analyzovat systém, který nemůžeme vypnout.
- *Dead Analysis* (Off-line Analysis) – sběr dat v předem připraveném prostředí, data jsou pro tento účel dodána v původním, nepozměněném stavu (například je doručen pevný disk vyjmutý z kompromitovaného systému).

Pro volbu analýzy živého systému mohou být různé důvody. Například systém může při komunikaci používat dočasné šifrovací klíče, nebo data na pevném disku jsou šifrována, kdežto v operační paměti se s nimi pracuje v nešifrovaném tvaru, zajímají nás soubory v dočasných souborových systémech na RAM discích (jakési virtuální disky v operační paměti, po vypnutí systému jsou taková data smazána), také nás mohou zajímat běhové informace (momentální stav systému), atd. Dalším důvodem může být nutnost zaznamenat kromě jiného i momentální datum a čas či získat logy, ve kterých nebude zaznamenáno vypnutí systému. Důvodem bývá také nutnost analýzy systému, který z nějakého důvodu nemůžeme vypnout (server či celý cluster serverů, aktivní síťové prvky v rozsáhlejší síti, apod.), nebo nechceme útočníka informovat o zjištění útoku vypnutím.

Příklad provedení live a dead analýzy najdeme například na [13].

Při live i dead analýze je především důležité zamezit *kompromitaci zkoumaného materiálu.* Zkoumaný materiál je kompromitován, pokud se jeho stav změní oproti stavu, ve kterém byl před započetím zkoumání – například omylem došlo k vymazání, změně obsahu nebo změně parametrů souboru (datum posledního přístupu k souboru, přístupová oprávnění, vlastnictví souboru apod.) na paměťovém médiu, vytvoření nového souboru, přidání dalších záznamů do logu na zkoumaném zařízení, instalaci nové aplikace, odstranění některého ovladače či dokonce celého periferního zařízení, apod.

Ke kompromitaci může dojít omylem či technickou nebo lidskou chybou kdykoliv během zkoumání, proto (pokud je to možné) je třeba předem vytvořit několik kopií datového obsahu. Pak zkoumáme některou z kopií, nikoliv originál, aby nedošlo k jeho pozměnění. Pokud porušíme *integritu* kopie (kompromitujeme ji), není problém přejít k další kopii, která je zatím nedotčená. Vytvoření dostatku kopií nám usnadní také dodržení podmínky *opakovatelnosti.*

Poznámka:

Jak později zjistíme, systém Windows nám s vyhýbáním se kompromitaci zkoumaných dat moc nepomáhá. Například při průzkumu pevného disku (třeba přeneseného z jiného zařízení) Windows automaticky tento disk zpřístupňují hned po fyzickém připojení, a to v režimu čtení i zápisu. Při tomto typu zpřístupnění nejen hrozí kompromitace „omylem“, ale také se automaticky provádí inicializační a logovací zápisy do některých datových struktur souborového systému na disku. Podobně se bohužel chovají i mnohé rozšířené distribuce Linuxu a jiných UNIX-like systémů.

Proto je lepší v podobných případech použít některou distribuci Linuxu specializovanou pro tento účel (na některé z nich se později podíváme), případně existují tzv. write-blockery.

1.3.3 Metodiky

Metodika (angl. methodology) je obecně popis pracovního postupu. Metodika popisující postupy z oboru ICT nebo příbuzných obvykle stanoví tyto parametry:

- z jakého důvodu (za jakým účelem) se postup provádí, motivace,
- do jakých fází má být postup členěn,
- jaké základní zásady je třeba dodržovat jak po celou dobu, tak v jednotlivých fázích,
- kdo má daný postup nebo jednotlivé fáze provádět,
- kde má být postup proveden, případně konkrétně pro jednotlivé fáze,
- co a jak se v jednotlivých fázích zpracovává,
- jaké prostředky a nástroje jsou pro jednotlivé fáze povoleny,
- jak se má provádět dokumentace a jak má vypadat výstup,
- jakým způsobem má být výstup předán či prezentován.

To je pouze orientační (a příliš podrobný) výčet. Ve skutečnosti mohou být některé parametry předem jasné, tudíž nemusejí být zvášť uváděny, nebo naopak natolik nejasné, že nemohou být konkretizovány.

Pokud jde o digitální forenzní analýzu, existuje více metodik (některé obecné a některé naopak pro konkrétní typ zkoumaných objektů). Některé jsou určeny pro live nebo dead analýzu, jiné zahrnují obojí. Ne každá metodika je také použitelná přímo na ten typ zkoumání, který musíme provést, specifický přístup například vyžaduje zkoumání dat v cloudu nebo v mobilních telefonech.

Účel je obvykle dán a základní zásady jsou dány tímto účelem (především je nutné zamezit kompromitaci zkoumaných dat, obecně zajištění náležitostí uvedených na straně 10).

Podíváme se na několik možných metodik – to ovšem není vyčerpávající výčet. U těchto metodik se soustředíme jen na rozdělení činností do fází, protože především v tom se liší. Osobu provádějící zkoumání budeme označovat pojmem „znalec“, ovšem ve skutečnosti se nemusí jednat o soudního znalce (obecně by to měla být osoba znalá potřebných postupů).

Jednoduchá metodika. Rozlišujeme tři fáze:

Sběr dat. Znalec stanoveným způsobem získá objekty zkoumání – digitální stopy (například pevný disk ze zkoumaného počítače doručený důvěryhodnou osobou, případně je hardware získán na fyzickém místě činu, záběry kamer jsou vyžádány od správce těchto zařízení, atd.). Už

tato fáze musí být dostatečně dokumentována (například fotografie nebo videozáznam dokumentující zařízení na místě činu, zvukový záznam s popisem relevantních skutečností,...).

V této fázi také pořizujeme kopie zkoumaných dat, například bitové obrazy disků, snímáme nestálá data ze systémů a vytváříme jejich kopie. U získaných dat vytváříme digitální otisky (obdobu kontrolních součtů, budeme probírat později), aby bylo možné kdykoliv později odhalit kompromitaci těchto dat.

Analýza. Získaná data podrobujeme analýze. Procházíme veškeré získané digitální stopy a hledáme data relevantní k probíhajícímu vyšetřování (soubory, záznamy v log souborech, e-maily obsahující určitý druh informací či s konkrétními adresami, procházíme historii webového prohlížeče, cookies, atd.). Většinu práce tvoří hledání a určování, nicméně sem řadíme také případné dešifrování, prolamování klíčů, hledání souvislostí mezi nalezenými údaji, používání statistických metod, pokusy o obnovu zničených dat, apod.

V této fázi můžeme dodatečně zjistit, že některá potřebná digitální stopa chybí. Proto se můžeme vrátit do první fáze a opakovaně provést sběr s tím, že se budeme soustředit i na takové stopy, které jsme dřív nepovažovali za potřebné.

Cílem fáze je analýzou stop zjistit, co a jak se na (někdy virtuálním) místě činu stalo, případně kdo a proč co provedl – rekonstrukce události. Také v této fázi je třeba každý krok podrobně zdokumentovat.

Prezentace. Znalec stanoveným způsobem informuje příslušné osoby – Policii ČR, státní zastupitelství, nadřízeného v organizaci, apod. Pokud se případ dostane k soudu, znalec je povinen o svých zjištěních ústně referovat před soudem.

Prezentace výsledků zkoumání musí splňovat určitá kritéria, například mohou být vyžadovány vyplněné formuláře, doložena kvalifikace znalce, apod.

✎ **Podrobnější metodika.** V této metodice je pět fází, ale vpodstatě jde o rozčlenění fází předchozí metodiky.

Identifikace. Je třeba určit, co konkrétně má být zkoumáno, určujeme digitální stopy.

Sběr. Tato fáze představuje samotné shromažďování digitálních stop, včetně získávání dat z živého systému. V případě potřeby se vracíme k fázi identifikace.

Uchování. Vytváříme kopie, bitové obrazy, u všech dat digitální otisky (hash), apod. V případě potřeby se vracíme k předchozím fázím.

Analýza. Procházíme digitální stopy (resp. kopie), určujeme, třídíme, zpracováváme atd. Zhruba odpovídá fázi analýzy v předchozí metodice.

Prezentace. Totéž jako poslední fáze podle předchozí metodiky.

✎ **Metodika podle [4].** Tato metodika je velmi podrobná – popisuje celkem 17 fází, které jsou rozděleny do pěti skupin. Soustředíme se na tyto skupiny, podrobnosti najde čtenář v [4].

Přípravné (Readiness) fáze. Cílem těchto dvou fází (operační příprava a příprava infrastruktury) je zajistit přípravu na zkoumání tak, aby byly včas k dispozici všechny prostředky, znalosti a dovednosti potřebné pro zkoumání. Předem je nutné připravit vše, co by mohlo být potřebné na místě činu nebo v laboratoři – personál provádějící zkoumání musí být proškolen,

vybaven nejnovějšími informacemi v oboru, hardware a software zprovozněn a aktualizován, dokumentační prostředky taktéž, atd.

Tyto fáze jsou v předchozích metodikách vynechány, resp. považovány „za samozřejmé“.

Detekční (Deployment) fáze. Ve fázi detekce je bezpečnostní incident zjištěn a oznámen (buď člověkem nebo detekčním systémem, jako je firewall nebo IDS). Následuje fáze potvrzení a autorizace, ve které je třeba získat předběžné informace o místě činu a přesunout na toto místo pověřené osoby. Do této fáze patří také případné formální úkony jako je získání povolení k domovní prohlídce, pokud je zapotřebí. Ve firemním prostředí prověřené osoby ověří, zda opravdu došlo k narušení bezpečnosti a které systémy byly zasaženy. Místo činu je vhodně označeno (fyzické místo činu třeba hraniční páskou či pečetí) a ochráněno (i před počasím), musíme zajistit, aby nedošlo ke kompromitaci místa činu a zničení (digitálních) stop.

Fáze průzkumu fyzického místa činu. V těchto 6 fázích procházíme a dokumentujeme fyzické místo činu, cílem je sběr a analýza fyzických stop a rekonstrukce událostí, které se staly těsně před incidentem. Také sem řadíme zjištění osob, které by mohly mít s incidentem cokoliv společného, jsou formálně odpovědní, případně mohou při vyšetřování pomoci.

V této fázi se vytvářejí předběžné hypotézy, které pomáhají při odhadu důležitosti nalezených stop a při určování, co je třeba zdokumentovat. Během této fáze se tyto hypotézy mohou měnit a upravovat podle momentálního stavu průzkumu místa činu.

Fáze průzkumu digitálního místa činu. Tyto fáze probíhají v součinnosti s fázemi z předchozí skupiny, protože digitální stopy se často získávají právě z nalezených fyzických důkazů (notebooků, kamer, atd.) a navíc hypotézy tvořené v obou skupinách fází se navzájem doplňují a ovlivňují. Rozhodně je vhodné, aby alespoň někdo z těch, kdo provádějí digitální průzkum, byl přítomen při průzkumu fyzického místa činu, aby zabránil případné kompromitaci digitálních důkazů z neznalosti.

Digitální místo činu může být fyzicky ohraničeno (pokud se incident udál pouze v rámci jednoho počítače či mobilního zařízení), ale často fyzické hranice nemá (například pokud se jednalo o čin spáchaný přes Internet). Navíc se digitální stopy hůře získávají a analyzují, snadno se kompromitují (vytváření kopií také patří do této fáze). Proto jsou tyto fáze považovány za časově, prostorově a znalostně náročnější. Je třeba prozkoumat a vytřídit jak digitální stopy, tak i celé digitální místo činu. Také zde se vytvářejí hypotézy, ověřují se, srovnávají se s hypotézami jiných fází, upravují se. I v těchto fázích je důležitá dokumentace a prezentace.

Fáze zhodnocení výsledku zkoumání. V této fázi hodnotíme nejen zjištěné skutečnosti, ale také průběh celého zkoumání. Důsledky jsou závislé na účelu zkoumání a konkrétní situaci – například při zkoumání narušení bezpečnosti v organizaci může být důsledkem úprava infrastruktury, konfigurace, přístupových oprávnění, změna bezpečnostních politik, předepsání školení odpovědných osob, napomenutí či propuštění zaměstnance nebo dokonce předání případu úřadům či policii.

Může se však ukázat, že „něco skřípalo“ při samotném vyšetřování. Pak může být důsledkem úprava metodiky, školení osob, dovybavení určitými prostředky apod.

Stanovení metodiky je důležité jak při forenzním zkoumání, tak i u bezpečnostního auditu (metodika by měla být součástí smlouvy) nebo vyšetřování narušení bezpečnosti v organizaci. Důsledkem nedodržení tohoto pravidla bývá nepoužitelnost výsledků zkoumání, nemožnost revize/přezkoumání výsledků, neschopnost dojít ke správnému výsledku, (neúmyslné) ničení či přehlížení digitálních stop, nedotáhnutí vyšetřování do konce.

Další informace:

Pro metodiky dokonce existuje i standard: RFC 3227 (dostaneme se k němu buď tak, že označení standardu zadáme Googlu, nebo si to najdeme rovnou na webu IETF: https://datatracker.ietf.org/doc/html/rfc3227).

V dokumentu najdeme doporučení o tom, čeho si při zkoumání všímat, čemu se vyhnout, na co si dát pozor, jaké kroky by měly být provedeny, jak si počínat při dokumentování, ale taky doporučené prostředky (spíše v Linuxu nebo jiném UNIXovém systému).

Chain of custody („kontrolní řetězec“) je řetězec zainteresovaných osob nebo organizací podílejících se na dodání důkazů. Ke každému článku tohoto řetězce musí být dodány i související informace – co konkrétně subjekt prováděl, s čím, kde, kdy (časová razítka) apod., kdo měl jaká přístupová oprávnění. Mělo by být jasné, kdo konkrétně měl jaký přístup k důkazům či jiným prostředkům, a co s nimi dělal, případně co (nedokumentovatelného) mohl.

Atribuce kybernetického útoku znamená vystopování původce útoku. Pokoušíme se zjistit, jak konkrétně útok probíhal, stopujeme ho na cestě zpět přes jednotlivá zařízení, přes která se šířil v síti, a snažíme se dostat až k původci. Úspěch atribuce závisí na tom, jak moc důkladně hacker za sebou „zametal“.

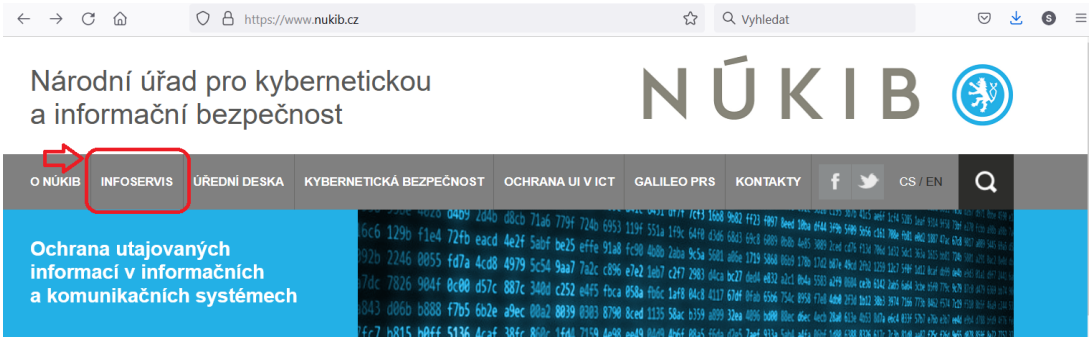
1.4 Bezpečnostní týmy a národní podpora bezpečnosti

Každá země (nebo skoro každá) má nějaký úřad nebo státní organizaci zaštiťující bezpečnost a speciálně kyberbezpečnost, obvykle je těchto institucí celá řada s rozdílnými pravomocemi a určením. U nás máme samozřejmě bezpečnostní služby (asi každý zná BIS), ale to není zdaleka všechno. Na BIS se totiž asi těžko budeme přímo obracet, pokud budeme potřebovat pomoci se zvládnutím kybernetického útoku.

Národní úřad pro kybernetickou a informační bezpečnost (NÚKIB) je zastřešující institucí pro kybernetickou bezpečnost v České republice. Má na starosti také například bezpečnost v projektu Galileo (to je evropská obdoba GPS) a sídlí v Brně (má taky pobočku v Praze).

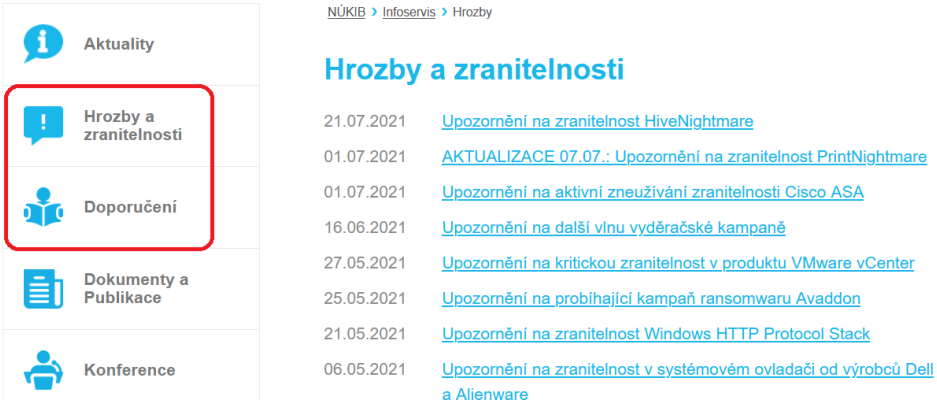
Jeho výkonná sekce (tj. ti, kdo se aktivně podílejí na řešení velkých bezpečnostních incidentů, provádějí výzkum, vývoj, pořádají kyberbezpečnostní cvičení a školeni, atd.) se nazývá *Národní centrum kybernetické bezpečnosti* (NCKB). NCKB byl původně samostatný, později se stal součástí NÚKIB. Pod NCKB patří také vládní bezpečnostní tým, jak zjistíme v následujících odstavcích.

Proč je pro nás NÚKIB důležitý? Kromě jiného na webu tohoto úřadu máme dostupné různé zajímavé informace. Na obrázku 1.1 je naznačeno, kde se dostaneme do menu pro získání různých informací. Na získané webové stránce vlevo najdeme v menu možnost zobrazit si bezpečnostní



Obrázek 1.1: Web NÚKIB

varování, doporučení, aktuality apod., viz obrázek 1.2.



Obrázek 1.2: Web NÚKIB, část pro infoservis

Pokud bychom chtěli přejít na sekci NCKB, jde o odkaz Kybernetická bezpečnost v horním menu (obrázek 1.1). V tomto menu se přes *Kontakty* také dostaneme k odkazu pro hlášení bezpečnostních incidentů. Především organizace podléhající Zákonu o kybernetické bezpečnosti zajímá taky *Úřední deska*, na které jsou například vyhlášky, varování apod.

Úkol

Projděte si web NÚKIB (https://www.nukib.cz/), ověřte si varování před nejnovějšími bezpečnostními incidenty, podívejte se na nejnovější aktuality a doporučení. Podívejte se, co je momentálně na webu NCKB (v menu nahoře *Kybernetická bezpečnost*, resp. https://www.nukib.cz/cs/kyberneticka-bezpecnost/).

Svůj bezpečnostní tým zasahující (nejen) proti kyberkriminalitě by dnes měla mít každá větší organizace. Setkáváme se zde se dvěma důležitými zkratkami:

- CERT (Computer Emergency Response Team),
- CSIRT (Computer Security Response Team).

V obou případech jde víceméně o totéž – tým odborníků určený k boji proti kyberkriminalitě, tedy lidé aktivně se zabývající bezpečností v oblasti ICT. Obě zkratky se používají pro označení bezpečnostních týmů jak na národní úrovni, tak i na úrovni soukromých společností. Ještě něco

mají tyto týmy společného – spolupráci. V této oblasti je spolupráce mimořádně důležitá, není zde místo pro nějakou konkurenční nenávist. CERT je ve skutečnosti chráněnou značkou ve vlastnictví Carnegie-Mellon University, proto se v praxi setkáme spíše se zkratkou CSIRT nebo nějakou obdobou zkratky CERT.

CSIRT.CZ je CSIRT týmem provozovaným sdružením CZ.NIC (to je správce domény .CZ – české národní domény). Jedná se o hlavní CSIRT tým České republiky, a to od roku 2012, kdy bylo uzavřeno memorandum mezi CZ.NIC a Národním centrem kybernetické bezpečnosti (resp. Ministerstvem vnitra ČR), které má ze zákona povinnost zajistit provoz bezpečnostního týmu bojujícího proti kyberkriminalitě. Nicméně tento tým ve skutečnosti existuje už od roku 2007 (v této zemi se totiž na státní úrovni o bezpečnostních týmech dlouho jen jednalo, a teprve potom, co vznikly iniciativou v soukromém sektoru, byla jejich činnost posvěcena zákonem).

Tým CSIRT.CZ plní tyto úkoly:

- řešení bezpečnostních incidentů ve spolupráci s českými provozovateli sítí a zahraničními partnery,
- pomoc provozovatelům sítí při zřizování vlastních CSIRT týmů,
- poskytuje reaktivní i proaktivní služby, poradenství,
- provozuje honeypoty („vějičky“ pro útočníky – zařízení záměrně vypadající jako slabě zabezpečená),
- mohou zjišťovat možné autory DoS útoků a předávat informace orgánům, které pak na toto oznámení reagují, nebo informovat provozovatele sítě o napadení této sítě.

Na mezinárodní bezpečnostní scéně tým spolupracuje především s organizacemi TERENA (Trans-Europen Research and Education Networking Association) a FIRST (Forum for Incident Response and Security Team).

Proaktivními službami se rozumí služby v oblasti bezpečnostní prevence, například informační pomoc, nabídky školení, provoz IDS/IPS (systémů detekujících možnost nebo pravděpodobnost napadení) apod. *Reaktivní služby* naopak reagují na již proběhlý nebo probíhající bezpečnostní incident, v tomto případě například přijímání oznámení a podnětů o incidentech a reakce na ně.

K novějším službám týmu patří služba bezplatného penetračního testování webu (tzv. Skener webu), a dále taktéž bezplatná služba zátěžových testů odpovídajících svou intenzitou DDoS útokům, které na několik dnů roku 2013 silně zpomalily české sítě.

Úkol

Projděte si web našeho národního bezpečnostního týmu: https://www.csirt.cz. V menu pod logem je zajímavá například položka *Kyberbezpečnost*, podívejte se také na *Projekty*.

Všimněte si, že tam jsou i elektronické knihy ke stažení (v horním nenápadném menu položka *Edice*). Z pohledu získávání informací je zajímavá také položka *Akademie*

GovCERT.cz je vládní bezpečnostní tým, který dohlíží na síť orgánů vlády, veřejné správy a samosprávy, včetně části kritické infrastruktury provozované státem (kdežto národní tým má na starosti komerční a akademickou sféru – trochu zmatek). GovCERT je součástí NCKB.

CESNET-CERTS je bezpečnostní tým společnosti CESNET, provozovatele sítě propojující akademické a výzkumné organizace (především vysoké školy a ústavy Akademie věd). Pracuje už od

roku 2004 a nese odpovědnost za bezpečnost především v sítích nějak souvisejících s akademickou a vědeckou sférou (včetně domény eduroam.cz). Povinnosti a pravomoci jsou vpodstatě podobné jako u CSIRT.CZ, jen vztažené na sítě, kde má tým pravomoc.

Poznámka:

CSIRT týmy (ani celostátní) nemají žádné pravomoce k řešení bezpečnostních incidentů na jiné než technické úrovni, nejde o represivní orgány. Mohou pouze monitorovat a radit, pravomoci k razantnějším reakcím mají pouze ve své vlastní síti.

Další: Jak bylo výše napsáno, svůj bezpečnostní tým by měly mít všechny větší organizace. Vlastní CSIRT týmy působí například v sítích ISP (poskytovatelů Internetu), bank, poskytovatelů energií, provozovatelů energetických společností (jako je u nás například ČEZ), velkých automobilek, velkých poskytovatelů internetových služeb (Google, Seznam, Facebook), atd.

Bezpečnostní týmy velkých antivirových firem a dalších organizací působících v oboru kybernetické bezpečnosti jsou také důležitým zdrojem zajímavých informací jak o momentální situaci, tak i o trendech, statistikách apod.

Každý takový tým se určitým způsobem prezentuje – musí mít někde uvedeny kontaktní informace, vymezení pole působnosti a odpovědnosti, seznam služeb, které poskytuje (především v rámci sítě svého provozovatele). Především v případě národních týmů musí být zřejmé, kdo se na tým může obrátit, s jakým problémem k řešení a jakým způsobem (může to být třeba formulář na webu). Pro tyto účely obvykle tým pořádá i různá školení (to je vpodstatě jedno z proaktivních opatření).

Další informace:

- GovCERT.cz: https://www.nukib.cz/cs/kyberneticka-bezpecnost/vladni-cert/
- https://www.csirt.cz/
- http://www.cesnet.cz/sluzby/reseni-bezpecnostnich-incidentu/, https://csirt.cesnet.cz/
- https://www.cisco.com/c/en/us/products/security/cybersecurity-reports.html
- https://research.checkpoint.com/category/threat-research/, https://pages.checkpoint.com/cyber-security-report-2021.html
- https://www.fireeye.com/current-threats/annual-threat-report.html
- https://www.mcafee.com/enterprise/en-us/threat-center/mcafee-labs/reports.html
- https://www.sophos.com/en-us/labs/security-threat-report.aspx

Úkol

V seznamu odkazů nad tímto úkolem jsou taky odkazy na weby různých organizací, které se zabývají kybernetickou bezpečností. Prozkoumejte jejich obsah.

1.5 Legislativa, standardy, normy

V každé zemi existuje legislativa – souhrn zákonů (a dalších dokumentů, včetně prováděcích předpisů) určujících, co je trestné a co je přestupkem, a jak se má postupovat v případě, že určitý trestný čin nebo přestupek byl spáchán. Kromě toho v každém státě máme určitou strukturu entit, která zajišťuje kontrolu, provádění a případné restrikce, pokud jsou zákony porušeny. Navíc existuje mezinárodní právo a mezinárodní smlouvy.

Občané by tyto zákony měli dodržovat (a samozřejmě také firmy, organizace, úřady apod.). Je důležité, aby ti všichni měli určité *právní povědomí*, tj. znalost zákonů alespoň do té míry, aby věděli, co je zákonem zakázáno. Víceméně každý ví, že nemá vraždit, vykrádat kapsy, pomalovávat bez dovolení cizí domy, házet zápalné lahve na cizí pozemek apod., ale kolik lidí ví, že by neměli sdílet soubory, jichž nejsou vlastníkem a které jejich licence nedovoluje sdílet?

Vedle právního povědomí je u občanů důležitá i jiná znalost – *zneužitelnost určitého jednání*, které není původně jako škodlivé míněno (například účetní, která na Dropbox nahraje nešifrovaný soubor s údaji o výplatách zaměstnanců, aby si zjednodušila přenos mezi různými počítači, porušuje zákon, třebaže ne úmyslně). A pozor, nedbalost může být taky trestná.

Nebudeme zde procházet celý právní systém České republiky, jen se stručně podíváme na několik souvisejících zákonů.

V *Trestním zákoníku* 40/2009 Sb. (viz [15]) účinném od 1. 1. 2010 najdeme hned tři paragrafy vztahující se se kyberkriminalitě:

- § 230 Sb. – Neoprávněný přístup k počítačovému systému a nosiči informací,
- § 231 Sb. – Opatření a přechovávání přístupového zařízení a hesla k počítačovému systému a jiných takových dat,
- § 232 Sb. – Poškození záznamu v počítačovém systému a na nosiči informací a zásah do vybavení počítače z nedbalosti.

Mnoho lidí si neuvědomuje, že v kyberprostoru mohou porušit i *Autorský zákon* (dostupný například na [1], výborně vysvětleno v knize [3]).

Studenti by se měli o svých právech a povinnostech vyplývajících z Autorského zákona informovat minimálně před tím, než začnou pracovat na své závěrečné (bakalářské, diplomové...) práci, a taky by měli vědět, že v Autorském zákoně se striktně rozlišuje použití pro vzdělávací a komerční účely. Pro vzdělávací účely je možné používat obsah vytvořený třetí osobou i bez jejího souhlasu (ovšem je nutné uvést zdroj), ale pro komerční účely je již souhlas nezbytný.

Pro ochranu osobních údajů máme v současné době dvě normy: *Zákon o zpracování osobních údajů* dostupný například na [17] a direktivu Evropské unie *GDPR* (General Data Protection Regulation, Obecné nařízení na ochranu osobních údajů). Zákon o zpracování osobních údajů nahradil v roce 2019 starší normu Zákon o ochraně osobních údajů. Direktiva GDPR začala platit v roce 2018 a novelizací se jeho obsah víceméně promítl do Zákona o ochraně/zpracování osobních údajů a navíc oproti GDPR obsahuje také „místně specifické“ prvky (jako například konkrétní způsoby řešení ochrany evidovaných údajů a podrobnosti o fungování Úřadu na ochranu osobních údajů). V těchto normách se stanovuje, co je osobní a citlivý údaj, jak a komu je povoleno s těmito údaji disponovat, jaké jsou povinnosti správce a zpracovatele osobních a citlivých údajů; pro nás je to důležité, protože tyto údaje mohou být poskytovány, evidovány a transportovány i elektronicky.

Uvedení GDPR do praxe bylo hodně náročné jak ve veřejné, tak i v komerční sféře, protože oproti původnímu znění Zákona o ochraně osobních údajů se způsob zacházení s osobními daty hodně změnil. Situace také zneužily některé firmy, které si z poradenských služeb na toto téma udělaly velmi výhodný byznys.

Za osobní údaje jsou považovány například jméno, pohlaví, věk, datum narození, zdravotní stav, filozofické či náboženské vyznání, biometrické údaje, fotografie, e-mailová adresa, telefonní číslo a IP adresa, ale také například cookies. Obecně jde o údaje, které jsou pro danou osobu specifické a mohou být použity pro její identifikaci, případně jejich zveřejnění by mohlo být pro danou osobu nepříjemné či dokonce poškozující. Speciální podmnožinou osobních dat jsou citlivá osobní data, jejichž zpracování podléhá přísnějšímu režimu – jde například o zdravotní údaje, biometrické a genetické údaje, ale také veškeré osobní údaje dětí.

Úkol

O GDPR se dočteme na webu Ministerstva vnitra (https://www.mvcr.cz/gdpr/clanek/co-je-gdpr.aspx), nicméně velmi přehledně a pěkně pohromadě jsou informace na webu https://www.gdpr.cz/, zvláště pokud v menu použijeme položku *GDPR?*. Projděte si tento web, zjistěte například, za jakých okolností musí správce osobních údajů (tj. ten, kdo dotyčná data drží a může za určitých okolností používat) určitá data vymazat.

Dále se zaměříme na nejnovější *Zákon o kybernetické bezpečnosti* 181/2014 Sb.[16]. Tento zákon se vztahuje na tyto subjekty:

a) poskytovatel služby elektronických komunikací a subjekt zajišťující síť elektronických komunikací, kromě subjektů uvedených pod písmenem b),
b) orgán nebo osoba zajišťující významnou síť, pokud nejsou správcem komunikačního systému podle písmene d),
c) správce informačního systému kritické informační infrastruktury,
d) správce komunikačního systému kritické informační infrastruktury,
e) správce významného informačního systému.

Jak vidíme, Zákon o kybernetické bezpečnosti se nevztahuje přímo navšechny občany a organizace, ale vztahuje se pouze na organizace a správce kybernetických systémů nějakým způsobem důležitých pro bezpečnost státu – například provozovatele databází obsahujících data o občanech, velké poskytovatele internetu a provozovatele telekomunikačních služeb, provozovatele systémů, jejichž narušení by mohlo znamenat omezení nebo ohrožení státu, provozovatele systémů pro řízení dopravy, provozovatele energetických sítí a velkých energetických systémů apod.

Subjektům, na které se zákon vztahuje, se ukládá povinnost provedení technických a organizačních opatření. K předepsaným *technickým opatřením* patří zajištění fyzické bezpečnosti (například fyzické zabezpečení serverů a aktivních síťových prvků), používání nástrojů pro ověřování identity uživatelů, řízení přístupových oprávnění, ochrana před škodlivým kódem, používání kryptografických prostředků, používání nástrojů pro proaktivní a reaktivní zabezpečení infrastruktury, logování, vyhodnocení a sběru informací o kybernetických bezpečnostních událostech, zajištění bezpečnosti průmyslových a řídicích systémů, atd. *Organizační opatření* znamenají stanovení vhodných bezpečnostních politik (zásad), řízení rizik, řízení přístupu, provádění auditu, apod. Subjekty

jsou povinny hlásit kybernetické bezpečnostní incidenty provozovateli národního CERT týmu nebo Národnímu bezpečnostnímu úřadu, a to předem daným způsobem.

Velkou změnou k dosavadní praxi je i výše zmíněná povinnost hlásit bezpečnostní incidenty. Ve skutečnosti totiž není v zájmu organizace (jakékoliv), aby se o bezpečnostním incidentu kdokoliv dozvěděl (zejména pokud jde o zákazníky, akcionáře či strategické partnery).

V každém případě je Zákon o kybernetické bezpečnosti velmi důležitým dokumentem a obdobu takového zákona najdeme v legislativě většiny vyspělých zemí.

Další související zákony a dokumenty jsou:

- *Úmluva o počítačové kriminalitě* (The EU Cybersecurity Act, Budapešť, 2001, vstoupila v platnost r. 2004) Rady Evropy byla až v roce 2013 předložena k ratifikaci Senátu ČR[1], stát se ratifikací zavazuje k zajištění souladu svých zákonů s touto úmluvou,
- *Zákon o elektronických komunikacích*[2] 127/2005 Sb. částečně zasahuje i do oblasti počítačových sítí,
- *Zákon o ochraně utajovaných informací a o bezpečnostní způsobilosti*[3] 412/2005 Sb. – důležitý zejména pro státní správu, ale pozor – součástí je i definice „Újmy zájmu České republiky a nevýhodnosti pro zájmy České republiky“, což se za určitých okolností může týkat i soukromých firem.

Jak bylo výše řečeno, problémem kyberprostoru je jeho *globalita*. V kyberprostoru neexistují žádné hranice a ani není možné vymezit menší oblasti, které by se lépe kontrolovaly. Naproti tomu legislativa vždy nějaké hranice má, je platná pouze na území daného státu. Proto je v oblasti kyberkriminality velmi důležitá mezinárodní spolupráce.

Situaci komplikuje to, že co je v jedné zemi trestné, může být v jiné zemi posuzováno mnohem mírněji, nebo v dané zemi stíhán čin, jehož důsledky se projevují pouze v jiné zemi. Vyšetřovatelé určité země mají omezený přístup k dokumentům a informacím pocházejícím z jiné země, třebaže jsou tyto informace pro vyšetřování nezbytné. Taktéž není zcela samozřejmé, že podezřelý bude z jedné země vydán k trestnímu stíhání probíhajícímu v jiné zemi. I v případě, že se podaří mezinárodní spolupráci při vyšetřování přece jen realizovat (existuje Interpol, Europol, s mnoha zeměmi máme navázánu spolupráci například v boji proti organizovanému zločinu), jedna komplikace zůstává – vysoká časová náročnost takové komunikace.

ENISA (the European Union Agency for Cybersecurity – neřešte, že zkratka neodpovídá) je agentura v EU, jejímž úkolem je podpora co nejvyšší úrovně kybernetické bezpečnosti ve státech Evropské unie Poskytuje podporu jednotlivým zemím například při digitalizaci a tvorbě bezpečnostní infrastruktury, řešení kybernetických útoků a jejich prevenci, sdílení bezpečnostních informací (a obecně spolupráci v této oblasti), poradenství, podpoře vzdělávání, spolupráci na certifikacích, atd. Propojuje také národní CSIRT týmy, o kterých byla řeč dříve.

ENISA stojí také za výše zmíněným The EU Cybersecurity Act, pracuje na certifikačním rámci pro ICT produkty a služby.

[1] Na http://www.senat.cz/xqw/webdav/pssenat/original/66810/56264 je český překlad Úmluvy včetně předkládací zprávy pro Parlament ČR.

[2] http://www.zakonyprolidi.cz/cs/2005-127

[3] http://www.zakonyprolidi.cz/cs/2005-412

Úkol

Na https://www.enisa.europa.eu/topics najdeme seznam témat, kterými se ENISA zabývá. Projděte si tento seznam (je dostupný také v horním menu) a zjistěte, jestli se tam nachází i něco velmi „aktuálního“. V horním menu si také najděte položku *Publications* – najdete tam bezpečnostní reporty, bezpečnostní upozornění, stanoviska (opinion papers) a další.

Bezpečnost v organizaci se neřídí pouze legislativou konkrétní země, je praktické dodržovat i určité technické normy a standardy.

Pro větší firmy je důležitý standard ISO/IEC 27001, který popisuje *požadavky na management bezpečnosti informací*. Tento standard byl v ČR přejat jako norma ČSN ISO/IEC 27001:2014. Je zde specifikován *Information Security Management System* (ISMS) analyzující bezpečnostní rizika související se správou informací – stanovuje rámec, ve kterém organizace identifikuje, analyzuje a pojmenovává bezpečnostní rizika vznikající při zpracování, přenosu a evidování informací.

Standard neřeší řízení těchto rizik (tedy stanovení následných kroků), to najdeme až ve standardu ISO/IEC 27002 (u „jedničky“ jsou pouze naznačeny v příloze), naopak před prostudováním ISO/IEC 27001 je dobré si pročíst standard ISO/IEC 27000, kde najdeme formální definice pojmů v „jedničce“ používaných. Celá skupina těchto vzájemně provázaných standardů (a dalších souvisejících) se označuje jako ISO27k (neformálně, znamená ISO 27 tisíc).

Poznámka:

GDPR a ISO27k mají mnoho společného, třebaže jejich vlastní účel je trochu jiný. V obou případech jde o ochranu dat a mnohé firmy zjistily, že pokud prošly certifikací ISO27k, nemají s iplementací GDPR moc práce.

Další informace:

- Text standardu ISO/IEC 27001:2013 (tj. z roku 2013) v originálu je částečně dostupný na http://www.iso.org/iso/home/store/catalogue/_ics/catalogue_detail/_ics.htm?csnumber=54534
- Stručný popis obsahu standardu ISO/IEC 27001: http://www.iso27001security.com/html/27001.html
- Informace o ČSN ISO/IEC 27001:2014 na http://www.cqs.cz/Normy/CSN-ISO-IEC-270012014-Bezpecnosti-informaci.html

Certifikace systému v oblasti bezpečnosti je proces, po jehož absolvování je daný systém prověřen vzhledem k určitému stupni bezpečnosti. Organizace (případně úřad) pořizující takový systém má jistotu, že systém splňuje určitý bezpečnostní standard. Problémem těchto certifikací je jejich vysoká cena a časová náročnost certifikačního procesu (běžně půl roku i několik let). K nejznámějším certifikacím patří *TCSEC* (Trusted Computer Evaluation Criteria, také Oranžová kniha), což je americký certifikační systém (původně byl určen pro systémy používané ve státní správě, ale pronikl i do soukromého sektoru). Známá je také kanadská obdoba – *CTCPEC* (Canadian Trusted Computer Product Evaluation Criteria), evropská *ITSEC* (Information Technology Security

Evaluation Criteria) a také mezinárodní pokus o sjednocení všech těchto kritérií *CC* (Common Criteria). Podrobnější informace najde čtenář ve zdroji [6].

Další standardy v oblasti bezpečnosti informací a popisy typických bezpečnostních praktik jsou například BS 7799-2 a BS 7799-1, ISO/IEC TR 13335, ISO/IEC 17799, ISO/IEC 15408 a další.

Souhrn kapitoly

V této kapitole jsme se zaměřili na základní pojmy a postupy z oblasti digitální forenzní analýzy a podívali jsme se na organizace, týmy, standardy a normy důležité pro oblast kybernetické bezpečnosti.

Literatura

[1] Autorský zákon: Zákon č. 121/2000 Sb [online]. *Zákony pro lidi.cz*. 2000 [cit. 2015-06-15]. Dostupné z: http://www.zakonyprolidi.cz/cs/2000-121

[2] Baldwin, Hayden B., M/Sgt. Crime Scene Processing Protocol [online]. *Forensic Enterprises, Inc.* [cit. 2018-03-23]. Dostupné z: http://www.feinc.net/cs-proc.htm

[3] Bartoš, Aleš. *Autorské právo v otázkách a odpovědích*. Praha: Pierot, 2012. ISBN 978-80-7353-223-9.

[4] Carrier, Brian a Eugene H. Spafford. Getting Physical with the Digital Investigation Process. In: *CERIAS Tech Report 2003-29* [online]. 2003 [cit. 2015-06-18]. Dostupné z: https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/2003-29.pdf

[5] Cichonski, Paul (NIST), Thomas Millar (DHS), Tim Grance (NIST), Karen Scarfone (Scarfone Cybersecurity). *Computer Security Incident Handling Guide* [online]. National Institute of Standards and Technology (NIST) [cit. 2021-07-22] US Department of Commers and NIST: 2012. Dostupné z: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

[6] Doseděl, Tomáš. *Počítačová bezpečnost a ochrana dat*. Vyd. 1. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.

[7] Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector. *SecretService.gov* [online]. 2004 [cit. 2015-06-17]. Dostupné z: http://www.secretservice.gov/ntac/its_report_040820.pdf

[8] Jirovský, Václav. *Kybernetická kriminalita: nejen o hackingu, crackingu, virech a trojských koních bez tajemství*. 1. vyd. Praha: Grada, 2007, 284 s. ISBN 978-80-247-1561-2. Některé stránky dostupné na: https://books.google.cz/books/about/Kybernetick%C3%A1_kriminalita.html?id=0TxDiektLJQC

[9] *ManagementMania Knowledge* [online]. Sekce Informatika a řízení IT, Řízení bezpečnosti. [cit. 2015-06-15]. Dostupné z: https://managementmania.com/cs/

[10] Mozayani, Ashraf a Carla Noziglia. *The forensic laboratory handbook: procedures and practice*. 2nd ed. New York, N.Y.: Humana Press, 2011, xii, 600 p. ISBN 16-076-1872-9. Některé stránky dostupné na: https://books.google.cz/books?id=Kqhj3ugmYhQC&pg=PA225&lpg=PA225&dq=Digital+Evidence:+Standards+and+Principles+Forensic+Science+Communications&source=bl

[11] Net Losses: Estimating the Global Cost of Cybercrime [online]. *Center for Strategic and International Studies, McAfee.com*. 2014 [cit. 2015-06-19]. Dostupné z: http://www.mcafee.com/mx/resources/reports/rp-economic-impact-cybercrime2.pdf

[12] Palmer, Gary (ed.). *A Road Map for Digital Forensic Research: Report From the First Digital Forensic Research Workshop (DFRWS)* [online]. Utica, New York, 2001 [cit. 2015-06-15]. Dostupné z: http://www.dfrws.org/2001/dfrws-rm-final.pdf

[13] Pearce, Craig. Computing Forensics: A Live Analysis [online] *Linux.Conf.Au Security Miniconf*. 2005 [cit. 2015-06-18]. Dostupné z: https://www.linux.org.au/conf/2005/security_miniconf/presentations/crpearce-lca2005.pdf

[14] Ponnemon, Larry. The Cost of a Lost Laptop [online]. *Intel.com*. 2009 [cit. 2015-06-19]. Dostupné z: http://www.intel.com/content/dam/doc/white-paper/enterprise-security-the-cost-of-a-lost-laptop-paper.pdf

[15] Trestní zákoník: Zákon č. 40/2009 Sb [online]. *Zákony pro lidi.cz*. 2009 [cit. 2015-06-15]. Dostupné z: http://www.zakonyprolidi.cz/cs/2009-40

[16] Zákon o kybernetické bezpečnosti: Zákon č. 181/2014 Sb [online]. *Zákony pro lidi.cz*. 2014 [cit. 2015-06-15]. Dostupné z: http://www.zakonyprolidi.cz/cs/2014-181

[17] Zákon o zpracování osobních údajů: Zákon č. 110/2019 Sb [online]. *Zákony pro lidi.cz*. 2000 [cit. 2021-09-12]. Dostupné z: https://www.zakonyprolidi.cz/cs/2019-110/

[18] Zákon o znalcích a tlumočnících: Zákon č. 444/2011 Sb. [online]. *Unie soudních znalců* [cit. 2015-06-16]. 2011. Dostupné z: http://uniesoudnichznalcu.cz/public/media/files/zakon-444-2011-sb-o-znalcich-a-tlumocnicich.pdf

Kapitola 2

Základní průzkum systému

Rychlý náhled: V této kapitole navážeme na předchozí kapitolu a zaměříme se na prostředky k průzkumu systému, zejména co se týže průzkumu paměťových médií. Vyzkoušíme si různé nástroje ve Windows a Linuxu (či jiných UNIXových systémech) a dále specializované operační systémy vybavené potřebnými nástroji už od „výrobce“. Na konci kapitoly nás čeká odlehčení ve formě exkurze do světa velkých dat (big data) a jejich zpracovatelů.

Klíčová slova: Bitová kopie disku a oddílu, dead analýza disku, write blocker, ISO obraz, Správce zařízení, skryté zařízení, Správce disků, `mountvol`, `diskpart`, atribut read-only, režim offline, `fsutil`, DuBaron Disk Image, ImDisk Virtual Disk Driver, Recuva, TestDisk, PhotoRec, souborový systém, speciální soubor, `/etc/mtab`, `lsblk`, `mount`, `dd`, `dcfldd`, `ddrescue`, Caine Linux, Kali Linux, Autopsy, NetCat, integrita, `Get-FileHash`, `shasum`, skrytá oblast disku, HPA, DCO, `hdparm`, RM bit, datový sklad, big data, data mining, GFS, MapReduce, Hadoop.

Cíle studia: Cílem této kapitoly je naučit se bezpečným způsobem vytáhnout maximum dat především z paměťových médií. Účelem je orientovat se v potřebných nástrojích pro různé platformy, včetně celých operačních systémů a specializovaných nástrojů přímo určených pro tyto účely. Dalším cílem je rozšířit své povědomí o různých úložištích o úložiště určená pro big data.

2.1 Příprava a možnosti

V předchozí kapitole jsme probrali několik metodik digitální forenzní analýzy, a tedy víme, že (ať už budou námi získané výsledky použity pro jakýkoliv účel) je třeba nejdřív provést sběr dat takovým způsobem, aby tato data nebyla kompromitována, a teprve potom můžeme analyzovat.

2.1.1 Nastínění základního postupu

Předpokládejme, že chceme analyzovat data v konkrétním systému. Co vše nás bude zajímat?

- soubory na pevném disku, konkrétní zaměření záleží na tom, co hledáme (systémové logy a logy aplikací, spustitelné soubory, konfigurační soubory, odcizené fotografie či videa, atd.),
- historie, cookies a další informace pro jednotlivé webové prohlížeče,

- data z komunikačních programů (e-maily apod.), kontakty,
- obsah operační paměti (obecně pamětí, jejichž obsah nebude po vypnutí systému přístupný),
- hesla, bezpečnostní klíče, certifikáty apod.,
- seznam instalovaného softwaru, případně jeho legálnost,
- nastavení a zabezpečení systému, nastavení sítě, síťová aktivita, historie síťové aktivity a obecně síťové komunikace,
- seznam periferních zařízení, která byla v poslední době k systému připojována (například USB flash disků),
- soubory či oddíly se zálohami, případně skryté oddíly, smazané soubory,
- škodlivý software, zranitelnosti, bezpečnostní problémy, signály možného napadení systému,
- atd.

V jiném typu systému než je běžný desktopový počítač nás zase budou zajímat trochu jiné informace. Co z toho plyne? Prvním krokem je ujasnit si, co nás vlastně zajímá. Samozřejmě můžeme později svůj seznam rozšiřovat, také můžeme na něco důležitého narazit „náhodou“, ale základní představu bychom si měli na začátku vytvořit.

Záleží také na tom, jakým způsobem má být proveden sběr dat a případně i analýza – za běhu systému (live) nebo po jeho vypnutí (dead). Druhá možnost je samozřejmě jednodušší (z pohledu možné kompromitace systému) a bezpečnější, ale v některých případech je třeba zvolit live variantu (jak bylo diskutováno v předchozí kapitole).

Zde se budeme zabývat převážně paměťovými médii – pevnými disky, SSD, optickými disky (CD, DVD, BluRay), USB flash disky apod. Z těchto médií lze vytvořit jejich obraz – bitovou kopii.

Bitová kopie (obraz, image) paměťového média je soubor s kopií dat z původního paměťového média bit po bitu, z tohoto souboru je možné obsah tohoto paměťového média plně rekonstruovat. Obraz může být vytvořen i z oddílu pevného disku, nemusí jít o obraz celého média.

Obrazy lze ukládat v různých formátech – nejběžnější je ISO, ale můžeme se setkat i s formáty IMG, BIN (starší Windows), DMG (v MacOS), TIB (vytvořeno v Acronis True Image), MDF, CUE, CDI, DAA a dalšími. Pro práci s obrazy – jejich vytvářením, připojováním apod. – existuje mnoho různých nástrojů, každý z nich podporuje určitou podmnožinu těchto formátů (formát ISO je podporován asi nejběžněji).

Postup (Dead analýza pevného disku)

Předpokládejme, že můžeme provést dead analýzu a máme k dispozici pevný disk ze zkoumaného systému vymontovaný a dodaný důvěryhodnou osobou. Protože potřebujeme provést analýzu tak, aby naše výsledky byly použitelné pro vyvození případných důsledků (dopadnutí pachatele, zjištění insidera, optimalizaci vytížení sítě apod.), měli bychom postupovat takto:

- ze zkoumaného pevného disku vytvoříme ISO obraz,
- pro jistotu je dobré vytvořit kontrolní součet (podrobně v kapitole o šifrování),
- získaný ISO obraz si nakopírujeme (vícekrát), zálohujeme,
- pevný disk bezpečně uložíme,
- analyzujeme některou kopii a pokud dojde k problémům či kompromitaci, přejdeme k další kopii, atd.

Takže potřebujeme program, který umí podle připojeného pevného disku vytvořit jeho ISO obraz, a dále program, který nám umožní na ISO souboru provádět analýzu.

Poznámka:

Předně si musíme uvědomit, že pokud analyzované paměťové médium nechceme kompromitovat, musíme mít toto médium během pořizování obrazu připojeno „pouze pro čtení“ – použijeme hardwarový nebo softwarový *write blocker*.

To je důležité, protože jak Windows, tak i některé další systémy včetně některých linuxových distribucí hned po připojení média toto médium připojí i pro zápis (nejen USB flash disk, ale i externí disky nebo interní disky připojované přes rámeček) a hned po připojení mohou do souborového systému zapisovat. Tomuto problému se budeme věnovat později.

Sehnat kvalitní write blocker není nic jednoduchého a někdy ani levného. V českém prostředí se také můžeme setkat s názvy *blokátor zápisu* nebo *forenzní blokátor*, případně *forenzní blokátor zápisu*, ale i tak v českých e-shopech moc na výběr nemáme. Taky záleží na tom, jaké rozhraní k paměťovému zařízení vyžadujeme – buď USB (teď záleží který konkrétně), SATA, případně M.2 pro SSD, existují také forenzní blokátory speciálně na paměťové karty. Další kritérium je rozsah funkcí zařízení.

Úkol

Pokuste se na Internetu najít několik různých write blockerů. Pokud nenajdete vhodný e-shop, můžete zkusit například https://firewire-revolution.eu/produkt-kategorie/forenzni-hw/blokatory/. Na tomto webu jsou nejen hardwarové forenzní nástroje, ale také jiný související sortiment.

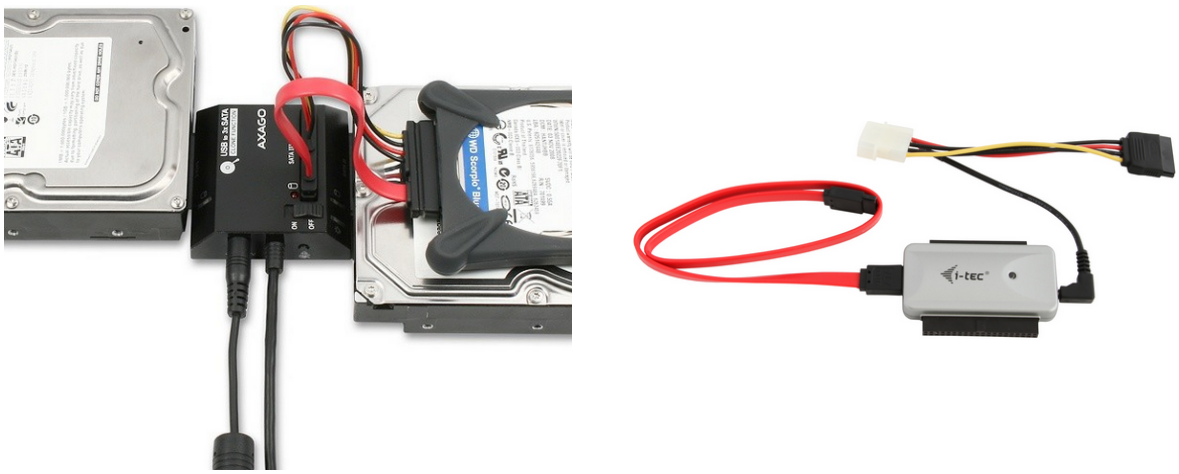
2.1.2 Jak získat ISO obraz paměťového média

Paměťová zařízení mají různá komunikační rozhraní. Pokud jde o optické médium, USB flash disk nebo běžný USB externí disk, obvykle není problém (prostě použijeme optickou mechaniku, USB port, Thunderbolt apod., podle konkrétního rozhraní). Ovšem u interních pevných disků nebo SSD už je to něco jiného.

Pokud při dead analýze potřebujeme připojit pevný disk nebo SSD k vlastnímu počítači či notebooku, můžeme samozřejmě otevřít case svého počítače a připojit zařízení k příslušnému rozhraní (SATA, IDE, PCIe, mSATA, M.2, SATA Express podle konkrétního zařízení), ale je i jiná možnost. Existují konvertory (adaptéry, rámečky, boxy) umožňující převod těchto rozhraní na rozhraní USB. V obchodech je obvykle najdeme pod názvem Externí box, SATA adaptér apod., cena těch levnějších je v řádu několika stokorun. Existují dokonce takové externí boxy, které dokážou z připojeného disku vytvořit obraz i bez připojení k počítači, včetně blokace zápisu.

Další možností je pořídit obraz disku přímo na místě (na zkoumaném zařízení), pro tento účel je lepší mít po ruce bootovací distribuci Linuxu takovou, která neprovádí automatické připojování paměťových médií při startu nebo je alespoň připojuje pouze pro čtení, případně také write blocker.

Pro pořízení ISO obrazu paměťového média nestačí jen toto médium připojit (pokud samo-



Obrázek 2.1: Externí boxy pro připojení SATA/IDE disků od výrobců AXAGO a I-tec

zřejmě nemáme specializované zařízení určené pro tento účel), ale potřebujeme také software, který pořídí bitovou kopii paměťového média nebo oddílu disku a uloží ji, třeba do souboru s příponou iso. V tomto směru je situace jednodušší v Linuxu a jiných UNIXových systémech, tam si vystačíme s „palubními prostředky“ a je to víceméně otázka jednoho či dvou příkazů. Ve Windows potřebujeme další aplikaci, a navíc těžko přinutíme systém, aby připojované zařízení nezpřístupnil pro zápis.

Když už máme hotový obraz a v dostatečném počtu kopií zálohovaný, vezmeme jednu z kopií obrazu a můžeme s ní pracovat. Ovšem k tomu potřebujeme nástroj, který dokáže připojit obraz disku jako virtuální disk, a navíc pouze pro čtení (opět musíme dát pozor, abychom data nekompromitovali). V Linuxu a UNIXových systémech s tím není problém, pro Windows potřebujeme aplikaci, která to bude umět.

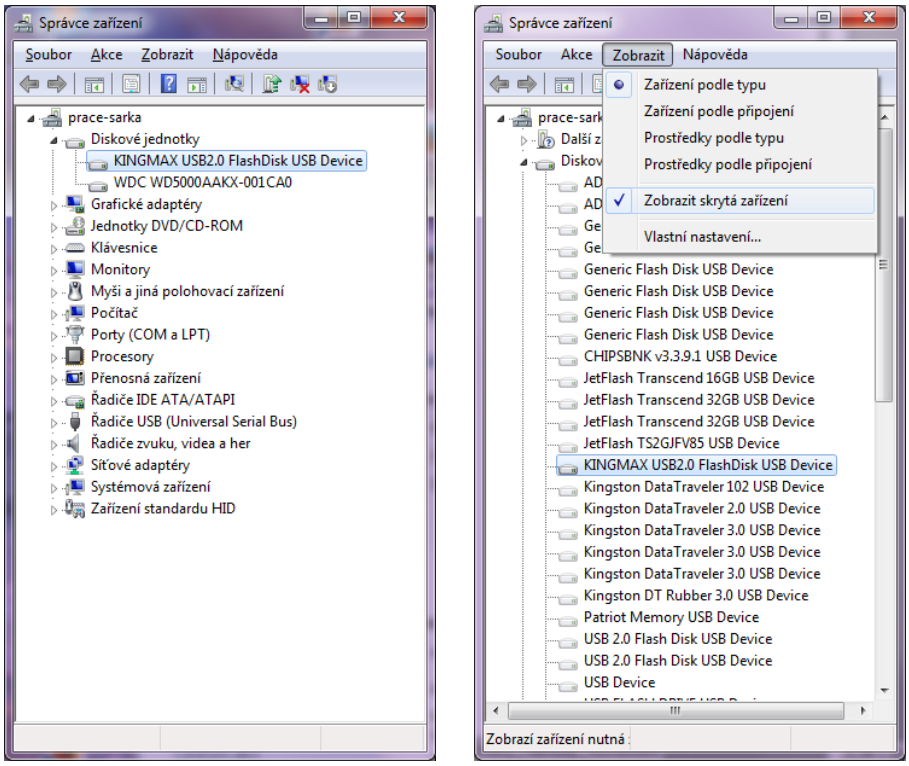
2.2 Postupy ve Windows

Sekce o Windows bude mnohem delší než následující sekce o Linuxu – Windows jsou sice uživatelsky přívětivější pro běžného uživatele (alespoň si to většina uživatelů myslí), ale pro tak speciální typy úloh jako je například analýza disku bez nebezpečí jeho kompromitace zde nejsou příslušné nástroje a některé problémy prakticky nelze řešit.

Pro většinu úloh, které nás zajímají, ve skutečnosti dokážeme sehnat nástroje, které budou pracovat i ve Windows, dokonce i takové, které jsou volně šiřitelné a mnohé i s otevřeným zdrojovým kódem (open-source programy). U několika málo úloh si vystačíme i s nástroji integrovanými ve Windows.

2.2.1 Zobrazení seznamu připojovaných zařízení

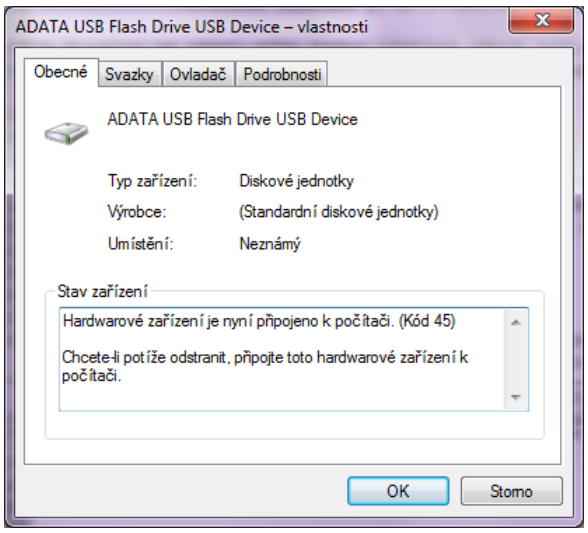
Ve Windows se někdy může hodit přehled všech zařízení, která byla v poslední době připojována nebo jsou právě připojena. Takový přehled se může hodit třeba tehdy, když máme podezření, že se k určitému počítači za nepřítomnosti vlastníka dostal někdo cizí a například si něco kopíroval na USB flash disk.



Obrázek 2.2: Správce zařízení – vpravo po zapnutí zobrazení skrytých zařízení

Nástroj *Správce zařízení* ukazuje seznam všech aktivních ovladačů zařízení, ovšem v základním nastavení jen těch zařízení, která jsou právě připojena a běží bezchybně. Ovšem my tento nástroj můžeme nakonfigurovat tak, aby ukazoval i ovladače těch zařízení, která tyto podmínky nesplňují. Nástroj je kromě jiného dostupný i v Ovládacích panelech, případně ho můžeme spustit z jeho souboru: `devmgmt.msc`.

Na obrázku 2.2 vidíme rozdíl mezi stavem, kdy jsou zobrazena pouze právě připojená zařízení (vlevo) a kdy jsou zobrazena všechna zařízení, která byla v minulosti připojena a zatím je nikdo nesmazal ze seznamu (vpravo). Tato „skrytá“ zařízení zobrazíme pomocí volby v menu, jak je na obrázku naznačeno. Poklepáním na konkrétní zařízení získáme okno s podrobnějšími informacemi.



V případě nepřipojeného zařízení pravděpodobně na první záložce tohoto okna bude poněkud zavádějící a sama sobě si odporující informace, jak vidíme na obrázku vpravo. Pro nás je důležitá hlavně poslední záložka – „Podrobnosti“, kde v seznamu vybíráme parametr, který nás zajímá, a zjišťujeme příslušné hodnoty. Dá se tam například zjistit *datum instalace* a *datum první instalace* – tak zjistíme, kdy bylo příslušné zařízení poprvé připojeno, například kdy dotyčný použil svůj USB flash disk.

Zobrazit skrytá zařízení se vyplácí i na systému, který primárně nechceme zkoumat, ale jed-

noduše jen vyčistit – stačí při vybraném „nadbytečném“ ovladači (třeba USB flash disku, o kterém víme, že už ho nebudeme připojovat) zmáčknout klávesu DEL, nebo v kontextovém menu zvolit „Odinstalovat“. Taktéž v případě, že Windows dělají problémy při připojování určitého USB flash disku (nebo jiného zařízení), můžeme jeho ovladač takto smazat a při příštím připojení se nainstaluje (již správný, opravený) ovladač.

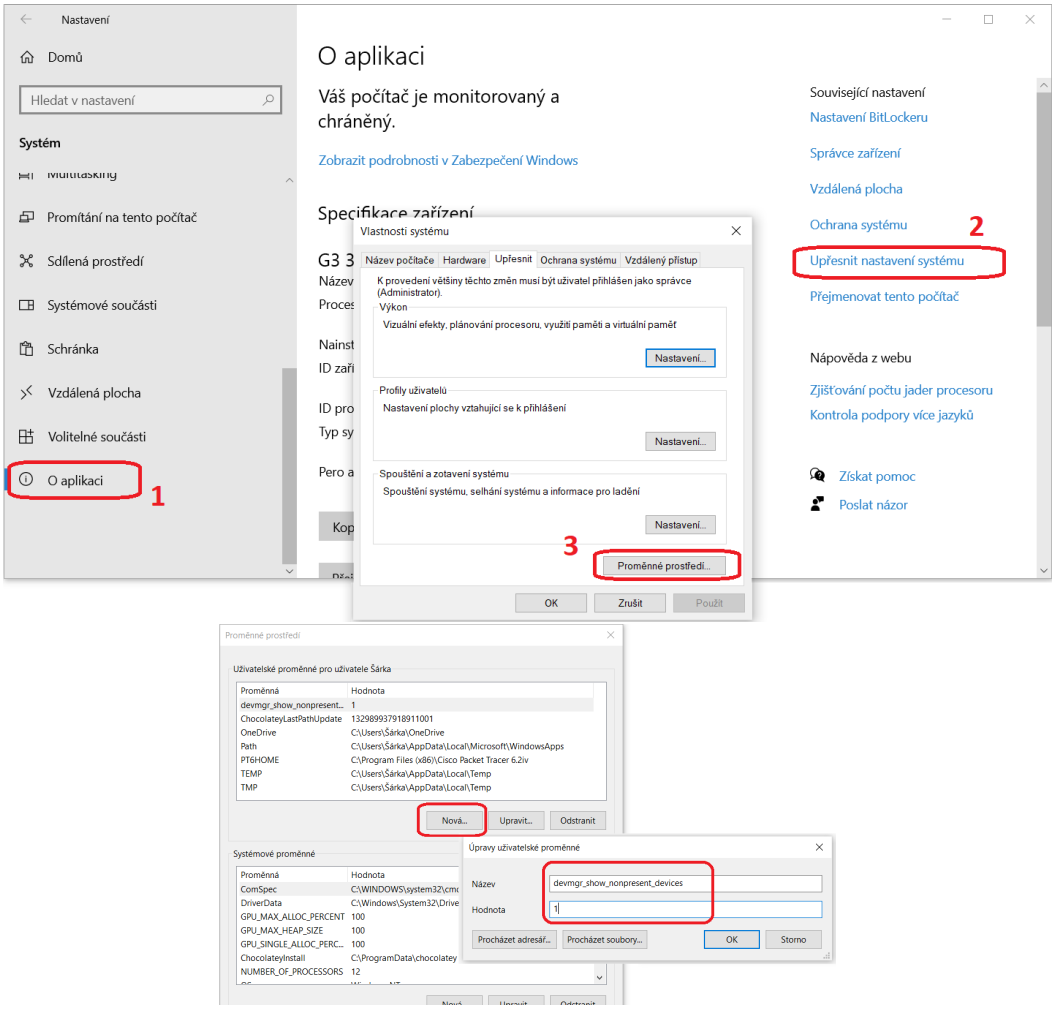
Jenže prosté zapnutí zobrazování skrytých zařízení přímo v nástroji nemusí být účinné.

Postup (Zobrazení opravdu všech skrytých zařízení)

Pokud se zařízení, které nás zajímá, v seznamu pořád nezobrazuje, třebaže jsme zapnuli zobrazování skrytých zařízení, je třeba vytvořit jednu speciální proměnnou. Provedeme to takto:

- pravým tlačítkem na ikonu *Počítač* vyvoláme kontextové menu, vybereme „Vlastnosti“,
- v panelu vlevo klepneme na „Upravit nastavení systému“,
- karta „Upřesnit“, tlačítko „Proměnné prostředí“, dole tlačítko „Nová“,
- zadáme název proměnné `devmgr_show_nonpresent_devices`, hodnotu nastavíme na číslo 1.

Postup je ukázán na obrázku 2.3. Pak by po zvolení zobrazení skrytých zařízení ve Správci zařízení už měla být zobrazena opravdu všechna zařízení včetně právě nepřipojených a poškozených.



Obrázek 2.3: Vytvoření proměnné pro zobrazení všech zařízení

Kde jinde se dá dostat k seznamu (již připojených) paměťových médií:

- *Správce disků* – dá se spustit například spuštěním souboru `diskmgmt.msc`,
- *Zařízení a tiskárny* – v Ovládacích panelech,
- mnoho dalších míst (Průzkumník, `mountvol`, `diskpart`, atd.), něco probereme dále.

Na těchto různých místech můžeme získat i další informace o připojovaných médiích a jsou dostupné další možnosti jejich konfigurace, proto se vyplatí je znát.

2.2.2 Změna způsobu připojení disku

Windows bohužel přistupují odlišně k pamětem pevným a výměnným (výměnná paměťová média jsou například USB flash disky nebo optické disky, podrobněji se tím budeme zabývat na konci kapitoly). Projevuje se to i v možnostech konfigurace jejich připojování. Pokusíme se Windows přinutit, aby připojované disky nebyly automaticky zpřístupňovány.

Postup (Zákaz automatického připojování disků)

Správně bychom měli postupovat takto:

- spustíme Příkazový řádek s oprávněními správce:
  klepneme na Start, vyhledáme `cmd`, na ně pravým tlačítkem myši, spustit jako správce
- zakážeme automatické připojování nových svazků
  `mountvol /N`
- až tuto vlastnost nebudeme potřebovat, znovu automatické připojování povolíme:
  `mountvol /E`

Moc nefunguje, Windows si výměnné disky stejně připojují dle vlastního rozmaru. Jiná možnost by byla použít nástroj diskpart (viz dále) s vnitřním příkazem `automount disable`. Jenže ten na výměnná média taky nefunguje.

Teď už je nám jasné, že nezabráníme automatickému zpřístupnění disku hned při jeho připojení. Je však možné to provést dodatečně, tedy smířit se s tím, že disk byl hned po připojení zpřístupněn, a pak teprve nastavit vlastnost „pouze pro čtení“. K tomu použijeme nástroj Diskpart, který pracuje v textovém režimu (v Příkazovém řádku).

Postup (Atribut read-only)

Pokusíme se nastavit u připojeného disku vlastnost „pouze pro čtení“.

- spustíme Příkazový řádek s oprávněními správce:
  klepneme na Start, vyhledáme `cmd`, na ně pravým tlačítkem myši, „Spustit jako správce“
- spustíme programové prostředí Diskpart
  `diskpart`
- v prostředí `diskpart` zjistíme, jaké disky jsou připojené, vybereme ten, který potřebujeme:

```
DISKPART> list disk

  Disk ###  Stav           Velikost Volné    Dyn  Gpt
  --------  -------------  -------- -------  ---  ---
  Disk 0    Online          465 GB  1024 KB
```

```
  Disk 1    Online           29 GB      0 B
  Disk 2    Online           37 GB  6144 KB

DISKPART> select disk 2

Nyní je vybrán disk 2.
```

- změníme atributy vybraného disku:
  `attributes disk set readonly`
- konec práce s Diskpartem: příkaz `exit`

Problém je, že ve skutečnosti jsme zablokovali jen změnu *vlastností* oddílů (ještě to není ono), tj. musíme nastavit readonly spíše u oddílů disku. Použijeme podobný postup, jen musíme jít hlouběji, k oddílům.

Postup (Nastavení atributu read-only pro oddíl na disku)

Tento postup již bude stručnější, začíná podobně jako předchozí:

- `diskpart`
  `list disk`
  `select disk 2`
- zobrazíme seznam oddílů na vybraném disku:
  `list volumes`
- vybereme ten oddíl, který potřebujeme:
  `select volume xxxx` (doplníme podle toho, který oddíl chceme)
- vybraný oddíl převedeme do režimu „pouze pro čtení“:
  `attr volume set readonly`
- až nebudeme tento atribut potřebovat, můžeme ho zrušit: `attr volume clear readonly`

Po provedení tohoto postupu již budou data na takto zpracovaném oddílu přístupná jen v režimu „pouze pro čtení“, tedy pokud chceme zkoumat přímo tento disk, nenapácháme neúmyslné škody.

Zkusíme již připojený disk (ať už s atributem read-only nebo bez něj) převést do režimu offline, abychom mohli vytvořit jeho ISO obraz. K tomu opět použijeme nástroj Diskpart. Kdyby disk nebyl v režimu offline, nebylo by možné vytvořit jeho obraz.

Postup (Převod disku do režimu offline)

Máme připojen disk, který chceme převést do režimu offline:

- spustíme programové prostředí Diskpart (jako v předchozím postupu)
- v prostředí `diskpart` zjistíme, jaké disky jsou připojené, vybereme ten, který potřebujeme:

```
DISKPART> list disk

  Disk ###  Stav           Velikost Volné    Dyn  Gpt
  --------  -------------  -------- -------  ---  ---
  Disk 0    Online          465 GB  1024 KB
  Disk 1    Online           29 GB      0 B
  Disk 2    Online           37 GB  6144 KB
```

```
DISKPART> select disk 2

Nyní je vybrán disk 2.
```

- tam zvolený (bohužel už připojený) disk vybereme a označíme *offline*:

```
DISKPART> offline disk

Program DiskPart úspěšně převedl vybraný disk do offline režimu.

DISKPART> list disk

  Disk ###  Stav           Velikost Volné    Dyn  Gpt
  --------  -------------  -------- -------  ---  ---
  Disk 0    Online          465 GB  1024 KB
  Disk 1    Online           29 GB      0 B
* Disk 2    Offline          37 GB  6144 KB
```

- konec práce s Diskpartem: příkaz `exit`

Na výpisu vidíme, že námi vybraný disk (vybraný je označen hvězdičkou) je ve stavu *Offline* (hodnota je v druhém sloupci tabulky).

Poznámka:

Pokud bychom se v diskpartu neorientovali, můžeme „zavolat o pomoc“:

- `help`
- `help attr`
- `help attr volume`

Disk se dá odpojit pomocí myši, to všichni víme. Podíváme se, zda lze disk odpojit v Příkazovém řádku (toho by se dalo využít třeba ve skriptu v rámci automatizace údržby). K tomuto účelu se dá použít nástroj `fsutil`.

Postup (Odpojení svazku v Příkazovém řádku)

Nástroj `fsutil` (na rozdíl od diskpartu) nepracuje v interaktivním režimu, tedy pokaždé zadáváme i příkaz, nejen argumenty. Nejdřív si ukážeme, jak zjistit informace o oddílech (abychom omylem neodpojili něco, co nechceme odpojit). Předpokládáme například, že náš svazek je připojen pod písmenem f:, a chceme zjistit další informace:

- spustíme Příkazový řádek s oprávněními správce
- `fsutil fsinfo drivetype f:`
- `fsutil fsinfo volumeinfo f:`

Teď už jsme přesvědčeni, že právě tento oddíle chceme odpojit, tedy to provedeme:

- zadáme (zde například pro oddíl připojený jako F:)
  `fsutil volume dismount f:`

Tento postup funguje pouze na místní disky, ne na výměnná média.

Nápověda k `fsutil` se získává následovně:

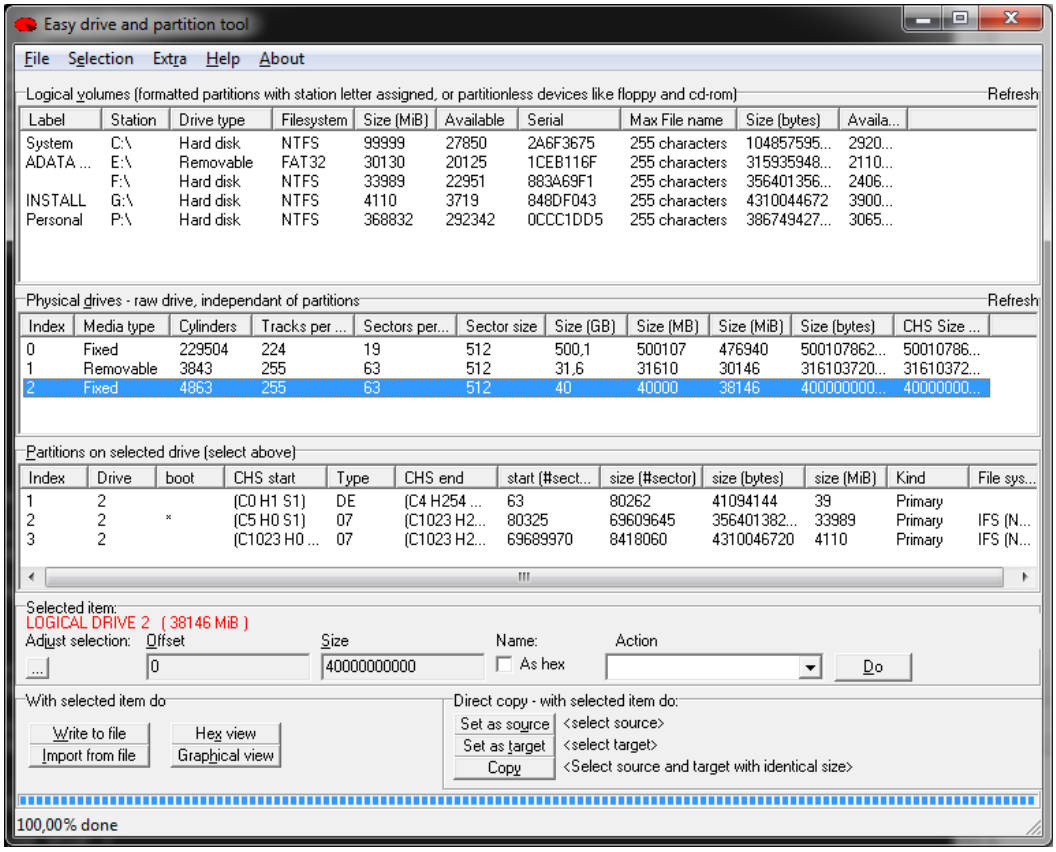
```
fsutil ?
fsutil volume ?     atd.
```

Tento postup má jeden závažný nedostatek – funguje pouze na oddíly pevného disku (místní disky), co je poněkud paradoxní. Výměnná média takto odpojit nelze.

2.2.3 Vytvoření obrazu disku

Předpokládejme, že máme připojen disk, a tento disk je v režimu offline. Pro vytvoření obrazu disku můžeme použít například nástroj *DuBaron Disk Image*. Tento nástroj je volně dostupný na adrese http://www.dubaron.com/diskimage/ a je distribuován pod open-source licencí GNU GPL.

DuBaron Disk Image umí zpřístupnit všechny fyzicky připojené disky (i offline), jejich oddíly a také vlastnosti, umí vytvořit ISO obraz takového disku nebo jeho jednotlivých oddílů, uložit v souboru s příponou .iso nebo v komprimovaném formátu. Kromě toho dokáže vytvořit kontrolní součet disku a nabízí grafický i hexadecimální náhled disku.



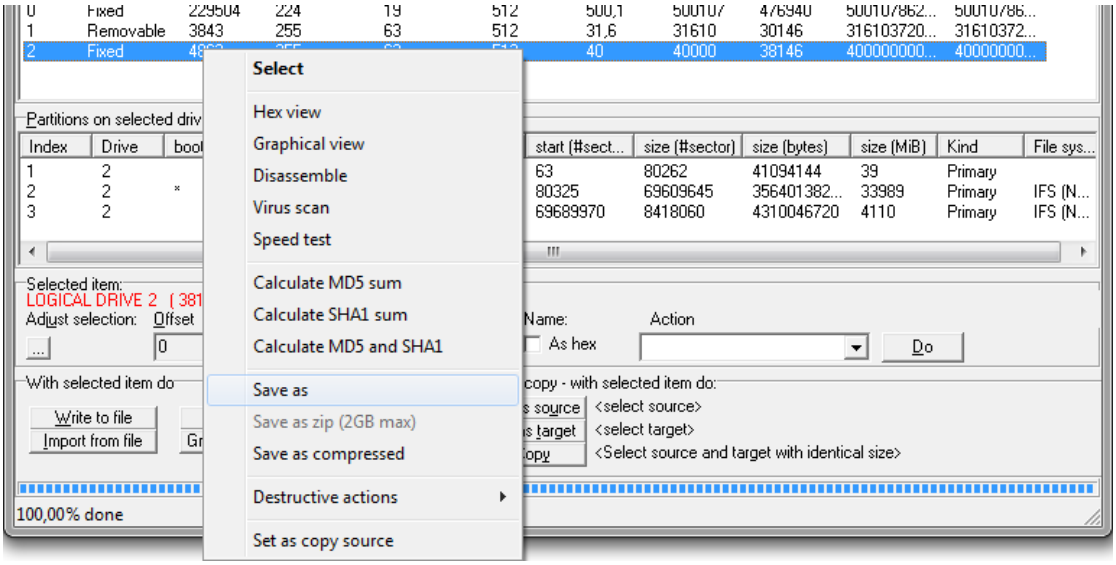
Obrázek 2.4: Nástroj DuBaron Disk Image

Na obrázku 2.4 je náhled okna nástroje.

- V horní části vidíme seznam jednotek, jak je mapují Windows (pod písmeny). Podle obrázku je zřejmé, že máme připojen jeden USB flash disk (pod písmenem E:), je označen jako

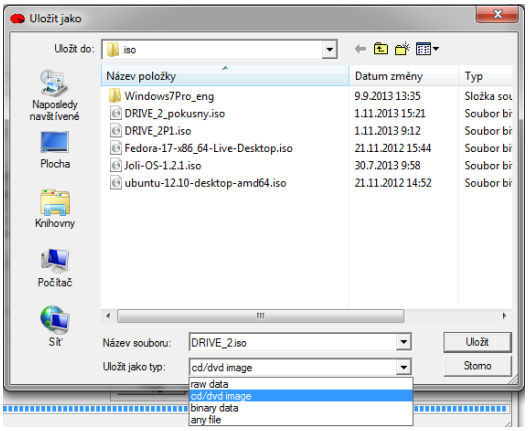
Removable (tedy výměnné médium). Ostatní položky jsou oddíly na pevných discích.

- Následuje seznam fyzických paměťových médií. Teď už je jasné, že kromě jednoho výměnného média jsou připojeny dva fyzické pevné disky, z nichž nás bude zajímat ten s indexem 2 (zde to není vidět, ale je připojen externě přes USB box). Tento disk jsme vybrali (klepli myší), což ovlivnilo zbytek obsahu okna.
- V třetí části okna je seznam oddílů na vybraném disku (zde na disku s indexem 2). Všechny tři oddíly jsou primární (sloupec Kind), prostřední je bootovací (sloupec Boot).



Obrázek 2.5: Možnosti práce s diskem v nástroji DuBaron Disk Image

Postup (Vytvoření ISO obrazu disku)



Na obrázku 2.5 je obsah kontextového menu získaného klepnutím pravým tlačítkem myši na příslušný disk (zde na disk s indexem 2). Po vybrání volby „Save as“ se zobrazí běžné „ukládací“ okno (jak vidíme zde vpravo), ve kterém můžeme určit formát. Pro vytvoření ISO zvolíme „cd/dvd image“ – původně totiž ISO soubory sloužily spíše k ukládání obsahu optických médií. Po nějaké době (záleží na velikosti disku) bude ISO soubor uložen tam, kde jsme si určili.

Podobně se vytváří ISO obraz některého oddílu na disku, jen volbu hledáme v kontextovém menu pro řádek s příslušným oddílem (oddíly vidíme na obrázcích 2.4 a 2.5 v tabulce hned pod seznamem fyzických disků).

Poznámka:

U představení tohoto programu je uvedena i licence, pod kterou je distribuován – GNU GPL. Údaj o licenci může být velmi důležitý, protože takto může autor vymezit možnosti použití produktu.

Pokud se jedná o software distribuovaný pod otevřenou licencí (GNU GPL, GNU LGPL, BSD apod.), nebývá problém s téměř jakýmkoliv způsobem použití včetně komerčního, případně úprav zdrojového kódu minimálně pro vlastní využití.

Freeware je často distribuován pod licencí EULA, u které se můžeme (nemusíme) setkat s určitými omezeními – například autor může povolit bezplatně pouze nekomerční použití, tedy pokud chceme program využívat ve firmě, můžeme, ale až po zaplacení.

V každém případě si při instalaci jakéhokoliv nástroje dostupného (nejen) zdarma dáváme pozor, jestli se nám nepokouší do systému propašovat ještě něco dalšího.

Úkol

Pokud máte možnost, vyzkoušejte si nástroj DuBaron Disk Image nebo jiný podobný. V případě prezentovaného nástroje máme výhodu přítomnosti dalších užitečných funkcí včetně kontroly integrity, ale samozřejmě pokud potřebujeme opravdu jen vytvořit obraz disku či oddílu, dá se najít spousta jiných.

Upozornění: pokud chcete vytvořit obraz oddílu disku nebo celého disku, počítejte s tím, že moc rychlé to tedy nebude. Kratší dobu strávíte nad USB flash diskem (záleží ovšem na jeho kapacitě).

Další informace:

Vlastně není úplně pravda, že přímo ve Windows není k dispozici nic na vytvoření obrazu disku podle paměťového média. Jedna možnost by tu byla, ale není zrovna uživatelsky příjemná. V PowerShellu si můžeme vytvořit funkci, která to zvládne. Postup je na http://thelowercasew.com/create-an-iso-file-with-powershell.

2.2.4 Připojení ISO obrazu disku

Pokud chceme již vytvořený (a dostatečněkrát zkopírovaný) obraz disku prozkoumat, potřebujeme software, který nám umožní se k souboru s obrazem disku (oddílu na disku) chovat jako ke skutečnému disku. Ve Windows od verze 8 je možné připojit ISO nebo IMG soubor jednoduše myší (pravé tlačítko myši, atd.), ovšem nelze si zvolit parametry připojení, například možnost připojení pouze pro čtení.

Pro tento účel existuje víc různých nástrojů, můžeme vyzkoušet například *ImDisk Virtual Disk Driver*, který je dostupný na http://www.ltr-data.se/opencode.html/#ImDisk a je distribuován pod licencí GNU GPL.

Tento software pracuje jako ovladač (protože se potřebuje napojit do jádra na mechanismus správy paměťových médií).Po instalaci najdeme v Ovládacích panelech ikonu programu (jak vidíme na obrázku 2.6), přes který můžeme ovladač konfigurovat a například určovat ISO soubory, které mají být připojeny jako virtuální disky. Pokud tuto operaci děláme častěji, je lepší si na Ploše vytvořit zástupce tohoto programu nebo po jeho spuštění si program připnout na Hlavní panel.

Postup (Připojení ISO obrazu disku)

Předpokládejme, že máme obraz disku, oddílu na disku, USB flash disku nebo optického média s názvem `obrazdisku.iso`. Nainstalovali jsme ovladač ImDisk a teď chceme pomocí tohoto ovladače připojit obraz jako virtuální disk:

- spustíme ovládací program ImDisku z Ovládacích panelů nebo jiným způsobem, otevře se okno, které vidíme na obrázku 2.7,
- v menu zvolíme *File – Mount new virtual disk*, otevře se konfigurační okno, které vidíme na obrázku 2.8,
- do pole „Image file“ načteme náš .iso soubor, vedle můžeme určit písmeno jednotky, pod kterým má být nový virtuální disk přístupný,
- v druhém sloupci zvolíme pro „Image file access“ druhou nebo třetí možnost (aby do samotného souboru nebylo za žádných okolností zasahováno),
- zatrhneme položku „Read-only media“ (připojíme pouze pro čtení), potvrdíme.

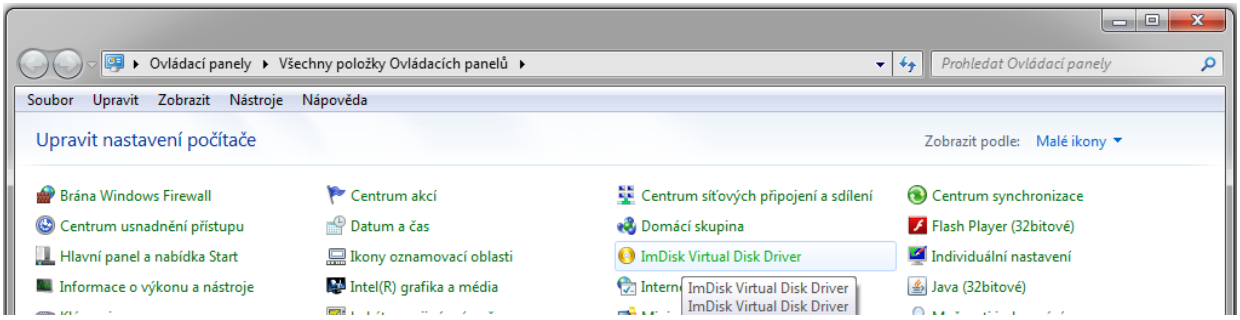
Od té chvíle je virtuální disk přístupný pod zadaným písmenem jednotky (jak na Příkazovém řádku, tak i v aplikaci Průzkumník či jiném souborovém manažerovi).

2.2.5 Záchrana poškozených dat

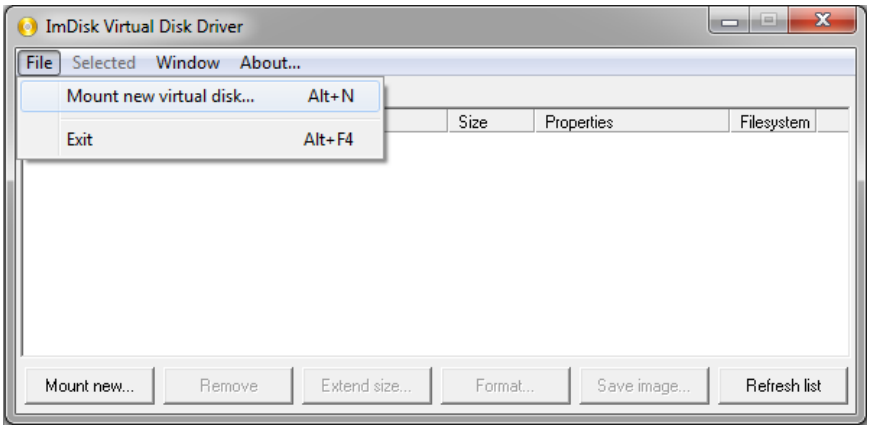
Nyní se budeme zabývat případem, kdy máme paměťové médium, s poškozenými soubory, které chceme (pokud možno) zachránit. Přímo ve Windows sice máme program `chkdsk`, ten však moc možností a schopností nenabízí, tedy je třeba použít nástroje třetích stran.

Asi nejlepším nástrojem pro Windows (alespoň z těch, které mají grafické rozhraní) je *Recuva* od firmy Piriform (od této firmy je například i velmi známý nástroj *CCleaner*). Tento program je v základní verzi volně dostupný, existuje i komerční varianta s dalšími možnostmi. Má jednoduché grafické rozhraní s průvodcem, nic složitého.

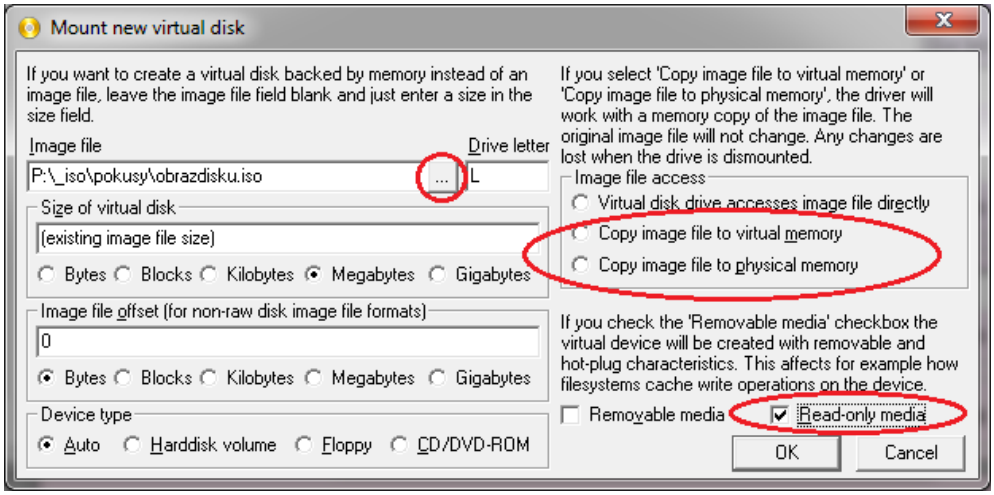
Dále existují nástroje bez grafického rozhraní, z nichž je zřejmě nejkvalitnější dvojice nástrojů *TestDisk* (pro zjištění poškozených oblastí disku) a *PhotoRec* (pro záchranu dat z těchto oblastí), obojí od firmy CGSecurity. Tyto nástroje jsou volně ke stažení ve variantách pro různé operační systémy včetně Windows. Recuva se sice lépe ovládá, ale PhotoRec je úspěšnější.



Obrázek 2.6: Ovladač ImDisk v Ovládacích panelech



Obrázek 2.7: Ovládací program ImDisku



Obrázek 2.8: Vytvoření virtuálního disku v ImDisku

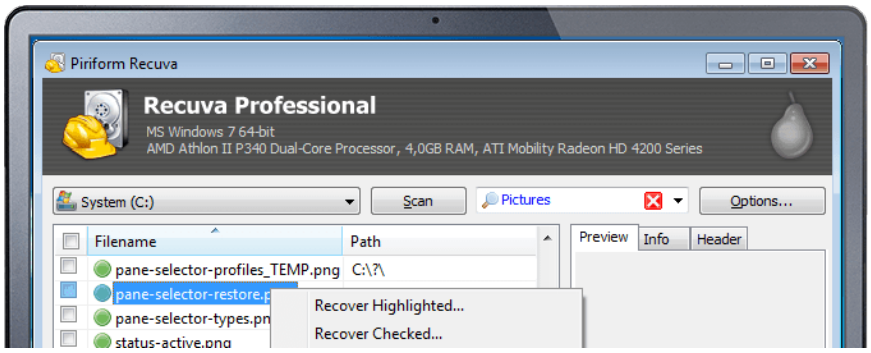
Další informace:

- https://www.piriform.com/recuva
- http://www.cgsecurity.org/wiki/TestDisk
- http://www.cgsecurity.org/wiki/PhotoRec
- http://www.svethardware.cz/photorec-a-recuva-jak-na-obnovu-dokumentu/29502-2

2.2.6 Přístup k různým souborovým systémům

Na každém paměťovém médiu najdeme souborový systém, jehož účelem je udržovat adresářovou strukturu na médiu, evidovat v ní soubory a další informace (atributy, přístupová oprávnění, údaje o názvech souborů, časové údaje apod.). Abychom v daném operačním systému mohli přistupovat k danému paměťovému médiu (včetně virtuálních vytvořených z ISO souborů), musí tento operační systém rozumět souborovému systému na tomto médiu, jinak je to pro něj jen změť bitů.

V každém operačním systému je v jádře komponenta, přes kterou prochází komunikace se souborovými systémy na připojených paměťových médiích, ve Windows to je modul IFSM (Installable



Obrázek 2.9: Rozhraní programu Recuva

File Systems Manager). IFSM si rozumí s těmito souborovými systémy:

- FAT16, VFAT, FAT32, NTFS – souborové systémy pro pevné disky, USB flash disky,
- exFAT – souborový systém pro výměnná média (od Windows XP),
- ISO9660, UDF – souborové systémy pro optická média (CD, DVD, BluRay).

Pro všechny tyto souborové systémy mají Windows ovladač (například pro NTFS to je soubor *ntfs.sys*). Pokud potřebujeme prozkoumat paměťové médium s jiným než podporovaným souborovým systémem, musíme pro něj sehnat a nainstalovat ovladač.

V Linuxu se obvykle používají souborové systémy rodiny extXfs – ext2fs, ext3fs a ext4fs (různá čísla jsou verze). Na Internetu můžeme najít dokonce několik ovladačů extXfs pro Windows, nejznámější jsou:

- Ext2Read dostupný na https://sourceforge.net/projects/ext2read/
- Ext2Fsd dostupný na https://sourceforge.net/projects/ext2fsd/

Oba nástroje jsou volně ke stažení, open-source projekty zveřejněné pod licencí GNU GPL v2. Oba zvládnou dnes nejběžnější verzi ext4fs včetně extentů, ten první umí bez problémů i LVM (logické svazky, něco jako dynamické svazky ve Windows). Přístup k souborům na daném diskovém oddílu pak funguje stejně jako u běžného NTFS.

Další možností je nástroj DiskInternals Linux Reader, který funguje spíše jako samostatný prohlížeč podobný Exploreru, je dostupný na https://www.diskinternals.com/linux-reader/access-ext4-from-windows/.

Pro souborový systém HFS a jeho potomky (HFS+ a HFSX), s nimiž se setkáváme v Apple MacOS, existuje pro Windows aplikace *HFSexplorer*. Je naprogramovaná v Javě, musíme mít instalováno Java Runtime Environment. Je dostupná na http://www.catacombae.org/hfsexplorer/ a distribuovaná pod licencí GNU GPL v3. Další možností je výše zmíněný DiskInternals Linux Reader, který taky zvládá HFS+ a HFSX.

Od verze High Sierra používá MacOS souborový systém APFS. Kromě DiskInternals Linux Readeru existují pro přístup k APFS z Windows spíše komerční nástroje (MacDrive APFS apod.).

2.3 Postupy v Linuxu

Většina linuxových distribucí v současné době zpřístupňuje paměťová zařízení hned po připojení i pro zápis, protože pro uživatele je to tak jednodušší a běžný uživatel to vyžaduje. Tedy pokud

chceme používat Linux při práci s „citlivými“ paměťovými médii (tím je míněno, že je potřebujeme mít připojené jen pro čtení, ideálně až ve chvíli, když to sami budeme chtít), pak máme dvě možnosti – buď si ve své oblíbené distribuci upravíme automount tak, aby vyhovoval této naší potřebě, anebo zvolíme distribuci, která je v tomto směru už přízpůsobená.

Pokud se nám nechce moc pátrat, můžeme zvolit některou z distribucí, které budou popisovány dále v sekci o forenzních distribucích od strany 46. Pokud chceme přece jen používat některou distribuci, která standardně připojuje média i pro zápis, obvykle stačí prohledat nápovědu k dané distribuci (například pro Ubuntu je postup na https://help.ubuntu.com/community/Mount/USB).

2.3.1 Jak Linux eviduje zařízení

V Linuxu je paměťové médium charakterizováno třemi prvky:

1. *ovladač* – modul v jádře, jako u kteréhokoliv jiného systému,
2. *speciální soubor* – soubor sloužící k nízkoúrovňovému přístupu k zařízení, speciální soubory najdeme v adresáři `/dev`, v angličtině se více setkáme s názvem *device file*,
3. *bod připojení* (platí pouze pro paměťová média) – adresář, ve kterém je přístupný obsah paměťového média (oddílu na disku, USB flash disku, optického média apod.), tyto adresáře najdeme většinou v adresáři `/media` nebo `/mnt` (podle distribuce).

Ovladač nás při průzkumu obvykle moc nezajímá, název speciálního souboru obvykle potřebujeme při připojování paměťového média, bod připojení (přípojný bod) potřebujeme znát kvůli tomu, že právě přes něj se dostaneme k datům na médiu. Ve Windows máme místo toho „písmenka“.

2.3.2 Připojení disku

U starších distribucí se ještě vzácně můžeme setkat s tím, že pro správu speciálních souborů (obecně zařízení) je používán systém *devfs*. V tom případě může být nutné nejdřív vytvořit speciální soubor pro paměťové zařízení, které chceme připojit. K tomu slouží příkaz `mknod`, parametry najdeme v manuálových stránkách. Dnes se však téměř výhradně setkáváme v distribucích se systémem *udev*, který vytváří speciální soubory automaticky, takže stačí po fyzickém připojení média zjistit, který ze speciálních souborů je ten pravý.

Postup (Průzkum speciálních souborů)

Svůj speciální soubor má jak každý disk (pevný disk, SSD, optický disk, USB flash disk apod.), tak i každý oddíl na disku (to se týká především pevných disků). Pokud je médium rozděleno na oddíly, připojujeme příslušný oddíl, jinak připojujeme disk.

Pokud naše distribuce používá pro správu zařízení systém *udev*, jsou speciální soubory paměťových médií pojmenovány následovně:

- `/dev/sda` je speciální soubor prvního pevného disku, USB flash disku či jiného paměťového média,
- `/dev/sdb` je speciální soubor druhého pevného disku či jiného média, atd. (podle písmene na konci),
- `/dev/sda1` je speciální soubor prvního oddílu na prvním pevném disku,
- `/dev/sda2` je speciální soubor druhého oddílu na prvním pevném disku,

- `/dev/sdb1` je speciální soubor prvního oddílu na druhém pevném disku, atd. (písmeno značí disk, číslo oddíl na tomto disku).

USB flash disky a další paměťová média, která nejsou přímo pevnými disky, používají totéž označení, včetně označení oddílů (ve skutečnosti i na USB flash disku či paměťové kartě můžeme mít několik oddílů, jen s nimi budeme mít trochu problém ve Windows).

Konkrétní název zjistíme například tak, že (po fyzickém připojení média) porovnáme obsah adresáře `/dev` (položky sdxxx) s tím, co je uvedeno v souboru `/etc/mtab` (to je seznam právě připojených paměťových médií). Speciální soubor, který je v adresáři `/dev` a zároveň není v souboru `/etc/mtab`, bude zřejmě ten, který hledáme.

Jednodušší asi bude použít následující příkaz:

```
lsblk -o name,label,size,fstype,model
```

Tento příkaz vypíše všechna bloková zařízení (což jsou obvykle paměťová média) – název speciálního souboru, popisek, velikost, souborový systém a Model number. Podle těchto parametrů obvykle dokážeme média odlišit, hlavně podle velikosti.

Postup (Připojení paměťového média pouze pro čtení)

Předpokládejme, že máme spuštěnou linuxovou distribuci, která neprovádí automatické připojování paměťových médií. Právě jsme fyzicky připojili USB flash disk a zjistili jsme, který speciální soubor k tomuto médiu přísluší. Dále potřebujeme přípojný bod pro naše zařízení. Může už být vytvořen, ale když není, je třeba ho vytvořit. V adresáři `/media` vytvoříme nějak vhodně pojmenovaný soubor, například `usbflash`:

```
mkdir /media/usbflash
```

(nebo můžeme totéž provést pomocí myši v grafickém prostředí).

Teď už můžeme provést softwarové připojení. Předpokládejme, že příslušný speciální soubor je `/dev/sdb` (což je pravděpodobné, jestli máme jen jeden HDD) a přípojný bod je `/media/usbflash`. Pak připojíme náš USB flash disk *pouze pro čtení* takto:

```
mount -o ro /dev/sdb /media/usbflash
```

Poté by v adresáři `/media/usbflash` měl být přístupný celý obsah připojeného USB flash disku.

Přepínač `-o` určuje parametry připojení, stanovili jsme parametr `ro` (read-only, pouze pro čtení). Jako další parametry připojení je možné zadat znakovou sadu (aby se správně zobrazovaly názvy souborů) a další, podle potřeby.

Pokud je v souboru `/etc/fstab` uveden řádek obsahující naši dvojici speciálního souboru a přípojného bodu, nemusíme do příkazu psát oba tyto údaje, stačí jeden z nich.

Paměťové zařízení odpojíme tímto příkazem (stačí uvést jen přípojný bod):

```
umount /media/usbflash
```

2.3.3 Vytvoření obrazu disku

V předchozím textu jsme si ukázali, jak připojit obraz disku v režimu pouze pro čtení. Pokud však chceme pouze vytvořit soubor s obrazem tohoto disku, tento postup nebudeme potřebovat.

Poznámka:

Obraz paměťového média vytváříme zásadně tak, že toto médium sice bude fyzicky připojeno, ale nikoliv softwarově. Pokud jsme předtím toto médium připojili, zase je odpojíme (například pomocí příkazu `umount`). Paměťové médium nebude přístupné přes žádný přípojný bod, budeme používat pouze název speciálního souboru tohoto média.

Je možné, že budeme mít k dispozici nástroj s grafickým rozhraním, záleží na konkrétní distribuci, použitém grafickém prostředí a případně na tom, zda důvěryhodný nástroj najdeme. Nicméně zde si ukážeme postup, který funguje v jakémkoliv UNIXovém systému včetně Linuxu, i kdyby zrovna neměl žádné grafické rozhraní.

Použijeme příkaz `dd`, který slouží k rychlým přesunům dat, s určitými parametry může provádět i základní konverze. Pro nás jsou zajímavé tyto parametry příkazu:

- `if=xxx` je zkratka z „input file“, určuje vstup, ze kterého chceme kopírovat data (speciální soubor oddílu disku nebo jiného pam. média, název souboru apod.; když neuvedeme, bere se standardní vstup – třeba přes rouru),
- `of=yyy` je zkratka z „output file“, určuje výstup, na který chceme data kopírovat (když neuvedeme, bere se standardní výstup),
- `bs=zzz` je velikost bloku (block size), který se kopíruje v jednom kroku (výchozí je 512 B), různé hodnoty mohou zrychlit/zpomalit kopírování, můžeme použí například `bs=4k` pro práci s bloky 4 KiB (typická velikost clusteru na discích),
- `count=bbb` v případě, že jsme určili velikost bloku, takto stanovíme počet bloků, které mají být kopírovány, zbytek vstupu bude ignorován,
- `conv=aaa` za rovnítkem následuje seznam argumentů pro nízkoúrovňové konverze, nás může zajímat například
  - `noerror` pokud dojde k chybě, příkaz se neukončí, ale pokusí se o zotavení a pokračuje za chybně přečtenými daty,
  - `sync` pokud jsme použili předchozí argument, pak chybná data nahradí nulovými daty (s ascii kódem 0),
  - `notrunc` jestliže je výstup delší než vstup, zachová se délka výstupu (nebude „useknut“); používá se například při klonování jednoho diskového oddílu na jiný,

  takže parametr se všemi třemi argumenty by byl `conv=noerror,sync,notrunc`.

Příklad

Použití příkazu `dd` si ukážeme na příkladech:

- `dd if=/dev/sdb of=~/kopie/mujdisk.iso`
  vytváříme ISO obraz (pravděpodobně optického) datového média, jehož speciální soubor je `/dev/sdb`, výsledek bude uložen v domovském adresáři
- `dd if=/dev/sdb1 of=/disky/mujdisk.iso bs=4k`
  vytváříme kopii prvního oddílu na disku se speciálním souborem `/dev/sdb` (může být i externí), výsledek uložíme do místního souboru

- `dd if=/dev/sdc of=/media/externi/kopie.iso`
  vytváříme kopii (zřejmě) USB flash disku, výsledek uložíme do souboru na externím disku, který je přístupný přes přípojný bod `/media/externi` (musí být samozřejmě přístupný i pro zápis)
- `dd if=/dev/sda2 of=/dev/sda5 bs=4k conv=notrunc,noerror`
  obsah druhého oddílu prvního pevného disku zálohujeme do pátého oddílu téhož disku (cílový oddíl by měl být pokud možno stejně velký nebo větší)
- `dd if=/dev/zero of=/dev/sda4 bs=4k conv=notrunc`
  přepíšeme celý čtvrtý oddíl na pevném disku nulami (ascii kód 0, tedy všechny bity nastavíme na 0)
- `dd if=/dev/random of=/dev/sda4 bs=4k conv=notrunc`
  přepíšeme celý čtvrtý oddíl na pevném disku náhodnými čísly (pokud toto provedeme vícekrát, pravděpodobně už nebude možné žádná data z oddílu zachránit – provádějte před prodejem disku)
- `dd if=/dev/zero of=~/pomocne/novy.dat bs=4k count=10`
  vytvoříme nový (pokud ještě neexistoval) soubor o délce $1024 * 4 * 10 = 40\,960$ B, vyplněný nulami

Co se týče parametru `bs`, sice není povinný, ale v určitých případech může urychlit proces (zkopírování celého oddílu na disku je celkem časově náročné). Obecně platí, že spíše vyšší hodnota znamená vyšší rychlost (delší bloky ⇒ méně kroků), u vnějších paměťových médií je dobré volit délku clusteru (pokud ji známe).

Může se stát, že potřebujeme vytvořit obraz datového média, který se nám nevejde na cílové místo, případně je nad možnosti příslušného souborového systému (u každého souborového systému existuje limit na maximální délku souboru). Pak je třeba použít parametry `count` (určuje počet bloků), `skip` (přeskočí určitý počet bloků na začátku vstupu) a `seek` (přeskočí určitý počet bloků na výstupu, použijeme pro případné budoucí kompletování).

Poznámka:

U příkazu `dd` si dávejte velký pozor a raději si ho dvakrát přečtěte, než klepnete na Enter. Pokud uděláte chybu v parametrech, mohou být následky velmi vážné. Například když zaměníte parametry `if` a `of`, přepíšete disk, ze kterého jste ve skutečnosti chtěli kopírovat!

Další informace:

- http://linux.die.net/man/1/dd
- http://www.abclinuxu.cz/clanky/navody/unixove-nastroje-11-split-dd
- http://www.computerhope.com/unix/dd.htm
- http://www.linuxquestions.org/questions/linux-newbie-8/learn-the-dd-command-362506/

Kromě příkazu `dd` existují i další příkazy, které jsou již přímo určeny pro konkrétní účely (berme je jako varianty či rozšíření tohoto příkazu), například pro zálohu dat nebo pro použití při digitální forenzní analýze.

Příkaz `dcfldd` je právě rozšířením příkazu `dd` o možnosti, které mohou hodit při digitální forenzní analýze, například dokáže zároveň s kopírováním vytvořit kontrolní součet, vytvářet víc kopií zároveň a zaznamenávat průběh své činnosti. Není běžnou součástí distribucí, v běžné distribuci ho musíme doinstalovat (v repozitáři hledáme balíček *dcfldd*).

Většina parametrů příkazu `dcfldd` je stejná jako u `dd`, jen některé jsou navíc a některé lze použít i jiným způsobem. Například:

- `of=xxx` lze v jednom příkazu použít i vícekrát, pokud chceme, aby bylo zároveň vytvářeno více výstupů,
- `errorlog=yyy` určuje soubor, do kterého se má zapisovat průběh vytváření obrazu (jinak se vypíše jen na obrazovku),
- `hash=zzz` stanovuje typ kontrolního součtu, který má být vypočten (viz další kapitolu, příklady hash funkcí jsou na straně 111),
- `hashlog=aaa` určuje soubor, do kterého se má zapisovat případná chyba při vytváření hashe,
- `split=bbb` je parametr zjednodušující to, na co u `dd` potřebujeme kombinaci `skip` a `count`, tedy rozporcuje vstup do více výstupů.

Parametrů je více, například pro generování logů lze místo souboru zadat program, kterému se mají logová data posílat k dalšímu zpracování, případně můžeme určit délku sekvencí výstupu, na které postupně bude hash funkce používána, taky může být stanoveno více hash funkcí s určením, jak mají být kombinovány.

Příklad

Typické použití příkazu `dcfldd` je následující:

```
dcfldd if=/dev/sdb1 of=/img/img2015-09-04.dd bs=2k conv=noerror,sync \
   errorlog=/img/log2015-09-04.txt hash=sha256 hashlog=/img/haslog2015-09-04.txt
```

Nejdřív jsme určili vstup (první oddíl druhého fyzického disku) a výstup (soubor s příponou .DD), délka bloku pro kopírování je 2 KiB, log s případnými chybami najdeme v zadaném souboru, bude vypočten hash šifrou SHA-2 o délce 256 B, případné chyby z generování hashe najdeme opět v zadaném souboru.

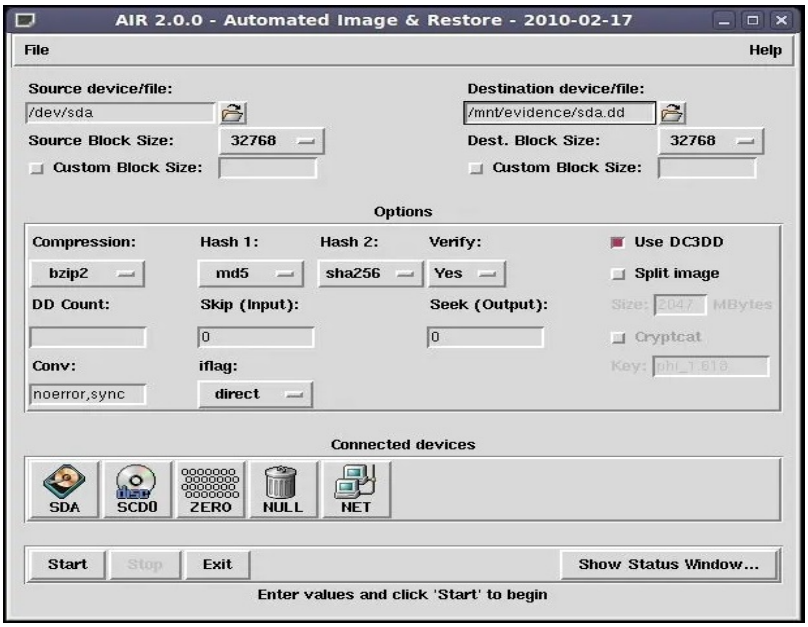
Podobné možnosti jako `dcfldd` má také program `dc3dd` (také dokáže vytvářet hashe a směrovat logy do souborů), jeho možnosti jsou však o něco menší než u `dcfldd`.

Další informace:

- http://linux.die.net/man/1/dcfldd
- http://forensicswiki.org/wiki/Dcfldd
- http://www.obsidianforensics.com/blog/imaging-using-dcfldd
- http://sourceforge.net/projects/dc3dd/
- http://forensicswiki.org/wiki/Dc3dd

- http://www.linuxcertif.com/man/1/dc3dd/

Pro příkazy `dd` a `dcfldd` existuje také grafický frontend (tj. ovládací aplikace s grafickým rozhraním) – open-source nástroj AIR (Automated Image & Restore). Tato aplikace však v některých distribucích nebude fungovat, případně v jiných může mít omezenou funkcionalitu (na webu projektu je seznam distribucí, ve kterých aplikace plně funguje, ale pokud tam některá konkrétní distribuce není, neznamená to nutně, že by v ní aplikace nebyla použitelná).



Obrázek 2.10: Rozhraní nástroje AIR[1]

2.3.4 Připojení ISO obrazu disku

Už víme, jak vytvořit ISO obraz paměťového média, zbývá se naučit, jak tento obraz připojit takovým způsobem, abychom s ním mohli běžně pracovat podobně jako přímo s původním fyzickým médiem (ovšem s připojením pouze pro čtení, abychom obraz nekompromitovali).

Postup (Připojení obrazu disku pouze pro čtení)

Předpokládejme, že chceme připojit soubor `/obrazy/mujobraz.iso`, a to pouze pro čtení. Nejdřív vytvoříme přípojný bod (pokud již není vytvořen), například:

```
mkdir /media/isoobraz
```

Nebudeme potřebovat speciální soubor, místo toho použijeme smyčku (loop) z našeho ISO obrazu, tedy kromě parametru `ro` použijeme i parametr `loop`:

```
mount -o ro loop /obrazy/mujobraz.iso
```

Stojí za zvážení, jestli nepoužít i další parametry, například určení znakové sady, to však už necháme na čtenáři (viz manuálové stránky nebo výuka v předmětu Operační systémy).

[1] Zdroj: https://www.linuxlinks.com/automatedimageandrestore/

2.3.5 Záchrana poškozených dat

V Linuxu zachraňujeme data nejen z oddílů s unixovými souborovými systémy, optických médií, USB flash disků či SD karet, ale také samozřejmě z windowsovských oddílů.

Také v Linuxu můžeme použít dvojici nástrojů TestDisk a Recuva, které jsou zmíněny v části kapitoly o Windows na straně 37, ale kromě toho máme k dispozici i jiné nástroje.

Nástroj `ddrescue` je postaven na nástroji `dd`, přičemž byly přidány právě možnosti záchrany dat z poškozeného média. Parametry jsou zapisovány trochu jinak (tradičněji) než u `dd`.

Příklad

Použití programu `ddrescue` si ukážeme na příkladech.

- `ddrescue -d -r3 /dev/sdb /obrazy/obrazsdb.img /obrazy/obrazsdb.log`
  projde druhý pevný disk (`/dev/sdb`), výsledek uloží do souboru `/obrazy/obrazsdb.img`, při čtení vstupu použije přímý přístup k hardwaru bez asistence knihoven (přepínač `-d`), poškozené sektory se pokouší číst 3× (přepínač `-r3`), posledním parametrem je název výsledného log souboru
- `ddrescue -d -f -r3 /dev/sda /dev/sdb /media/usb/rescue.logfile`
  zde víceméně klonujeme první fyzický disk na druhý včetně opravy chyb, navíc je parametr `-f` zamezující výpisu různých dotazů (force), log soubor je třeba uložit jinam – zde například na USB flash disk (pozor, druhý disk by měl být minimálně tak velký jako první)
- `ddrescue -d -r3 /dev/sdc /obrazy/dvd.iso /obrazy/dvd.log`
  pokoušíme se opravit poškozené DVD (vytvořit opravený obraz)

Další informace:

- http://linux.die.net/man/1/ddrescue
- https://www.technibble.com/guide-using-ddrescue-recover-data/
- http://www.gnu.org/software/ddrescue/manual/ddrescue_manual.html

2.4 Hotová řešení

2.4.1 Forenzní distribuce

Forenzní distribuce jsou linuxové distribuce vytvořené právě pro účely digitální forenzní analýzy. V těchto distribucích najdeme především nástroje použitelné pro tyto účely (včetně těch s grafickým rozhraním), obvykle snadno přístupné obvykle přes menu systému.

Jedná se o live distribuce, to znamená, že dotyčný systém máme uložen na výměnném paměťovém médiu (DVD nebo USB flash disk) a z tohoto média ho spouštíme (musí být nastaveno a povoleno v BIOSu dotyčného počítače, je třeba bootovat z tohoto média, nikoliv z vnitřního pevného disku). Důležitou vlastností forenzních distribucí je, že během startu nedochází k automatickému připojování paměťových médií, alespoň ne pro zápis.

Typicky se v těchto distribucích setkáváme i s nástroji na analýzu počítačů s instalovanými Windows. Po nabootování z live CD s Linuxem pak máme možnost procházet jak data, tak i nastavení Windows bez toho, aby analyzovaný systém běžel (a mohl klást překážky).

Caine. Ve skutečnosti jde o zkratku C.A.IN.E. (Computer Aided Investigative Environment) a je to live distribuce pocházející z Itálie. Má jednoduché přehledné grafické rozhraní, k jednotlivým nástrojům se dostáváme přes menu systému.

Obrázek 2.11: Prostředí forenzní distribuce C.A.IN.E. (obrázek z webu projektu)

V distribuci jsou samozřejmě všechny běžné linuxové nástroje (`dd`, *GParted*, *ntfs-3g* atd.), nástroje výše uvedené (například `dcfldd`, `ddrescue`, `dc3dd`, `Photorec`, *AIR* atd.), programy pro vytváření kontrolních součtů, otevírání a přehrávání různých typů souborů včetně multimediálních (také pro práci s exif daty), síťové nástroje (včetně šifrování po síti – *Cryptcat*), nástroje pro práci s paměťovými médii (včetně SSD), nástroje pro analýzu oddílů s nainstalovanými Windows (včetně přístupu k heslům), součástí distribuce je také *Autopsy* – rozhraní k balíku forenzních nástrojů *Sleuth Kit* (viz dále).

Na níže uvedeném odkazu najdeme charakteristiku distribuce, seznam obsažených nástrojů, některé postupy a odkazy ke stažení.
Web: https://www.caine-live.net/

Paladin. Paladin je komerční distribuce založená na Ubuntu, jejím centrálním nástrojem je aplikace Paladin Toolbox zpřístupňující různé forenzní nástroje. Kromě jiného obsahuje Autopsy a údajně si dokáže poradit i s dešifrováním disků šifrovaných BitLockerem.
Web: https://sumuri.com/software/paladin/

Security Onion. Tato volně dostupná distribuce je určena k hledání zranitelností, prověřování napadení systému, monitoringu sítě a správě logů a výstupů bezpečnostních sond typu Snort.

Web: https://securityonionsolutions.com/software

Obrázek 2.12: Prostředí forenzní a penetrační distribuce Kali (obrázek z webu projektu)

Kali Linux. Jedná se o penetrační distribuci, tedy určenou spíše pro testování zabezpečení zařízení v síti, ale obsahuje i některé forenzní nástroje.

Při bootování volíme mezi několika možnostmi, mezi nimiž je „Forensics Live Boot“ (jinak volíme položku „Live amd64“, distribuce je sestavena pro 64bitové systémy). Existuje také varianta pro nepříliš běžné platformy, jako například pro Raspberry Pi.

Co se týče softwarového vybavení, nebudeme už vše jmenovat – v Kali Linuxu najdeme víceméně cokoliv, co bychom mohli potřebovat při využití ve forenzní analýze či penetračním testování. Pokud při bootování zvolíme využití pro forenzní analýzu, zcela vyřadíme automount.

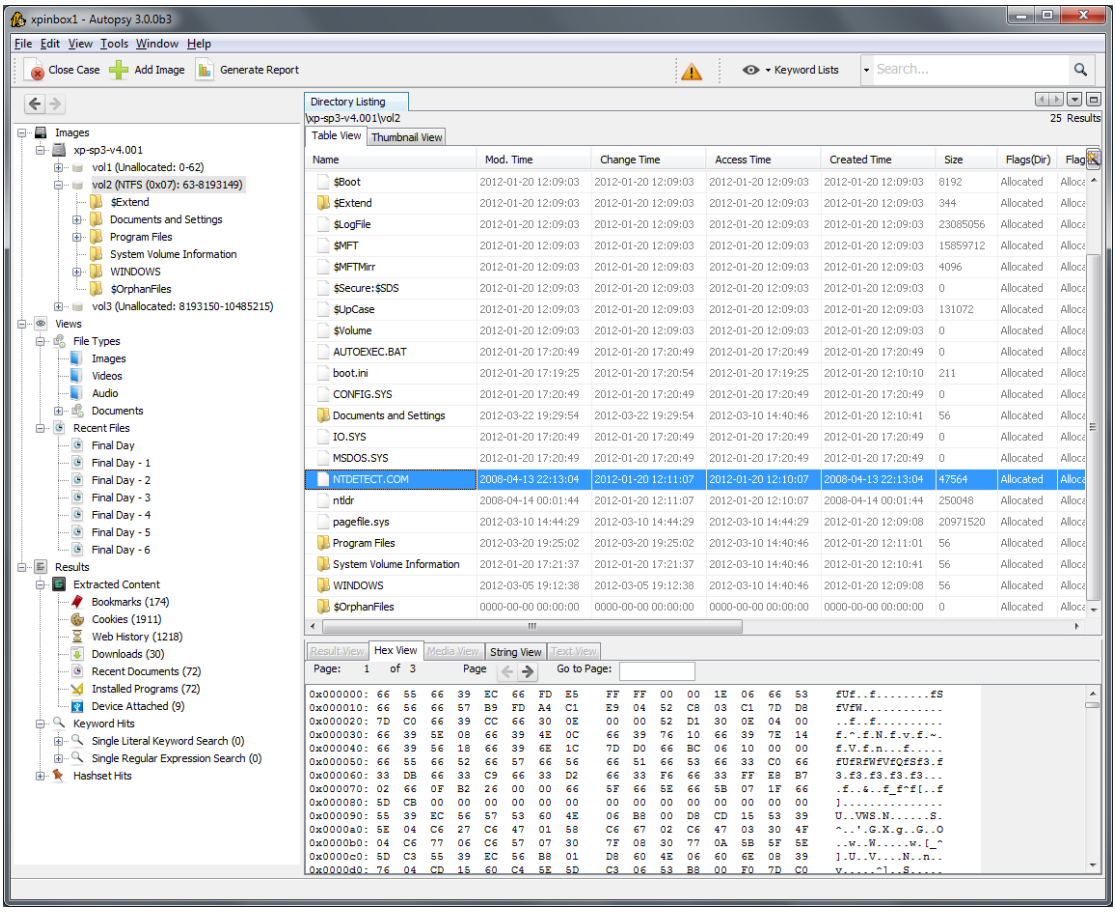
Zajímavá může být možnost úpravy ISO obrazu na míru, na webu projektu je dokonce k dispozici návod.

Kali Linux NetHunter je projekt penetrační platformy pro Android na některých zařízeních řady Nexus a OnePlus. K jeho využití samozřejmě potřebujeme podporované zařízení.
Web: https://www.kali.org/, http://tools.kali.org/tools-listing, https://www.kali.org/kali-linux-nethunter/

2.4.2 Forenzní prostředí

Existují aplikace ve formě speciálního prostředí, které běží pod Windows nebo některým unixovým systémem. Obvykle jde o sadu nástrojů, která je dostupná přes jednotné (většinou grafické) rozhraní.

Autopsy. Jedná se o aplikaci, která je rozhraním k sadě forenzních nástrojů *The Sleuth Kit*. Toto prostředí je možné používat ve Windows, v Linuxu i v MacOS X, je volně ke stažení. Ukázku použití Autopsy při analýze obrazu disku najdeme v příloze A.1 na straně 170.



Obrázek 2.13: Prostředí aplikace Autopsy (obrázek z webu projektu)

Pomocí Autopsy je možné procházet strukturu souborů a provádět její analýzu (včetně multimediálních dat s exif informacemi), u běžných webových prohlížečů se dostat k historii, záložkám, cookies, obnovovat smazané soubory (obsahuje nástroj PhotoRec), apod.
Web: http://www.sleuthkit.org/autopsy/, http://www.sleuthkit.org/sleuthkit/docs.php

SIFT (SANS Investigative Forensic Toolkit). SIFT je sada nástrojů použitelných pro digitální forenzní analýzu, případně pro atribuci útoku. Existuje v několika variantách: předně jde o applianci (tj. kompletní nainstalovaný obraz operačního systému) do virtuálního počítače, dále se dá získat ve formě softwaru intalovaného do Ubuntu Linuxu (ale zřejmě by to mohlo fungovat i u odvozených systémů), třetí možnost je instalace ve Windows do podsystému WSL (Windows Subsystem for Linux), ovšem to bychom nejdřív museli mít aktivován zmíněný subsystém. Vlastně to napůl patří do předchozí sekce s forenzními distribucemi.
Web: https://www.sans.org/tools/sift-workstation/

Digital Forensics Framework (DFF). Tato aplikace pro Windows a Linux je v odlehčené variantě zdarma ke stažení, za některé funkce je však třeba si připlatit. Jedná se o komplexní nástroj poskytující Write Blocker (jako většina předchozích), možnost analýzy souborů v nejrůznějších souborových systémech, různých formátů, registru Windows, apod. Placená verze navíc umí například analyzovat Skype a data webových prohlížečů (což většina předchozích umí i zadarmo), ale také obsahuje profesionální podporu pro forenzní záznam.

Web: http://www.arxsys.fr/

Další informace:

Podobných projektů je mnohem více. Některé nástroje jsou komplexnější, jiné zaměřené na konkrétní účel (například na analýzu mobilních zařízení, počítačových sítí, sociálních sítí). Seznamy forenzních nástrojů najdeme například na

- http://resources.infosecinstitute.com/computer-forensics-tools/
- https://forensiccontrol.com/resources/free-software/
- http://forensicswiki.org/wiki/Tools
- http://www.cert.org/digital-intelligence/tools/
- https://en.wikipedia.org/wiki/List_of_digital_forensics_tools

2.5 Lokální a vzdálená metoda dead analýzy

Podívejme se na postup získání ISO obrazu paměťového média podle toho, zda u zařízení přímo sedíme (tedy nemusíme obraz přenášet přes síť) nebo zda chceme obraz pořizovat na dálku přes síť. Jsou tyto možnosti:

- ze zařízení byl vyjmut pevný disk, bezpečně transportován a obraz bude pořízen na důvěryhodném zařízení,
- obraz pořizujeme přímo na kompromitovaném zařízení,
- obraz bude pořizován přes počítačovou síť (vzdáleně).

Postup (Pořízení ISO obrazu lokálně na důvěryhodném zařízení)

Tato možnost je nejbezpečnější a nejjednodušší. Na zařízení máme správně seřízen čas (kvůli reportování) a je zajištěno, že média nebudou připojována pro zápis – používáme některou formu Write Blockeru.

Disk připojíme například pomocí externího boxu, přičemž dáváme pozor, aby disk nebyl připojen pro zápis (tj. Write Blocking). Můžeme využít některou z dříve probíraných metod, ideálně ve forenzní distribuci.

Postup (Pořízení ISO obrazu lokálně na kompromitovaném systému)

Předpokládejme, že kompromitovaný systém můžeme vypnout. Je třeba zajistit, aby byl správně seřízen čas a aby nedošlo k dalšímu ovlivnění obsahu disku. Proto postupujeme takto:

- odpojíme všechny disky a jiná média, pokud to jde,
- počítač zapneme (či restartujeme) a přejdeme do BIOS Setup, tam zkontrolujeme především nastavení času a pořadí bootování (pak budeme bootovat z výměnného média),
- vložíme/připojíme médium s bezpečným systémem (například forenzní distribuci na optickém disku nebo USB flash disku),
- restartujeme a ověříme, zda bootování probíhá tak jak má, zajistíme Write-Blocking,
- připojíme ta datová média, která budeme potřebovat, začneme se snímáním obrazu.

Postup (Pořízení ISO obrazu přes síť – vzdáleně)

Tuto metodu použijeme, pokud na místě nemáme možnost ISO obraz zachytit, například kvůli jeho velikosti, případně disky nejsou fyzicky přístupné. Situaci komplikuje požadavek na zajištění bezpečnosti, obraz nesmí být při cestě sítí pozměněn a pokud možno ani zachycen. Typicky přenášíme v rámci lokální sítě, ale i přesto bychom měli šifrovat. Postupujeme takto:

- na kompromitovaném zařízení spustíme bezpečnou live distribuci (třeba některou forenzní) a zajistíme Write-Blocking,
- na přijímajícím počítači také musí běžet prověřený systém,
- na obou zařízeních máme zjištěny IP adresy (adresu přijímajícího zařízení budeme potřebovat do příkazu, zde například 10.6.25.187),
- stanovíme číslo portu, který bude využit při přenosu (například 3000, nesmí být používán jinými procesy), tento port musí být povolen na mezilehlých síťových zařízeních (například firewallech),
- použijeme program NetCat (příkaz `nc`):
  - na přijímajícím počítači zapneme naslouchání na portu 3000:
    `nc -l 3000 | dd of=/mount/images/getimage2015-09-09.dd`
  - na kompromitovaném počítači bude vytvářen obraz a průběžně odesílán na síť:
    `dd if=/dev/sda1 bs=4k | nc 10.6.25.187 3000`

Správně bychom měli navíc na odesílající straně šifrovat a na přijímající straně dešifrovat. To se dá zajistit několika způsoby. Asi nejméně úprav vyžaduje zapojení některého šifrovacího nástroje do roury, například `gpg` nebo `ssl` (případně `openssl`, podle konkrétní implementace). V našem případě potřebujeme pouze jednosměrný transfer, tedy bychom si s těmito možnostmi vystačili.

Postup (Pořízení ISO obrazu přes síť – vzdáleně a šifrovaně)

Variace předchozího příkladu při použití GPG na straně příjemce:

```
nc -l 3000 | gpg -d | dd of=/mount/images/getimage2015-09-09.dd
```

a na straně odesílatele:

```
dd if=/dev/sda1 bs=4k | gpg -e | nc 10.6.25.187 3000
```

Ovšem tento způsob komunikace nebude zcela neinteraktivní, sem tam bude třeba v cíli opakovaně zadávat heslo.

Další možnost je místo příkazu `nc` použít příkaz `ncat`, což je speciální implementace NetCatu „říznutá“ programem `nmap`, která má v sobě zabudovanou podporu SSL. Další speciální implementace je `socat` opět s vestavěnou možností šifrování.

2.6 Ověření integrity

Při vytváření, uchovávání nebo transportu obrazu disku (či jiného balíku dat) může dojít k jejich pozměnění, což by znehodnotilo jakékoliv závěry, ke kterým jsme pomocí těchto dat došli.

Abychom dokázali včas pozměnění dat odhalit, musíme vygenerovat *kontrolní součet* původních dat a následně i nových dat. Existuje více různých druhů kontrolních součtů – MD5, SHA-1, SHA-256, atd., nástroj volíme podle toho, který kontrolní součet chceme vygenerovat.

Například pokud vytváříme obraz USB flash disku, vytvoříme si kontrolní součet tohoto disku a následně kontrolní součet vytvořeného obrazu. Pokud tento obraz budeme transportovat po síti, v cíli opět můžeme vygenerovat kontrolní součet. Ve všech třech případech bychom měli dostat tentýž výsledek – kdyby ne, znamenalo by to, že došlo k pozměnění obrazu.

Poznámka:

Generování kontrolních součtů úzce souvisí s kryptografií, kterou se budeme zabývat především v kapitole 4. O různých typech kontrolních součtů (tedy o hash funkcích pro zajištění integrity dat) se píše od strany 104.

Postup (Vygenerování kontrolního součtu ve Windows)

Potřebujeme vytvořit kontrolní součet souboru `C:\obrazy\soubor.iso`. Máme tyto možnosti:

- Spustíme PowerShell, v něm existuje cmdlet `Get-FileHash`:
  `Get-FileHash C:\obrazy\soubor.iso -Algorithm MD5`
  Tento cmdlet je však až v PowerShellu od verze 4, tedy není použitelný ani ve Windows 7. V PowerShellu verze 4 jsou dostupné pouze algoritmy MD5 a SHA1, ve vyšší verzi i další.
- Jednodušší je použít nástroj třetí strany, ostatně aplikace popsané v této kapitole obvykle generování kontrolního součtu bez problémů zvládnou. Podívejme se například na *DuBaron Disk Image*, kterým jsme se zabývali na straně 34 (je tam také odkaz na web projektu). Stačí klepnout myší na příslušný disk či oddíl na disku a v menu zvolit *Calculate...*.

Postup (Vygenerování kontrolního součtu v Linuxu)

Linux obsahuje nástroje prakticky na cokoliv, a najdeme zde také programy na generování kontrolních součtů. Jako parametr zadáváme buď název souboru, ze kterého chceme vygenerovat kontrolní součet, nebo v případě disku či oddílu příslušný speciální soubor.

- vygenerování MD5 hashe souboru a oddílu na disku:
  `md5sum /obrazy/dd2015-09-10.dd`
  `md5sum /dev/sdb1`
- podobně, jen navíc oboje uložíme do souboru – pod sebe, ať se dobře porovnávají:
  `md5sum /obrazy/dd2015-09-10.dd > /obrazy/kontrola.txt`
  `md5sum /dev/sdb1 » /obrazy/kontrola.txt`
- pokud zadáme oba vstupy najednou, výsledky se vypíšou na samostatné řádky:
  `md5sum /dev/sdb1 /obrazy/dd2015-09-10.dd`
- pokud chceme spíše SHA-1 nebo 256bitový SHA-2, zadáme:
  `sha1sum /obrazy/dd2015-09-10.dd`
  `sha256sum /obrazy/dd2015-09-10.dd`
  `shasum -a 256 /dev/sdb1` (pro SHA2 s délkou klíče 256; umí i jiné délky klíče)

Použití je podobné jako u `md5sum`. Ve všech případech lze vstup poslat i přes rouru.

Jestliže místo původního disku či oddílu máme k dispozici jen soubor s kontrolním součtem a chceme ověřit, jestli třeba během transportu nedošlo k pozměnění obrazu, postupujeme takto:

- jak obraz, tak i soubor s jeho kontrolním součtem uložíme do téhož adresáře, soubor s kontrolním součtem se bude zřejmě nazývat stejně jako původní soubor, jen bude přidána přípona podle typu použité hash funkce při generování kontrolního součtu, například `obraz.dd` a `obraz.dd.md5` nebo `obraz.dd.sha1`,
- přesuneme se do dotyčného adresáře:
  `cd /obrazy/testdd`
- otestujeme:
  `md5sum -c obraz.dd.md5` nebo třeba pro SHA1:
  `shasum -a 1 -c obraz.dd.sha1`

V Linuxu existují i nástroje s grafickým rozhraním – záleží, jaké grafické prostředí máme nainstalováno a které balíčky jsme doinstalovávali. Například pro prostředí Gnome existuje nástroj *gtkhash*. Taktéž ve forenzních distribucích obvykle nějaký takový nástroj najdeme, případně je funkce generování kontrolního součtu součástí jiných nástrojů.

Postup (Vygenerování kontrolního součtu v Apple MacOS)

Taktéž pro systém od Applu existují nástroje pro tyto účely. Můžeme najít funkci generování hashe jako součást komplexnějších nástrojů, nebo můžeme použít příkaz textového režimu, například pro 256bitový SHA-2 hash:

```
shasum -a 256 /obrazy/dd2015-09-10.dd
```

Pro MacOS existuje také program s grafickým rozhraním – *HashTab*.

Úkol

V tom operačním systému, který zrovna máte spuštěn, si zjistěte, jaké máte možnosti generování kontrolních součtů. Dále si vytvořte malý textový soubor s několika řádky textu (je jedno, jaký má obsah). Předpokládejme, že dotyčný soubor se jmenuje `test.txt`.

- Nejdřív si vytvořte pro soubor `test.txt` kontrolní součty MD5, SHA1 a SHA-256 (případně i jiný, jestli budete chtít). Porovnejte délku výsledných hashů.
- V souboru `test.txt` proveďte drobnou změnu: přepište jedno písmeno písmenem následujícím v abecedě (je jedno, které písmeno zvolíte). Nyní znovu vytvořte kontrolní součty jako v předchozím bodě, porovnejte jejich obsah.

Zřejmě zjistíte, že i taková drobná změna daný hash zcela „překope“.

2.7 Překážky

2.7.1 Skryté oblasti na disku

Pokud si koupíme počítač s předinstalovaným systémem (a nejen tehdy), velmi pravděpodobně je na disku alespoň jedna skrytá oblast. Skrytou oblast nelze odhalit nástroji proběžné uživatele, dokonce ani jejich existenci, není patrná ani v BIOS Setup.

V lepším případě v nich najdeme Recovery partition, která se pak používá při případném přechodu systému do „továrního nastavení“, případně je tam záloha některých systémových souborů či ovladačů. V horším případě tam je zálohován či instalován takový software, který do systému nainstaloval distributor bez vědomí uživatele (může jít až o formu spywaru), v nejhorším případě tam najdeme škodlivý software, který se tam dostal dokonce i bez vědomí distributora.

Poslední jmenovaný případ ilustruje nebezpečnost existence skrytých oblastí – jsou zneužitelné. Další nevýhodou jejich existence je často zbytečně zabrané místo na disku.

HPA (Host Protected Area). Tento typ skryté oblasti používají distributoři tak, jak bylo výše naznačeno, může být na pevných discích komunikujících sadou ATA od verze 4. Do HPA se dá dostat pomocí speciálních ATA příkazů (ATA příkazy slouží k nízkoúrovňové komunikaci s paměťovými zařízeními připojenými např. přes sběrnici SATA). S HPA pracujeme takto:

- program *MHDD* umí HPA zviditelnit a pracovat s ní – jedná se o volně šiřitelný bootovací nástroj postavený na FreeDOSu,
- program `hdparm` používaný v textovém režimu dokáže HPA zjistit a zrušit – je volně šiřitelný, použitelný ve Windows i v Linuxu,
- veškeré forenzní distribuce a některé další forenzní nástroje (například SleuthKit v Autopsy) obsahují nástroje pro práci s HPA.

V postupech dále je používán program `hdparm` pro Linux.

Postup (Zjištění a odstranění skryté oblasti HPA)

Předpokládejme, že na počítači máme spuštěnu některou distribuci Linuxu (ideálně forenzní a nabootovali jsme z výměnného média). Dostali jsme disk, který dřív patřil do jiného počítače, chceme zjistit, jestli na něm není HPA. Analyzovaný disk má speciální soubor `/dev/sdb`.

```
hdparm -N /dev/sdb
```

Výstup:
```
/dev/sdb:
max sectors = 78125000/78165360, HPA is enabled
```

To znamená, že HPA na tomto disku opravdu je, velikost této oblasti bychom zjistili odečtením dvou čísel, která jsou na výstupu (40360). HPA nám na tomto disku bude zbytečně zabírat místo, tedy jsme se rozhodli tuto oblast odstranit, aby toto místo bylo využitelné pro běžné oddíly:

```
hdparm -N p78165360 /dev/sdb
```

Výstup:
```
/dev/sdb:
setting max visible sectors to 78165360 (permanent)
max sectors = 78165360/78165360, HPA is disabled
```

Pozor, tento příkaz může být destuktivní pro obsah celého disku!

Poznámka:
Po skrytých oblastech bychom se měli na disku porozhlédnout ještě před tím, než začneme vytvářet obraz disku (pokud chceme vytvořit obraz *celého disku*).

DCO (Defice Configuration Overlay). Možnost vytvořit DCO byla do standardu ATA přidána ve verzi 6.

Jestliže je na disku pouze DCO, forenzní nástroje s ní obvykle nemají problém, vpodstatě má význam podobný jako HPA. Horší situace nastává, pokud jsou na disku obě tyto oblasti – pak bývá nejdřív HPA a pak DCO, jak vidíme na obrázku 2.14. V této situaci mohou mít i profesionální forenzní nástroje problém vytvořit plný obraz disku.

Obrázek 2.14: Struktura disku s oběma typy skrytých oblastí

Postup (Zjištění a odstranění skryté oblasti DCO)
Skrytou oblast DCO zjistíme takto:

```
hdparm -dco-identify /dev/sdb
```

Výstup:
```
/dev/sdb:
DCO Revision: 0x0001
The following features can be selectively disabled via
DCO:
Transfer modes:   ........
```

Skrytou oblast DCO zlikvidujeme tímto příkazem (pozor, taktéž destruktivním):

```
hdparm -dco-restore /dev/sdb
```

Teď se zobrazí varovné hlášení s dotazem, jestli opravdu chceme provést to, co jsme zadali, tedy musíme přitvrdit:

```
hdparm -yes-i-know-what-i-am-doing -dco-restore /dev/sdb
```

Poznámka:
Pokud bychom chtěli ovlivňovat disk přes ATA rozhraní v roli programátorů, můžeme pro ten účel využít API funkci `DeviceIOControl(...)`, informace o parametrech jsou níže.

Další informace:
O skrytých oblastech na disku a programu `hdparm` najdeme informace zde:

- http://www.linuxexpres.cz/praxe/nastaveni-disku-hdparm
- http://blog.asiantuntijakaveri.fi/2012/07/restoring-full-capacity-of-sata-disk-on.html
- http://forensicswiki.org/wiki/DCO_and_HPA

O ATA příkazech pro komunikaci s ovladačem disku:

- http://www.t13.org/documents/uploadeddocuments/docs2006/d1699r3f-ata8-acs.pdf
- http://martin.lipinsky.cz/skola/pz/pz_08_ata-atapi-sata.pdf

2.7.2 Removable Media Bit a výměnná média

Pro připomenutí: výměnné médium je takové paměťové médium, které je přenositelného charakteru. Celkově je v tomto směru tak trochu chaos – zatímco každý bez problémů chápe, že například optický (CD/DVD/BluRay) disk, USB flash disk nebo SD karta je výměnné médium, externí disk výměnným médiem není (alespoň co se týče rozpoznání v operačním systému). Zde se budeme zabývat pouze paměťovými zařízeními připojovanými přes sběrnici USB, tj. externími disky a USB flash disky.

Jak operační systém pozná, že se jedná o výměnné médium? Ve firmwaru každého paměťového zařízení připojovaného přes USB jsou uložena různá data týkající se konfigurace daného zařízení. Je zde i jeden konkrétní bit, který nazýváme RMB – Removable Media Bit. Jedná se o bit 7 prvního oktetu výstupu příkazu *SCSI Inquiry Data Response*, kterým lze komunikovat právě s potenciálně výměnnými zařízeními, včetně USB flash disků. Pokud někoho zmátlo, že v příkazu je zkratka SCSI: se zařízeními připojenými přes USB se komunikuje právě příkazovou sadou SCSI.

V případě, že se jedná o výměnné zařízení, tento bit nastaven na 1, v opačném případě na 0. Ovladač v jádře potřebuje tuto informaci znát, proto si ji (kromě jiného) zjišťuje od jádra žádostí `StorageDeviceProperty`.

Linux se s RMB bitem „pere“ takovým způsobem, že to vpodstatě uživatele nijak neomezuje, ve Windows je však situace poněkud jiná. Pokud Windows zjistí, že se jedná o výměnné médium, takto nám komplikují situaci:

- standardně na takovém médiu „odmítají vidět“ více než jeden oddíl,
- výměnné médium nelze označit offline ani připojit pouze pro čtení, pokud vysloveně nepoužijeme sofistikovaný forenzní nástroj či hardwarový Write Blocker,
- když chceme vytvořit obraz, pak pouze oddílu, nikoliv celého disku.

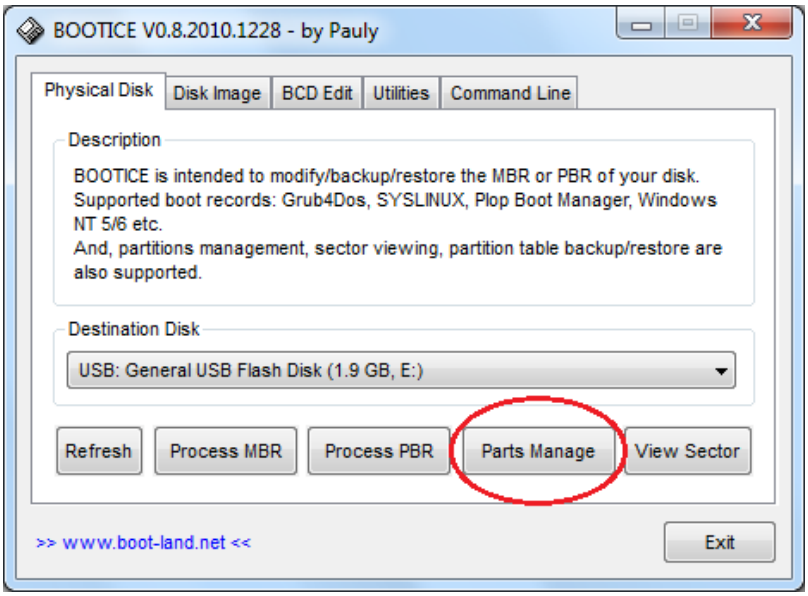
Změna hodnoty RMB bitu není bezpečná, můžeme tím zařízení poškodit (v každém případě by byla pravděpodobně poškozena data), navíc není řečeno, že postup změny bude opravdu fungovat – záleží na konkrétním firmwaru.

Další informace:

- http://msdn.microsoft.com/en-us/windows/hardware/gg487419.aspx
- http://www.t10.org/
- http://msdn.microsoft.com/en-gb/library/ff566971%28VS.85%29.aspx
- http://ubuntuforums.org/showthread.php?t=1020293
- http://www.computrogeek.com/2013/06/lexar-usb-formatting-tool-download-2013.html
- http://www.seagate.com/staticfiles/support/disc/manuals/scsi/100293068a.pdf

Protože v Linuxu lze s USB flash disky zacházet prakticky stejně jako s pevnými disky (včetně dělení na oddíly), budeme se dále věnovat pouze postupům ve Windows.

Rozhodně je lepší tento problém spíše obejít. Určité řešení spočívá v použití nástroje *Bootice*. Tento nástroj sice nemění RMB, ale jeho možnosti mohou dostačovat. Bootice slouží především k zálohování nebo modifikaci MBR (kde kromě jiného najdeme i rozdělení disku na oddíly) a PBR sektorů na disku, dokáže také rozdělit USB flash disk na oddíly (to přímo ve Windows nejde) a určit, který z takto vytvořených oddílů bude ve Windows běžně viditelný – může to být kterýkoliv, ale pouze jeden.



Obrázek 2.15: Prostředí programu Bootice

Kdybychom chtěli, aby ve Windows bylo na USB flash disku viditelných více oddílů, musíme buď nastavit RMB na 0 nebo sehnat a nainstalovat speciální ovladač fungující jako filtr v jádře.

Další informace:

http://agnipulse.com/2011/11/partition-usb-flash-drive/

Postup

Pokud chceme o USB flash disku zjistit co nejvíc informací – například pro práci s RMB bitem obvykle potřebujeme zjistit identifikátory VID (VendorID) a PID (ProductID) – můžeme použít tyto volně dostupné nástroje:

- *Nirsoft USBDeview* je program pro průzkum instalovaných a/nebo připojených USB flash disků, podrobný popis a odkaz ke stažení najdeme na
  http://www.nirsoft.net/utils/usb_devices_view.html
- *ChipGenius* je aplikace pro průzkum a opravu USB flash disků, použití je popsáno na
  http://usb-fix.blogspot.cz/

2.8 Datové sklady a BigData

2.8.1 Skladování velkých objemů dat

V této sekci se budeme zabývat datovými sklady. Toto téma nepatří přímo do zaměření předmětu, proto se ho dotkneme spíše jen okrajově.

Definice (Big Data, Datový sklad, Data Mining)

Pojem *Big Data* označuje rozsáhlé soubory dat, jejichž uchovávání a zpracování je mimo možnosti běžných prostředků.

Datový sklad (Data Warehouse) je speciální databáze zaměřená na správu velkých objemů dat (Big Data). Může být uchovávána historie (časová dimenze), data se obvykle nemažou ani nepřepisují. Cílem není ani tak samotné skladování dat, ale spíše jejich zpracování, analýza, generování závěrů a poznatků.

Data Mining (dolování dat) je proces analýzy velkých objemů dat za účelem získávání nových užitečných poznatků.

Problematika Big Data a datových skladů je rok od roku důležitější, na toto téma existuje již mnoho článků jak v odborných periodikách, tak i na Internetu. Pozornost se zaměřuje jak na státní správu (stačí zadat heslo „Open Data“), tak i na velké poskytovatele datových služeb, zejména Google, protože právě takové firmy nutně potřebují speciální fyzické i softwarové prostředky pro zvládnutí obrovských objemů dat.

Zaměřme se tedy na datové centrum Googlu. V tomto směru je Google velmi otevřený, na webu můžeme kromě „suchých“ informací najít také fotografie a videa.

Další informace:

Datové centrum Googlu (resp. jeho část) si můžeme prohlédnout na fotografiích a videích, tak i prostřednictvím služby StreetView.

- Hlavní stránka: http://www.google.com/about/datacenters/
- Přehled všech datových center: http://www.google.com/about/datacenters/inside/locations/index.html
- Fotografie: http://www.zive.cz/bleskovky/google-nafotil-sve-datove-centrum-do-street-view/sc-4-a-165935/default.aspx (nejdřív zobrazí areál – doporučuji v přehledovém okýnku vlevo dole zvětšit rozlišení, ať se objeví přístupy do budovy, přes ně se dostanete dovnitř)
- Video: http://www.youtube.com/watch?v=avP5d16wEp0

2.8.2 Jak na Big Data

Když potřebujeme uložit velké množství dat, musíme nejen mít paměťová média v dostatečné kapacitě, ale je třeba zajistit jejich propojení a souhrnnou funkčnost (skladování, organizování, ukládání, vyhledávání, srovnávání, kombinování, evidenci vlastností a vztahů, atd.).

Souborový systém určuje způsob, jakým jsou organizována data na paměťových médiích. Je to speciální databáze, ve které evidujeme soubory, adresáře/složky, v nichž se soubory nacházejí, vzájemné vazby, přístupová oprávnění, atributy (vlastnosti) souborů, atd., umožňuje také vyhledávání, srovnávání apod. *Propustnost souborového systému* je ukazatel stanovující, jak rychle systém umožňuje provádět operace v této databázi (včetně hledání).

Na každém paměťovém médiu je souborový systém (na běžných pevných discích například NTFS, FAT32, ext3fs apod.), ovšem pro souhrn paměťových médií ve formě datového skladu běžné souborové systémy nestačí – jejich propustnost není dostačující, protože je potřeba distribuovat data a velmi rychle (paralelně) vyhledávat a zjišťovat vztahy mezi daty.

Distribuovaný souborový systém je souborový systém rozprostřený na více fyzických zařízeních (vpodstatě síťový systém), který vytváří transparentní přístupovou vrstvu k těmto fyzickým zařízením. Poskytuje podobné funkce jako jiné souborové systémy plus další specifické funkce, přičemž uživatel není nucen starat se o skutečné umístění dat a manipulaci na nižší úrovni.

GFS (Google File System). Jedná se o distribuovaný souborový systém, který používá Google ve svých datových centrech. Je optimalizován na tyto požadavky:

- zápis není prováděn moc často, když už, tak spíše na konce souborů,
- vyhledávání je prováděno velmi často a je časově kritické,
- bezpečnost a stabilita, integrita dat.

Takže na úrovni souborového systému (a kousek nad ní) máme GFS. Jenže to nestačí, ještě potřebujeme algoritmus, který dokáže efektivně zpracovávat velké objemy dat a framework, který implementuje jak tento algoritmus, tak i přístupový systém k našemu souborovému systému.

MapReduce. MapReduce je programovací model pro paralelní zpracování velkých objemů dat, je dostupný ve formě knihovny pro programování v jazyce C++. Algoritmus funguje takto:

- data, která mají být zpracována, se na serveru rozdělí do malých zvládnutelných celků,
- tyto celky se rozešlou na podřízené uzly, které je zpracují – funkce `map`,
- podřízené uzly pošlou výsledek serveru, ten je sloučí – funkce `reduce`.

Algoritmus se stará o řízení distribuovaných serverů, paralelizaci, komunikaci mezi uzly v síti. Programátor napíše dvě funkce:

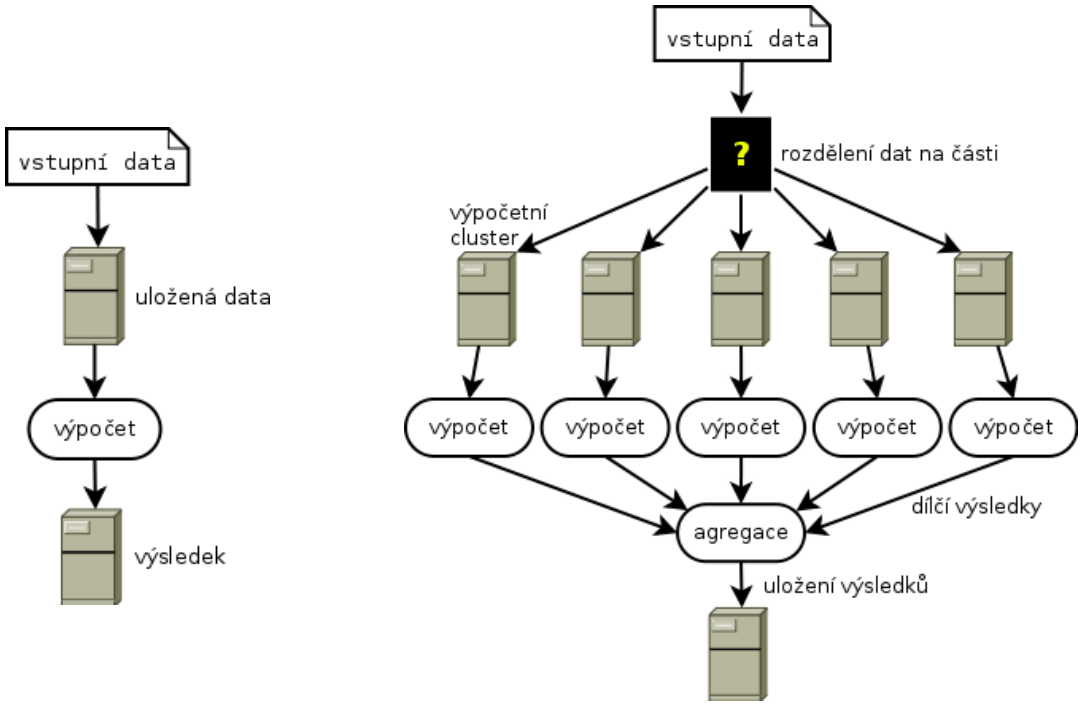
- `map()` – určí, co mají podřízené uzly provést s daty,
- `reduce()` – určí, jak se mají výsledky dát dohromady včetně redukce duplicit.

2.8.3 Hadoop

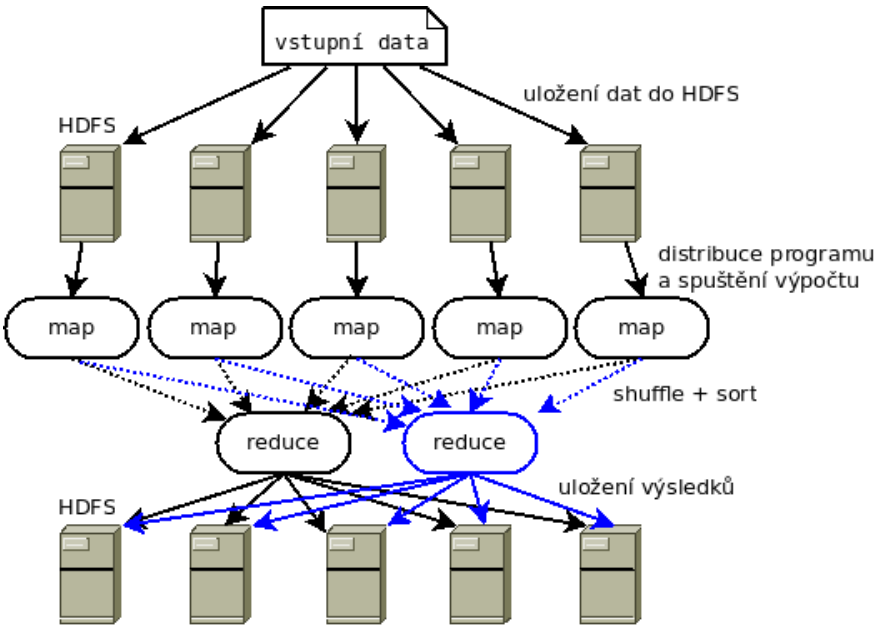
Hadoop je sada nástrojů (framework) pro zpracování velkých objemů *nestrukturovaných* dat s využitím algoritmu MapReduce. Je distribuován pod open-source licencí. Samotný Hadoop je volně dostupný, ale jeho zprovoznění není až tak úplně jednoduché, proto firmy často volí spíše na něm založené komerční produkty – od IBM, Amazonu, Microsoftu a dalších.

Jedná se o sadu nástrojů, mezi kterými dominuje především knihovna *MapReduce* a distribuovaný souborový systém *HDFS* (Hadoop Distributed File System) s cílenou redundancí dat inspirovaný systémem GFS od Googlu, typický tím, že se v něm data zásadně nemažou ani nepřepisují (co je jednou zapsáno, to tam už zůstane). Do sady také patří například NoSQL systémy

Cassandra a *HBase*, *Sqoop* pro efektivní přenos dat mezi Hadoop a datovými sklady, speciální jazyky *Pig* a *Cascading* a další.



Obrázek 2.16: Proces zpracování dat klasický (vlevo) a distribuovaný (vpravo) v běžném systému[2]



Obrázek 2.17: Proces zpracování dat v systému Hadoop[2]

Hadoop se používá tehdy, když je třeba v krátkém čase zpracovávat obrovské množství dat, přičemž již zapsaná data se nemění a nová data jsou přidávána. Na obrázku 2.16 vidíme, jak proces

[2]Zdroj: http://fulltext.sblog.cz/2012/01/16/hadoop-1-kam-s-nimi/

zpracování vypadá, pokud nevyužijeme Hadoop ani žádné podobné řešení – buď výpočet probíhá na jediném stroji (což je neefektivní), nebo sice výpočet rozložíme mezi víc strojů, ale je třeba „nějak“ vyřešit rozdělení práce mezi tyto stroje a po výpočtu provést agregaci (sloučení) výsledků, aby bylo možné tyto výsledky uložit.

Na obrázku 2.17 je naznačeno, jak takový výpočet probíhá v Hadoop. Distribuce není součástí výpočtu, nenásilně proběhne vlastně už při ukládání dat do HDFS, výpočet (určený funkcí `map`) pak probíhá na těch zařízeních, kde jsou data reálně uložena (omezuje se transport dat). Následně proběhne sloučení výsledků (určené funkcí `reduce`), které nemusí nutně znamenat celkovou agregaci – slučuje se jen to, co bude uloženo na tomtéž místě, výsledek je uložen opět do distribuovaného úložiště. Celkově to znamená úsporu času a dalších nákladů spojených s transportem dat, částečně i s distribucí výpočtu a sloučením výsledků.

Systém Hadoop v současné době používá mnoho velkých společností – například Facebook či Yahoo, firmy se s ním setkávají i v cloud řešeních (například Amazon Web Services, Microsoft Azure, Google Compute Engine).

Jak si Hadoop pořídit? Sice je možné stáhnout všechny potřebné komponenty na webu a vše si vlastními silami nainstalovat a zprovoznit (vše je volně dostupné), ale nasazení tohoto systému je vcelku náročné – technicky i časově. Proto většina společností volí tzv. *Hadoop distribuce*, které jsou doplněny o různé další nástroje a včetně služby nasazení. Například:

- *Apache Hadoop* – podpora různých OS, obsahuje Zookeeper, Hive, Pig, atd. – je vybaven dalšími nástroji z projektů Apache, vyznačuje se vysokou dostupností (to znamená, že odstavení jakékoliv součásti nesmí způsobit nefunkčnost celku), je volně dostupný na http://hadoop.apache.org/,
- *Cloudera Hadoop* – komerční distribuce (zřejmě nejpoužívanější), která se hodně soustředí také na podporu a školení zákazníků, v základním open-source softwaru provádí proprietární změny, mezi jejími zákazníky je například také armáda USA,
- *Amazon Web Services Elastic MapReduce* – výhodou je především propracovaná implementace do cloudu
- *IBM InfoSphere BigInsights* – podpora operačních systémů RHEL (RedHat Enterprise Linux), SLES (SUSE Linux Enterprise Server), trochu jiná skladba komponent, podpora POSIX přístupu pro HDFS, vysoká dostupnost, základní edice je ke stažení,
- *Microsoft HDInsight Server* – podpora pouze Windows, nejmenší množství nástrojů, nezajišťuje vysokou dostupnost,
- cloudové služby různých poskytovatelů.

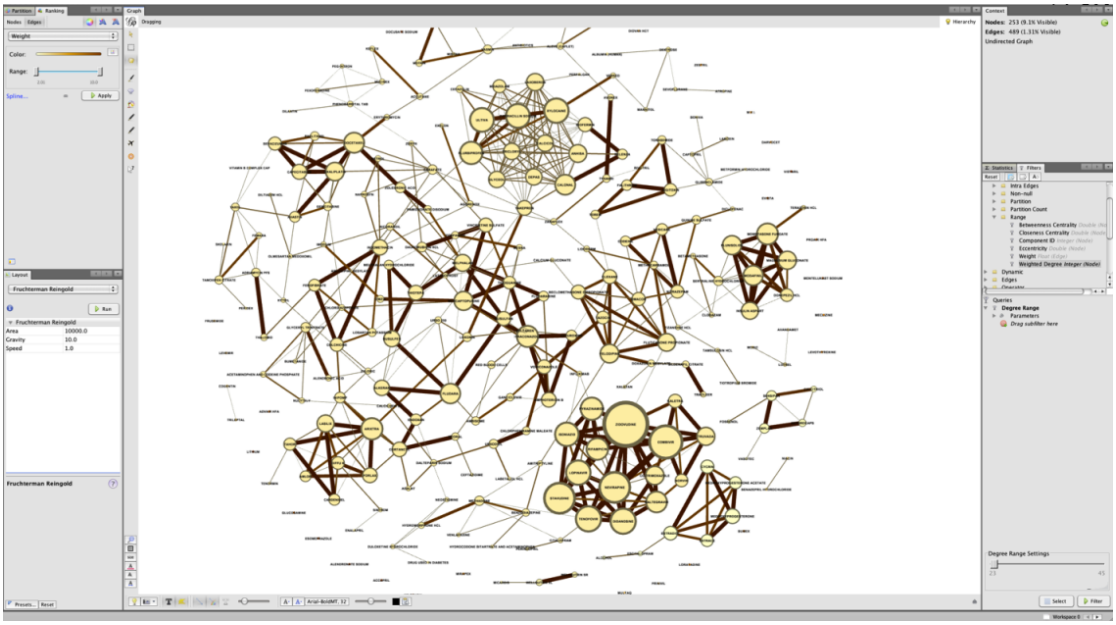
Další informace:

- http://www.linuxexpres.cz/software/kdyz-se-rekne-hadoop
- https://www.predictiveanalyticstoday.com/top-free-apache-hadoop-distributions-hadoop-appliance-hadoop-managed-services/
- https://hadoopilluminated.com/hadoop_illuminated/Distributions.html
- http://hadoop.apache.org/
- http://www-01.ibm.com/software/data/infosphere/hadoop/

- http://fulltext.sblog.cz/2012/01/16/hadoop-1-kam-s-nimi/
- http://fulltext.sblog.cz/2012/01/23/hadoop-2-zakladni-soucasti-souborovy-system/
- http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Pro lepší představu si ukážeme, jak konkrétně se Hadoop využívá v praxi při řešení úkolu zpracování reportů o negativních účincích v případě kombinace různých léků.

Velkým problémem současného zdravotnictví je velké množství existujících léků včetně duplicit používaných pro totéž onemocnění, přičemž existuje obrovské množství jejich možných kombinací. Určité kombinace mohou být nebezpečné. Pro určení a vizualizaci vztahů mezi léky byl použit systém Hadoop v kombinaci s vizualizačním nástrojem. Celý projekt je podrobně popsán na webu http://blog.cloudera.com/blog/2011/11/using-hadoop-to-analyze-adverse-drug-events/.

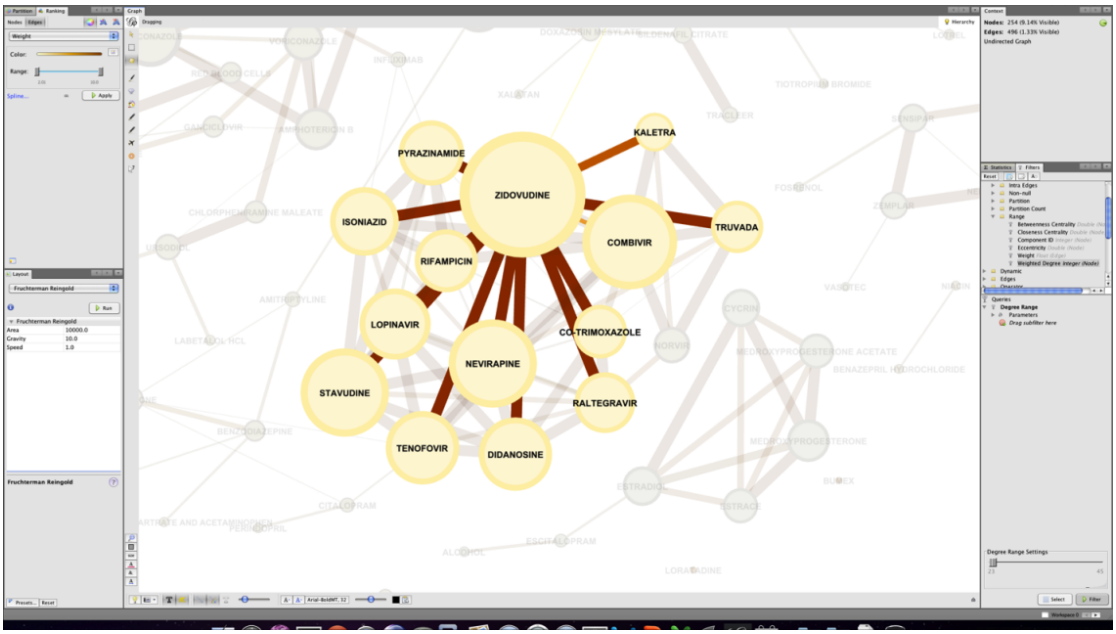


Obrázek 2.18: Hadoop – schéma vztahů se všemi léky[3]

Souhrn kapitoly

V této kapitole jsme se naučili, jak v různých operačních systémech zacházet s obrazy disku či oddílu na disku, ať už jde o bezpečné získání obrazu, jeho připojení, analýzu a ověření kontrolního součtu. Podívali jsme se taky na různé překážky analýzy pevných disků ve formě skrytých oddílů a RM bitu. Důležitou součástí kapitoly byly sekce o specializovaných nástrojích, ať už pro instalaci do operačních systémů (například Autopsy), nebo kompletních distribucí. Na konci jsme se zaměřili na Big data a nástroje pro jejich zpracování, zejména Hadoop distribuce.

[3] Zdroj: http://blog.cloudera.com/blog/2011/11/using-hadoop-to-analyze-adverse-drug-events/



Obrázek 2.19: Hadoop – vybrány pouze léky související s HIV[3]

Kapitola 3

Ochrana dat

Rychlý náhled: Navážeme na předchozí kapitolu a v možnostech získávání a analýzy dat „překročíme hranice“ paměťových médií. Podíváme se na to, jak se obchoduje s daty, kde například mohou data unikat, a také jak si ověřit, zda nám někdo neodcizil identitu, nebo jak co nevíce snížit rizika spojená s odcizením či ztrátou zařízení. V kapitole se také zaměříme na různé druhy cookies a jejich vlastnosti, a taky jejich nástupce FLoC.

Klíčová slova: Leads, fullz, PRISM, licence, smluvní podmínky, sociální sítě, cookie, FLoC, tracking cookie, SQLite, supercookie, evercookie, anonymní režim, technologie anti-theft.

Cíle studia: Cílem této kapitoly je seznámit se s problematikou ochrany osobních údajů, ať už se jedná o data určující vlastní identitu, nebo o další data důležitá z osobních i pracovních důvodů.

3.1 Obchod s daty

3.1.1 Kdo a proč chce uživatelská data

Běžnou reakcí běžného uživatele je: Kdo by se chtěl nabourat do mého počítače? Koho zajímá můj počítač a můj router? Koho by zajímala moje data? No, koho asi:

Úřady a zpravodajské služby se zajímají o:

- osobní údaje, kontakty (hlavně na podezřelé či „chronicky známé“ osoby), zájmy (hlavně ty „závadnější“), politická příslušnost a politické ambice, údaje v kalendáři, instalované aktualizace systému a aplikací a nastavení bezpečnostních produktů včetně firewallu (přehled o zranitelných místech systému), platební neschopnost, stěhování, IP adresa, poloha, hovory, navštívené weby, fotografie, videa, pojmy pro vyhledávání, instalovaný SW, licence, co nakupujeme, ...

Webové služby (Facebook, Google, Amazon a spol.) *a obchody* se zajímají o:

- mail, telefon, IP adresa, zájmy, co nakupujeme, zdravotní stav, poloha, jazyk, časové pásmo, kontakty, přátelé, fotografie, videa, údaje v kalendáři, navštívené weby, cookies, pojmy pro vyhledávání, instalovaný SW (hlavně webový prohlížeč), ...

Hackeři včetně náhodných zájemců se zajímají o:

- mail, telefon, IP adresa či jiné identifikátory, přihlašovací údaje kamkoliv (běžný uživatel používá totéž heslo pro různé účely), momentální poloha, číslo platební karty, zájmy, údaje v kalendáři, příbuzní a přátelé a údaje o nich (často se dá použít v heslech nebo jako odpovědi na kontrolní otázky), navštívené weby, cookies, pojmy pro vyhledávání, instalovaný SW, jakákoliv data, která mohou mít hodnotu (fotografie, videa, dokumenty apod.) osobní i pracovní, instalované aktualizace systému a aplikací a nastavení bezpečnostních produktů včetně firewallu (přehled o zranitelných místech systému), ...

To není ani zdaleka vyčerpávající výčet – záleží, o co konkrétně dotyčnému zvědavci jde. Účelem intervence v cizím počítači může být také třeba využití jeho zdrojů (místa na disku, procesorového času, přenosového pásma v síti apod.) pro vlastní účely, nebo přímo krádež konkrétních dat.

Ať už jde o jakéhokoliv zájemce, možnosti využití uživatelských dat se dají shrnout do těchto kategorií:

- personalizovaná reklama na webových stránkách (může být chápána i pozitivně),
- obchody: personalizované nabídky „co ještě koupit“, „ostatní k tomuto produktu obvykle kupují“, účelem je přesvědčit zákazníka k doplnění dalšího zboží do košíku, oficiálním důvodem je pomoc při výběru zboží,
- získávání dat za účelem jejich kompletace a dalšího prodeje, vytváření a používání *profilů uživatelů*,
- policie: vytipování potenciálních teroristů nebo ohrožených míst,
- malware: finance, vydírání (ransomware), zneužití certifikátů a hesel, čísel kreditních karet, osobních informací, špionáž, adresný spam, adresný malware (phishing apod.),
- atd.

Práva a povinnosti subjektů, které mohou získávat a využívat osobní údaje jiných osob, určuje *Zákon o zpracování osobních údajů*, a samozřejmě direktiva GDPR – o obou byla řeč v první kapitole. V Zákoně o zpracování osobních údajů jsou definovány pojmy osobní údaj, citlivý údaj, shromažďování a uchovávání osobních údajů, jejich likvidace, správce a zpracovatel, příjemce, atd., a dále kdo jakým způsobem které údaje může získávat, uchovávat a zpracovávat – práva a povinnosti zúčastněných subjektů. Většina správců údajů má povinnost oznámit Úřadu pro ochranu osobních údajů, jaká osobní data za jakým účelem shromažďují a jaká opatření provedli k jejich ochraně. Zákon striktně určuje, jakým způsobem může správce s daty nakládat, k čemu je potřeba získat souhlas subjektu (osoby) a o čem je správce povinen subjekt informovat.

Tolik o právu a teorii, praxe však bývá trochu jiná. Mnohé firmy naprosto nemají problém se získanými osobními údaji obchodovat, a když se zákazník (subjekt apod.) zeptá, co se děje s jeho osobními daty, často se mu dostane odpovědi typu

- „Data jsou určena pro naši vlastní potřebu, poskytujeme je pouze důvěryhodným spolupracujícím firmám.“
- „Data shromažďujeme za účelem zlepšení kvality našich služeb.“

Na co máme právo? Teoreticky můžeme:

- Můžeme si vyžádat vyřazení z databáze. Ale kontrolu toho, zda to firma provedla, obvykle nemáme.

- Můžeme požádat o seznam subjektů, kterým byla naše osobní data poskytnuta, ale nemusíme ho dostat (zvláště když jde o firmu se sídlem v zahraničí).

Poznámka:
V oblasti ochrany osobních údajů vždy platí zákony té země, ve které má firma sídlo. Opatrný člověk, po kterém chce někdo osobní údaje, si tedy vždy předem zjistí, kde má dotyčný subjekt sídlo.

Ještě nedávno platilo, že když chtěl člověk něco nakoupit v internetovém obchodě, byl donucen k registraci, a v rámci registrace byl podroben výslechu s rozsahem otázek téměř odpovídajícím zájmům předlistopadové StB (pro mladší ročníky: Státní bezpečnost, tajná služba ČSSR). Dnes se sice ještě pořád najdou firmy, které se snaží ze zákazníka dostat co nejvíc „zajímavých“ informací (alespoň v nepovinných položkách, přičemž počítají s tím, že si sem tam někdo nevšimne, že je položka nepovinná), ale v rámci konkurenčního boje už tyto praktiky ustupují.

3.1.2 Jak se s osobními daty obchoduje

Trh s osobními daty je poměrně rozsáhlý a z části také legální.

Definice (Leads, fullz, CVV, dumps)
Leads jsou balíky s osobními daty, typicky jde o marketingově zajímavý obsah. O uživateli je evidováno zejména jméno a příjmení, kontaktní informace (adresa, e-mail, telefon apod.) a další informace, které se buď okamžitě nebo dodatečně podaří o uživateli získat.

Fullz jsou balíky nelegálně získaných osobních dat, především s finančním využitím. Typicky zahrnují osobní informace o uživateli, čísla kreditních karet, číslo CVV z karty, PINy či jiné bankovní informace, přístupové údaje k různým službám včetně platebních nebo sociálních sítí, atd.

Pokud používáme platební kartu pro platbu na Internetu, obvykle zadáváme číslo karty, datum expirace a číslo *CVV* (nebo CVC). Při placení na terminálu nebo komunikaci s bankomatem se místo toho načítají údaje uložené na čipu nebo v magnetickém pruhu karty, čemuž se říká *dumps* a případně se potvrzuje zadáním PINu.

Leads si k danému uživateli typicky vytvářejí internetové obchody, sociální sítě, telekomunikační operátoři a jiné firmy při registraci uživatele (údaje z registračního formuláře), údaje jsou pak průběžně doplňovány dalšími informacemi například v průběhu nákupu uživatele či procházení strukturou webových stránek. O těchto údajích platí, že uživatel (zákazník) by měl být o využití svých osobních dat informován ještě před tím, než je poskytne. Jiným způsobem vytvářením leads je jejich generování (leads generation) z nejrůznějších dostupných databází (včetně telefonních seznamů a sociálních sítí).

K odcizení čísla karty, data expirace a čísla CVV/CVC může dojít například při komunikaci s nedůvěryhodným nebo napadeným internetovým obchodem během platby za zboží. Proto se z bezpečnostních důvodů dnes běžně setkáváme s ochranou pomocí platební brány, přičemž se obvykle používá protokol *3D Secure*. 3D Secure znamená zapojení tří dalších subjektů (kromě

plátce a příjemce): vydavatele karty (banky, která kartu vydala), zpracovatelské banky a karetní asociace (Visa, MasterCard apod.). Karetní údaje jsou při použití 3D Secure předávány šifrovaně, přičemž obchodník by se k nim vůbec neměl dostat

Další informace:

- Srovnání platebních bran na českém trhu z roku 2020 najdeme na https://www.fastcentrik.cz/blog/velke-srovnani-10-platebnich-bran a novější uz roku 2021: https://www.gopay.com/blog/velke-srovnani-platebnich-bran/ (ovšem pozor, toto srovnání pochází od jednoho z provozovatelů, tedy není nestranné). Další srovnání z roku 2021 je na https://www.upgates.cz/a/srovnani-platebnich-bran-2020.
- Fungování 3D Secure je popsáno na https://www.gopay.com/blog/jakou-zmenu-v-placeni-prinese-3d-secure-2-0/

Příklad

V mnoha zemích je naprosto legální s leads obchodovat a některé firmy se na to dokonce specializují. Například v USA je to společnost *LeadsPlease*, která zprostředkovává leads s různým složením podle požadavků zákazníka (adresy, koníčky, roční příjem, milovníci zvířat, apod.).

Celosvětově působí například společnost *Acxiom*, která je na trhu už od roku 1969. K zákazníkům Acxiomu patří i Facebook. Z legálních jsou nejdražší medicínská data (data lidí s určitým onemocněním).

Jedním z největších obchodníků s osobními daty je Facebook (tzv. Facebook Lead Ads), ale není ani zdaleka jediný. Taky existují návody a nástroje pro získávání vlastních tzv. sales leads, stačí si na webu vyhledat například heslo „B2B leads“ nebo „sales leads“.

Získávání dat pro reklamní společnosti je výnosné také v mobilním světě. Mnohé mobilní aplikace existují v bezplatné variantě, přičemž buď je uživateli zobrazována reklama, nebo uživatel musí „zaplatit“ určitými informacemi (a mnohdy o tom ani neví), případně obojí. Získaná data jsou shromažďována a dále zpracovávána nebo distribuována. Data mohou být anonymizována (za účelem zjišťování nějakých obecnějších zákonitostí) nebo obsahovat i osobní údaje.

Příklad

Společnost *Adelphic* se zabývá analýzou vzorců chování uživatelů mobilních telefonů, zjišťuje, jak uživatel reagoval na předchozí reklamní nabídky (pro tyto účely vyvinula vlastní platformu). Společnost *Drawbridge* zase provádí statistickou analýzu anonymních údajů, také z cookies. Dokáže dát dohromady různá zařízení, která pravděpodobně patří témuž uživateli, což je užitečné zejména pro cílení reklamních sdělení.

V šedé zóně, v některých zemích už za hranicí zákona (jako u nás), je prodávání leads coby údajů získaných od vlastních zákazníků. Zcela za hranicí zákona je pak prodávání leads získaných nelegálně (tedy fullz), například napadením systému některé firmy či státní instituce.

Některé státní instituce (především tajné služby, vč. NSA) se pokoušejí doplňovat, kompletovat a využívat leads propojováním různých zdrojů a nasazováním speciálních algoritmů, které slouží

k rozpoznávání zájmových či podezřelých osob nebo kritických situací, což může mít i pozitivní následky (dopadení hledaného zločince), ale někdy může způsobit i osobní tragédii. Například Murat Kurnaz byl algoritmy od NSA označen za podezřelého z účasti na teroristických akcích a neprávem uvězněn v Guantanamo Bay.

3.1.3 PRISM alias Velký Bratr

7. června roku 2013 se ve Washington Post a Guardian objevily první informace o zadních vrátkách do různých systémů (společností Microsoft, Yahoo!, Google, Facebook, YouTube, Skype, AOL, Apple), která si zajistily americké služby NSA a FBI. O několik dní později (10. června) tuto informaci všechny jmenované firmy popřely a uvedly (na CNET), že se ve skutečnosti jedná o obrovskou databázi, do které zmíněné společnosti předávají data na základě soudního rozhodnutí. V té době bylo zveřejněno jméno Edward Snowden (externí spolupracovník FBI). 4. července se však podobné informace objevily i v Le Monde s tvrzením, že podobné aktivity provádějí i další zpravodajské agentury. Zprávy vyvolaly znepokojení nejen u běžných občanů, ale i u politiků na celém světě.

19. července byl doručen veřejný dopis mnoha nevládních organizací a internetových společností prezidentu Obamovi s žádostí o zveřejňování informací, této kauze a o žádostech NSA k přístupu k utajeným informacím. Načež 22. července šéf hostingové společnosti XMission prohlásil, že na základě příkazu tajného soudu musela tato společnost dovolit instalaci monitorovacího síťového zařízení.

1. srpna zveřejnil Edward Snowden informaci o projektu XKeyscore – existence stovek serverů po celém světě, které sbírají a třídí informace o uživatelích. Odmítl tvrzení, že sledování probíhá pouze v případech povolených soudem.

9. srpna rozhodl šéf NSA o výměně 90 % lidských administrátorů za speciální software z důvodu nutnosti omezení úniku tajných informací. Obama vysvětluje a obhajuje činnost NSA, navrhuje legislativní řešení. Nicméně Obamův návrh na změny v zákonech jsou kritizovány.

Postupně vyplynuly na povrch další kauzy, například kompromitace mobilního telefonu německé kancléřky A. Merkelové. Americká legislativa týkající se bezpečnosti země a pravomocí NSA a FBI se postupně mění, ale s občasnými hrozbami teroristických útoků spíše směrem k podpoře jejich větších pravomocí.

✎ NSA vytvořila technologii *Accumulo* založenou na GFS od Googlu (viz str. 59). Accumulo slouží k evidování obrovského množství dat (Big Data), jejich spolehlivému indexování a následnému velmi rychlému vyhledávání a kombinování. Mohou být řešeny například tyto úkoly:

- Vytvořte kompletní profil vybraného uživatele.
- Vyhledejte konkrétní klíčová slova v e-mailech pocházejících z určitých IP adres.

Princip získávání a využívání dat je podle NSA odůvodněn tímto výrokem: „Kdo nemá co skrývat, tomu sledování nebude vadit.“ Podobné výroky se dokonce objevují i o šifrování – kdo prý šifruje, má co skrývat. NSA zapomíná, že některé informace je třeba skrývat i z naprosto legálních důvodů (ať už osobních nebo například pro ochranu výrobního či obchodního tajemství firmy, z konkurenčních důvodů apod.). Navíc ani servery NSA a FBI nejsou plně imunní proti činnosti hackerů, což se v minulosti bohužel nejednou ukázalo.

Postup (Ochrana vlastní komunikace)

Co můžeme dělat pro ochranu své vlastní komunikace:

- Šifrovat, ale není řečeno, že to pomůže. Pokud služba, se kterou šifrovaně komunikujeme, „spolupracuje“, ... Údajně má NSA přístup k datům na Outlook.com ještě před jejich zašifrováním.
- Můžeme používat alternativy k produktům „spolupracujících“ společností.
- Neposílat po síti nic, co je soukromé či dokonce tajné. Také cloudové služby mohou „spolupracovat“.

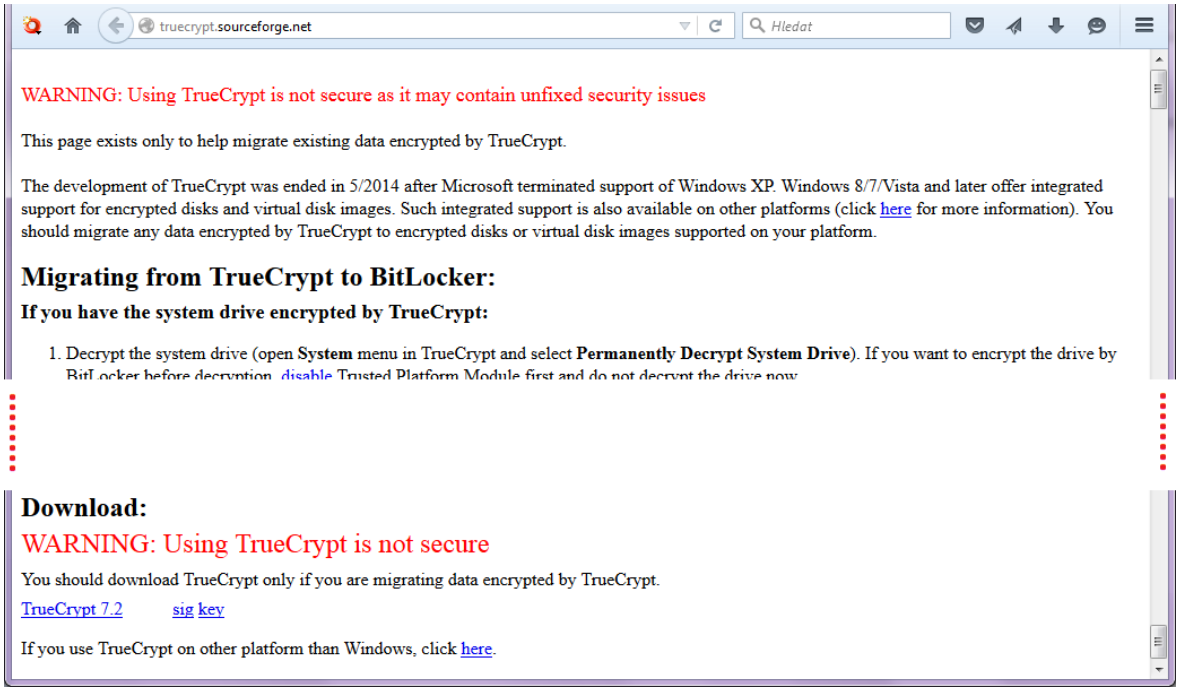
Další informace:

Alternativy k produktům „spolupracujících“ společností: https://prism-break.org/

Tento web se hodí i tehdy, když z jakéhokoliv důvodu nechceme či nemůžeme používat určitý produkt a hledáme vhodnou náhradu. Ovšem vhodnost je relativní, záleží, co od produktu očekáváme.

Pokud šifrovat, jaký nástroj pro tento účel použít? Šifrováním se budeme zabývat v následujících kapitolách, zde jen upozorníme na problém s šifrováním související.

Jedním z nejoblíbenějších (navíc volně šiřitelných) nástrojů na šifrování disků byl po léta *TrueCrypt*. Uživatelé si ho mohli stáhnout i se zdrojovým kódem ve variantě pro svou platformu včetně klíče na ověření integrity a digitálního podpisu, také z toho důvodu byl považován za bezpečný a důvěryhodný. Je také zajímavé, že dodnes nejsou známi autoři tohoto softwaru.



Obrázek 3.1: Varování na webu projektu TrueCrypt

Jenže na webu projektu se najednou objevila zpráva „WARNING: Using TrueCrypt is not

secure as it may contain unfixed security issues“ (od května 2014), přičemž je zde doporučen přechod na jiné nástroje (na Windows jde o BitLocker, který však na některých edicích Windows není dostupný) – viz obrázek 3.1 na straně 69. Autoři v anonymním e-mailu jako odpovědi na veřejný dotaz dokonce ani nedoporučují vytváření forku (odvozené varianty) původního TrueCryptu, údajně je bezpečnější začít od podlahy při současném studiu kódu TrueCryptu.

Okamžitě se vynořila řada spekulací o tom, co se vlastně stalo. Je podivné, že předně k něčemu takovému vůbec došlo (zdrojový kód je zveřejněný, tedy pod neustálou kontrolou, softwarový problém nebyl nikým nalezen), autoři zůstali i nadále neznámí, pro Windows je jako nástupce doporučen ne zcela vhodný produkt (proprietární, s uzavřeným zdrojovým kódem, navíc pro běžné uživatele s „home“ edicemi není dostupný), na webu je vcelku nesmyslná zmínka o souvislosti s ukončením podpory Windows XP (přitom TrueCrypt není zdaleka určen jen pro Windows do verze XP). Web projektu byl analyzován a prohlášení včetně odkazu na poslední verzi TrueCrypt určenou jen pro dešifrování bylo prohlášeno za pravé. Vlastně ani není možné skutečný stav věci zjistit, protože nikdo neví, na koho se obrátit (autoři jsou neznámí). I kdyby se někdo prohlásil za autora TrueCrypt, bude mít problém dokázat, že mluví (píše) pravdu.

Za vcelku pravděpodobnou se pokládá domněnka, že na autory byl činěn nátlak ze strany některé tajné služby a autoři raději ukončili projekt, než by „zradili“ své uživatele.

Další informace:

- http://www.lupa.cz/clanky/truecrypt-konci-za-ponekud-nejasnych-okolnosti/
- http://krebsonsecurity.com/2014/05/true-goodbye-using-truecrypt-is-not-secure/
- http://www.hacker10.com/encryption-software-2/list-of-truecrypt-compatible-encryption-software/
- http://www.root.cz/zpravicky/vyvojar-truecryptu-nedoporucuje-jeho-forkovani/

3.2 Smluvní podmínky – s čím souhlasíme

Naprosto běžnou situací při instalaci nového softwaru či registraci k určité službě či v internetovém obchodě je odsouhlasení smluvních (příp. licenčních) podmínek. Běžný uživatel jednoduše klepne na OK, Souhlasím, Agree, apod., a vlastně ten text ani nečte. Ve většině případů takové jednání nemívá negativní následky, ale, jak se říká, výjimka potvrzuje pravidlo.

Některé případy jsou spíše úsměvné, nicméně stojí za zamyšlení:

Příklad

Společnost GameStation přidala do svých licenčních podmínek pasáž o tom, že uživatel souhlasí s prodejem své duše. Uživatelům naprosto nevadilo tyto podmínky odsouhlasit (po upozornění byli velmi překvapeni).

V jiných případech mělo k zamyšlení rozhodně dojít ještě před potvrzením:

Příklad

Nejmenovaný slovenský web s freewarem ke stažení podmínil stažení softwaru registrací. Ve smluv-

ních podmínkách, které bylo třeba během registrace odsouhlasit, stálo, že se uživatel zavazuje provozovateli webu platit 60 eur měsíčně po dobu 2 let, odstoupení od smlouvy bylo také finančně vázáno. Uživatelé byli pak po uběhnutí určité doby od registrace důrazně upozorněni, že mají zaplatit – část uživatelů radši fakturu zaplatila, zbytek se dodnes soudí.

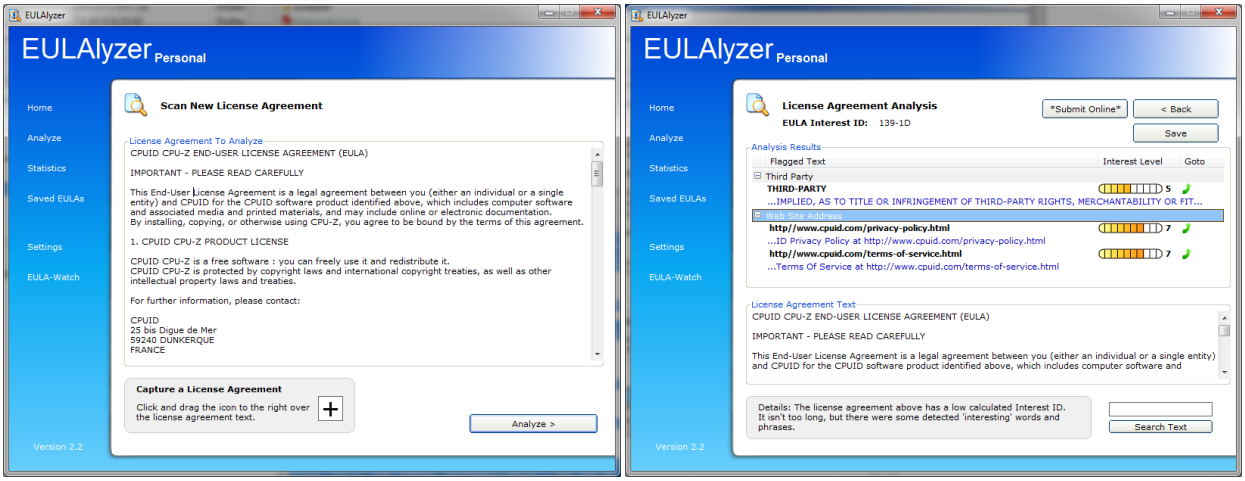
Z toho vyplývá, že když s něčím máme souhlasit, pak bychom měli především vědět, s čím souhlasíme – prostě si to přečíst, ať už se jedná o souhlas s používáním osobních údajů (se smluvními podmínkami) při registraci k službě či v internetovém obchodě nebo o souhlas s licenčními podmínkami při instalaci nového softwaru.

Mnoho softwaru, zejména pro Windows, je distribuováno pod licencí EULA (End-User License Agreement). Licence EULA se vyznačuje nejen tím, že se používá pro proprietální (nesvobodný) software, jehož používání je vždy nějakým způsobem omezováno (minimálně tím, že se nemůžeme prohrabovat ve zdrojových kódech, případně nemožností použít pro komerční účely či dále distribuovat), ale také tím, že je poměrně pružná. Producent softwaru, který tuto licenci zvolí, si může přidávat vlastní podmínky používání dotyčného softwaru, a právě v těch může být problém.

EULAlyzer je volně šiřitelný software, který v licencích typu EULA hledá „háčky“. Je třeba zkopírovat text licence (pokud se dostaneme k jeho kopírovatelné verzi) a pak vložit do příslušného okna programu. Problém může nastat tehdy, když je licence ve špatně kopírovatelném formátu nebo je textu příliš hodně, což je například u produktů společnosti Adobe – tam jsou licenční podmínky v PDF souboru o 467 stranách, z toho 16 stran je anglicky.

Další informace:

EULAlyzer je dostupný na http://www.brightfort.com/eulalyzer.html.



Obrázek 3.2: EULAlyzer s načtenou licencí CPU-Z a výsledkem analýzy

Na obrázku 3.2 je okno aplikace EULAlyzer s načtenou licencí programu CPU-Z, vedle je pak výsledek analýzy. Je zřejmé, že se především v textu hledají zmínky na „third-party“ práva, software apod. (tedy na cokoliv, co se odkazuje na někoho dalšího), a dále adresy na webu, na kterých by případně mohly být vypsány další podmínky používání. Program CPU-Z „prošel“;

něco sice bylo nalezeno, ale nic závadného.

V každém případě bychom si měli dávat pozor na to, co odsouhlasíme. V případě softwaru pak bychom navíc měli dávat pozor na to, co konkrétně je nám dovoleno s dotyčným produktem provádět.

knihy.nic.cz, Cyberkrime, str. 176 tisknutá

3.3 Sociální sítě a vyhledávače

3.3.1 Facebook

Jednou z nejpoužívanějších sociálních sítí je Facebook, o kterém je známo (ani Mark Zuckerberg se tím netají), že je horlivým sběratelem dat o uživatelích (a nejen o nich). Používání Facebooku je sice zdarma, ale ve skutečnosti uživatel platí svými daty a povolením zobrazování reklam.

Poznámka:

Už před časem proběhla médii zpráva, že Facebook vstoupil na burzu, a to velmi úspěšně. V každém případě se jedná o firmu, která musí vydělávat, aby mohla fungovat, a bez toho by ani vstup na burzu nebyl možný. Až 88 % obratu Facebooku pochází z prodeje reklamy (z toho čtvrtina z reklam na mobilních zařízeních), zbytek z prodeje aplikací (30 % si nechává Facebook, zbytek vývojáři) – údaj z Chipu 10/2014. Facebook nejen získává informace přímo od svých uživatelů (Facebooku, Instagramu, WhatsApp, Oculus atd.), ale navíc je nakupuje od firem specializovaných na získávání leads, tato data kombinuje a kromě jiného je využívá k cílení reklamních sdělení.

Rozhraní a struktura menu se na Facebooku sem tam mění, ale obvykle bychom někde vpravo nahoře měli najít (třeba i zanořeně) volby jako *Nastavení účtu* a *Nastavení soukromí*. Tam někde se dostaneme k důležitým volbám a informacím.

Také zde by se mnohý uživatel velmi divil, kdyby si pořádně přečetl *Podmínky použití*, se kterými souhlasil, když si na Facebooku vytvářel účet. K těmto podmínkám se dostaneme kdykoliv – přes odkaz *Podmínky použití* na dolním konci webové stránky. Pár zajímavých bodů – citujeme z https://www.facebook.com/legal/terms, údaje jsou z roku 2018:

- *„Jste vlastníkem veškerého obsahu a informací, které na Facebooku zveřejníte, a pomocí nastavení soukromí a aplikací můžete určit, jakým způsobem budou sdíleny.“*
  Pozn.: Tím je dána i odpovědnost uživatele za vkládaná data (pokud uživatel zveřejní něco, co je vlastnictvím někoho jiného, není to problém Facebooku).
  Uživatel by měl také vědět, že výchozí nastavení soukromí je „veřejné“.
- *„K obsahu chráněnému právy k duševnímu vlastnictví, jako jsou fotografie a videa (... DV), nám výslovně udělujete následující oprávnění, v souladu s vaším nastavením soukromí a nastavením aplikací: udělujete nám nevýhradní, přenosnou, převoditelnou, celosvětovou bezúplatnou (royalty-free) licenci na použití veškerého obsahu podléhajícího DV, který zveřejníte na Facebooku nebo v návaznosti na něj (Licence k DV).“*
  Pozn.: Všimněte si, co vše uživatel Facebooku dovoluje v manipulaci se svým obsahem. Udělená licence je dokonce přenosná a převoditelná, tedy Facebook může (nemusí) s uživatelovými fotografiemi dokonce obchodovat.

- *„Jestliže obsah podléhající DV odstraníte, bude odstraněn obdobným způsobem, jako při přesunutí do Koše v počítači. Berete však na vědomí, že odebraný obsah může existovat v záložních kopiích po přiměřeně dlouhou dobu (nebude však dostupný ostatním).“*
  Pozn.: Jinými slovy – ve skutečnosti není smazáno nic, oficiálně pro případ, že by uživatel chtěl obsah obnovit.
  Co se týče dostupnosti smazaného obsahu – pod pojmem „ostatní“ není zahrnut Facebook.
- *„... jak vaše ohlasy, tak i vaše návrhy můžeme použít bez nároku na honorář pro vás (stejně jako vy nemáte povinnost nám je poskytovat).“*
  Pozn.: Na tomto ustanovení je zajímavý ten tak trochu alibistický dodatek.
- *„Při shromažďování informací od uživatelů jste povinni dodržovat následující pravidla: musíte si vyžádat jejich souhlas, vysvětlit, že jejich informace shromažďujete vy (ne Facebook) a zveřejnit zásady ochrany osobních údajů, kde bude uvedeno, jaké informace shromažďujete a k jakému účelu budou použity.“*
  Pozn.: Pokud někdo přes Facebook shromažďuje data, Facebook jen stojí stranou a za nic neodpovídá (nicméně shromážděná data vlastně taky může využít).
- *„Povolujete nám použít vaše jméno, profilovou fotku, obsah a informace ve spojení s komerčním, sponzorovaným nebo souvisejícím obsahem (...), který poskytujeme nebo zprostředkováváme. To třeba znamená, že nás tímto opravňujete, abychom mohli za poplatek firmám nebo jiným subjektům zobrazovat vaše jméno nebo profilový obrázek s vaším obsahem nebo informacemi, aniž bychom vás museli jakkoli finančně odškodnit.“*
  Pozn.: Zde Facebook dává příklad, jak chce využívat licenci udělenou podle druhé odrážky tohoto seznamu.
- *„Neposkytneme váš obsah ani informace inzerentům bez vašeho souhlasu.“*
  Pozn.: Ve výchozím nastavení souhlas udělujete. Pokud nechcete, přenastavte si soukromí.
- *„Pokud si stáhnete nebo použijete náš software (například samostatný softwarový produkt, aplikaci nebo modul plug-in do prohlížeče), jste srozuměni s tím, že tento software může od nás stahovat a instalovat upgrady, aktualizace a další funkce za účelem zlepšení, zdokonalení a dalšího vývoje softwaru.“*
  Pozn.: Bylo by zajímavé zjistit, co je míněno těmi „dalšími funkcemi“.
- *„Souhlasíte s přenesením a zpracováním svých osobních údajů ve Spojených státech amerických.“*
  Pozn.: Tady pozor, nelze dávat na Facebook údaje, u kterých nechceme (nebo nesmíme, třeba z legislativních důvodů) dopustit, aby opustily území Evropské unie.
- *„Vyhrazujeme si všechna práva mimo práv, která vám výslovně přísluší.“*
  Pozn.: Jinými slovy – co Facebook nemá výslovně zakázáno, to mu dovolujeme.

Ustanovení v podmínkách je více, také je zajímavé, že existují i zvláštní ustanovení platná pouze pro občany Německa.

Úkol

Pokud máte u Facebooku účet, projděte si volby nastavení soukromí. Zkontrolujte také seznam zařízení, ze kterých probíhá přihlašování, taktéž najděte nastavení výstrah při nerozpoznaném

přihlášení. Dále se pokuste najít místo, kde je možné stáhnout si soubor s vlastními daty, která o vás Facebook eviduje.

Další informace:

- http://www.tyinternety.cz/2011/10/03/clanek/co-vsechno-o-vas-facebook-vi-a-jak-to-z-nej-dostat/
- http://europe-v-facebook.org/EN/Data_Pool/data_pool.html (souhrn dat o sobě, která lze od Facebooku získat, když člověk ví jak)

Poznámka:

Měli bychom si uvědomit, že o sobě hodně prozrazujeme (jak samotnému Facebooku, tak i těm, kdo na něm vyhledávají) tím, jak klepeme na „Like“ (To se mi líbí), co sdílíme apod. Další případ, kdy je uživatel zdrojem informací jak pro Facebook, tak i pro příslušnou webovou službu, je využívání možnosti „přihlásit se přes Facebookový účet“.

Informace je možné z Facebooku dostat i externími nástroji. Zajímavým řešením je použití nástroje Wolfram Alpha Personal Analytics for Facebook (http://www.wolframalpha.com/facebook/) – je třeba buď mít účet u Wolframu nebo si ho při této příležitosti vytvořit. Hlavní obrazovka služby je na obrázku 3.3 (tlačítko *Get Your Report*). Tento nástroj přistupuje k serverům Facebooku přes API, které Facebook zpřístupnil třetím stranám. Nicméně možnosti tohoto API jsou postupně omezovány, v současné době lze přes ně stáhnout již méně informací než ze začátku.

Obrázek 3.3: Wolfram Alpha Personal Analytics for Facebook

3.3.2 Google

Dalším horlivým sběratelem dat je Google. Oficiálním účelem je opět snaha o vylepšení služeb poskytovaných zákazníkům. V médiích bylo propíráno už více kauz, ve kterých Google figuruje právě pro svůj zájem o data až na hraně zákona (například když automobily mapující města pro projekt StreetView monitorovaly provoz ve veřejných Wi-fi sítích, kolem jejichž přístupových bodů projížděly).

Podobně jako u Facebooku, i zde je dobré si důkladně projít možnosti nastavení. Po přihlášení na svůj Google účet se v pravém horním rohu objeví sada prvků pro nastavení a používání účtu, a podobně jako u Facebooku, i zde se průběžně mění názvy i struktura voleb. Obvykle tam bývá položka Můj účet, Spravovat účet Google nebo něco podobného, přes kterou se dostaneme k různým možnostem nastavení včetně ochrany soukromí a nastavení zabezpečení.

Nicméně – Google je v trochu jiné pozici vzhledem ke svým zákazníkům než Facebook, protože nepůsobí přímo v oblasti sociálních sítí (tedy víceméně), ale zato to vyvažuje svými aktivitami na poli vyhledávání a dalších služeb. Při vyhledávání toho na sebe uživatel prozrazuje velmi hodně, a pokud je přihlášen na svém Google účtu, může takto získané údaje Google agregovat do příslušných leads. Ovšem pozor – není úplně pravda, že když zrovna nejsme přihlášeni, tak data nejsou předávána Googlu: data jsou ve skutečnosti shromažďována lokálně a předána až v okamžiku přihlášení.

Úkol

Pokud máte účet u Googlu, přihlaste se a projděte si možnosti nastavení. Najděte také seznam zařízení, přes která teď k účtu přistupujete. Propátrejte ovládací prvky aktivity – jsou užitečnější než by se podle názvu dalo usuzovat.

Přejděte k obrazovce pro vyhledávání (stačí zadat nějaký pojem pro vyhledání). Pokud jste přihlášeni, měla by být (zřejmě vpravo nahoře) ikona pro nastavení. Projděte si položky, které v zobrazeném menu najdete, včetně vašich dat ve vyhledávání. Všimněte si, kde je nastavena personalizace vyhledávání (a zda ji máte zapnutou).

3.3.3 Jiné sociální sítě

Všechny větší sociální sítě (někde) informují své uživatele o jejich právech a povinnostech, nabízejí možnost konfigurace soukromí a bezpečnosti a také umožňují udělat si přehled o datech, která o uživateli shromažďují, případně si je přímo stáhnout. Tyto možnosti najdeme jak na LinkedIn, Twitteru, Instagramu, atd., stačí jen se v příslušných nastaveních trochu „prohrabat“.

3.3.4 Nástroje k řízení sociálních sítí

Existují nástroje, které slouží k hromadnému řízení sociálních sítí, případně k hromadnému publikování příspěvků na různých sociálních sítích nebo upozorňování na nové příspěvky publikované sledovanými stránkami. Ať už se jedná o jakýkoliv nástroj (aplikaci, mobilní aplikaci, webovou aplikaci běžící ve webovém prohlížeči), vždy je třeba tento nástroj propojit s příslušnou sociální sítí, tedy poskytnout své přihlašovací údaje (tudíž stojí za úvahu, jestli k tomu máme odvahu).

Ke správě publikovaných dat a sledování obsahu slouží aplikace či služba *My Permissions*. Umožňuje konfigurovat to, co síť „může“, včetně určení notifikací e-mailem. Dokáže se napojit na různé sociální sítě, přes Facebook, Twitter, Google+, Účet Microsoft přes Foursquare až po Dropbox.

Zveřejňované příspěvky na různých sociálních sítích lze spravovat pomocí různých dalších aplikací či služeb, jde spíše o marketingové nástroje než o produkty ke správě osobních dat. Známé jsou například Hootsuite, Buffer, Kontentino, Later nebo ZoomSphere.

Další informace:

- http://mypermissions.org/
- https://www.domena.cz/blog/sikovne-nastroje-na-socialni-site/

3.4 Webové prohlížeče

3.4.1 Cookies

Cookie („sušenka“) je ve světě počítačových sítí soubor nebo jinak reprezentovaný malý kousek dat, ve kterém je uložena konkrétní informace o uživateli, je uložen na počítači tohoto uživatele. Cookie je na počítač uložen na žádost serveru, se kterým uživatel komunikuje, během komunikace se odesílá serveru při žádostech o konkrétní stránku na serveru uloženou. Podpora cookies se poprvé objevila v prohlížeči Netscape Navigator v 90. letech.

Cookies se obvykle používají právě při komunikaci s webovými servery přes protokol `http` nebo některý odvozený. Komunikace pomocí tohoto protokolu je *bezstavová*, což znamená, že při vyřizování uživatelova požadavku (na zobrazení určité stránky z webu) server netuší, zda už předtím nějaká komunikace s tímto uživatelem proběhla nebo jestli si „povídají“ poprvé – nemá nikde uložen *stav* komunikace, proto bezstavová. Cookies do komunikace přinášejí možnost uložit stav komunikace s určitým uživatelem, například:

- identifikační údaje uživatele (ať server pozná, o koho jde), a to buď nějaký identifikační řetězec (pak hovoříme o session cookies) nebo přímo jméno a heslo (to už dnes snad ani ne),
- položky nákupního košíku, pokud brouzdáme v internetovém obchodě,
- seznam článků, které jsme si na portálu přečetli, naše preference u prodávaného zboží, konkrétní nastavení prostředí webové aplikace, atd.

V současné době jsou velmi pravděpodobně v cookies pouze identifikační řetězce; cokoliv dalšího se už nachází v databázi provozovatele internetového obchodu, cloudové služby, reklamní sítě apod.

Při komunikaci se serverem je tomuto serveru zasláno také příslušné cookie, podle kterého si server „vzpomene“, s kým komunikuje, a vhodně upraví odesílaná data. K čemu cookies slouží:

- k identifikaci uživatele a přiřazení některých informací, jak je výše naznačeno,
- reklamní účely – personalizace zobrazovaných reklam,
- lze zjistit, na jakých stránkách se pohybujeme, kolik času tam trávíme, co nás zajímá, apod.

Kromě toho mohou být údaje v cookies i zneužity, tedy jejich občasné promazání (alespoň těch, která nám nejsou potřebná) je na místě. *Tracking cookies* jsou právě cookies sloužící ke sledování uživatelů (ukládají v sobě data užitečná pro marketingovou analýzu).

V každém cookie mohou být uloženy tyto položky:

- název a hodnota cookie – tyto položky jsou povinné, všechny ostatní volitelné,
- doba platnosti cookie (po uplynutí této doby je zlikvidováno),
- doména, pro kterou je cookie platné,
- cesta na příslušném webovém serveru, pro kterou je cookie platné (většinou je nastavena na `/`, tedy kořenový adresář v doméně),
- bezpečnostní parametr; je možné vyžadovat, aby bylo cookie přenášeno pouze přes zabezpečené připojení.

Cookies jsou sice v textové podobě, ale není řečeno, že se se dají snadno přečíst. Hodnotou může být libovolný řetězec pro běžného uživatele naprosto nesrozumitelný, to platí i o využití pro identifikaci uživatele (také se jim říká *magic cookies*).

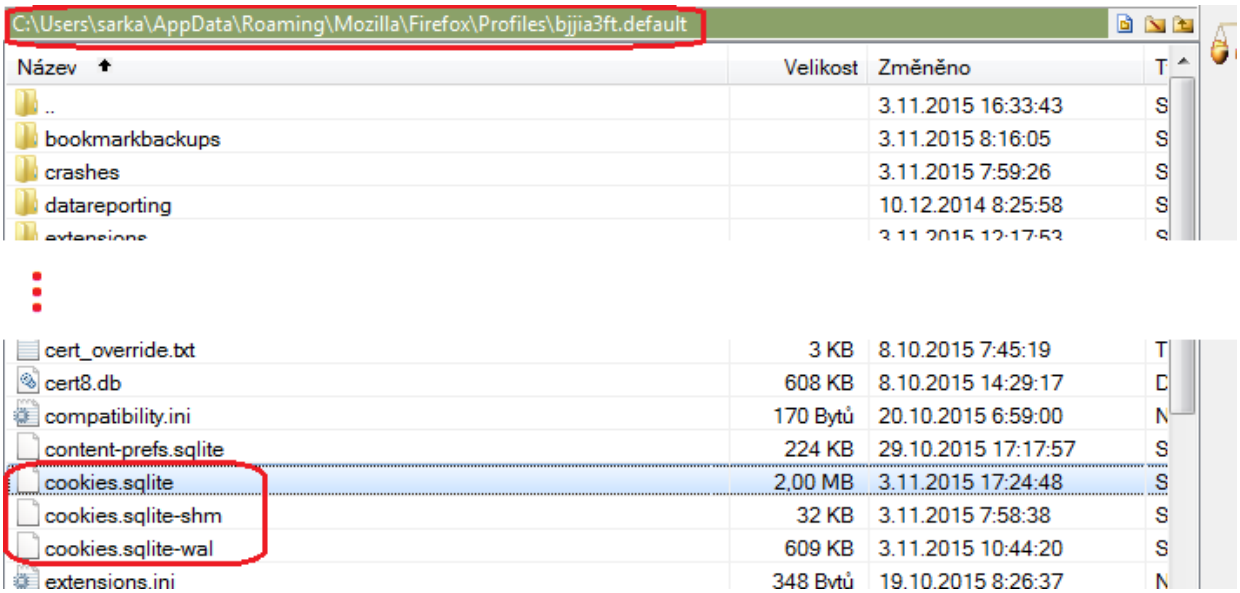
Poznámka:
Pro ukládání cookies, ale také záložek (oblíbených stránek), historie prohlížení a dalších podobných dat jsou mnoha webovými prohlížeči používány SQLite databáze. SQLite databáze je nepříliš velký binární soubor většinou s příponou .sqlite, mechanismus k jeho prohlížení těchto konkrétních souborů je běžně nabízen právě v rozhraní webových prohlížečů nebo pomocí doplňků do prohlížečů.

Klasické HTML cookies. O těchto cookies platí vše z předchozích odstavců. Jsou to buď textové soubory nebo data v malé SQLite databázi, typicky ve velikosti do 4 KiB. Na obrázku 3.4 vidíme umístění souboru s SQLite databází obsahující cookies pro prohlížeč Firefox (je použit souborový manažer FreeComander), v záhlaví je vyznačena cesta.

Nevýhodou klasických HTML cookies je, že se odesílají s každým `http` požadavkem na server, tedy zbytečně zahlcují provoz na síti.

Poznámka:
Další nepříjemnou vlastností je, že některé weby mají ve zvyku do cookies ukládat citlivější údaje (včetně identifikace uživatele) – pokud není použito zabezpečení přenosu (šifrování), jsou tato data odesílána bez jakékoliv ochrany a kdokoliv je může zachytit. Pak hrozí krádež identity (někdo jiný se může na daném serveru za uživatele vydávat) nebo útoky typu man-in-the-middle.

HTML5 cookies. Ve specifikaci HTML verze 5 je další typ cookies určený pro interaktivní aplikace vyžadující ukládání většího množství dat lokálně. Účelem je vytvořit lokální úložiště (na straně uživatele) pro data, která jsou na straně uživatele zapotřebí a zároveň není nutné je neustále přenášet po síti (typicky pro větší objemy dat). Jiný název pro tento typ cookies je *Local Storage* nebo *DOM Storage*.



Obrázek 3.4: SQLite databáze s cookies pro Firefox

HTML5 cookies jsou obvykle ukládány v SQLite databázi, obvyklá velikost jednoho cookie bývá 5 nebo 10 MB. V rámci webové stránky se k těmto cookies přistupuje pomocí JavaScriptu.

Flash cookies. Jedná se o binární soubory s příponou .sol, které najdeme v lokální složce Flash Playeru. Účelem původně bylo umožnit flash aplikacím (hrám, videím) lokálně ukládat konfiguraci a jiné potřebné údaje, dnes však jsou typicky používány ke sledování a evidování uživatele (flash je zatím pořád hodně používán reklamními servery). Flash cookies jsou také schopny sledovat HTML cookies a v případě jejich smazání je obnovovat.

Limit pro velikost tohoto typu cookie je 100 KiB, ale uživatel může být požádán o rozšíření vymezeného prostoru (je dobré si důkladně rozmyslet, jestli to opravdu chceme dovolit). Na rozdíl od předchozích typů flash cookies nejsou závislé na webovém prohlížeči (tj. úložiště je jen jedno a přistupují k němu všechny prohlížeče a jiné aplikace), nemají žádný limit platnosti a také je náročnější je zlikvidovat.

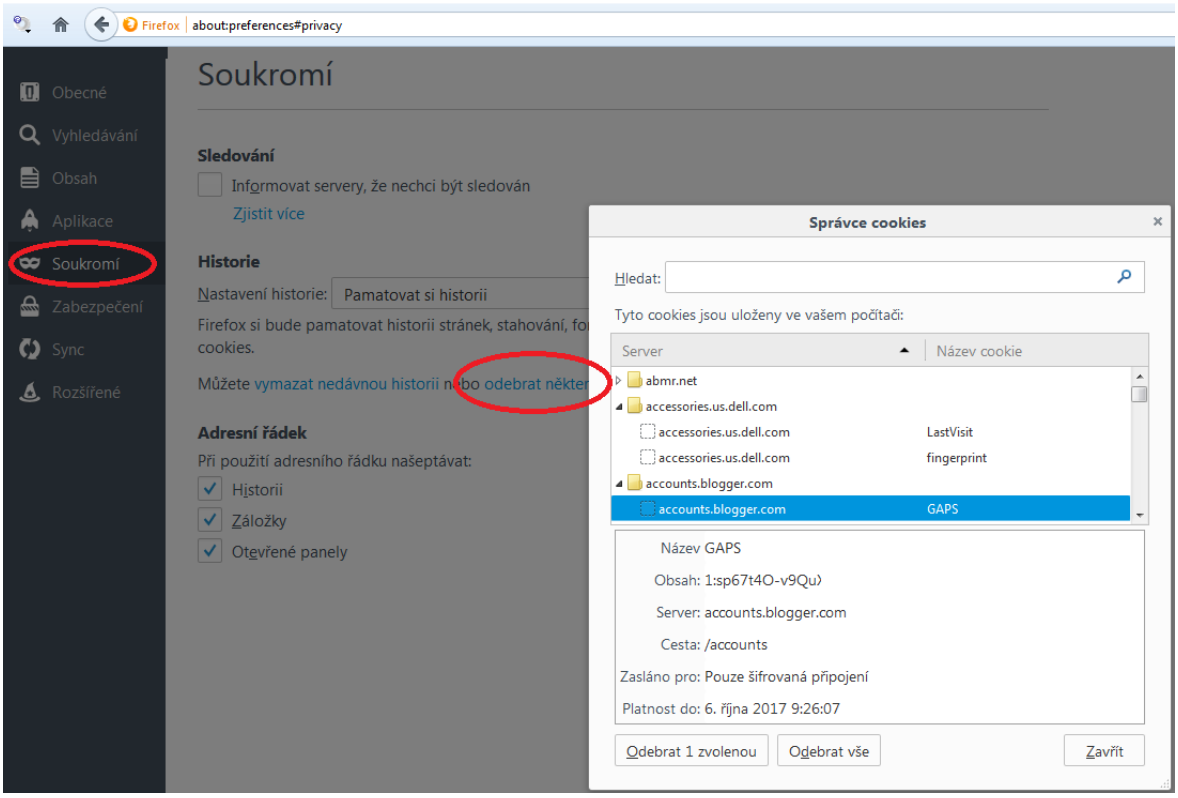
Supercookies. Je to kombinace HTML, HTML5 a Flash cookies, přičemž je zajištěno jejich vzájemné propojení. Pokud je některé z takto propojených cookies smazáno, bude zbývajícími cookies obnoveno. Zlikvidovat supercookie je poměrně náročné.

Evercookie. Tentokrát se nejedná o typ cookie, ale o demonstrační „multicookie“, které bylo vytvořeno jako ukázka, kam všude je možné data o uživateli zastrčit a jak lze tato úložiště provázat za účelem vzájemné „obnovy“. Evercookie je uloženo na 13 různých místech v systému. Informace můžeme najít na http://samy.pl/evercookie/.

Postup (Odstranění klasických HTML cookies)

HTML cookies lze odstranit přímo z prostředí webového prohlížeče:

- IE: *Nástroje – Odstranit historii procházení – Cookies – Odstranit*
- Firefox: *Nástroje – Možnosti – Soukromí*, odkaz *Odebrat některá cookies*



Obrázek 3.5: Odstranění cookies ve Firefoxu

- Chrome: *Přizpůsobení a ovládání Google Chrome – Historie – Vymazat všechny údaje o prohlížení – Smazat soubory cookie*

Dále je možné především tracking cookies odstranit například pomocí nástroje *CCleaner* od Piriformu.

Postup (Odstranění flash cookies)

S flash cookies nám také může pomoci *CCleaner* (záložka *Applications*, *Multimedia*, vybereme Adobe Flash Player). Pro některé webové prohlížeče existují doplňky, které kromě jiného dokážou také mazat všechny běžné druhy cookies včetně flash cookies:

- Firefox: doplněk BetterPrivacy
- Chrome: doplněk Click & Clean

Tyto doplňky by měly umožnit smazat i supercookies.

Postup (Zakázání používání HTML5 cookies)

Pokud máme dojem, že HTML5 cookies nepotřebujeme, můžeme vypnout DOM Storage. Ve Firefoxu se to dělá následovně:

- do adresního řádku zadáme about:config,
- parametr `dom.storage.enabled` (případně vyfiltrujeme dom.storage),
- poklepeme, nastavíme na `False`.

Webové stránky pak nebudou moci DOM Storage používat.

Proti zneužívání cookies také mohou pomoci různé doplňky do webových prohlížečů, například Adblock Plus, NoScript, Tracking protection list, atd.

Další informace:

- https://tools.ietf.org/html/
- http://www.piriform.com/docs/ccleaner/ccleaner-settings/choosing-which-cookies-to-keep
- http://www.zdrojak.cz/clanky/minulost-soucasnost-a-budoucnost-lokalniho-uloziste-pro-html5-aplikace/
- http://www.aspnet.cz/articles/344-local-storage-sbohem-cookies-sbohem-session
- http://stackoverflow.com/questions/13897472/how-do-third-party-tracking-cookies-work
- http://www.root.cz/zpravicky/zombie-cookies-preziji-diky-html5/
- http://earchiv.chip.cz/cs/earchiv/vydani/r-2011/chip-05-2011/mazani-cookies.html
- http://www.ictmanazer.cz/2011/12/pripravte-se-na-hrozby-spojene-s-html5/
- http://en.wikipedia.org/wiki/Zombie_cookie
- http://www.cookiecentral.com/
- http://www.earchiv.cz/a96/a638k130.php3

3.4.2 Nástupce cookies: FLoC

Cookies jsou čím dál víc vnímány negativně, zakazovány a mazány, proto Google (coby jeden z největších koncernů žijících z reklamy) přichází s něčím jiným – Federated Learning of Cohorts (FLoC).

FLoC (podobně jako cookies) jsou marketingový prostředek, který má přispívat ke správnému zacílení reklamy. Zatímco cookie umožňuje „vynést“ směrem k reklamní společnosti nejrůznější osobní informace, ze kterých se dají preference uživatele odvodit, FLoC na to jde jinak.

Účelem FLoC je neevidovat na serverech uživatele jako individualitu, ale zařadit ho do skupiny (kohorty) podle jeho zájmů, preferencí, obvyklých položek v nákupech, atd. Konkrétní údaje (například to, že uživateli se líbí určitá značka auta, nedávno koupil skříň ve světlé barvě, má rodinu s malými dětmi a psa, pro kterého kupuje krmení,...), zůstávají lokálně na počítači uživatele, kdežto reklamní weby dostávají pouze označení (ID) kohorty, do které byl uživatel zařazen. Podle této informace může daný reklamní web usoudit na to, co asi tohoto uživatele bude zajímat, a podle toho přizpůsobit výběr zobrazovaných reklam. Evidované zájmy by se měly pravidelně (každý týden) přehodnocovat, i to samozřejmě lokálně. V jedné kohortě by měli být sdruženi uživatelé v souhrnu s podobnými zájmy, přičemž každá kohorta by měla čítat pravděpodobně jednotky tisíc uživatelů.

Zatím je FLoC podporováno pouze v Google Chrome verze 89 a vyšší (a tam se musí zapnout – taky zatím), ostatní prohlížeče nejsou tímto mechanismem zrovna nadšeny a některé ho přímo odmítají (především typicky alternativní prohlížeče prosazující ochranu osobních údajů jako DuckDuckGo, Vivaldi, Brave).

Zdánlivá výhoda plynoucí z toho, že v žádném případě nejsou přes síť přenášeny přímo osobní informace, je však vyvažována nevýhodami. Za velkou nevýhodu je považováno to, že uživatel nad FLoC nemá naprosto žádnou kontrolu, má vliv dokonce menší než na cookies. Taky je otázka, jak velký rozdíl je mezi FLoC ID a identifikátorem uloženým v cookie; FLoC ID je pro hackery dokonce užitečnější, protože je „obecně platné“, kdežto ID v cookie má smysl pouze pro daný obchod či službu. Odborníci mají obavy, že jde o další prvek použitelný pro fingerprinting (o čemž se dozvíme dále).

Další informace:

- https://www.root.cz/clanky/novou-sledovaci-techniku-google-floc-odmitaji-vivaldi-brave-i-duckduckgo/
- https://web.dev/floc/
- https://amifloced.org/ (včetně otestování prohlížeče, zda podporuje a má zapnuté FLoC)

3.4.3 Sledování a propojení

Weby, které navštěvujeme, jsou propojeny s jinými weby. Většinou se jedná případy, kdy je část stránky webu „propůjčena“ pro reklamní účely (reklamní banner) nebo jiná adresa jakýmkoliv dalším způsobem spolupracuje na dotváření stránky.

Víme, že v cookies mohou být speciální identifikátory, které například určují konkrétního uživatele. Cookie může být posláno na naše zařízení nejen přímo tou stránkou, kterou chceme zobrazit, ale může jít i o cookie z adresy na stránku napojené (third-party).

Příklad

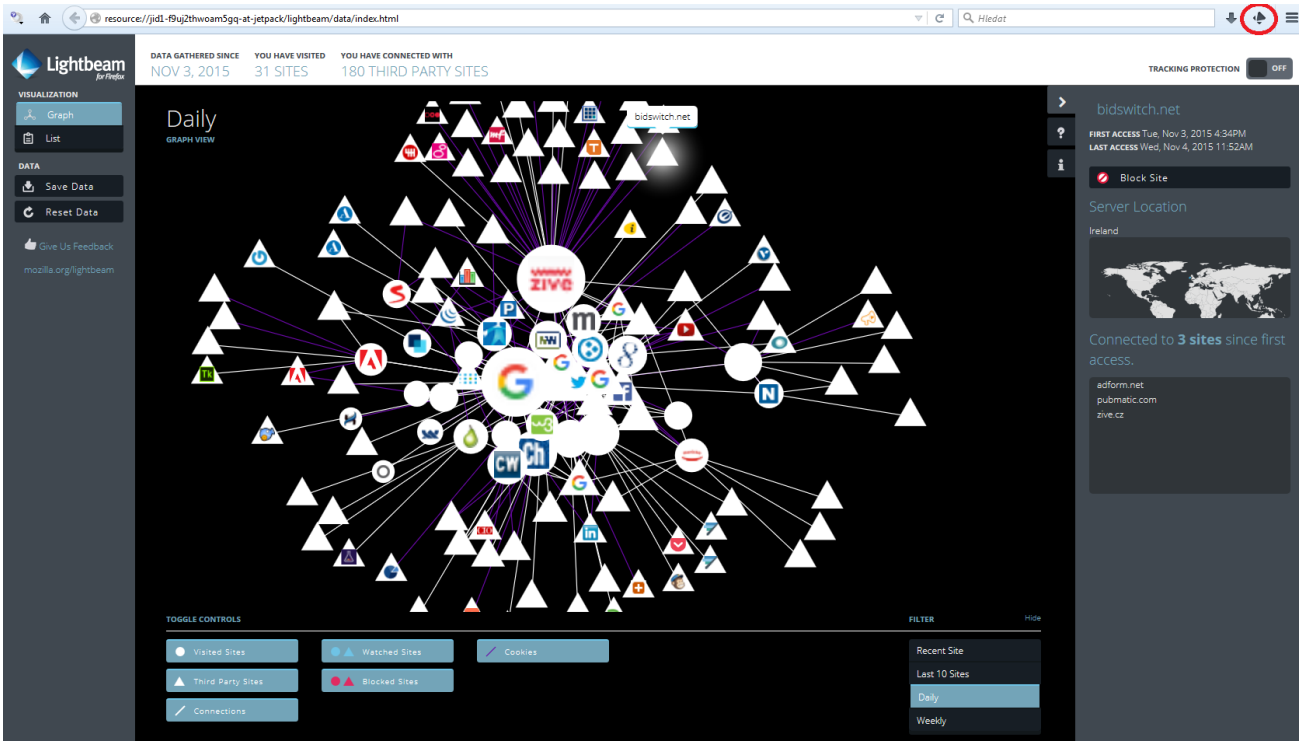
Představme si situaci, kdy určitá reklamní společnost XYZ má bannery umístěné na dvou různých webech AB a CD, přičemž oba bannery se načítají ze stejné adresy (adresy příslušející společnosti XYZ). Pokud některý uživatel navštěvuje stránky obou těchto webů, bude mít pravděpodobně na svém počítači všechna cookies z těchto webů – nejen cookies webů AB a CD (jejichž adresy zadal do adresního řádku), ale i z napojených adres (tj. včetně cookie dotyčné reklamní společnosti XYZ). To znamená, že kdykoliv uživatel navšíví například web AB, jsou odeslána cookies na adresy AB a XYZ. Podobně pro web CD.

Pokud tedy uživatel navštíví web AB, společnost XYZ se o tom dozví. Pokud uživatel navštíví web CD, společnost XYZ se o tom také dozví. Navíc má důležitou informaci: uživatel s tímto ID navštěvuje web AB a zároveň web CD, z čehož se dá lépe usuzovat na jeho zájmy a příslušně cílit reklamu.

Postup (Vizualizace propojení webů)

Propojení mezi weby, které navštěvujeme, můžeme vidět například pomocí nástroje *Lightbeam*. Jedná se o doplněk webového prohlížeče Firefox, tedy získat se dá stejně jako jakýkoliv jiný doplněk.

Po instalaci se v liště objeví ikona nástroje, po klepnutí na tuto ikonu se zobrazí rozhraní (jako na obrázku 3.6; ikona je vyznačena vpravo nahoře). Lightbeam načítá vztahy za běhu při načítání stránek, takže ze začátku neuvidíme nic, až po chvíli surfování se začnou v grafu objevovat



Obrázek 3.6: Doplněk Lightbeam spuštěný ve Firefoxu

jednotlivé uzly. Kruhové uzly jsou přímo weby, které jsme navštívili, trojúhelníkové patří webům „třetích stran“ – jsou navázány na ty kruhové. Po najetí myší nad uzel se objeví k němu příslušná adresa, po klepnutí myší máme v pravém podokně informaci o uzlu – adresa, lokace a seznam navázaných webů.

Vzhled grafu můžeme ovlivňovat – kolečkem myši (či jiným zoomovacím způsobem) zvětšujeme/zmenšujeme, do určité míry funguje také tažení (třeba myší). Vlevo se volí typ zobrazení. Implicitně je zvoleno „Graph“, další možností je „List“, tedy seznam všech zaznamenaných adres s příslušnými údaji včetně výpisu provázání.

Lightbeam také umožňuje konkrétní stránku zablokovat (zvolíme příslušný uzel v grafu nebo položku v seznamu, v pravém podokně je tlačítko *Block Site*). Tím zamezíme dalšímu sledování z této adresy, příslušné cookie již na ni není posíláno.

Uvedení nástroje Lightbeam v roce 2013 rozpoutalo diskusi o rozsahu a smyslu sledování uživatelů webu. Podobný doplněk zobrazující vztahy mezi weby až tak komplexně zatím neexistuje pro jiné prohlížeče, třebaže doplňky, které alespoň částečně tyto informace nabízejí, k dispozici jsou: například Ghostery je doplněk pro většinu běžných prohlížečů, který pro právě zobrazenou webovou stránku vypíše seznam napojených adres.

Další informace:

- https://addons.mozilla.org/en-US/firefox/addon/lightbeam/?src=search
- http://thenextweb.com/apps/2013/10/25/mozillas-lightbeam-firefox-add-lets-users-visualize-sites-tracking/
- http://www.independent.co.uk/life-style/gadgets-and-tech/news/mozillas-lightbeam-tool-will-expose-who-

is-looking-over-your-shoulder-on-the-web-8902269.html

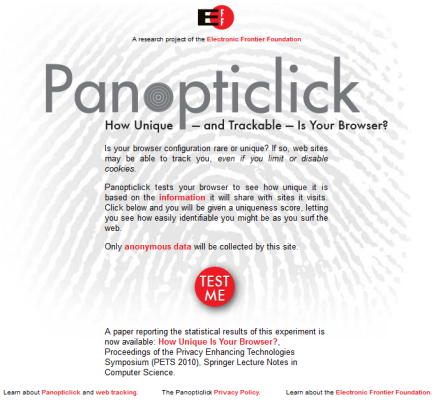
- http://www.pcworld.com/article/2057926/hands-on-mozillas-lightbeam-is-info-porn-for-privacy-geeks.html
- https://www.ghostery.com/

Při sledování uživatele je především důležité tohoto uživatele co nejlépe identifikovat, aby bylo možné k jeho profilu přidávat další dodatečně získané informace. Cookies nejsou jedinou možností, jak uživatele identifikovat – ve skutečnosti o nás hodně prozrazuje náš webový prohlížeč. Právě tato aplikace totiž spoluvytváří pakety, které pak počítačovou sítí putují ke svému cíli, a určuje, co vše bude v záhlaví paketu uvedeno. Součástí paketu jsou samozřejmě cookies, ale kromě toho tam mohou být informace prohlížeči a jeho verzi, instalovaných doplňcích, časovém pásmu, rozlišení obrazovky, instalovaných fontech, atd. Souhrn toho všeho je do značné míry unikátní, a tedy pomáhá uživatele identifikovat.

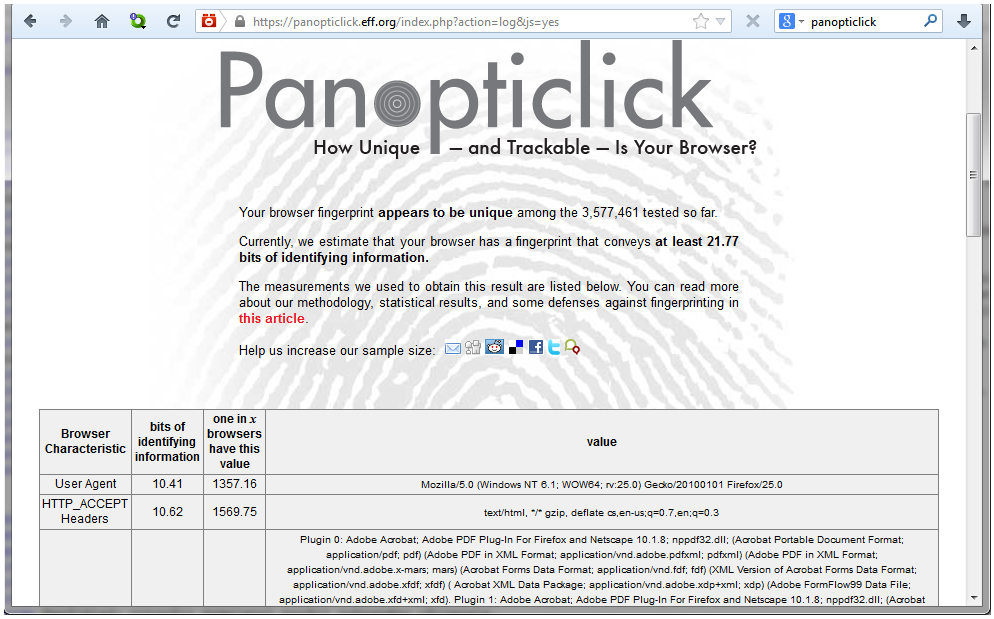
Souhrn dat charakteristických pro určitý systém a prohlížeč, která jsou uvedena v záhlaví `http` paketu, se nazývá *webový otisk prstu*.

Postup (Ověření unikátnosti dat posílaných webovým prohlížečem)

Ve svém prohlížeči otevřeme stránku https://panopticlick.eff.org/. Jedná se o stránku projektu PanoptiClick, která dokáže zobrazit náš webový otisk prstu. Stiskneme červené tlačítko „Test me“ uprostřed stránky (jak vidíme na obrázku vpravo), čímž požádáme o výpis některých dat ze záhlaví `http` paketu, a tedy našeho webového otisku prstu.



Část výstupu je na obrázku 3.7 – začíná určením použitého webového prohlížeče s uvedením verze jádra prohlížeče, a systému, ve kterém prohlížeč běží. Následuje položka



Obrázek 3.7: Výpis webového otisku prstu uživatele

`HTTP_ACCEPT Headers`, což je seznam akceptovaných MIME typů
(zjednodušeně – jaké datové formáty dokáže prohlížeč přijmout a zobrazit). Následuje seznam pluginů i s jejich verzemi a knihovnami, určení časové zóny (o kolik minut jsme posunuti oproti nultému poledníku) a další údaje.

Zajímavý je sloupec „one in x browsers have this value“ (jeden z x prohlížečů, které mají tuto hodnotu), který ukazuje unikátnost našeho prohlížeče právě vzhledem k vlastnosti uvedené v dotyčném řádku.

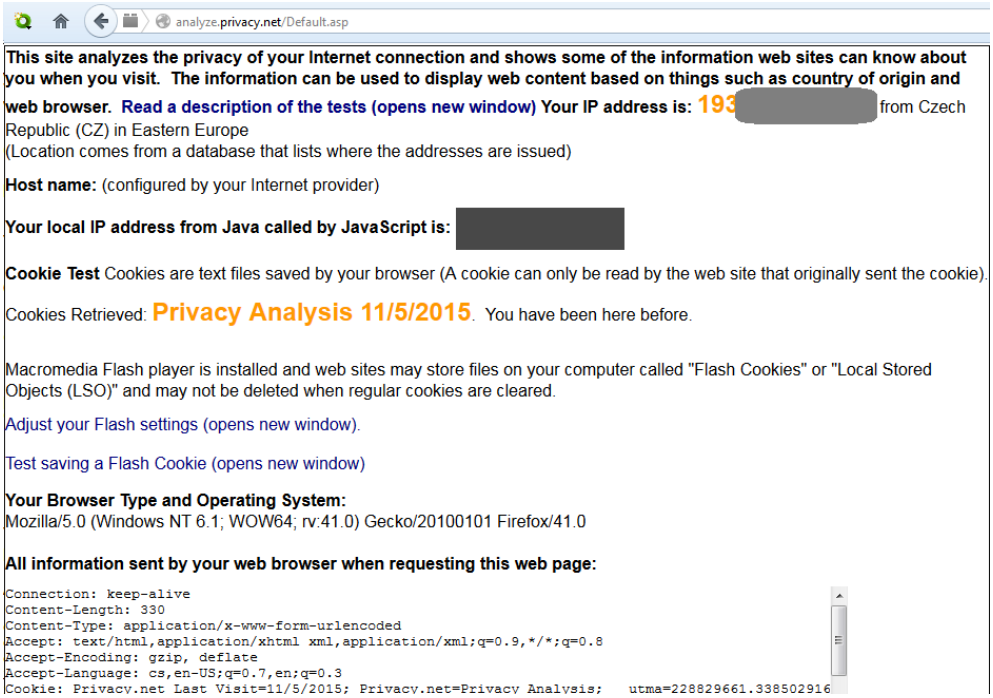
Další informace:

- https://panopticlick.eff.org/browser-uniqueness.pdf
- http://www.newmediacampaigns.com/blog/browser-rest-http-accept-headers

Trochu jiný výpis informací poskytovaných serverům naším prohlížečem získáme na další adrese:

Postup (Výpis údajů ze záhlaví `http` paketu)

V prohlížeči přejdeme na stránku http://analyze.privacy.net/. Automaticky se spustí test a objeví se následující okno (obrázek 3.8):



Obrázek 3.8: Výpis údajů ze záhlaví `http` paketu

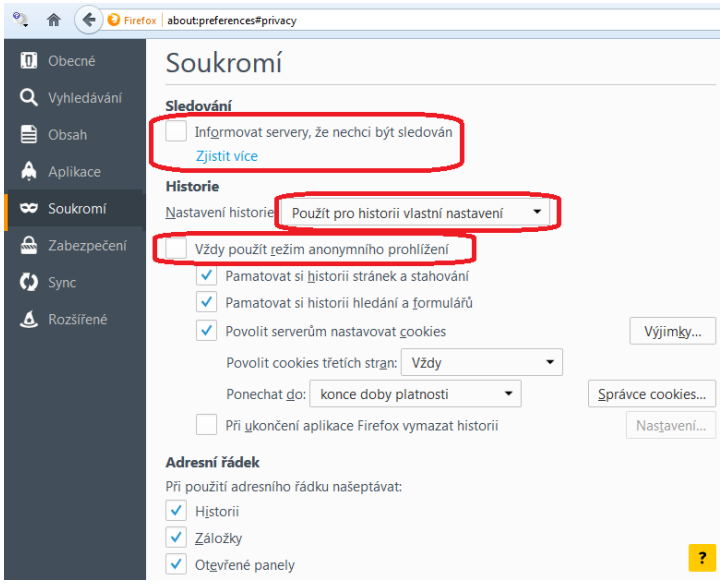
3.4.4 Anonymní surfování a anonymní režimy

I běžný uživatel se vyděsí, když zjistí, co vše putuje (nejen) k serveru, jehož stránku si chce zobrazit. Jak bylo výše uvedeno, z těchto informací lze vytvořit unikátní identifikátor dotyčného uživatele (resp. jeho zařízení+systému+prohlížeče), což je zneužitelné.

V používaných prohlížečích je obvykle možné zapnout volbu „Do not track“. Zapnutím této volby sdělujeme serverům, že si nepřejeme být sledováni. Ovšem – na dotyčném serveru záleží, jestli tuto naši žádost bude brát v úvahu. Touto volbou si tedy nevynucujeme vyšší soukromí a mnohé servery ji ignorují.

Postup (Funkce „Do not track“ ve Firefoxu)

V prohlížeči Firefox přejdeme do *Možností*, karta *Soukromí*. Zaškrtneme volbu „Informovat servery, že nechci být sledován“ – obrázek 3.9 nahoře. Pokud se jedná o Firefox pro Android, najdeme v panelu *Předvolby* položku *Požádat webové stránky, aby nesledovaly vaši aktivitu.*

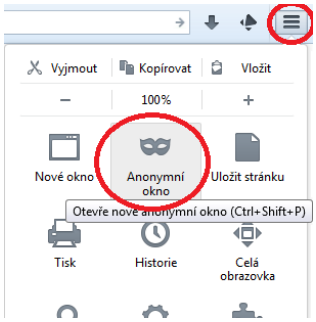


Obrázek 3.9: Nastavení soukromí ve Firefoxu

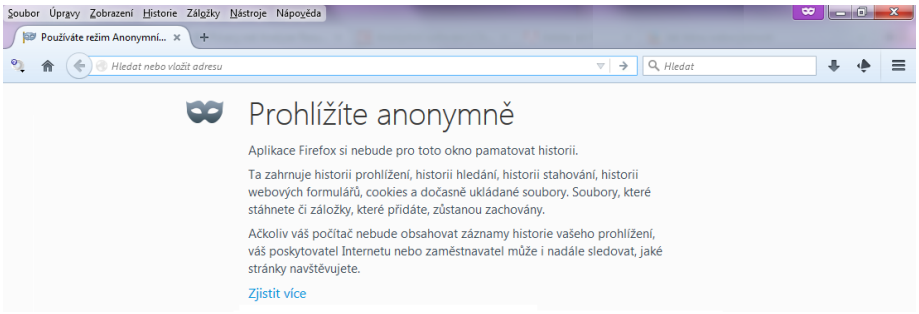
Prohlížeče také nabízejí anonymizační režimy. U každého prohlížeče je tento režim jinak „soukromý“, obvykle jde o vypnutí zaznamenávání historie, cookies, stažených souborů. Nejde ani tak o to, aby servery uživatele nesledovaly, ale spíše o to, aby nezůstalo „zbytečně moc“ informací pro příliš zvědavého člověka přehrabujícího se v počítači.

Postup (Nastavení anonymního režimu)

V prohlížeči Firefox:



Na obrázku 3.9 vidíme, že když v možnostech nastavení zvolíme *Použít pro historii vlastní nastavení*, objeví se hned o řádek níže volba *Vždy použít režim anonymního prohlížení*. Jestliže však chceme anonymní režim použít jen pro jedno okno, v *Možnostech* klepneme na *Anonymní okno* (postup vidíme na obrázku vpravo), případně můžeme použít klávesovou zkratku Ctrl+Shift+P – otevře se nové okno, které již poběží v anonymním režimu, tedy nebudou se ukládat cookies, historie, dočasné soubory apod. Stránka je na obrázku 3.10. Pokud budeme chtít v tomto režimu stáhnout soubor a uložit na počítači, půjde to, nic dalšího by se nemělo ukládat.



Obrázek 3.10: Okno v anonymním režimu ve Firefoxu

V prohlížeči Chrome:
Zde je to v *Možnostech* volba *Nové anonymní okno*, případně klávesová zkratka Ctrl+Shift+N. V Chrome pro Android je to volba *Nová anonymní karta*. Funguje to podobně jako u Firefoxu – pokud budeme chtít z webu stáhnout soubor, není problém ho uložit, ale historie prohlížení, cookies a další podobná data se ukládat nebudou. Navíc se deaktivují všechny doplňky, které by jinak mohly ukládat data bez ohledu na zapnutí anonymního režimu.

V prohlížeči Opera:
V Opeře je anonymní režim velice důkladně propracován – můžeme si otevřít nejen nové anonymní okno (klávesovou zkratkou Ctrl+Shift+N), ale může jít i o záložku (list, panel). Kromě neukládání historie, cookies apod. se také deaktivují doplňky.

Poznámka:
Pozor – účelem anonymních režimů ve webových prohlížečích není zamezení sledování na internetu, ale zametení stop přímo na počítači. K tomu účelu je lepší používat jiné možnosti, například ve Firefoxu máme k dispozici doplňky NoScript, AdBlock Plus, Better Privacy, Ghostery a další.

Naopak stojí za úvahu vypnout modul Shockwave Flash – zejména kvůli častým bezpečnostním problémům a mnohdy zbytečnému zpomalování načítání stránek, tak i proto, že na něm stojí mnohé reklamní systémy a souvisí s flash cookies.

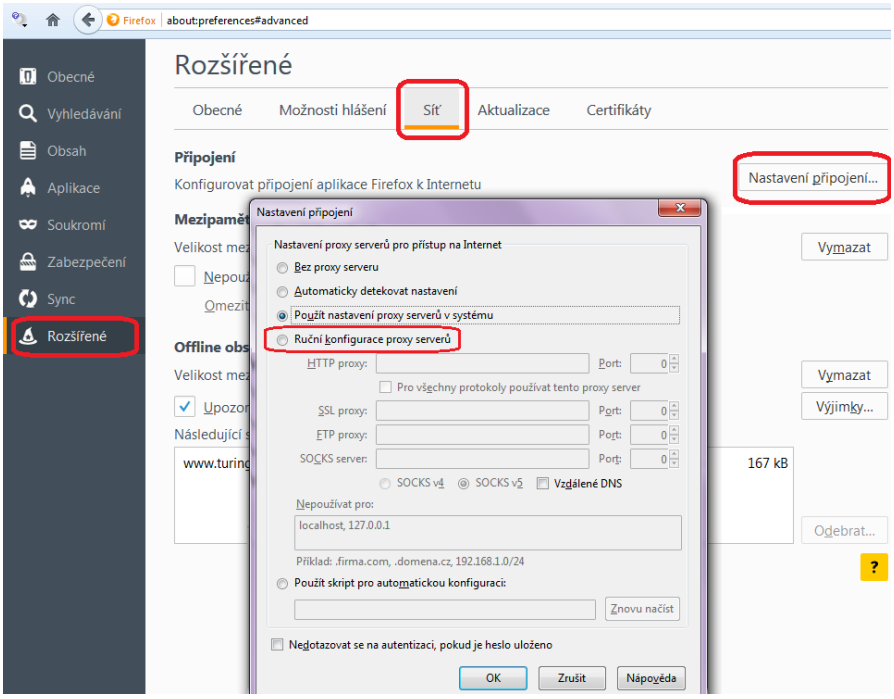
Anonymitu „směrem ven“ mohou zajistit *proxy servery*. Používají se nejen tehdy, když uživatel chce jednoduše skrýt identitu, ale také v případě, že chceme využívat službu s lokálním omezením kontrolovaným podle IP adresy (potřebujeme vystupovat pod IP adresou z jiné země). Proxy může fungovat prostě tak, že v paketu po odeslání z našeho počítače zamění naši IP adresu v poli odesílatele za některou svou IP adresu (navíc provede změnu v poli s číslem portu) a pak odešle k cíli, v paketu posílaném v opačném směru udělá přesně opačnou operaci.

Webové proxy servery jsou jednodušší variantou – otevřeme stránku takového proxy serveru, do pole zadáme tu adresu, s níž bychom chtěli přes proxy komunikovat. Veškerá konfigurace se provádí přes webové rozhraní, odtud název. Problémem je výraznější zpomalení surfování (záleží i na momentálním vytížení vybraného proxy serveru), horší stabilita a v některých případech i bezpečnost, čas od času jsou buď tyto servery napadeny nebo se dokonce objevují nastrčené falešné proxy. Uživatel bohužel nemá možnost ověřit si, co vlastně proxy server dělá s jeho komunikací

o co vše si ukládá. Navíc se může stát, že jako reálný proxy slouží napadený počítač uživatele, který o tom nemá ani ponětí.

Typickými zástupci jsou Hide My Ass (http://www.hidemyass.com), Proxify (http://proxify.com/) nebo Anonymouse (http://anonymouse.org/anonwww.html), seznam aktivních webových proxy serverů je například na adrese http://proxy.org/cgi_proxies.shtml, http://free-proxy.cz/cs/web-proxylist/ nebo http://www.publicproxyservers.com/proxy/list1.html.

Pokud chceme proxy server využívat více než jen pro jednu stránku, je lepší to ručně nastavit přímo v prohlížeči. Obvykle potřebujeme znát především adresu a číslo portu, přes které se bude komunikovat. Na obrázku 3.11 je provedení pro Firefox. V Internet Exploreru se nastavení provádí přes *Možnosti Internetu* – panel *Připojení* – *Nastavení místní sítě*; v Chrome přes *Zobrazit pokročilá nastavení* – *Změnit nastavení proxy*.



Obrázek 3.11: Ruční nastavení proxy ve Firefoxu

Klasické proxy servery se o něco složitěji nastavují, ale údajně nabízejí větší úroveň anonymity. Nicméně problémem zůstává zpomalení komunikace, horší stabilita připojení a problematické zabezpečení včetně možnosti nechtěného zneužívání počítačů jiných uživatelů. Seznam klasických proxy najdeme například na webu SamAir (http://www.samair.ru/proxy) nebo NNTime (http://nntime.com/).

Rozšířením a znásobením principu proxy je využití *anonymizačního softwaru* umožňujícího napojení na kaskádu proxy serverů. Anonymita je zde kromě určité substituce adres (podobné té u běžných proxy) zajišťována šifrováním, účelem je tedy zajistit soukromí po cestě (nejen v cíli). Toto řešení přináší vyšší úroveň anonymity a bezpečnosti (víc proxy serverů za sebou, relativně důvěryhodných, žádné divoké proxy servery), ale problémům se bohužel taky nevyhýbá. Komunikace se zpomaluje a navíc se také objevují zprávy o bezpečnostních problémech.

Nejznámějšími zástupci jsou anonymizační sítě TOR (The Onion Routing) a JAP, případně

potomek druhé jmenované sítě Jondo. Podrobněji se budeme sítí TOR zabývat až po prostudování kapitoly o kryptografii.

Další informace:

- http://www.chip.cz/casopis-chip/earchiv/rubriky/technika/anonymni-surfovani/
- http://pcworld.cz/internet/jak-a-proc-surfovat-na-webu-v-utajeni-2-dil-45149

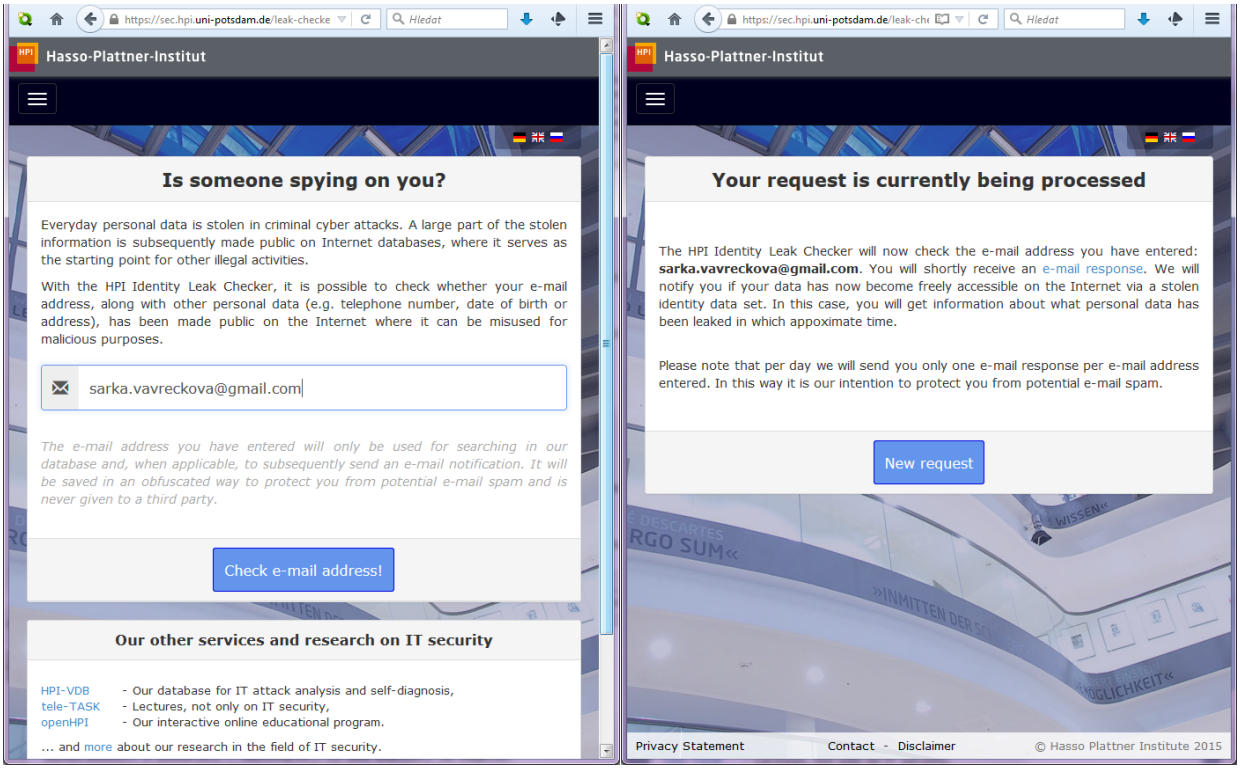
3.5 E-mail – ověření odcizení identity

Nepříjemným faktem je, že s osobními daty se čile obchoduje, a to i na černém trhu. Pokud se někomu podařilo nelegálně získat přístup k našemu e-mailovému účtu, může být tato adresa zneužita například k rozesílání spamu, případně mohou být odcizeny a zneužity údaje uložené v e-mailech ve schránce účtu.

Postup (Ověření napadení e-mailového účtu)

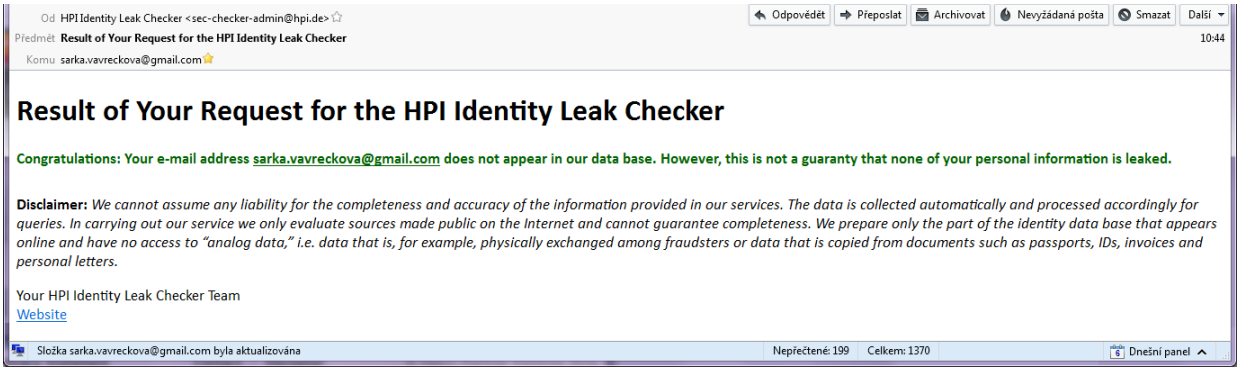
Institut softwarových technologií Hassa Plattnera v Postupimi schromažďuje databázi odcizených identit. K těmto údajům se dostává více různými cestami včetně prohledávání nabídek obchodníků s osobními údaji. Provozuje webovou aplikaci *HPI Identity Leak Checker*, ve které si každý může ověřit, zda jeho osobní údaje (zejména e-mailová adresa nebo telefonní číslo) nejsou někde na internetu zneužívány k nelegálním účelům.

Na adrese https://sec.hpi.uni-potsdam.de/leak-checker/search je třeba zadat příslušný prověřovaný údaj (třeba e-mailovou adresu) do k tomu určeného pole, viz obrázek 3.12.



Obrázek 3.12: Ověření napadení e-mailového účtu

Po klepnutí na potvrzující tlačítko je uživatel informován o odeslání e-mailu, v němž pak najde informaci o výsledku průzkumu, náhled je na obrázku 3.13. Všimněte si, že v tomto e-mailu jsme informováni o tom, že se naše adresa nenachází v příslušné databázi, ale to ještě neznamená, že tato adresa není někým zneužívána. Tedy i když aplikace nic nenajde, nemáme jistotu, že problém neexistuje.



Obrázek 3.13: E-mail s informací o výsledku prověření účtu

3.6 Ztracené nebo odcizené zařízení

3.6.1 Eliminace následků

Zejména u přenosných zařízení existuje riziko, že je uživatel někde zapomene nebo budou odcizena. Následné problémy mohou být finančního rázu (viz odhad ceny odcizeného notebooku na straně 9), ale také další:

- s financemi souvisí jak cena nového zařízení, kterým potřebujeme nahradit to ztracené či odcizené,
- v zařízení mohou být cenné a zneužitelné údaje osobního rázu – certifikáty, přístupové údaje k různým službám, fotografie, videa, domácí úkoly dítěte, atd.
- v zařízení mohou být cenné údaje z pohledu zaměstnavatele – přístupové údaje do firemní sítě, údaje o vyráběných produktech (výrobní tajemství), návrhy, patenty (včetně těch zatím nepodaných), účetnictví, obchodní tajemství, atd., to vše je zneužitelné a také někdy těžko nahraditelné,
- seznamy kontaktů na přátele, příbuzné, kolegy, obchodní partnery apod. s dalšími přidruženými informacemi jsou také velmi cenné, zneužitelné a „obchodovatelné".

Pokud jsme určité zařízení ztratili a nedokážeme je běžnými metodami najít, pak mohou být následky vpodstatě stejné jako u odcizení.

Jak tedy postupovat? Hlavně rychle. Čím dřív reagujeme, tím menší škoda může vzniknout. U odcizeného či nedohledatelného zařízení je třeba nahlásit vše Policii ČR. Nahlašujeme takovou hodnotu odcizené věci, jakou má pro nás – nejen samotný hardware, ale započítáváme i hodnotu dat na zařízení uložených. Stojí za úvahu zohlednit také případné negativní důsledky zneužití těchto dat, to se ovšem zdůvodňuje velmi špatně.

Také je třeba co nejdřív změnit přístupové údaje u všech služeb, jejichž přístupy byly uloženy na dotyčném zařízení.

Pokud jsme s předstihem mysleli na to, že taková situace může nastat (tedy provedli jsme určitá preventivní opatření), můžeme s pomocí prostředků, které budou diskutovány dále, pomoci s vyhledáním zařízení nebo alespoň destrukcí zneužitelných dat na dálku.

3.6.2 Prevence

Jak bylo výše zmíněno, existují určité možnosti prevence. Předně jde o chování, které snižuje pravděpodobnost odcizení nebo ztracení zařízení, případně alespoň snižuje následné ztráty.

Předně bychom si měli na své či propůjčené věci *dávat pozor*, nepřenášet to, u čeho přenášení není nutné a nenechávat nic „zajímavého“ viditelně v automobilu (notebooky, mobily, tablety, navigace, kabelky, atd.). Dalším opatřením, které sice krádeži nezabrání, ale může eliminovat následné ztráty, je *zálohování*. Měli bychom si uvědomit, že existují data, která už nelze znovu vytvořit (například fotografie dětí) nebo je jejich vytvoření náročné (účetnictví, diplomka).

Souhrnně k možnostem pro notebooky:

- zámek Kensington umožňuje přichytit notebook ke stolu, což ovšem funguje jen v součinnosti se stolem či jiným podobným kusem nábytku,
- zaheslovaný BIOS: toto zabezpečení se dá u konkrétních notebooků jednoduše obejít, u jiných je to složitější (záleží na konkrétním výrobci a modelu),
- uživatelské heslo do systému: dá se obejít, zvláště ve Windows
- čtečka otisků prstů: postupně se zanáší, tedy je nutné ji čistit, pokud to umíme,
- další biometrické metody (snímání oka, hlasu apod.): drahé, taky může stávkovat.

Ovšem důkladnějším preventivním opatřením je instalace softwaru, který dokáže pomoci dohledat odcizené zařízení nebo alespoň na dálku smazat uživatelská či systémová data nebo dokonce zneaktivnit zařízení. Jak takové řešení funguje?

Předně je nutné mít tento software (případně doprovázený hardwarem) předem nainstalován a aktivován. Možnosti jsou různé podle toho, jak „hluboko“ je řešení integrováno a jakým způsobem dokáže naše zařízení komunikovat. Existují dva základní přístupy:

1. zařízení se v pravidelných intervalech hlásí, a pokud se po několik za sebou jdoucích intervalů hlášení neobjeví, je považováno za odcizené nebo ztracené,
2. mechanismus aktivujeme na dálku co nejdřív po odcizení či ztracení.

Zařízení by předně mělo nahlásit svou polohu a případně odeslat další informace (například snímky z kamery), a na vzdálený příkaz (či při pokusech o násilný průnik do zařízení) zkartovat důležitá data. Lokalizace se provádí podle možností zařízení:

- GPS čip – tento údaj může být přesný, jen bohužel ne všechna zařízení tento čip mají,
- mobilní sítě – podle toho, ke které základnové stanici se zařízení přihlásilo (to je ovšem použitelné jen u zařízení s GSM modulem, například u mobilních telefonů),
- Wi-fi Access Pointy – poloha se dá zjistit podle polohy hlášené nejbližšími Access Pointy v dosahu.

Zjištěné a vyžádané informace pak zařízení odešle opět podle svých možností – webovým protokolem na server, přes SMS, e-mailem apod.

Poznámka:

Pokud se tímto způsobem podaří získat například snímky zloděje, v žádném případě je nesmíme sami zveřejňovat (tím bychom se vystavili postihu, i zloděj má právo na ochranu osobnosti). Je třeba tyto informace co nejdřív předat Policii ČR.

Poznámka:

Níže uvedená řešení mohou být bezpečnostním benefitem, pokud chceme dohledat či na dálku řídit své ztracené nebo odcizené zařízení, ale na druhou stranu mohou být zneužita ke špehování uživatele nebo dokonce k zásahům do jeho zařízení. Na to je třeba brát ohled – přístup ke svému „sledovacímu“ účtu dobře chránit a doufat, že společnost zajišťující tuto službu nebude napadena.

Computrace je služba provozovaná společností Absolute, uživatelům ji pak poskytují výrobci notebooků. Musíme mít podporované zařízení, ovšem u většiny velkých výrobců najdeme modely touto technologií vybavené. Nepracuje na úrovni hardwaru, ale na úrovni BIOSu (firmwaru). V systémovém procesu běží skrytá služba *Computrace Agent*, která v pravidelných intervalech odesílá na server IP adresu a další informace (podle konfigurace), případně přijímá instrukce ke zkartaci dat nebo systému. Řešení je určeno pouze pro Windows (včetně starších verzí) a funguje i v případě, že zloděj nabootuje z jiného média či se pokusí přeinstalovat systém nebo vyměnit disk.

Další informace:

- http://notebook.cz/clanky/technologie/2008/Computrace
- http://www.absolute.com/en/products/absolute-computrace
- http://www.dell.com/learn/us/en/555/vsl-absolute

Poznámka:

Princip nesmazatelného a neviditelného sledování je bohužel zneužíván i samotnými výrobci elektroniky. V časopise Chip 11/2015 se objevila zpráva o aféře týkající se společnosti Lenovo (nicméně je možné, že tyto pochybné aktivity zdaleka nevyvíjí jen tato společnost) – do notebooků a stolních počítačů Lenovo integrovalo produkt Lenovo Service Engine (LSE), který zjišťoval některé konkrétní informace o systému a uživateli a posílal vše Lenovu. Po odhalení společnost předstala tento nástroj do zařízení integrovat a postiženým majitelům nabídla nástroj na odstranění.

Lenovo využilo funkci Windows Platform Binary Table (WPBT) v UEFI (to je nástupce BIOSu) určené právě pro nástroje použitelné při krádeži zařízení, tudíž není možné tento nástroj běžnými prostředky odhalit a deaktivovat (ani přeinstalováním systému či výměnou disku).

Prey je univerzálnější řešení – pro notebooky, tablety a smartphony. Je to open-source software a většina kódu pro klienta je volně dostupná. Také Prey funguje tak, že na serveru máme

zřízen účet (ve verzi zdarma až pro tři zařízení) a na zařízení instalujeme agenta, který očekává instrukce. Po obdržení instrukcí agent shromáždí veškeré dostupné informace, které by mohly pomoci najít zařízení (také včetně snímků, pokud má zařízení například kameru), a následně odešle na server.

Lokalizace se provádí pomocí GPS nebo zjišťováním dostupných Wi-fi AP, komunikovat může pomocí SMS nebo přímo webovým protokolem přes síť.

Prey toho umí celkem hodně:

- uzamknout zařízení,
- skrýt data a aplikace,
- případně odstranit hesla,
- detekovat výměnu SIM karty,
- hlasitý alarm, ochrana před odinstalací,
- vydávat se za neškodnou hru a tedy skrýt sebe sama.

Další informace:

- http://preyproject.com/
- http://www.svetandroida.cz/prey-najdete-svuj-ztraceny-ci-odcizeny-telefon-201306
- http://mac.appstorm.net/reviews/security/prey-a-powerful-recovery-tool-for-your-missing-laptop/

Existují i další řešení pro mobilní zařízení, například:

- pro Android: IOBit, Cerberus a další,
- produkty antivirových firem, často se jedná o součásti bezpečnostních balíků (Norton, Eset, avast!, Lookout, AVG, Zoner, atd.).

Další informace:

- https://play.google.com/store/apps/details?id=com.lsdroid.cerberus&hl=cs
- Jak přidat ochranu proti odcizení telefonu do ROM? [online] *Android Fórum*, 2011. Dostupné na: http://androidforum.cz/jak-p-idat-ochranu-proti-odcizeni-telefonu-do-rom-t21135.html

3.6.3 Webkamera

Webkameru dnes najdeme jak na mnohých noteboocích a různých mobilních zařízeních, tak i například u smart televizí a dalších „chytrých“ zařízení. Webkamery jsou často využívány i pro jiné účely než jen sledování zabezpečení místnosti či jiného prostoru. Některé z těchto možností jsou legální (nasnímání zloděje našeho zařízení a přeposlání policii), ale bohužel jiné nikoliv.

Příklad

Klienti jedné z amerických firem půjčující notebooky a prodávající je na splátky zjistili, že na notebooku koupeném na splátky právě od dotyčné společnosti je nainstalován software PC Rental Agent od společnosti Designerware. Tento software běží na počítači skrytě a dokáže na vyžádání

posílat nejrůznější informace včetně fotografií získaných webkamerou. Firma program používala za účelem dohledání půjčených zařízení, která nebyla řádně vrácena, nebo prodaných zařízení, u kterých nebyly uhrazeny poslední splátky (to je v USA právně v pořádku, pokud firma informuje nabyvatele, jenže to se nestalo). Informace najdeme například na
http://www.pcworld.com/article/227031/rental_companies_watch_pc_renters_via_webcam.html.

Dokonce se ukázalo, že v některých firmách byl takový software zneužíván zaměstnanci, přičemž klienti vůbec netušili, že jsou natáčeni a snímky putují mimo zařízení. Navíc se takový software dá považovat za bezpečnostní mezeru zařízení.

Pro zneužití webkamery ani není nutné instalovat „hlídací“ software – objevily se případy *utajené aktivace webkamery* (ono totiž nemusí platit, že když nesvítí světýlko, nenatáčí se) škodlivým softwarem. Metoda je oblíbená také u ransomwaru (vyděračského softwaru), přičemž uživateli je zobrazeno hlášení typu váš počítač byl uzamknut/šifrován, napravíme to po odeslání částky xxxx na účet yyyy. Pokud je u takové hlášky i uživatelův snímek, hlášení působí důvěryhodněji a výhružněji, uživatel pravděpodobněji zaplatí.

Příklad

Place Raider (z roku 2012) je trojský kůň pro smartphony, který pravidelně vytváří fotografie a ukládá na serveru (nijak na tuto svou činnost neupozorňuje). Fotografie týkající se konkrétní místnosti či bytu lze zkompletovat a metodou podobnou skládání panoramatických snímků vytvořit 3D panoramatický zoomovatelný model tohoto prostoru. Běžný uživatel si totiž neuvědomuje, že se při pohybu místností často otáčí (kličkuje mezi nábytkem, zabočuje k jiným dveřím, s někým hovoří a tedy se k němu natočí, apod.), toutéž cestou jde následně v opačném směru, přičemž stačí, aby svůj mobil držel v ruce.

Tento trojský kůň je ve fázi proof-of-concept, je určen pouze pro výzkumné účely a je dílem týmu z University of of Indiana. Je zřejmé, že něco takového by ocenili zejména vykradači bytů a kanceláří nebo zájemci z řad státních služeb.

Postup (Zabránění zneužití webkamery)

Vypnutí webkamery pomocí klávesové zkratky (jak se to dělá na většině notebooků) *nemusí nutně vést k její deaktivaci* (tyto signály jsou softwarově zachytitelné a ovlivnitelné). U mnohých kamer se dá nastavit ovládání tak, aby bylo nutné aktivaci zaznamenávacího módu ručně odsouhlasit, případně se při aktivaci záznamu může ozvat zvukový signál, ovšem taky je otázka, jak moc účinné toto opatření bude.

Pokud se jedná o webkameru připojenou přes USB kabel, pak stačí vysunout příslušný konektor. V jiných případech (webkamera bývá integrovaná do monitoru notebooku) je řešením prostě zakrýt přeloženým papírem (některé kamery jsou přímo vybaveny krytkou). U mobilních zařízení typu mobil či tablet je možné kameru přelepit barevnou lepicí páskou (pokud ji moc často nepoužíváme), ale aby nedošlo k poškození lepidlem, měli bychom předem na objektiv přiložit třeba kousek papíru. Nejdůležitějším opatřením je dávat si pozor, na co kameru zaměřujeme.

Škodlivý software přistupující k webkameře bývá řazen do kategorie spywaru, a tedy mnohé

antispywarové programy dokážou toto napadení zjistit, například *SpyBot Search & Destroy*.

Je třeba si uvědomit, že podobně může být napaden a zneužit i mikrofon a různé další senzory, kterými jsou dnes vybaveny zejména tablety a chytré telefony.

Další informace:

- http://www.eweek.com/security/researchers-demonstrate-3d-spy-trojan-for-mobile-phones
- http://arxiv.org/pdf/1209.5982.pdf

3.7 Operační systémy coby sběrači dat

3.7.1 Windows 10

Většina uživatelů Windows 7 a 8 (plus 8.1) je přesvědčována k upgradu na Windows 10 neodbytnou (nesnadno odstranitelnou) ikonou okýnka v Oznamovací oblasti Hlavního panelu a mnozí z nich už volání poslechli. Poměrně velké procento těchto „poslušných“ pak zjistilo, že ovladače některých periferních zařízení se s touto změnou nedokázaly vyrovnat, další museli řešit problémy s kompatibilitou aplikací, o které rozhodně nechtějí přijít... Někteří uživatelé zase řešili (v tom případě spíše softwarové než hardwarové) problémy při koupi nového zařízení s předinstalovanými Windows 10.

Jenže s Windows 10 souvisí ještě úplně jiný druh problémů. Na webu se postupně začaly objevovat více či méně poplašné články o tom, že se vlastně jedná o obrovský spyware, odesílají se megabajty dat na servery Microsoftu, Microsoft chce vědět naprosto vše, co v systému děláme, atd. Jak je to doopravdy?

Předně upozorňuji, že Microsoft verzi od verze stále více slučuje varianty Windows pro běžné počítače a pro mobilní zařízení, přičemž se u běžných Windows čím dál víc přiklání k metodám, jaké jsou zcela běžné v mobilním světě. Uživatelé mobilních operačních systémů jako je Android nebo iOS jsou na tom vlastně celkem podobně jako uživatelé Windows 10 – systém i aplikace sbírají všemožná data, ke kterým se dokážou dostat, a odesílají je na servery svých distributorů. Především autoři bezplatných aplikací si mnohdy *vydělávají* buď zobrazováním reklam, nebo právě komerčním využíváním získaných údajů (ať už je to jakkoliv zavrženíhodné, do určité míry je to legální).

Poznámka:

Z toho vyplývá jeden důležitý poznatek – každý systém (ať už Windows 10 nebo mobilní zařízení) sbírá a odesílá především taková data, k jakým ho pustíme a která mu dovolíme odesílat. Tedy základní obranou je správná konfigurace. Každý systém má pro tento účel své vlastní metody a nástroje, třebaže jsou tyto nástroje často dovedně skryty. Znalý uživatel ale ví, po čem se dívat – hledá možnosti nastavení soukromí, oprávnění aplikací, nastavení pro síťový provoz atd.

Na své soukromí je třeba myslet už při instalaci a důkladně si všímat všeho, co zaškrtáváme a kterou cestu volíme. Pokud během instalačního procesu zvolíme „Expresní instalaci“, bude

především přeskočena obrazovka, na které máme první možnost konfigurovat volby související se soukromím. Takže pozor, Expresní instalaci je lepší se vyhnout.

Microsoft se neustále stará o to, aby uživatelům nezakrněly neurony, tedy v různých verzích máme odlišnosti v tom, co se kterým nástrojem nastavuje, jak různé nástroje vypadají, co umí, jakým způsobem se k nim dostat,... Windows 10 rozhodně nejsou výjimkou.

Jednou ze změn je objevení se nového souhrnného nástroje, který v sobě sdružuje různé možnosti konfigurace – nástroje *Nastavení* dostupného přes nabídku Start. Takže máme dvě nejdůležitější místa, která jsou „rozstřelem" k různým nastavením: *Ovládací panely* a *Nastavení*. Právě přes *Nastavení – Soukromí* se nastavuje většina parametrů souvisejících s odesíláním osobních dat.

Čeho je dobré si všimnout a zvážit přenastavení – například (vše zde uvádět nebudeme):

- Vypnutí voleb souvisejících se službou *Telemetry*: tato služba odesílá data pro *Microsoft Customer Experience Improvement Program*, jehož oficiálním účelem je „zlepšování zákaznických zkušeností". Stojí za úvahu, co o sobě chceme takto pravidelně prozrazovat. V *Nastavení – Soukromí* na záložkách *Obecné*, *Řeč...*, *Informace o účtu*, *Další zařízení*, *Zpětná vazba...* jsou tyto volby dostupné.
- Služba *Wi-fi Sense* slouží ke sdílení hesel bezdrátových sítí, což funguje tak, že posílá všem dostupným kontaktům heslo každé sítě, u které to odsouhlasíme. V principu to je zneužitelné, nebezpečené i proto, že mnozí uživatelé „odklepnou" vše, co je jim naservírováno. Heslo vlastní Wi-fi sítě je VELMI soukromou informací.
- *Zjišťování polohy*: opět zneužitelná volba, navíc v mnoha zařízeních nemá smysl (hlavně stacionárních). Najdeme ji v *Nastavení – Poloha*.
- *Cortana* je „osobní asistentka" naprogramovaná s využitím umělé inteligence, jakási vyšší úroveň sémantického vyhledávání. Ovšem zatím funguje jen v některých zemích (u nás ne a na nastaveném jazyce to nezávisí), což jí ale nevadí v shromažďování a odesílání informací, které údajně ke své činnosti potřebuje (personifikaci odpovědí). Rozhodně je lepší ji vypnout (jak kvůli transportu osobních dat, tak i pro zbytečné zatěžování sítě): v *Nastavení – Soukromí – Řeč...*; podrobněji o vypnutí Cortany na
  https://365tipu.wordpress.com/2015/08/04/jak-vypnu-cortanu-ve-windows-10-je-cortana-bezpecna/

Samozřejmě je dobré (alespoň u znalého uživatele) provést podobné změny jako u předchozích verzí – zobrazení ikony Počítač (Tento počítač) na ploše, ať máme přes jeho kontextové menu snadný přístup k nejdůležitějším nástrojům, zapnutí zobrazování přípon známých typů souborů, zobrazování skrytých a chráněných souborů, vypnutí zjednodušeného sdílení, atd.

Bohužel někteří uživatelé zjistili, že u konkrétních voleb i přes jejich vypnutí určitá komunikace probíhá dál, to však jako uživatelé již nejsme schopni ovlivnit. Ovlivnit můžeme ovšem to, s jakými weby komunikujeme, která data svěříme samotným Windows, které aplikace si nainstalujeme,...

To byl jen krátký výčet voleb, které by nás mohly zajímat. Podrobnější údaje najdete na odkazech uvedených níže.

Další informace:

- http://www.lupa.cz/clanky/jak-je-to-s-windows-10-a-zasadnim-utokem-na-nase-soukromi/
- http://www.mrpear.net/cz/blog/885/ochrana-soukromi-jak-nastavit-windows-10-pri-instalaci-a-po-ni

- http://www.root.cz/clanky/windows-10-analyza-probihajici-komunikace/ (ovšem pozor – analýza probíhala po vypnutí naprosto všech voleb jakkoliv souvisejících se soukromím)
- http://aeronet.cz/news/analyza-windows-10-ve-svem-principu-jde-o-pouhy-terminal-na-sber-informaci-o-uzivateli-jeho-prstech-ocich-a-hlasu/

Souhrn kapitoly

V této kapitole jsme se zabývali tématy souvisejícími s ochranou osobních dat. Nejdřív jsme se zaměřili na obchod s osobními daty, ať už legální nebo nelegální, podívali jsme se na smluvní podmínky, se kterými často souhlasíme i bez jejich přečtení, dotkli jsme se ochrany dat na sociálních sítích, dále jsme se podívali na fungování cookies a technologie FLoC, zjistili jsme, jak se dá ověřit odcizení identity, jak eliminovat následky odcizení zařízení, a na konci jsme si prodiskutovali ochranu osobních dat v operačních systémech.

Kapitola 4

Úvod do kryptografie a ověřování dat

Co se bezpečnosti týče, jedním ze základních kamenů je kryptografie, jejíž metody používáme zejména při šifrování, ověřování integrity dat (zda data při transportu nebyla poškozena, pozměněna či podvržena) a ověření identity odesílatele.

4.1 Co je to kryptografie

Nejdřív vysvětlíme několik základních pojmů.

✎ Pojem *kryptografie* pochází z řečtiny – ze slov kryptós (skrytý) a gráphein (psát). V současné češtině tak označujeme vědu o metodách utajení smyslu zprávy převodem do takového tvaru, jehož smysluplné čtení vyžaduje určitou utajenou znalost. Je to tedy věda o procesu šifrování.

✎ *Kryptoanalýza* je oproti tomu věda o metodách získávání obsahu šifrované zprávy bez přístupu ke zmíněné utajené znalosti, která by za jiných okolností byla k dešifrování nutná. Je to tedy věda o prolamování šifrování.

Kryptografie a kryptoanalýza spolu velmi úzce souvisejí – nejsou to jen „protivníci“, jak by se na první pohled mohlo zdát. Například algoritmy kryptografie se běžně ověřují pomocí algoritmů kryptoanalýzy, aby byly včas odchyceny potenciální slabiny šifrovacího kódu.

Moderní kryptografie využívá různé matematické prvky, které mají znesnadnit kryptoanalýzu, například jednosměrné funkce a hash funkce.

✎ *Jednocestná funkce* (také jednosměrná) je taková funkce $f : x \to y$, že je sice relativně snadné pro dané x vypočíst příslušné $y = f(x)$, ale naopak už je to horší. Pokud známe y, je pro nás technicky neproveditelné zjistit x takové, že $f(x) = y$. To, že je tento proces pro nás technicky neproveditelný, ještě neznamená, že je nemožný, jen při současném stavu znalostí a vybavení není realizovatelný v dostatečně krátkém čase.

Jednocestné funkce se často používají právě v šifrovacích alogoritmech.

✎ *Hash funkce* je taková jednocestná funkce $f : x \to y$, která je bezkolizní: je prakticky nemožné podle výsledku funkce y zpět sestrojit příslušný obraz (x takové, že $f(x) = y$). Výsledek hash funkce se také nazývá *digitální otisk* (nebo taky jednoduše „hash“). Další specifickou vlastností hash funkcí je rychlost (zejména oproti jiným jednocestným funkcím).

Výsledek y hash funkce je typicky jen krátký (často má fixní délku, ať už je délka vstupu x jakákoliv), dokonce pro více různých hodnot x může vyjít tatáž hodnota $f(x)$ (ovšem pravděpodobnost je velmi blízká nule), ale zároveň i malá změna v hodnotě x způsobí hodně velkou změnu ve výsledné hodnotě $f(x)$. Proto je nerealizovatelné modifikovat původní data x tak, aby po modifikaci měla stejný digitální otisk $f(x)$.

Důsledkem je, že je prakticky nemožné zfalšovat digitální otisk daného řetězce – stačí na testovaná data znovu spustit příslušný algoritmus, vytvořit kontrolní digitální otisk a hned víme, jestli data nebyla modifikována.

Poznámka:

Hash funkce se dnes používají i při autentizaci uživatele při přihlašování do systému. Jestliže při přihlašování zadáme heslo, z tohoto hesla je vytvořen hash. Systém si ve speciálním souboru či databázi eviduje hashe všech uživatelů, kteří se do systému mohou přihlásit, tedy stačí srovnat hash zadaného hesla s tím uloženým. Například v Linuxu jde obvykle o soubor `/etc/shadow`.

Z toho je zřejmé, že nikde v systému nejsou samotná hesla nikde uložena. Pokud uživatel heslo zapomene a systém má uloženy jen hashe, není možné jednoduše požádat administrátora o znovuzaslání hesla (protože ani admin se k heslu nedostane), ale po ověření jiným způsobem si uživatel může nastavit nové heslo (je tedy vygenerován a uložen nový hash).

Jinými slovy – pokud (například u nějaké služby na webu) je možné v případě zapomenutí hesla požádat o znovuzaslání (původního) hesla, zajímejte se o to, jakým způsobem jsou vlastně hesla chráněna.

4.2 Šifrování – o co jde

Laici mají dojem, že v kryptografii jde pouze o utajení přenášené zprávy, ale ve skutečnosti jsou metody kryptografie využívány především k dosažení těchto cílů:

- zajištění *důvěrnosti dat* – utajujeme data před neoprávněnými osobami, takovým mají být data nesrozumitelná,
- zajištění *integrity dat* – data nemají být během přenosu pozměněna (ať už úmyslně nebo neúmyslně), resp. pokud jsou pozměněna, poznáme to,
- zajištění *nepopiratelnosti původu* – je zřejmé a ověřitelné, kdo zprávu odeslal (odesílatel nemůže být zfalšován),
- *autentizace entity* – ověřuje se identita dané entity (například uživatele při přihlašování, zařízení při komunikaci mezi zařízeními, procesu, služby apod.),
- *autentizace zprávy* – ověřujeme, zda k příjemci dorazila tatáž zpráva, kterou odeslal odesílatel (vpodstatě odpovídá testování integrity dat).

Vedle toho se kryptografie používá také při důvěryhodném monitorování (například správce sítě monitoruje svou síť, případně v digitální forenzní analýze), při řízení přístupu a autorizaci (přidělování konkrétních oprávnění), důvěryhodném přidávání časového údaje v logu, atd. Nás budou zajímat především postupy pro zajištění důvěrnosti a integrity dat a nepopiratelnosti původu.

Otevřeným textem nazýváme data, která mají být šifrována. *Ciphertext* (šifrovaný text) je výsledek šifrování, tedy zašifrovaná data.

Při šifrování potřebujeme:

- data k šifrování (otevřený text),
- šifrovací algoritmus, resp. nástroj (třeba program), který tento algoritmus implementuje,
- kryptografický klíč.

Šifrovací (kryptografický) algoritmus je *statická část* celého procesu, nemění se. Určuje, jak konkrétně budou šifrovaná data transformována. Kód nebo pseudokód šifrovacích algoritmů je dokonce obvykle veřejný. Pokud bychom použili pouze šifrovací algoritmus, data by byla vždy zašifrována naprosto stejným způsobem a šifrování by nemělo valný smysl – bylo by snadné šifru prolomit a postup by nebyl použitelný například pro ověřování původu dat.

Proto potřebujeme *proměnnou část* šifrovacího procesu – kryptografický klíč. Kvalita šifrování je samozřejmě hodně ovlivněna použitým algoritmem a nastavením parametrů tohoto algoritmu (módem), ale zásadní vliv má i kvalita klíče. Obecně platí, že čím delší klíč, tím lépe, ale záleží i na jeho struktuře. Při běžném použití je totiž klíč často generován z textového hesla, a pokud si zvolíme slabé heslo, pak ani vygenerovaný klíč není kvalitní.

Rozlišujeme tyto *druhy šifrování*:

- symetrické šifrování – používá se tentýž klíč pro šifrování i dešifrování, tzv. symetrický klíč,
- asymetrické šifrování – potřebujeme dvojici klíčů – jeden klíč je určen pro šifrování, druhý pro dešifrování.

Tyto dva druhy šifrování tedy především rozlišujeme podle toho, zda lze či nelze pro dešifrování použít tentýž klíč jako pro šifrování, rozdílů je však více. V obou případech potřebujeme algoritmus pro šifrování E a algoritmus pro dešifrování D (tyto dva algoritmy jsou navzájem inverzní), ale rozdíl je v časové náročnosti těchto algoritmů – obvykle platí, že symetrické algoritmy jsou mnohem rychlejší než asymetrické, a jsou tedy použitelnější pro větší kvanta dat.

4.3 Symetrická kryptografie

Symetrické šifry mohou být *proudové* nebo *blokové*. Proudová šifra se aplikuje postupně na všechny znaky otevřeného textu (tj. na proud dat), bloková šifra nejdřív rozdělí otevřený text na bloky o stanovené velikosti a pak pracuje s těmito jednotlivými bloky. Obvykle potřebujeme algoritmus šifrování E a dešifrování D, který může mít „proměnné“ parametry (jako funkce).

4.3.1 Proudové šifry

Proudová šifra obvykle funguje takto: Máme k dispozici *klíč*. Tento klíč je použit k vygenerování posloupnosti tvořící pseudonáhodné *heslo* $h_1 \cdots h_n$. Pokud máme k dispozici klíč, dokážeme tutéž posloupnost (stejnou) vygenerovat kdykoliv znovu.

Každý (i-tý) znak otevřeného textu je šifrován funkcí $\mathrm{E}(i, h_i, zn)$ závislou na pořadí znaku a i-té pozici hesla. Podobně dešifrování probíhá tak, že každý i-tý znak (nebo skupinu znaků) šifrovaného textu dešifrujeme funkcí $\mathrm{D}(i, h_i, zn)$.

Postup

Vzorec pro proudovou šifru je následující:

- vstupem je otevřená posloupnost znaků $m_1 m_2 \cdots m_n$ a klíč k,
- generátor hesla vygeneruje z klíče k heslo $h(k) = h_1 h_2 \cdots h_n$,
- výstupem je posloupnost znaků $c_1 c_2 \cdots c_n$ taková, že $c_i = \mathrm{E}(i, h_i, m_i)$ pro $1 \leq i \leq n$.

Dešifrování probíhá přesně naopak:

- vstupem je zašifrovaná posloupnost znaků $c_1 c_2 \cdots c_n$ a klíč k,
- generátor hesla vygeneruje z klíče k heslo $h(k) = h_1 h_2 \cdots h_n$,
- výstupem je posloupnost znaků $m_1 m_2 \cdots m_n$ taková, že $m_i = \mathrm{D}(i, h_i, c_i)$ pro $1 \leq i \leq n$.

Funkce E a D jsou navzájem inverzní, tedy platí $\mathrm{D}(i, h(i), c_i) = \mathrm{E}^{-1}(i, h(i), m_i)$.

4.3.2 Módy blokových šifer pro zajištění důvěrnosti dat

Zaměřme se nyní na blokové šifry. S bloky se dá zacházet různými způsoby, tedy algoritmy mohou pracovat v různých módech (mód, tedy režim, je obvykle určen konkrétní dynamicky linkovanou knihovnou, kterou programátor do aplikace přilinkuje).

Velikost bloku obvykle odpovídá velikosti klíče, ale ne vždy. Některé módy používají velké bloky (wide-block encryption), jiné spíše menší bloky (narrow-block encryption).

Mód ECB (Electronic Cipher Book). Jedná se o nejjednodušší mód pro blokové šifrování, je definován v NIST SP 800-38A[3]. V tomto režimu nejsou brány v úvahu žádné vztahy mezi bloky – je jedno, jestli je daný blok na začátku, uprostřed nebo na konci, je jedno, jakými bloky je obklopen.

Máme k dispozici klíč k. K danému bloku otevřeného textu na vstupu je šifrovací funkcí E_k vytvořen šifrovaný blok, bez ohledu na kontext (okolní bloky). V opačném směru je na šifrovaný blok použita dešifrovací funkce D_k a výsledkem je dešifrovaný blok.

Postup

Pro ECB je vzorec velice jednoduchý:

- vstupem je otevřená posloupnost bloků $b_1 b_2 \cdots b_n$ a klíč k,
- výstupem je posloupnost šifrovaných bloků $c_1 c_2 \cdots c_n$ taková, že $c_i = \mathrm{E}_k(b_i)$ pro $1 \leq i \leq n$.

Dešifrování probíhá přesně naopak:

- vstupem je zašifrovaná posloupnost bloků $c_1 c_2 \cdots c_n$ a klíč k,
- výstupem je posloupnost bloků $b_1 b_2 \cdots b_n$ taková, že $b_i = \mathrm{D}_k(c_i)$ pro $1 \leq i \leq n$.

Problém je, že pokud je v otevřeném textu více stejných bloků, na výstupu budou tyto bloky taky stejné, a případný útočník má podstatnou nápovědu k odhadnutí obsahu původního otevřeného textu. Další nebezpečí je v tom, že bez sledování návaznosti mezi bloky je snadné některý blok zaměnit za jiný a podstrčit tak do konverzace data, která tam původně nebyla.

Poznámka:

Jedná se o nejjednodušší režim práce blokových algoritmů, bohužel na mobilních zařízeních je až příliš běžný. Například v aplikacích pro Android, které používají šifrování, se jedná o výchozí mód (včetně případu, kdy použijeme jinak silný algoritmus AES) – pokud explicitně neurčíme mód, použije se ECB. Programátoři se často nenamáhají mód určit, tedy ECB je v androidích aplikacích zastoupen v cca 70 % (údaj ze začátku roku 2015).

Mód CBC (Cipher Block Chaining). Tento mód (NIST SP 800-38A[3]) je dnes běžný v oblastech, kde opravdu jde o bezpečnost (například při autentizaci v internetovém bankovnictví). Používá se také hodně pro šifrování datových médií, třebaže pro tento účel není příliš vhodný.

První blok se šifruje stejně jako v módu ECB, ale u všech následujících bloků je zajištěna vazba na předcházející blok tímto způsobem: vezmeme předchozí blok (už zašifrovaný) s právě šifrovaným blokem a použijeme na ně funkci XOR po jednotlivých bitech – logickou nonekvivalenci, tedy $(x \text{ XOR } y = 1) \Leftrightarrow (x \neq y)$, jinak $(x \text{ XOR } y = 0)$.

První blok (který není podroben funkci XOR) by mohl být Achillovou patou této metody – pokud odesíláme více zpráv, které stejně začínají, zvýšíme pravděpodobnost napadení. Proto se používá *IV vektor – inicializační vektor* (nějaké náhodné číslo, například časové razítko), přičemž se provede funkce XOR na IV vektor a první blok. U druhého a dalších bloků použijeme místo IV vektoru šifrovaný předchozí blok. Tedy pro šifrování i-tého bloku je použita funkce $\mathrm{E}_k(\mathrm{st}_i)$ závislá na klíči a předchozím zašifrovaném bloku (u prvního bloku IV vektoru).

Ovšem pro dešifrování je třeba znát IV vektor, který je obvykle posílán (nezašifrovaný) zároveň se zprávou. Nemusí být šifrovaný, protože hlavního cíle (odlišení prvního bloku) bylo dosaženo. Navíc se vlastnost náhodnosti zprostředkovaně dostává i k dalším blokům (protože výsledek pro každý následující blok je ovlivněn výsledkem předchozího bloku).

Postup

Postup pro mód CBC:

- vstupem je otevřená posloupnost bloků $b_1 b_2 \cdots b_n$ a klíč k,
- je vygenerován IV vektor, který pro účely vzorce označíme c_0 (bude použit jako „virtuální“ nultý blok, ale skutečné zašifrované datové bloky jsou až od indexu 1),
- výstupem je posloupnost šifrovaných bloků $c_1 c_2 \cdots c_n$ taková, že $c_i = \mathrm{E}_k(b_i \text{ XOR } c_{i-1})$ pro $1 \leq i \leq n$.

Posíláme posloupnost bloků $c_1 \cdots c_n$ a IV vektor (tj. c_0). Dešifrování probíhá takto:

- vstupem je zašifrovaná posloupnost bloků $c_1 c_2 \cdots c_n$, klíč k a IV vektor c_0,
- výstupem je posloupnost bloků $b_1 b_2 \cdots b_n$ taková, že $b_i = c_{i-1} \text{ XOR } \mathrm{D}_k(c_i)$ pro $1 \leq i \leq n$.

Díky navazování bloků má tento mód kromě lepšího zabezpečení ještě jednu důležitou výhodu – schopnost zotavení po chybě: pokud jsou některé bloky poškozené (například technickými problémy při přenosu), je možné při dešifrování získat bezchybný text. Je je potřeba, aby alespoň dva některé po sobě jdoucí bloky byly bezchybné.

Módy CFB (Cipher Feedback Mode) a OFB (Output Feedback Mode). Tyto módy (def. v NIST SP 800-38A[3]) přepínají blokovou šifru do proudového režimu – používá se blokový algoritmus, ale místo použití přímo pro šifrování je jím generováno heslo (po blocích), se kterým se následně zachází stejně jako při proudovém šifrování (funkce XOR na heslo a otevřený text).

Nejdřív se vygeneruje náhodné číslo – IV vektor. Z něj je vytvořen první (vpodstatě nultý) blok hesla. Zbytek hesla je již u módů CFB a OFB generován odlišně – u OFB je každý další blok hesla závislý jen na předchozím bloku hesla, kdežto u CFB se přidává vliv předchozího zašifrovaného bloku (podobně jako u CBC).

Postup

Šifrování v módu OFB:

- vstupem je otevřená posloupnost bloků $b_1 b_2 \cdots b_n$, klíč k a IV vektor (poč. heslo) h_0,
- v každém kroku (pro každý blok dat) nejdřív vytvoříme heslo pro daný krok $h_i = \mathrm{E}_k(h_{i-1})$, i-tý blok b_i šifrujeme takto: $c_i = b_i \text{ XOR } h_i$; pro $1 \leq i \leq n$.

Dešifrování v módu OFB:

- vstupem je zašifrovaná posloupnost bloků $c_1 c_2 \cdots c_n$, klíč k a IV vektor (heslo) h_0,
- v každém kroku nejdřív vytvoříme heslo pro daný krok $h_i = \mathrm{E}_k(h_{i-1})$, i-tý blok c_i dešifrujeme takto: $b_i = c_i \text{ XOR } h_i$; pro $1 \leq i \leq n$.

Oproti předchozím blokovým šifrám je zde zajímavé, že pro šifrování i dešifrování používáme tentýž algoritmus – E. Opravdu to není překlep, tentýž algoritmus slouží i při dešifrování. Proč? Protože tento algoritmus používáme ke generování posloupnosti hesla, která musí být stejná pro šifrování i dešifrování, kdežto pro samotný proces zpracování dat využíváme funkci XOR, která je přirozeně symetrická: můžete si jednoduše tabulkovou metodou ověřit, že $(A \text{ XOR } B) \text{ XOR } B = A$.

Postup

Šifrování v módu CFB:

- vstupem je otevřená posloupnost bloků $b_1 b_2 \cdots b_n$, klíč k a IV vektor c_0,
- výstupem je posloupnost šifrovaných bloků $c_1 c_2 \cdots c_n$ taková, že $c_i = b_i \text{ XOR } \mathrm{E}_k(c_{i-1})$ pro $1 \leq i \leq n$.

Dešifrování v módu CFB:

- vstupem je zašifrovaná posloupnost bloků $c_1 c_2 \cdots c_n$, klíč k a IV vektor c_0,
- výstupem je posloupnost bloků $b_1 b_2 \cdots b_n$ taková, že $b_i = c_i \text{ XOR } \mathrm{E}_k(c_{i-1})$ pro $1 \leq i \leq n$.

Jak vidíme, mód CFB má s módem OFB společný způsob uplatněnění funkce E (slouží k vytváření hesla, tudíž není nutné použít při dešifrování inverzní funkci). Ovšem rozdíl je v tom, že vytváření hesla je propojeno s šifrováním a dešifrováním tím způsobem, že heslo pro následující krok je závislé na výsledku šifrování/dešifrování z předchozího kroku. Tedy se vlastně jedná o kombinaci OFB a předchozího CBC.

To, že při použití módů CFB a OFB není nutné mít zvláštní inverzní algoritmus pro dešifrování, je výhodou při *hardwarové implementaci* – šetříme integrované obvody (stačí jejich poloviční množství oproti implementaci jiných módů).

Mód CTR (Counter). Tento mód vznikl modifikací módu OFB, kde se parametr funkce E_k pro generování hesla pro další krok odvozuje postupnými aritmetickými úpravami IV vektoru (místo závislosti na hesle z předchozího kroku). Dobře se paralelizuje, je rychlý. Je také definován v NIST SP 800-38A[3] a vzorec s komentářem najdeme například ve zdroji [10] (společně s předchozími zde uvedenými).

Solení (Salting). Jedná se o drobnou úpravu módů používajících IV vektor (tj. všech předchozích kromě ECB). Účelem je snížit na minimum nebezpečí plynoucí z toho, že IV vektor je třeba přeposlat zároveň s šifrovaným textem. Samotný IV vektor c_0 se sice volně přikládá ke zprávě, ale při šifrování a dešifrování se místo něj použije hodnota c'_0 vzniklá úpravou původního IV vektoru předem domluveným způsobem.

Mód EDE (Encryption-Decryption-Encryption). Jedná se o vylepšení módu CBC přidáním jednoho nebo dvou klíčů (tedy máme celkem dva nebo tři klíče) a kombinací algoritmu šifrování a dešifrování (viz poznámku). Použijí se vždy tři klíče s tím, že pokud jsme zadali dva, pak je třetí totožný s prvním (tedy buď k_1, k_2, k_3 nebo k_1, k_2, k_1).

Postup

Šifrování v módu EDE probíhá takto:

- vstupem je otevřená posloupnost bloků $b_1 b_2 \cdots b_n$ a klíče k_1, k_2, k_3, přičemž může platit $k_1 = k_3$ (záleží, jestli máme dva nebo tři klíče), dále IV vektor,
- máme algoritmy E (šifrovací) a D (dešifrovací), které jsou navzájem inverzní,
- výstupem je posloupnost šifrovaných bloků $c_1 c_2 \cdots c_n$, kde $c_i = \mathrm{E}_{k_3}\Big(\mathrm{D}_{k_2}\big(\mathrm{E}_{k_1}(b_i)\big)\Big)$ pro $1 \leq i \leq n$ (plus použití IV vektoru).

Vzorec možná vypadá složitě, ale jen nám říká, že blok je nejdřív zpracován algoritmem E s využitím klíče k_1 (tj. nejvnitřnější část vzorce), pak algoritmem D s klíčem k_2 a nakonec algoritmem E s klíčem k_3 (nebo k_1).

Dešifrování je pak přesně opačné – nejdřív algoritmus D_{k_3}, pak E_{k_2} a následně D_{k_1}.

Poznámka:

Pokud v módu EDE použijeme všechny tři klíče stejné, metodu „degradujeme“ na CBC.

Módy LRW (Liskov, Rivest, Wagner) a XEX (Xor-Encrypt-Xor). Tyto režimy jsou určeny pro šifrování dat na paměťových médiích, zejména pevných discích. Otevřený text (zde obsah disku) je rozdělen na malé bloky stejné velikosti tak, aby se do jednoho sektoru vešlo více bloků (metody se označují „tweakable narrow-block encryption“, tedy bloky jsou opravdu menší než sektor).

Do procesu šifrování vstupuje šifrovaný blok, dále tzv. *tweak* (index tohoto bloku na paměťovém médiu nebo jiná specifikace bloku) a dva klíče. První klíč je šifrovací, druhý klíč (tweakovací) má stejnou velikost jako blok a plní podobnou úlohu jako IV vektor u předchozích módů.

Pro každý blok b_i se nejdřív pomocí tweakovacího klíče vygeneruje z tweaku pomocná hodnota x_i (různě podle konkrétního módu), která se pak využije k vytvoření výsledného šifrovaného bloku následovně: $c_i = \mathrm{E}_k(b_i \text{ XOR } x_i) \text{ XOR } x_i$.

Tyto módy byly v praxi nahrazeny módem XTS, viz dále.

Mód XTS (XEX-based Tweaked-codebook mode with ciphertext Stealing). Tento mód vznikl úpravou módu XEX (je již uváděn přímo jako mód pro algoritmus AES, ale použitelný i pro jiné algoritmy), je definován v NIST SP 800-38E[5] a standardizován jako IEEE 1619-2007. V současné době je pro účely šifrování paměťových médií používán asi nejčastěji.

Je velmi podobný režimu XEX, ale jiným způsobem se zachází s klíčem – celkový klíč zadaný uživatelem se rozdělí na poloviny, tím získáme dva různé klíče nezávislé jak navzájem, tak i na šifrovaných datech či jejich pozici na paměťovém médiu. Do procesu šifrování vstupuje také index sektoru, index šifrovaného bloku v sektoru (v jednom sektoru je víc šifrovaných bloků, podobně jako u XEX).

Mód GCM (Galois/Counter Mode). Tento mód je již obecnější, existuje i varianta pro využití při autentizaci (GMAC). Jak název napovídá, vznikl úpravou módu Counter (CTR), přičemž se používá Galoisovo pole (GF – Galois Field). Je popsán v NIST SP 800-38D[4], setkáme se s ním také například v RFC 4106 (použití GCM v IPSec) a dalších standardech a doporučeních.

Tento mód je poměrně rychlý a přitom považován za bezpečný, proto se s ním setkáváme zejména při šifrování dat přenášených v počítačových sítích, je také používán v TLS v. 1.2.

4.3.3 Módy pro zajištění integrity dat

Zatímco u důvěrnosti dat jde o to, jak případnému útočníkovi data znečitelnit, u integrity není ani tak důležité, aby si útočník zprávu nepřečetl, ale aby ji nemohl nepozorovaně pozměnit nebo podstrčit jinou (případně chceme zjistit, zda data nebyla během přenosu pozměněna vlivem poruchy).

Některé z předchozích módů mají i tuto vlastnost, která je často dovedena až na úroveň samoopravitelnosti (například už módy CBC a CFB odhalí a do určité míry dokážou při dešifrování opravit poškozená data, pokud jsou před nimi alespoň dva bloky nepoškozené), ale za určitých okolností nemusí být chyba zjištěna. Existují však módy, jejichž hlavním účelem je právě hlídat integritu dat.

Výstupem algoritmu běžícího v módu pro zajištění integrity je krátký řetězec – *autentizační kód*. Tento řetězec připojíme ke zprávě (původnímu otevřenému textu nebo šifrovanému textu vzniklému podle jiného módu). Příjemce použije stejný postup, a pokud získá tentýž řetězec, pak lze říct, že zpráva nebyla během přenosu modifikována.

Pokud však bychom použili pouze takovýto jednoduchý postup, moc by to proti záměrnému pozměnění nepomohlo – šifrovací algoritmy (včetně těch, které vytvářejí autentizační kódy) jsou veřejně známy, tedy pokud útočník zachytí naši zprávu, může ji pozměnit, vytvořit autentizační kód pozměněné zprávy, přiložit a poslat dál. Proto i zde potřebujeme *klíč*, který mají pouze odesílatel a příjemce, nikoliv třetí osoba. Jestliže však chceme tuto metodu použít pro zjištění případného

poškození vlivem přenosových chyb, klíč není potřeba.

Mód MAC (Message Authentication Code). Tento mód využívá ke generování autentizačního kódu mód CBC s nulovým IV vektorem. Otevřený text je rozdělen na bloky a zpracován tak jak je popsáno pro mód CBC na straně 101 s použitím klíče k_1, ale průběžně vytvářené šifrované bloky $c_1, \ldots, c_{n-1}$ jsou po použití zahazovány, výstupem je pouze poslední šifrovaný blok c_n.

Na rozdíl od původního CBC může proběhnout ještě jeden šifrovací krok navíc (vstupem je místo otevřeného bloku šifrovaný blok c_n), přičemž by měl být použit klíč k_2 odlišný od původního klíče k_1, a navíc se z výsledku bere jen určitý počet bitů.

Postup

Autentizační kód otevřeného textu $b_1 b_2 \cdots b_n$ získáme takto:

- potřebujeme klíče k_1 a k_2, otevřený text dorovnáme na násobek velikosti bloku,
- použijeme mód CBC s nulovým IV vektorem – $c_0 = 0 \cdots 0$:
  $c_i = \mathrm{E}_k(b_i \text{ XOR } c_{i-1})$ pro $1 \leq i \leq n$,
- blok c_n projde znovu šifrovacím algoritmem, tentokrát s druhým klíčem, výsledek usekneme na požadovanou délku:
  $mac = \mathrm{cut}(\mathrm{E}_{k_2}(c_n), \mathrm{delka})$

Odešleme otevřený text $b_1 b_2 \cdots b_n$, vytvořený kód mac a předpokládáme, že příjemce má oba klíče k_1 a k_2. Příjemce pak provede naprosto stejný proces, vypočte svou „verzi“ autentizačního kódu a ověří si, zda byla zpráva cestou modifikována.

Proč může být IV vektor nulový? Protože blok c_1 (který by právě byl problémem) se neposílá, je během generování kódu zahozen. Výhodou je, že není nutné se zprávou přenášet IV vektor.

Protože se jako základ používá mód CBC, je tento mód také označován jako CBC-MAC. Existuje více různých variant, všechny jsou součástí standardu ISO/IEC 9797-1. Některé varianty jsou dnes běžně používány.

Módy HMAC a CMAC. Mód CMAC (Cipher-based MAC, NIST SP 800-38B, RFC 4493, rok 2006) je modifikací původních CBC-MAC kódů, jedná se tedy o blokovou autentizační šifru. Oproti původním MAC kódům nabízí větší bezpečnost. Místo základního módu CBC byl zde původně použit mód XCBC, v novějších variantách jde o vylepšený mód OMAC (původní XCBC vyžadoval 3 klíče, novější OMAC si vystačí s jediným).

Mód HMAC (Hash-based MAC, zveřejněn jako RFC 2104, rok 1997) by správně neměl být řazen k blokovým módům, jedná se o schéma pro použití hash funkcí pro účely autentizace dat. Tento mód nepoužívá žádný IV vektor, navíc je díky využití hash funkcí rychlý. Mohou být použity různé hash funkce, příslušná hash funkce je pak součástí názvu: například HMAC-MD5.

Mód GMAC (GCM-MAC). Tento mód je založen na šifrovacím módu GCM, přičemž autentizační kód vytváříme z posledního vytvořeného šifrovaného bloku (jako u běžných MAC kódů). Na rozdíl od předchozích používá IV vektor (což je považováno za nevýhodu), ale na druhou stranu může fungovat s hardwarovou podporou, což jej činí rychlým. Zároveň s módem GCM je popsán v NIST SP 800-38D[4].

4.4 Příklady symetrických kryptovacích algoritmů

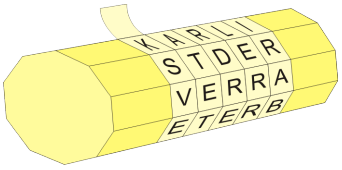
4.4.1 Historické algoritmy

Caesarova šifra. Jedná se o jednoduchou monoalfabetickou substituční šifru. Substituční znamená, že určité úseky textu nahrazujeme jinými speciálním postupem, který zná jak odesílatel, tak příjemce. Monoalfabetická znamená, že nahrazujeme každé písmeno jiným písmenem (délka úseku je 1 znak). Může jít o tabulku, kde je určeno, které písmeno má být kterým nahrazeno, nebo může jít o číslo určující, o kolik znaků v abecedě se máme při šifrování posunout jedním směrem a při dešifrování zase směrem opačným.

Julius Caesar často posílal po kurýrech depeše, které neměly být přečteny nikým nezasvěceným, proto zvolil druhou možnost. Při šifrování mělo být každé písmeno zprávy nahrazeno písmenem o 3 znaky dál v abecedě, dešifrování probíhalo opačně. Například při posloupnosti znaků v abecedě $A, B, C, D, E, F, G, \ldots$ napsal místo znaku B znak E, apod.

Monoalfabetické substituční šifry nejsou odolné proti *frekvenční analýze* – stačí si uvědomit, že v daném jazyce má každé písmeno určitou frekvenci výskytu (například v češtině je písmeno „a“ hodně časté, kdežto například „x“ spíše naopak) a tato frekvence je známá (dají se sehnat frekvenční tabulky daného jazyka). Proto stačí mít k dispozici dostatečně rozsáhlý šifrovaný text, který obsahuje relativně běžné věty daného jazyka, opatřit si frekvenční tabulku písmen, a po určitém tápání můžeme dešifrovat.

Homofonní substituční šifry nahrazují jedno písmeno jedním z definované množiny písmen (například pro jedno písmeno mohou být dána 3 cílová), což sice ztěžuje frekvenční analýzu, nicméně až tak velkou překážkou to není.



Skytalé. Je to typická transpoziční šifra, konkrétně z antické Sparty. Později byla oprášena za americké občanské války. Transpoziční šifry nezaměňují jeden znak za jiný, ale znaky v rámci zprávy přehazují, mění jejich pořadí.

Aby nebylo dešifrování příliš náročné, probíhá transpozice obvykle tak, že se text zapíše do řádků o stejné (předem dané) délce a pak se přepíše po sloupcích (nebo naopak).

V podání Skytalé se použil dřevěný válec o určitém průměru, na něj se spirálovitě namotal pásek pergamenu nebo jiného materiálu tak, aby smyčky nepřekrývaly, a následně se na smyčky v řádcích po délce válce zapsala zpráva – jak vidíme na obrázku vpravo[1] převzatém z Wikipedie. Po rozmotání jsme měli pás se zdánlivě náhodně uspořádanými znaky, pro dešifrování bylo nutné tento pás namotat na válec o stejném průměru.

Při kryptoanalýze tohoto typu transpozice stačí znát délku sloupce použitou při přepsání (např. průměr válce) – periodu.

Playfairova šifra. Tato šifra byla používána britskými jednotkami v druhé světové válce. Je to polygramová substituční šifra, tedy algoritmus nepracuje s jednotlivými znaky, ale s řetězci znaků (jeden řetězec je nahrazen jiným, zde jsou řetězce délky 2). Substituční tabulka je generována ze znaků hesla a znaků vyskytujících se v otevřeném textu a algoritmus stanovuje postup šifrování těchto řetězců.

[1] Zdroj obrázku: https://cs.wikipedia.org/wiki/Skytal%C3%A9

Vigenerova šifra. Je to zástupce polyalfabetických substitučních šifer. Tyto šifry jsou považovány za bezpečnější než předchozí, ale na druhou stranu je dešifrování složitější a časově náročnější (včetně kryptoanalýzy). Podobné šifry se také vyskytovaly ve starých špionážních filmech.

Potřebujeme tabulku znaků o rozměru abecedy (bez diakritiky 26×26 znaků). Řádky a sloupce jsou ohodnoceny stejně (jednoduše posloupnost písmen dané abecedy), přičemž označení řádků se chápe jako písmeno v otevřeném textu a označení sloupce jako znaky klíče. Jednotlivé sloupce tabulky obsahují také posloupnost znaků abecedy, ale nějakým způsobem modifikovanou, typicky posunutou o určitý počet písmen (obvykle jen o 1 písmeno), například první sloupec začíná písmenem A, druhý písmenem B, atd.

Při šifrování i dešifrování potřebujeme klíč. Je to posloupnost znaků určujících sloupce, které mají být v jednotlivých krocích použity. Například pokud je klíčem posloupnost $DUNAJ$, je první písmeno šifrováno podle sloupce D (výsledek je obsah buňky v řádku daného písmene a v sloupci D), druhé podle sloupce U, atd. Pro dešifrování stačí znát tabulku a klíč, ale při kryptoanalýze stačí k úspěchu už jen to, že známe délku klíče (tj. periodu, počet znaků, po kterém se opakuje použití jednotlivých sloupců).

Enigma. Jedná se o stroj používaný německými jednotkami v druhé světové válce (obrázek vpravo je ze zdroje [6], kde najdeme i simulátor původní Enigmy). Ve stroji byla sada rotorů, každý z nich prováděl určitou substituci (každý rotor se otáčel jinou rychlostí), přičemž otevřený znak určený k zašifrování postupně procházel všemi těmito substitucemi.

Enigma byla během války několikrát modifikována, přesto se pokaždé podařilo týmu britských a polských kryptoanalytiků její kód prolomit.

Vernamova šifra. O této šifře se mluví jako o neprolomitelné, což je nepochybně výhodou. Na druhou stranu je prakticky nemožné tuto šifru běžně používat.

Neprolomitelné šifry jsou založeny na jednorázovém hesláři (one-time pad), to znamená, že každý klíč může být použitý pouze jednou a navíc musí být absolutně utajen. Další podmínkou je, že klíč má stejnou délku jako šifrovaná zpráva (to souvisí s předchozí podmínkou – kdyby byl klíč kratší, bylo by nutné ho použít opakovaně na jednotlivé bloky otevřeného textu). Vernamova šifra splňuje tyto podmínky, přičemž při šifrování i dešifrování je na data a klíč použita operace XOR.

Proč je tato šifra v praxi nepoužitelná? Protože klíč musí být zcela náhodný (běžné generátory klíčů používají pouze pseudonáhodná čísla, což nedostačuje) a navíc musí být bezpečným způsobem distribuován mezi příjemcem a odesílatelem. Vzhledem k tomu, že pro každou zprávu bychom potřebovali unikátní klíč, byly by náklady na použití této šifry příliš velké.

Další informace:

První pokusy o šifrování se objevily už ve starověkém Egyptě a Mezopotámii, Caesar rozhodně primát nemá. Další náměty o historii šifrování najdeme například v prezentaci R. Pelánka

http://www.fi.muni.cz/~xpelanek/ucitele/data/janasifry/sifrovani_historicke_zajimavosti.pdf

4.4.2 Používané proudové šifry

Pro proudové šifry je typické, že je lze použít i v případech, kdy je k dispozici jen málo paměti (tj. nemají velké paměťové nároky), na rozdíl od blokových šifer, výhodou je také rychlost šifrování a dešifrování. S některými proudovými šiframi se setkáváme i v historii, například proudová byla Vernamova šifra.

RC4. Zkratka RC znamená Rivest Cipher (Rivestova šifra, podle Rona Rivesta). Jedná se o proudovou šifru (je zajímavé, že její předchůdce RC2 byl blokovou šifrou), která byla navržena roku 1987 pro firmu RSA Data Security, Inc.

Algoritmus byl původně utajen, ale dostal se na veřejnost roku 1994 metodou reverzního inženýrství (takto získaný algoritmus, vpodstatě verze RC4, je znám pod názvem Arcfour nebo ARC4 – Alleged RC4).

RC4 se používá jako jeden z možných algoritmů při šifrování webových stránek (HTTPS), obecně v SSL, v SQL, a dále ve Wi-fi sítích při volbě zabezpečení WEP. WEP je samo o sobě nepříliš bezpečné, RC4 je v něm implementována nekvalitně a navíc samotný algoritmus RC4 je považován za překonaný a nedoporučuje se jeho používání. Existuje množina slabých klíčů a samotný algoritmus má další slabiny.

Nástupcem RC4 je RC5, což je však bloková šifra, jejíž algoritmus je založen na principu Feistelovy šifry, viz dále.

A5. Tento algoritmus byl vyvinut pro šifrování GSM hovorů v mobilní síti (mezi základnovou stanicí a účastnickým zařízením, tedy na nejnižším stupni hierarchie sítě). I zde byl pokus o utajení algoritmu, který v případě prvních verzí nevyšel – opět díky reverznímu inženýrství. Klíč je uložen na SIM kartě telefonu.

Existuje několik variant: A5/1 je původní algoritmus využívající klíč o délce 64 bitů a v běžných implementacích je považována za ještě méně bezpečnou než DES. A5/2 je dokonce ještě slabší, účelem bylo zřejmě urychlit šifrování (používala se především ve východní Evropě a Asii). Pokračovatel A5/3 je již blokovou šifrou. Další (zatím neveřejná) varianta algoritmu je dodnes používána v sítích GPRS.

E0. Je to proudová šifra určená pro šifrování přenosu při použití Bluetooth. Při šifrování je využíván nejen klíč, ale také například BT adresa zařízení (48bitová adresa používaná při Bluetooth přenosu).

4.4.3 DES a 3DES

Algoritmus DES (Data Encryption Standard) vyšel jako vítěz ze soutěže uspořádané americkým NBS (National Bureau of Standards) v roce 1977 jako standard určený pro zabezpečení komunikace ve státní civilní sféře, přičemž byl standardizován jako FIPS PUB 46. Taktéž byl přijat jako standard ANSI X3.92 a ISO/IEC. Později se používání tohoto algoritmu rozšířilo i do soukromé sféry. Původní název byl DEA, byl vyvinut společností IBM.

Jedná se o blokovou šifru s délkou bloku 64 bitů. Šifrovací klíč má také délku 64 bitů, ale každý osmý bit klíče je paritní, tedy po odečtení parit získáme 56 „originálních“ bitů. Před lety to stačilo, v současné době jsou jak algoritmus, tak i délka klíče, považovány za nevyhovující.

Nyní se zaměřme na funkce E_k a D_k, jak byly používány v předchozím textu o módech šifrování. Algoritmus DES kombinuje postupy substitučních a transpozičních šifer – ve funkci E_k se na blok otevřeného textu používá jak substituce, tak i permutace (permutací je zde nazvána transpozice uplatněná na bity), to vše se opakuje 16×. Každé z 16 opakování se označuje jako *round* (česky runda).

Před první rundou je provede úvodní permutace, následuje 16 rund. V rámci rundy jsou zpracovávaná data půlena (na levou a pravou část), pravá část je funkcí XOR kombinována s určitou částí klíče (vybranými 48 bity, mění se pro každou rundu), substituována, permutována, pak funkcí XOR kombinována s levou částí, následně se levá a pravá část vymění a začíná další runda. Postup je podrobněji popsán a naznačen ve schématu například v [2] nebo [9].

Poznámka:

Tento algoritmus je typickou ukázkou tzv. *Feistelovy šifry* (podle Horsta Feistela, který princip použil v algoritmu Lucifer). Feistelova šifra určuje půlení zpracovávaného textu a opakované zpracovávání v rundách.

Tento princip je používán v mnoha symetrických blokových šifrách, protože příslušný algoritmus je snadněji implementovatelný.

DES již není považován za bezpečný z různých důvodů – především je to krátký klíč, ale také některé hodnoty klíče jsou považovány za slabé nebo poloslabé (pokud se v hexadecimálním zápisu klíče objevují určité vzory). Co se týče samotného algoritmu, jeho bezpečnost je hodně ovlivněna konkrétním módem použitým při šifrování – použitelné módy jsou definovány ve FIPS-81.

3DES (Triple DES) řeší slabiny algoritmu DES použitím módu EDE (viz str. 103, včetně vzorce a použití klíčů). Hlavním důsledkem použití tohoto módu je defacto ztrojnásobení délky klíče (pokud ovšem použijeme tři různé klíče), tedy místo 56 bitů máme až 168 bitů (plus paritní). Vše ostatní z algoritmu DES zůstává.

Tento algoritmus byl publikován jako NIST SP 800-67 Revision 1, také byl standardizován jako ISO/IEC 18033-3:2010 (Part 3), dále existují standardy ANSI a FIPS.

V současné době se postupně odkláníme i od tohoto algoritmu, třebaže je dosud používán. Jeho nevýhodou oproti novějším algoritmům je pomalost. Přesto se s ním setkáme například v bankovním sektoru nebo v některých e-mail klientech.

4.4.4 AES – Rijndael

Když algoritmus DES přestal vyhovovat, vypsal NBS další soutěž (roku 1997) – tentokrát mezinárodní a otevřenou. Po několika kolech zvítezil algoritmus Rijndael autorů Rijmena a Daemena z Belgie. Tento algoritmus je znám pod názvem AES (Advanced Encryption Standard), který byl vyhlašovateli soutěže předem připraven. AES byl roku 2002 standardizován jako FIPS PUB 197, taktéž ISO/IEC 18033-3. Informace jsou v [1], dále v [9] a [2].

Jedná se o blokovou šifru, která může běžet v různých módech a akceptuje tři různé délky klíčů – 128, 192 nebo 256 bitů, podle toho označujeme AES-128, AES-192 nebo AES-256. Délka bloku je 128 bitů (nezávisí na délce klíče).

Princip je podobný algoritmu DES, celý proces je také rozdělen do rund, počet rund závisí na délce klíče (čím delší klíč, tím víc rund proběhne: 10, 12 nebo 14 rund). I zde v rundě probíhá substituce, sada transpozic a kombinace s částmi klíče. Popis algoritmu AES-128 najdeme například v [13, str. 14–19].

Oproti DES (a 3DES) je AES odolnější proti prolomení – neexistují klíče, které jsou už z principu slabé, je také mnohem více odolný proti útokům typu brute-force. Ovšem záleží, který mód je pro šifrování algoritmem vybrán (programátorem příslušné aplikace), což už bylo diskutováno v sekci o módech.

S šifrováním AES se v běžném životě setkáváme poměrně často, protože je používáno například u Wi-fi, pokud máme nastaveno zabezpečení WPA2. Velkou výhodou je, že u většiny současných procesorů najdeme hardwarovou podporu tohoto šifrování, navíc je možné tento algoritmus paralelizovat.

Poznámka:
Šifrování AES podporuje i většina procesorů od Intelu, což si můžeme ověřit velmi snadno: na stránce http://ark.intel.com/ si najdeme příslušný procesor (jsou řazeny podle typu a generace), a když se „doklepeme“ až k podrobné specifikaci daného procesoru, najdeme na stránce řádek (parametr), jehož název obsahuje zkratku AES (bývá až někde ke konci stránky, „AES New Instructions“).

4.4.5 Další symetrické blokové šifry

Blowfish. Jedná se o symetrickou blokovou šifru navrženou roku 1993 jako nástupce šifry DES, v současné době konkuruje algoritmu AES. Jeho autorem je Bruce Schneier. Typickou vlastností je především to, že autor odmítl tento algoritmus patentovat, chtěl, aby zůstal volným dílem (public domain). Podporu Blowfish najdeme také v jádře Linuxu.

Algoritmus používá 64bitové bloky a proměnlivou délku klíče (od 32 bitů po 448 bitů). Funguje také na principu Feistelovy šifry, stejně jako DES. Aby šifrování a dešifrování bylo dostatečně rychlé (což tedy opravdu je), provádí se při každé výměně klíče jakési předzpracování. Z toho důvodu je při vytvoření nového klíče algoritmus pomalý, ale při každém dalším použití téhož klíče naopak velmi rychlý.

Nevýhodou algoritmu je, že podobně jako u DES, i zde existuje skupina klíčů považovaných za slabé, jinak je brán jako jeden z velmi bezpečných algoritmů. Setkáváme se s ním jako s jedním z více používaných algoritmů v některých bezpečnostních aplikacích (například GnuPG, Advanced Encryption Package, BestCrypt, DriveCrypt, CryptoDisk, atd.) a v programech pro správu hesel pro různé desktopové i mobilní platformy. Podrobnosti najdeme na stránkách autora [14].

Twofish. Tento algoritmus je potomkem algoritmu Blowfish. Má stejného autora (a další spoluautory) a byl jedním z konkurentů algoritmu AES ve finále soutěže vypsané NBS.

Oproti svému předchůdci je délka bloku 128 bitů a používají se delší klíče (až 256 bitů), komplexní kryptoanalýza ještě nebyla úspěšně provedena. Také není zatížen patenty a licencováním. Využití je vpodstatě podobné jako u Blowfish (třebaže se šíří pomaleji, i proto, že Blowfish je

vpodstatě ještě dostačující), a informace najdeme také na stránkách autora [14].

Existuje už i další generace – algoritmus Threefish, který používá tři různé délky bloků, klíč je stejně dlouhý jako blok a kromě běžného klíče využívá i tweakovací klíč (viz výše, str. 103).

Serpent. Je to další konkurent algoritmu AES ve finále soutěže NBS, skončil na druhém místě. Jedná se o blokovou šifru s blokem o velikosti 128 bitů, mohou být použity tři různé velikosti klíče. Algoritmus není patentován ani zatížen licenčními poplatky, je k volnému použití (public domain). Je považován za o něco méně bezpečný než AES, ovšem záleží na konkrétním nastavení (mód, klíč apod.).

IDEA. Původní název je IPES, vznikl roku 1991 ve Švýcarsku, současný název je zkratkou z International Data Encryption Algorithm. Používá délku bloku 64 bitů, klíč je dlouhý 128 bitů. Tento algoritmus je možné používat v prakticky jakémkoliv z běžných módů (ECB, CBC, OFB, ...), včetně trojkombinace EDE (Triple-IDEA) se dvěma klíči.

Algoritmus je poměrně jednoduše implementovatelný a používá se například jako jedna z možností v SSL a PGP.

4.4.6 Hash funkce pro zajištění integrity dat

Zde uvedené algoritmy nemají obvykle za cíl data utajit, ale cílem je vytvořit digitální otisk těchto dat použitelný pro kontrolu integrity (zda nebyla data modifikována). S digitálními otisky dat se můžeme setkat například tehdy, když si stahujeme ISO soubor s některou linuxovou distribucí (nebo čímkoliv jiným), přičemž je nám nabídnuta možnost stáhnout si také soubor s digitálním otiskem tohoto souboru. Pokud ze stáhnutého ISO souboru vypočteme digitální otisk a srovnáme s tím stáhnutým, zjistíme, zda data nebyla během stahování poškozena.

Další typické využití je při autentizaci – v systému nejsou uložena původní hesla, ale jejich hashe.

MD4. Tento algoritmus (MD je zkratka z Message Digest) je standardizován jako RFC 1320, ovšem tento standard je již označen jako obsolete, tedy zastaralý. MD4 vznikla roku 1990, autorem je Ronald Rivest. Ze svého vstupu vytváří digitální otisk o velikosti 128 bitů.

Přestože je algoritmus MD4 prolomitelný (a ani to nedá moc práce), je dodnes používán pro vytváření hashů hesel ve Windows, co se týče lokálního přihlašování (ve Windows se označuje jako NT hash).

MD5. Tento algoritmus je standardizován jako RFC 1321, je z roku 1991, autorem je opět Ronald Rivest. Oproti MD4 je sice o něco pomalejší, ale zato bezpečnější, algoritmus byl oproti předchůdci pozměněn. Zpracovává data v 64 rundách. Vytváří také hash o délce 128 bitů.

Třebaže je MD5 bezpečnější než MD4, je už dnes také považován za prolomitelný především v tom smyslu, že v některých případech nedokáže zajistit jednoznačnost – pokud pro určitý otevřený text vytvoříme hash, případný útočník může být schopen vytvořit pozměněný otevřený text, který má tentýž hash (tedy mohou existovat dva texty se stejným hashem, tzv. MD5 kolize).

V současné době je MD5 používán především pro ověřování integrity souborů stahovaných z Internetu (jak bylo výše naznačeno). Používá se také pro generování hashů hesel, přičemž se doporučuje zároveň použít i mechanismus Salting (solení, viz str. 103) s módem HMAC nebo jiné

možnosti zvýšení bezpečnosti.

SHS – algoritmy SHA. Do skupiny algoritmů SHS (Secure Hash Standard) řadíme algoritmy SHA (Secure Hash Algorithm) verze 0 až 3 (momentálně). Jsou standardizovány jako FIPS PUB 180 (původní SHA-0), FIPS PUB 180-1 (verze SHA-1), FIPS PUB 180-2 (souhrnně SHA-1, SHA-256, SHA-384 a SHA-512). Ve verzi draft (pracovní) je NIST FIPS Pub 202 pro SHA-3.

SHA-0 z roku 1993 se původně označoval jako SHA (bez verze) a produkoval hash o délce 160 bitů, používal velikost bloku 512 bitů a data zpracovával v 80 rundách.

Poměrně brzy byl nahrazen verzí SHA-1, která má stejné „vnější" charakteristiky (délka hashe, velikost bloku, počet rund), ale samotný algoritmus byl pozměněn. SHA-1 byl míněn jako nástupce MD5. Od roku 2010 NIST důrazně doporučuje nahradit verzi SHA-1 svými následovníky, zejména v amerických vládních agenturách, třebaže nalezené bezpečnostní problémy pravděpodobně ještě nebyly cíleně zneužity.

Verze SHA-2 má více variant (první varianty jsou z roku 2001), tyto varianty se liší délkou generovaného hashe (a tím také bezpečností – čím delší hash, tím náročnější je prolomení). Různé varianty se odlišují zápisem délky hashe, takže SHA-224 generuje hash o délce 224 bitů, kdežto SHA-512 generuje 512bitový hash. Varianty se také odlišují velikostí bloku a počtem rund (například nejjednodušší SHA-224 používá velikost bloku 512 bitů a počet rund 64 jako MD5, SHA-384 již má velikost bloku 1024 a počet rund 80). Algoritmus SHA-2 se ve skutečnosti prakticky neliší od algoritmu SHA-1, vyšší bezpečnost plyne spíše z jiných parametrů.

SHA-2 se dnes používá například v implementaci datových schránek ČR, při zajištění integrity elektronického podpisu, na webu při použití zabezpečení TLS.

S algoritmem SHA-3 (také Keccak, vysl. [ketchak]) se zatím nesetkáváme. Také existuje více variant lišících se délkou generovaného hashe, velikostí bloku a dalšími parametry. Označují se verzí a délkou hashe, například SHA3-256. Je zajímavé, že počet rund je u SHA-3 vždy 24, což je výrazně méně než u předchozích verzí (64 nebo 80). Naproti tomu SHA-3 pracuje s většími bloky. Účelem bylo zvýšit bezpečnost a zároveň nezhoršit rychlost algoritmu.

RIPEMD. Tento algoritmus byl v první verzi založen na MD4 (vznikl jeho úpravou), ale po oznámení prolomení MD4 byl přepracován a vznikla verze RIPEMD-160 a zjednodušená RIPEMD-128. Existují také další varianty s jinými délkami hashe. Zkratka názvu je z RACE Integrity Primitives Evaluation Message Digest, což je označení projektu Evropské unie, se kterým je spojován.

Ve srovnání s SHA-1 je RIPEMD-160 o něco pomalejší. Setkáme se s ním jako s jednou z možností generování hashe v PGP. Algoritmus byl vyvinut v univerzitním prostředí a není zatížen patenty, je chápán jako otevřená protiváha algoritmů SHA (u kterých existuje spojitost s americkou NSA).

Tiger. Algoritmus Tiger (resp. Tiger2) se používá například v P2P sítích (třeba Gnutella). Generuje hash o délce 192, 160 nebo 128 bitů, počet rund je 24 (jako u SHA-3).

Whirlpool. Je standardizován jako součást ISO/IEC 10118-3. Není zatížen patenty, je poskytován jako public domain (k volnému použití). Vytváří hash o délce 512 bitů, počet rund je 10 a je založen na podobném principu jako AES. Používá se například v aplikaci TrueCrypt.

4.5 Asymetrická kryptografie

Asymetrická kryptografie se dnes široce využívá nejen pro samotné šifrování, ale například také u certifikátů a digitálních podpisů. Je mladší než symetrická kryptografie, její počátky řadíme do 70. let 20. století.

4.5.1 Pár klíčů pro asymetrickou kryptografii

Zatímco u symetrických šifer se stejný klíč (nebo několik klíčů) používá pro šifrování i dešifrování, u asymetrických šifer potřebujeme pro každou z těchto operací jiný klíč.

Místo jediného klíče je třeba generovat pár klíčů:

- *soukromý klíč* musí být utajen, důkladně uschován, chráněn,
- *veřejný klíč* může být zveřejněn, předán cizí osobě.

Poznámka:
Po vygenerování je sice jeden z klíčů označen jako soukromý a druhý jako veřejný, ale ve skutečnosti je naprosto jedno, který z nich kterou roli plní, jsou v těchto rolích navzájem zaměnitelné. Pokud jakýkoliv z těchto klíčů použijeme pro šifrování, druhý z páru musíme použít pro dešifrování.

Další vlastností je, že tyto klíče sice k sobě patří a navzájem se bez sebe neobejdou, ale zároveň jeden z druhého nelze vygenerovat (nemělo by jít).

Doporučovaná minimální délka klíče je pro různé algoritmy individuální, obvykle to je 2048 bitů nebo více, ale velmi záleží na konkrétním algoritmu a jeho základu (například u algoritmů založených na eliptických křivkách můžeme při zajištění stejné bezpečnosti používat řádově kratší klíče).

4.5.2 Algoritmus

Algoritmy asymetrické kryptografie jsou založeny na jednocestných funkcích s padacími dvířky – trapdoor (tj. zpětné provedení takové funkce je prakticky neproveditelné, pokud nemáme k dispozici „hint“ ve formě veřejného klíče). Jsou sice samy o sobě v principu jednoduché, ale zároveň jejich kryptoanalýza je výpočetně velmi náročná. Bývají založeny na řešení konkrétního těžkého matematického problému (přesněji – pro kryptoanalýzu by mělo být nutné řešit těžký matematický problém), většinou:

- výpočet diskrétních logaritmů,
- faktorizace velkých čísel (rozložení čísla na součin dvou čísel nebo přímo řady prvočísel),
- diskrétní logaritmy na eliptických křivkách.

U všech bychom pro podrobnější výklad potřebovali určité znalosti z algebry (teorie cyklických grup), ale až tak do hloubky v našem výkladu jít nemusíme. Pěkný výklad zejména u faktorizace a diskrétních logaritmů nabízejí videa na [7].

Využití diskrétních logaritmů. Protože mnoho metod asymetrické kryptografie je nějakým způsobem spojeno s diskrétními logaritmy, uvedeme zde tuto definici:

Definice (Diskrétní logaritmus)

Nechť $g,\ m,\ Y \in \mathbb{N}$. Pak každé číslo $k \in \mathbb{N}$ takové, že platí

$$Y = g^k \bmod m \tag{4.1}$$

nazýváme diskrétním logaritmem o základu g z čísla Y vzhledem k modulu m.

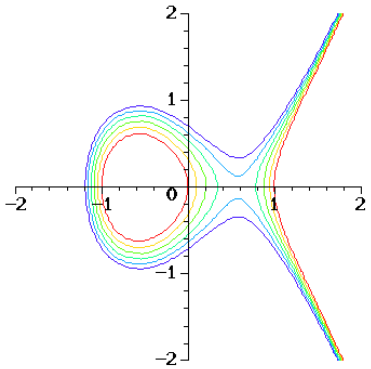
Vypočíst Y podle výše uvedeného vzorce je jednoduché. Ovšem pokud chceme zjistit k, přičemž známe všechny ostatní parametry, čeká nás (především v případě, že m je hodně velké číslo) při současných technických možnostech těžko proveditelný postup (prakticky s exponenciální časovou složitostí). Jinými slovy – na tomto základu se dá postavit těžko prolomitelný algoritmus.

Faktorizace. Faktorizace je proces rozkladu čísla na součin čisel, ale často bývá přímo chápána jako rozklad čísla na součin prvočísel. Například uvedená čísla můžeme faktorizovat takto:

- $24 = 2 \cdot 2 \cdot 2 \cdot 3$
- $50 = 2 \cdot 5 \cdot 5$
- $3162 = 2 \cdot 3 \cdot 17 \cdot 31$
- $551\,565\,283\,195 = 5 \cdot 337 \cdot 577 \cdot 691 \cdot 821$

Násobení samo o sobě je velmi jednoduchá operace, ale pokud to (v případě kryptoanalýzy) máme provést opačně (tj. pro zadané číslo provést rozklad), už rozhodně o jednoduchou operaci nejde, zvlášť když je dané číslo velmi velké. Při generování klíče volíme prvočísla (velmi velká, i několik set cifer), násobíme je a (s příp. další úpravou) získáme klíč.

Eliptické křivky. ECC (Elliptic Curve Cryptography) využívá jiný typ těžkého problému – vztahy mezi body na eliptických křivkách.



Eliptická křivka je křivka popsaná rovnicí

$$y^2 = x^3 + ax + b \tag{4.2}$$

Na obrázku vpravo[2] máme celou sadu křivek (různé barvy), u všech je $a = -1$, liší se hodnotou parametru b. Nad množinou bodů ležících na eliptické křivce je definována operace, kterou můžeme nazvat operací sčítání bodů (ať už to zní jakkoliv zvláštně). Grafická interpretace problému je zajímavá, ale z matematického hlediska je důležitější převod do jazyka algebry (prostě taky potřebujeme grupy a tělesa, abychom mohli operaci přepsat do rovnic a použít diskrétní matematiku). Pracuje se také s celými čísly s určitým ohraničením, což zajistíme použitím operace mod (modulo) a diskrétních logaritmů.

Další informace:

Podrobně a velmi zajímavě je využití eliptických křivek v kryptografii popsáno ve zdroji http://crypto-world.info/klima/2002/chip-2002-09-134-136.pdf.

[2] Obrázek vpravo je ze zdroje https://cosec.bit.uni-bonn.de/?id=644.

4.5.3 Hybridní šifrování

Jaký je rozdíl mezi symetrickými a asymetrickými metodami kryptografie? Tak především je rozdíl v počtu klíčů. Dále je rozdíl v podmínkách pro distribuci klíčů – v případě symetrického klíče musíme zajistit důvěryhodnou a bezpečnou cestu pro jeho transport druhé straně, kdežto u asymetrických metod nejsou na distribuci veřejných klíčů až takové bezpečnostní nároky. Jenže asymetrická kryptografie má i jednu nevýhodu – šifrování a dešifrování je výrazně (o několik řádů) pomalejší než u symetrických metod, což může celkem brzdit komunikaci.

To se obvykle řeší tak, že asymetrickou metodu použijeme na začátku komunikace, zajistíme bezpečnou výměnu symetrického klíče a dále se používá již jen symetrická metoda.

Hybridní šifrování je kombinování metod asymetrické a symetrické kryptografie – pro výměnu (nebo generování) symetrického klíče se použije asymetrická metoda a zbytek komunikace je již šifrován a dešifrován některou symetrickou metodou s využitím takto bezpečně transportovaného symetrického klíče. Typickým příkladem asymetrické metody sestavené pro účely hybridního šifrování je metoda Diffie-Hellman.

4.6 Příklady asymetrických kryptovacích algoritmů

4.6.1 Diffie-Hellman

Tento algoritmus je určen k bezpečnému navázání spojení dvou komunikujících stran. Při navazování spojení je používána asymetrická metoda (každá strana používá dva klíče, typickým výsledkem je vygenerování společného symetrického klíče (session key), který se nepřenáší a následně se používá v samotné komunikaci (tj. aplikace hybridního šifrování).

Postup

Celý postup generování klíčů je následující:

- jedna ze stran vygeneruje náhodné prvočíslo p (hodně velké, stovky cifer) a dále generátor multiplikativní grupy $(Z_p, \cdot)$, který označíme g, tato čísla jsou zveřejněna (předána druhé straně),
- první strana si zvolí náhodné číslo a, druhá strana si zvolí náhodné číslo b (tato čísla tvoří soukromé klíče),
- obě strany vypočtou své veřejné klíče:
  - první strana: $A = g^a \bmod p$
  - druhá strana: $B = g^b \bmod p$

  strany si vymění veřejné klíče A a B,
- obě strany vypočtou sdílené tajemství – klíč pro symetrické šifrování, který bude používán v další komunikaci:
  - první strana: $s = B^a \bmod p$
  - druhá strana: $s = A^b \bmod p$

  tím obě strany došly ke klíči, který je oběma známý, ale už nikomu dalšímu.

Postup je založen na tom, že platí $\left(g^a\right)^b \bmod p = \left(g^b\right)^a \bmod p$.

Tento postup se dá ve skutečnosti rozšířit i na více než dva účastníky.

Pokud jsou náhodná čísla a a b pro každou relaci generována znovu (tj. pokaždé jiná), označujeme tento postup jako *Ephemeral Diffie-Hellman* (zkratka DHE nebo EDH), zatímco jednodušší (méně bezpečná) varianta, kdy některá strana používá pořád totéž náhodné číslo, označujeme zkratkou DH.

Metoda byla patentována, ale patent již vypršel, tedy algoritmus je k volnému použití. Algoritmus vznikl již roku 1976 a je aplikací problému diskrétního logaritmu (o využití diskrétního algoritmu k šifrování včetně významu generátoru grupy je pěkné video na [7]).

Poznámka:
Samotný algoritmus Diffie-Hellman je napadnutelný útokem Man-in-the-Middle – pokud se hned na začátku komunikace za jednu ze stran prohlásí útočník, v klidu navážeme spojení s tím, s kým ve skutečnosti komunikovat nechceme (a útočník může totéž provést na obě strany). Proto při potřebě ověření protistrany je nutné kombinovat Diffie-Hellmana ještě s některým autentizačním mechanismem, například hesly či certifikáty.

4.6.2 RSA

Algoritmus RSA (Rivest, Shamir, Aleman – podle tvůrců algoritmu zaměstnaných v MIT) je založen na faktorizaci velkých prvočísel, vznikl roku 1977. Roku 1983 byl patentován, ale roku 2000 patent vypršel, tedy algoritmus je k volnému použití.

Postup (generování klíčů)
Postup generování klíčů k_1 a k_2 pro RSA je následující:

1. necháme vygenerovat dvě velká náhodná prvočísla p a q,
2. vypočteme $n = p \cdot q$,
3. vypočteme Eulerovu funkci $\varphi(n) = ((p-1) \cdot (q-1))$,
4. vygenerujeme náhodné číslo e z intervalu $(\max(p+1, q+1), \varphi(n))$, číslo e musí být s $\varphi(n)$ nesoudělné,
5. vypočteme číslo $d = e^{-1} \bmod \varphi(n)$, pokud vyjde d příliš malé (menší než cca $\log_2(n)$), vrátíme se do bodu 4 (vygenerujeme jiné číslo e),
6. náš pár klíčů je dvojice uspořádaných dvojic $k_1 = \{n, e\}$, $k_2 = \{n, d\}$.

Čísla p a q by měla být hodně velká a zároveň prvočísla, což není jednoduché zajistit najednou. Postup je takový, že vygenerujeme náhodné číslo, pak ověříme, jestli je prvočíslem (typicky metodou Eratosthenova síta, kdy začínáme ověřovat dělitelnost od nejmenších čísel), když ne, generujeme znovu, atd. – tak dlouho, až máme náhodné číslo.

Postup (šifrování a dešifrování)
Máme tedy dvojici klíčů. Klíč k_1 budeme brát jako veřejný, zajistíme jeho distribuci, kdežto klíč k_2 bude pro nás soukromý a naším úkolem je co nejlépe ho zabezpečit proti odcizení.

Označme M otevřený text a C jeho zašifrovanou variantu. Pokud chceme šifrovat klíčem k_1 a dešifrovat klíčem k_2, postup je tento:

- šifrování: $C = M^e \bmod n$
- dešifrování: $M = C^d \bmod n$

Jestliže naopak chceme šifrovat klíčem k_2 a dešifrovat klíčem k_1, jen zaměníme čísla d a e:

- šifrování: $C = M^d \bmod n$
- dešifrování: $M = C^e \bmod n$

Algoritmus RSA sice už slouží celkem dlouho, ale přesto je stále považován za bezpečný. Ovšem bezpečnost může být při neopatrnosti jednoduše prolomena.

Čísla p a q musí být hned po vygenerování dvojice klíčů *zničena*. Protože číslo n je součástí veřejného klíče (a tedy veřejně známé), bylo by při znalosti třeba jen jednoho z čísel p nebo q jednoduché odvodit soukromý klíč.

Poznámka:

V postupu generování klíčů se objevuje pojem *Eulerova funkce*. O co jde?

Jedná se o funkci $\varphi(n)$, jejímž argumentem může být celé nezáporné číslo (tj. je to zobrazení $\varphi : \mathbb{N}_0 \to \mathbb{N}_0$), a vrací počet celých nezáporných čísel k takových, která jsou menší nebo rovna n a zároveň jsou s ním nesoudělná, tedy $\mathrm{NSD}(k, n) = 1$. Je zřejmé, že platí:

- $\varphi(1) = 1$,
- $\varphi(p) = p - 1$, pokud p je prvočíslo (s prvočíslem jsou nesoudělná všechna čísla, která jsou menší než toto číslo),
- pokud x a y jsou nesoudělná, pak $\varphi(x \cdot y) = \varphi(x) \cdot \varphi(y)$,
- pokud x a y jsou prvočísla (ta jsou vždy nesoudělná), pak $\varphi(x \cdot y) = (x - 1) \cdot (y - 1)$.

Z toho plyne vztah, který jsme pro výpočet $\varphi(n)$ použili v postupu generování klíčů.

Hodně se diskutuje o bezpečnosti *generátoru náhodných čísel* používaného v RSA. Běžná zařízení mají k dispozici jen generátor pseudonáhodných čísel, která jsou teoreticky za určitých podmínek zpětně odvoditelná (ale pro běžné použití to naprosto postačuje), ovšem horší je podezření, že NSA se podařilo do knihovny tohoto generátoru zabudovat zadní vrátka (backdoor). Pokud je to pravda, pak tím strašákem není *jen* NSA, ale prakticky každý hacker tak schopný, že dokáže tato zadní vrátka objevit a zneužít.

Další informace:

Informace o RSA najdeme na stránkách RSA Labs, ale navíc je specifikace RSA zveřejněna v dokumentu RFC 3447:

- http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm
- RFC 3447. *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*. RSA Laboratories, 2003. Dostupné také z: https://tools.ietf.org/html/rfc3447

4.6.3 Elgamal

Algoritmus Elgamal byl představen roku 1984 a jeho autorem je Taher Elgamal (pochází z Egypta, t.č. pracuje na Stanford University v Kalifornii).

Postup je (podobně jako Diffie-Hellman) založen na problému výpočtu diskrétních logaritmů. Jeho zásadní nevýhodou z pohledu praxe je, že šifrovaná data jsou dvakrát delší než nešifrovaná, což je také důvodem menšího rozšíření než v případě konkurenčního RSA.

Tento algoritmus se používá jak pro samotné šifrování, tak i pro zajištění elektronického podpisu. Setkáme se s ním například v PGP nebo v GPG.

Další informace:

Podrobnější informace o algoritmu Elgamal najdeme v těchto zdrojích:

- https://kmlinux.fjfi.cvut.cz/ balkolub/Vyuka/Vlcek.pdf
- http://caislab.kaist.ac.kr/lecture/2010/spring/cs548/basic/B02.pdf

4.6.4 DSA

DSA (Digital Signature Algorithm) z roku 1991 je součástí sady DSS (Digital Signature Standard) popsané v FIPS 186-3 (aktualizace z roku 2009). Je sice patentován, ale zároveň uvolněn k volnému použití.

Tento algoritmus asymetrické kryptografie není přímo určen pro šifrování, je používán pro podepisování – v technologii digitálního podpisu. Je založen na řešení problému diskrétního logaritmu, podobně jako Elgamal.

V nastavení parametrů je třeba dbát nejen na vhodnou délku klíčů (klíče by měly být rozhodně nejméně 2048 bitů), ale také na volbu používané hash funkce (raději SHA-2 než SHA-1).

S DSA se setkáváme v různých běžných nástrojích (přestože jsou i zde obavy z kompromitace agenturou NSA), například GnuPG, OpenSSH.

4.6.5 ECC – eliptické křivky

Princip využití eliptických křivek (ECC, Elliptic Curve Cryptosystem) v kryptografii je velmi stručně popsán na straně 114.

Používáme eliptické křivky nad konečnými tělesy (tj. množina bodů je diskrétní – používáme celá čísla, a konečná – ohraničujeme s využitím operace modulo). Konečná tělesa se také nazývají Galoisova pole, proto se často používá značení GF_p (Galois Field modulo p). Nad tělesem je definovaná operace, kterou (poněkud nepřesně) označujeme jako součet bodů křivky.

Ve srovnání s algoritmem Diffie-Hellman o několik stránek výše je rozdíl především v tom, že místo multiplikativní grupy nad celými čísly používáme Galoisovo těleso s eliptickou křivkou a operaci součtu bodů. Je třeba distribuovat veřejný klíč a parametry křivky, dále obě strany z těchto informací postupně vypočtou společný klíč, který se již používá pro samotnou komunikaci.

V praxi se setkáváme s variantami výše popisovaných algoritmů, kde jsou místo diskrétních logaritmů používány eliptické křivky – například ECDHE je varianta algoritmu Ephemeral Diffie-Hellman používající ECC, také existuje ECDSA (varianta DSA s eliptickými křivkami).

Poznámka:

Je zajímavé, že kryptosystémy budované s využitím ECC si při zajištění stejné úrovně bezpečnosti vystačí s mnohem kratšími klíči – například ECC s klíčem o délce 160–256 bitů je obdobně bezpečný jako RSA s klíčem o délce 1024–3072 bitů.

Další informace:

Kromě zdroje uvedeného u základního popisu eliptických křivek mohou být zajímavé ještě další zdroje – například

- http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/
- http://www.cs.vsb.cz/arg/workshop/files/ecc_eli.pdf
- https://edux.fit.cvut.cz/oppa/MI-KRY/prednasky/prednaska11_12.pdf

Literatura

[1] *Annoucing the Advanced Encryption Standard (AES): Federal Information Processing Standards Publication 197* [online]. 2001 [cit. 2015-07-02]. Dostupné z: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[2] Doseděl, Tomáš. *Počítačová bezpečnost a ochrana dat.* Vyd. 1. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.

[3] Dworkin, Morris. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques* [online]. NIST Special Publication 800-38A. Gaithersburg: National Institute of Standards and Technology, 2001 [cit. 2015-06-30]. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[4] Dworkin, Morris. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC* [online]. NIST Special Publication 800-38D. Gaithersburg: National Institute of Standards and Technology, 2007 [cit. 2015-06-30]. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[5] Dworkin, Morris. *Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices* [online]. NIST Special Publication 800-38E. Gaithersburg: National Institute of Standards and Technology, 2010 [cit. 2015-06-30]. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

[6] *Enigma World Code Group* [online]. [cit. 2015-07-02]. Dostupné na: http://www.enigmaworldcodegroup.com/

[7] Hollas, Daniel. Moderní šifrování. *Khanova škola* [online]. [cit. 2015-07-27]. Dostupné z: https://khanovaskola.cz/blok/4/12-moderni-sifrovani

[8] Klíma, Vlastimil. Moderní kryptografie I [online]. *Crypto-World.info*, 2007 [cit. 2015-06-20]. Dostupné z: http://crypto-world.info/klima/mffuk/Symetricka_kryptografie_I_2007.doc

[9] Klíma, Vlastimil. Moderní kryptografie II: Symetrické šifrovací systémy [online]. *Crypto-World.info*, 2007 [cit. 2015-06-22]. Dostupné z: http://crypto-world.info/klima/mffuk/Symetricka_kryptografie_II_2007.doc

[10] Klíma, Vlastimil. Moderní kryptografie III: Mody činnosti blokových šifer a hašovací funkce [online]. *Crypto-World.info*, 2007 [cit. 2015-06-25]. Dostupné z: http://crypto-world.info/klima/mffuk/Symetricka_kryptografie_III_2007.doc

[11] Menezes, Alfred, Paul C. van Oorschot a Scott A. Vanstone. *Handbook of applied cryptography*. Boca Raton: CRC Press, c1997, xiii, 780 s. Discrete mathematics and its applications. ISBN 08-493-8523-7. Některé kapitoly jsou dostupné z: http://cacr.uwaterloo.ca/hac/

[12] Rogaway, Phillip. Evaluation of Some Blockcipher Modes of Operation. *University of California* [online]. Davis, 2011 [cit. 2015-06-30]. Dostupné z: http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf

[13] Rybka, Štěpán. *Skupina algoritmů NSA Suite-B Cryptography*. Brno, 2011. Dostupné také z: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=42057. Bakalářská práce. VUT v Brně. Vedoucí práce Ing. Jiří Sobotka.

[14] *Schneier on Security: Cryptography* [online]. [cit. 2015-07-23]. Dostupné z: https://www.schneier.com/cryptography.html

Kapitola 5

Kryptografie v praxi

V předchozí kapitole jsme se zabývali spíše kryptovacími algoritmy, v této se zaměříme na využití těchto algoritmů v různých nástrojích.

5.1 Digitální podpisy a certifikáty

5.1.1 Digitálně podepsaná zpráva

V předchozí kapitole se píše o hash funkcích pro zajištění integrity dat (viz str. 111). Digitální (elektronický) podpis zajišťuje nejen integritu, ale i nepopiratelnost (ověřujeme původce dat). Používá se asymetrická kryptografie, přičemž posílaná zpráva je na straně odesílatele zpracována soukromým klíčem tohoto odesílatele a na straně příjemce pak veřejným klíčem odesílatele. Předpokládá se, že příjemce má k dispozici veřejný klíč odesílatele, aby mohl jeho autentičnost ověřit.

U tohoto mechanismu není nutné šifrovat celou zprávu – nemusíme zajišťovat důvěrnost, ale pouze integritu a nepopiratelnost. Proto ve výsledku není celý šifrovaný text, ale pouze jeho hash (digitální otisk). Tento hash jednoznačně určuje zprávu (pokud bychom ve zprávě provedli třeba jen malou změnu, hash by se změnil velmi výrazně).

V souhrnu tedy potřebujeme minimálně následující:

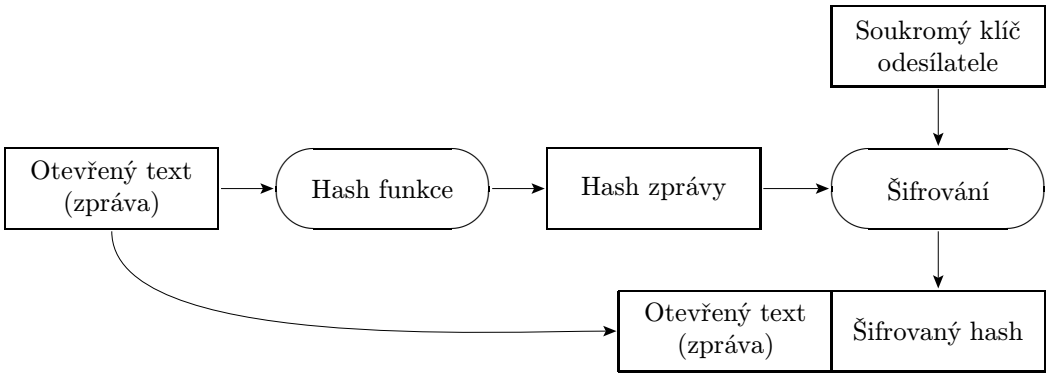
- hash algoritmus (rychlý symetrický algoritmus generující digitální otisk celého otevřeného textu), například MD5, SHA-1, SHA-2, RIPEMD-160,
- asymetrický algoritmus s párem klíčů pro zpracování výsledku hashování.

Postup (Používání digitálního podpisu)

Odesílatel:

- má otevřený text a svůj soukromý klíč,
- vygeneruje hash z otevřeného textu (některým symetrickým hash algoritmem),
- pomocí soukromého klíče vygeneruje z tohoto hashe jeho digitální podpis (asymetricky zašifruje, třeba pomocí DSA),
- odešle příjemci otevřený text a digitální podpis jeho hashe.

Postup je naznačen na obrázku 5.1.

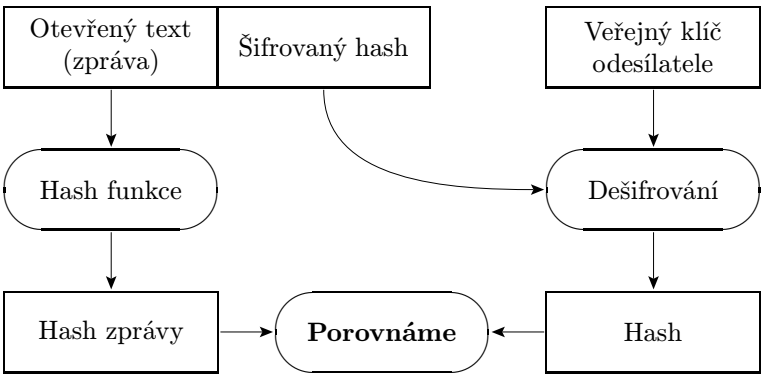


Obrázek 5.1: Digitální podpis na straně odesílatele

Příjemce:

- má veřejný klíč odesílatele, přijme otevřený text a digitální podpis jeho hashe,
- použije na otevřený text stejný hash algoritmus jako odesílatel ⇒ hash1,
- použije veřejný klíč odesílatele na digitální podpis ⇒ hash2,
- pokud hash1 a hash2 souhlasí, je vše v pořádku.

Postup je naznačen na obrázku 5.2.



Obrázek 5.2: Digitální podpis na straně příjemce

Výše uvedený postup zaručuje integritu a nepopiratelnost, ale nikoliv důvěrnost. Pokud chceme zajistit i důvěrnost zprávy (tj. aby si ji mohl přečíst pouze adresát a nikdo jiný), musíme zprávu doopravdy zašifrovat.

Postup (Používání digitálního podpisu a šifrování)

Pro šifrování se dají použít různé postupy (viz předchozí kapitolu), ať už půjde o symetrické nebo asymetrické algoritmy.

Jestliže použijeme symetrický algoritmus (třeba AES), musíme zajistit bezpečný transport klíče, výhodou je využitelnost i pro velké objemy dat. Pokud se rozhodneme pro asymetrický algoritmus, použijeme pro šifrování dat veřejný klíč příjemce – v takovém případě musí mít každá strana k dispozici veřejný klíč druhé strany. Případně se dá použít hybridní postup (symetricky šifrujeme data, asymetricky klíč pro symetrické šifrování).

Poznámka:

Teď už je jasné, kde je slabé místo tohoto postupu. Příjemce potřebuje veřejný klíč odesílatele (u šifrování také odesílatel potřebuje veřejný klíč příjemce), ale pak je tedy nutné nějakým dostatečně důvěryhodným způsobem tento veřejný klíč distribuovat. Pokud nezajistíme bezpečný transport klíče, nelze zaručit, že zpráva nebyla cestou pozměněna či dokonce podvržena.

Buď se tedy obě strany osobně setkají a vymění si své veřejné klíče, nebo využijí některou důvěryhodnou databázi veřejných klíčů, případně se spolehnou na garanci důvěryhodné entity. Rozhodně není řešením transportovat veřejný klíč zároveň se zprávou, na jejímž zabezpečení se podílí – případnému útočnikovi stačí zprávu pozměnit, podepsat vlastním veřejným klíčem a ten ke zprávě přiložit.

5.1.2 Certifikáty

Používání certifikátů alespoň částečně řeší výše naznačený problém. Veřejný klíč může být garantován důvěryhodnou (trusted) *certifikační autoritou* (CA).

Definice (Certifikát)

Certifikát je zabezpečený souhrn následujících informací o dané osobě či jiné entitě (vlastníkovi certifikátu):

- identifikační údaje samotného certifikátu (sériové číslo certifikátu, ID certifikační autority),
- údaje o platnosti (datum vydání certifikátu a do kdy je platný, může zde být i údaj o omezení způsobu využívání certifikátu),
- osobní údaje vlastníka certifikátu (jméno, příjmení, rodné číslo, příp. IČO apod.),
- údaje o algoritmech (použitý hash algoritmus, asymetrický algoritmus pro šifrování hashe, atd.),
- veřejný klíč, který je tímto certifikován (tj. veřejný klíč vlastníka certifikátu),
- může být také veřejný klíč certifikační autority,
- digitální podpis předchozích dat (k podpisu je použit soukromý klíč certifikační autority).

Skutečný obsah certifikátu může být trochu odlišný. Strukturu certifikátu pro konkrétní účel totiž stanovují různé standardy (zejména ITU X.509), ale i právní předpisy konkrétních zemí.

Certifikát v neintegrované formě je obvykle soubor se specifickou příponou. Když tedy chceme do aplikace importovat certifikát, jsme v dialogu požádáni právě o takový soubor.

Můžeme se setkat s certifikátem, jehož platnost již vypršela. To může znamenat, že vlastník ještě neobnovil platnost svého certifikátu, nebo může jít o bezpečnostní problém.

Platnost certifikátu může být dokonce odvolána i před vypršením platnosti – například tehdy, když byl kompromitován buď vlastník certifikátu nebo jeho CA (či autorita v hierarchii výše). Kompromitace může spočívat v odcizení soukromého klíče, který je potřebný pro digitální podpis, a tedy útočník může své vlastní zprávy podepisovat jménem napadeného a vydávat se za něj. Pokud vlastník certifikátu zjistí napadení, musí svůj certifikát odvolat, případně to provede příslušná CA, pokud ke kompromitaci došlo u ní.

Odvolaný certifikát je zařazen na *seznam odvolaných certifikátů* (CRL – Certificate Revocation List), a pokud kdokoliv (útočník nebo omylem vlastník) použije certifikát z takového seznamu, vystavuje se dokonce právnímu postihu (obdobně jako při falšování podpisu na úředním dokumentu). Seznam odvolaných certifikátů je aktualizován obvykle jednou za jeden až dva dny.

Existují dva způsoby ověření platnosti certifikátu:

- lze si od CA pravidelně stahovat buď celý seznam odvolaných certifikátů (což je poměrně velký balík dat) nebo jeho aktualizace; aktualizace jsou vydávány zpravidla jednou za jeden až dva dny,
- dynamická kontrola přímo u CA – vždy, když chceme použít certifikát, zkontrolujeme jeho platnost on-line.

U prvního způsobu se typicky častěji stahují pouze aktualizace, a pak jednou za pár týdnů celý seznam.

V definici je zmíněno, že součástí certifikátu může být údaj o omezení *způsobu využívání certifikátu*. Vlastník může mít pro různé účely různé certifikáty, tj. různé páry klíčů (například jeden pro podepisování e-mailů, dokumentů, další pro autentizaci, šifrování apod. Tato položka je implementována jako pole bitů, kde každé využití má jeden bit nastavený na 1 nebo 0 podle toho, zda je dotyčné využití povoleno. V některých případech to může zabránit zneužití certifikátu, především tehdy, když se u různých způsobů využití kříží role veřejného a soukromého klíče.

Není certifikát jako certifikát, záleží, kým byl vydán. Rozlišujeme čtyři třídy certifikátů, každá z nich zahrnuje certifikáty použitelné k určitému konkrétnímu účelu.

Class 1 je nejnižší třída sloužící typicky pro testování a výuku. Při pořizování certifikátu třídy 1 se prakticky nic neověřuje. Tyto certifikáty si také pořizují někteří lidé pro podepisování e-mailových zpráv, pokud se netrvá na opravdovém zabezpečení. Pokud je takovým certifikátem autentizován server, nepůsobí to důvěryhodně.

Class 2 je třída pro certifikáty, kde se údaje žadatele ověřují „třetí osobou“. Buď je třeba dodat notářsky ověřený vyplněný formulář žádosti o certifikát (ověřuje notář) nebo si CA ověřuje údaje dotazováním u důvěryhodné organizace, která příslušné údaje zná.

Class 3 jsou již certifikáty plně důvěryhodné jak ze strany úřadů, tak i ze strany větších firem. Žadatel o certifikát musí osobně navštívit kontaktní místo vybrané CA a předložit své dokumenty (občanský průkaz nebo pas a další podle typu údajů, které mají být v certifikátu uvedeny), kterými jednoznačně prokáže svou totožnost.

Class 4 jsou certifikáty, pro jejichž získání je žadatel prověřen stejně jako u třídy 3, ale navíc je ověřováno oprávnění žadatele k určité činnosti (pro kterou je certitikát vydáván). Takový certifikát svou vahou odpovídá využití oficiálního razítka organizace nebo jejího konkrétního oddělení.

Pokud potřebujeme certifikát ke komunikaci s úřady, podle zákonů ČR je třeba používat certifikát třídy 3.

S certifikáty se nejčastěji setkáváme ve dvou případech – u podepsaných e-mailů a zabezpečených webových stránek. Oba tyto případy si probereme v dalších částech kapitoly.

5.1.3 Certifikační autorita

Certifikační autorita (CA) je organizace vydávající certifikáty veřejných klíčů. Předpokládá se, že jde o důvěryhodnou organizaci, především musí být důvěryhodná pro adresáta zprávy podepsané takovým certifikátem.

V definici certifikátu na straně 123 stojí, že součástí certifikátu je digitální podpis zbývajícího obsahu (tj. všeho kromě tohoto podpisu), přičemž se pro podpis při udělování certifikátu využil soukromý klíč CA vydávající certifikát. Z toho vyplývá, že pouze ten, kdo má soukromý klíč této CA, může vydávat certifikáty garantované dotyčnou CA, a tedy soukromý klíč je nejchráněnějším vlastnictvím každé CA. Kdyby byl odcizen, pak by zločinec až do zneaktivnění (zařazení do seznamu CRL) mohl vytvářet a používat certifikáty, které by každý považoval za pravé a důvěryhodné.

Proto každá CA důkladně chrání své soukromé klíče – včetně fyzické ochrany, nastavení přístupových práv apod. Aby bylo možné na jednu stranu soukromý klíč ochránit a na druhou stranu ho i reálně používat, jedná se typicky o zařízení, které komunikuje pouze tímto způsobem: přijme balík dat, podepíše ho dotyčným soukromým klíčem a výsledek dá na výstup. Oficiálně nelze tomuto zařízení přikázat, aby na výstup dalo přímo soukromý klíč, vlastně od své aktivace (včetně uložení soukromého klíče) s jinými vstupy a výstupy odmítá pracovat. Při pokusu o získání soukromého klíče se zařízení fyzicky zničí.

Pokud přesto dojde k odcizení soukromého klíče, je třeba urychleně varovat všechny zainteresované. Ve skutečnosti je pak třeba zneaktivnit všechny certifikáty, které tato CA dosud vydala, získat nové soukromé klíče a všem zákazníkům vydat nové verze certifikátů podepsané již novým soukromým klíčem (vše na náklady CA, případně se náklady hradí z pojištění). V každém případě napadená CA ztrácí důvěryhodnost a měla by prokázat, že slabé místo, přes které došlo k napadení, bylo opraveno.

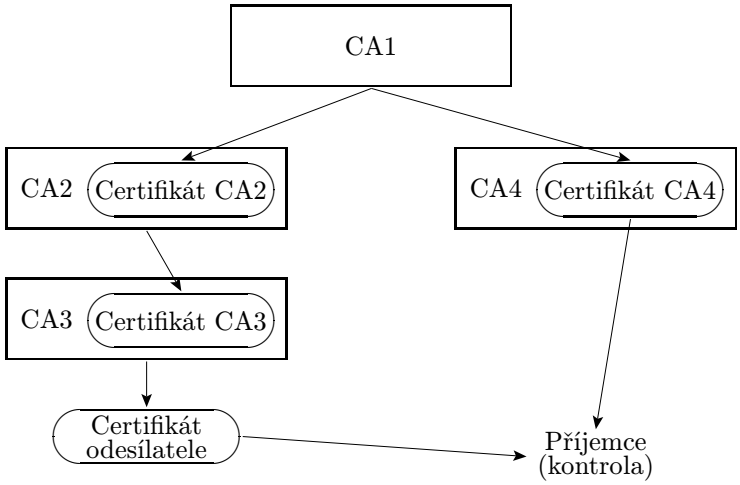
5.1.4 PKI

Každý uživatel počítače má určenu množinu certifikačních autorit, kterým *důvěřuje*. Údaje o jejich certifikátech včetně veřejného klíče má obvykle importovány v příslušné aplikaci – ve webovém prohlížeči, e-mailovém klientovi apod. Nejde jen o to – používat certifikát, ale jde o používání certifikátu, jemuž adresát důvěřuje. Svou vlastní certifikační autoritu si ve skutečnosti může vytvořit každý, a větší firmy to taky dělají (vlastní certifikáty používají pro zajištění důvěryhodnosti v rámci firmy). Jenže takovému certifikátu zřejmě nebude důvěřovat nikdo mimo firmu.

Pokud příjemce důvěřuje právě té CA, od které má odesílatel certifikát, není problém: ověření důvěryhodnosti bude spočívat v použití veřejného klíče této CA na certifikát odesílatele, ověření, zda není odvolaný, a dále se postupuje stejně jako u digitálního podpisu. Ale co když tomu tak není?

Proto jsou CA uspořádány do jakéhosi stromu, kdy důvěryhodnost CA z podřízeného uzlu je garantována tou CA, která je jí ve stromu nadřízena, a tedy „nižší“ CA vlastní certifikát vydaný její nadřízenou autoritou. V případě, že příjemce nedůvěřuje CA odesílatele, je potřeba v tomto stromě najít takový uzel (tedy certifikační autoritu), v jehož podstromě se nachází jak CA odesílatele, tak i kterákoliv CA z těch, jimž důvěřuje příjemce.

Situace je načrtnuta na obrázku 5.3. Odesílatel má certifikát od CA3, které však příjemce



Obrázek 5.3: Strom certifikačních autorit

nedůvěřuje. Důvěřuje však autoritě CA4, a tyto dvě autority jsou v podstromě autority CA1. Proto příjemce bude důvěřovat certifikátu odesílatele.

V takovém stromě musí existovat *certifikační cesta* od certifikátu odesílatele k CA v kořeni podstromu (zde CA1), což znamená, že všechny certifikáty na této cestě (při ověřování ve směru od kořene k listu) musí být platné. Hierarchie certifikátů veřejných klíčů se nazývá *PKI (Public Key Infrastructure)* – infrastruktura veřejných klíčů.

Jak se řeší případ, kdy takový podstrom nenajdeme (tj. CA odesílatele není v žádném vztahu k těm CA, jimž příjemce důvěřuje), a tedy máme dva nezávislé stromy CA? To je řešitelné *křížovou certifikací*, kdy některá CA z jednoho stromu certifikuje kteroukoliv CA z druhého stromu a naopak.

Každý stát Evropské unie má určenu instituci, která provádí akreditace certifikačních autorit v daném státě. V rámci Evropské unie funguje vzájemné uznávání CA akreditovaných těmito institucemi, podobné mezistátní smlouvy jsou uzavřeny i s mnoha mimoevropskými zeměmi.

5.1.5 Jak získat certifikát v ČR

Pokud chceme mít vlastní certifikát, měli bychom si především rozmyslet, jaký certifikát vlastně potřebujeme, k čemu bude využíván:

Vlastní certifikát organizace se používá pouze v rámci dané organizace. Organizace se sama sobě stává certifikační autoritou.

Komerční certifikát je použitelný pro podepisování e-mailů a bezpečné přihlašování, komerční doménový certifikát pak k zabezpečení komunikace na webu (SSL certifikáty).

Kvalifikovaný certifikát umožňuje vytvářet *zaručené elektronické podpisy*, tj. digitální podpisy jednoznačně a právně dokazatelně propojené s konkrétní osobou (vlastníkem), tedy obdobu skutečného „papírového“ podpisu. Žadatel (jako fyzická osoba) obvykle musí předložit dva osobní doklady totožnosti (třeba občanský průkaz a řidičský průkaz nebo rodný list).

Pokud chceme elektronicky podepisovat obchodní nebo občanskoprávní smlouvy, potřebujeme tento typ certifikátu.

Kvalifikovaný certifikát vydaný akreditovaným poskytovatelem certifikačních služeb je vyšší úroveň kvalifikovaného certifikátu. Tyto certifikáty jsou dražší, ale na druhou stranu jsou použitelné i pro zabezpečenou komunikaci s úřady včetně *datových schránek*. Oproti předchozímu typu je při žádosti rozdíl vpodstatě jen v tom, koho o certifikát žádá.

Postup (Žádost o kvalifikovaný nebo komerční certifikát obecně)

Pokud jsme se rozhodli pro určitý typ certifikátu (podle způsobu jeho využívání) a konkrétní certifikační autoritu, u které o certifikát požádáme, budeme dále postupovat takto:

- důkladně prostudujeme informace na webu vybrané CA – nabídku certifikátů, certifikační politiku, veškeré podmínky, požadavky a postupy, ceník, nabízené nástroje a způsob zveřejňování seznamu CRL,
- vygenerujeme pár klíčů (veřejný a soukromý klíč) – buď svými vlastními prostředky nebo využijeme nástroj na webu CA,
- podle pokynů na webu CA vyplníme formuláře a vygenerujeme žádost o certifikát,
- na registračním místě CA vyřídíme zbývající požadované úkony včetně naší identifikace.

Mezi informacemi, které by nás měly zajímat, je také využití získaného certifikátu (výše zmiňované účely použití), a doba platnosti certifikátu. Před uplynutím této doby je třeba vždy požádat o obnovení platnosti certifikátu.

Pokud potřebujeme kvalifikovaný certifikát vydaný akreditovaným poskytovatelem certifikačních služeb, musíme se obrátit na jednu z těchto certifikačních autorit:

- PostSignum provozovaný Českou poštou,
- První certifikační autorita, a.s.,
- eIdentity, a.s.

Pro mnohé žadatele je zřejmě nejvýhodnější první možnost, protože registrační místa jsou regionálně nejdostupnější. Následují kontakty na všechny tyto poskytovatele certifikační služby. Vždy jde o certifikát třídy 3, tedy musíme osobně prokázat svou totožnost.

Další informace:

- http://www.mvcr.cz/clanek/prehled-kvalifikovanych-poskytovatelu-certifikacnich-sluzeb-a-jejich-kvalifikovanych-sluzeb.aspx
- https://www.postsignum.cz/
- https://www.ica.cz/
- https://www.eidentity.cz/app

Je zajímavé, že pokud zadáme adresu kteréhokoliv z těchto tří poskytovatelů služby bez udání protokolu HTTPS, tedy až od zdvojeného lomítka, načte se pouze HTTP verze (bez jakéhokoliv zabezpečení), třebaže se jedná o společnosti zabývající se bezpečnostními otázkami a na webu se pomalu stává standardem zabezpečení už ve výchozím nastavení (vynucené přesměrování přes HTTPS). Pokud adresu zadáme i s protokolem HTTPS, u PostSignum je vše v pořádku, ale u První

certifikační jsou některé části stránky nezabezpečené (Firefox hlásí, že tyto části zablokoval) a v případě eIdentity prohlížeč hlásil neplatný SSL certifikát.

Poznámka:

Pozor, certifikáty našich tří slavných kvalifikovaných autorit naprosto nejsou vhodné pro jiný účel než pro komunikaci se státní správou. Pokud chceme certifikát pro webový server nebo jiné podobné „světové“ účely, zvolíme světově uznávaného poskytovatele komerčních certifikátů.

Pokud potřebujeme komerční certifikát, můžeme se obrátit také na výše uvedené poskytovatele, ale častěji se volí spíše jiné certifikační autority. Ti, kdo se dívali na adresy uvedené a komentované o odstavec výše, si určitě všimli, že tito poskytovatelé používají SSL certifikáty od společnosti avast!. Další poskytovatelé v ČR jsou Symantec, Zoner, atd. Obvykle nabízejí certifikáty garantované světovými certifikačními autoritami jako Thawte, GeoTrust, VeriSign, Terena, DigiNotar a další. Informace najdeme často také u našeho poskytovatele domény, pro kterou provozujeme server.

Další informace:

- http://www.blog.sslmarket.cz/ssl/jak-spravne-vyuzit-ssl-certifikat-dil-prvni/
- https://www.interval.cz/clanky/ssl-certifikaty-pro-seriozni-komercni-web-nutnost/
- https://www.alpiro.cz/jak-vybrat-ssl.html
- https://www.ca.cz/napoveda/

Poznámka:

Jak bylo výše řečeno, největší devízou certifikační autority je důvěryhodnost. Pokud budou kompromitovány soukromé klíče CA, má problém nejen dotyčná CA, ale i její klienti, protože těmito klíči jsou podepsány jejich certifikáty. V minulosti se bohužel již staly případy, kdy se útočníkovi podařilo u důvěryhodné CA získat certifikát na doménu, kterou ve skutečnosti nevlastnil, a tedy se jeho server mohl vydávat za server té domény. I to je samozřejmě bezpečnostní problém.

Další informace:

Další možností je stát se sami sobě certifikovanou autoritou. K tomu potřebujeme OpenSSL, o čemž najdeme informace například

- v seriálu *Jak na OpenSSL* na root.cz: http://www.root.cz/clanky/jak-na-openssl/,
- v článku *Bezpečný web s HTTPS* na linuxexpres.com: http://www.linuxexpres.cz/praxe/bezpecny-web-s-https.

To, že získání certifikátu nemusí být jednoduché, vyplývá také z těchto článků:

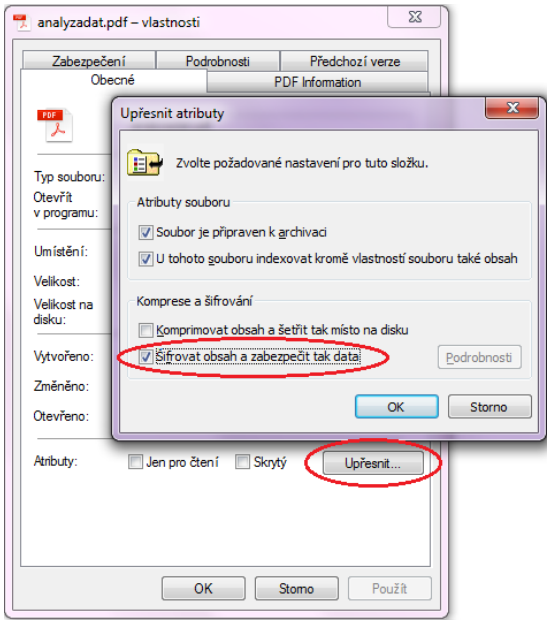
- http://www.lupa.cz/clanky/jak-jsem-si-poridil-elektronicky-podpis-ceske-posty/
- http://www.lupa.cz/clanky/kvalifikovany-certifikat-na-dve-veci/

5.2 Šifrování dat na paměťových médiích

5.2.1 Šifrovací nástroje vestavěné ve Windows

EFS (Encrypting File System). Tento šifrovací souborový systém je koncipován jako nástavba NTFS a je podporován ve vyšších edicích Windows (kromě Starter a Home. Nešifruje se celý disk ani oddíl, ale pouze konkrétní soubory. Šifrování a dešifrování při zápisu a čtení souboru probíhá transparentně na pozadí, je pouze mírně pomalejší.

EFS používá hybridní algoritmus, tedy kombinuje rychlejší symetrickou a bezpečnější asymetrickou šifru. Pro asymetrickou šifru je třeba, aby existovaly minimálně dva certifikáty – jeden patří uživateli, jehož soubor je šifrován, druhý tzv. Recovery agentovi (pokud je poškozen certifikát uživatele, pak s pomocí tohoto druhého certifikátu je možné data dešifrovat). Ke každému certifikátu pak přísluší jedna dvojice soukromého a veřejného klíče. Soukromý klíč je pak uložen v bezpečném úložišti v profilu uživatele.



Šifrování souboru začíná vygenerováním symetrického klíče (FEK – File Encrytion Key), kterým jsou zašifrována data, následně je FEK zašifrován veřejným klíčem uživatele uloženým v jeho profilu. Následně je FEK zašifrován veřejným klíčem Recovery agenta. Výsledný soubor má záhlaví skládající se z těchto dvou položek (FEK šifrovaného se dvěma různými veřejnými klíči) a případně dalších položek, následují samotná šifrovaná data.

Během dešifrování je z bezpečného úložiště vyzvednut soukromý klíč uživatele, s jeho pomocí je dešifrován FEK v záhlaví šifrovaného souboru, a následně jsou pomocí FEK dešifrována data.

Jak je to z pohledu uživatele: pokud máme edici Windows podporující EFS, stačí si zobrazit *Vlastnosti* souboru, který chceme šifrovat (viz obrázek vpravo), na kartě *Obecné* dole u atributů zvolíme *Upřesnit*, zatrhneme pole *Šifrovat obsah...* V případě, že se jedná o složku, bude aktivní i tlačítko vedle tohoto pole, v něm určíme, zda se má šifrování použít i rekurzívně (na podsložky). Když už nebudeme chtít mít tento soubor šifrovaný (například kvůli rychlosti práce se souborem), toto pole odtrhneme.

EFS používá pro symetrické šifrování algoritmus AES, hash algoritmy řady SHA a pro asymetrické šifrování ECC (eliptické křivky), ve starších verzích byl místo ECC používán algoritmus RSA. Běžným uživatelům stačí výchozí nastavení, případná konfigurace se pak provádí v nastavení Zásad skupiny (`gpedit.msc`).

Další informace:

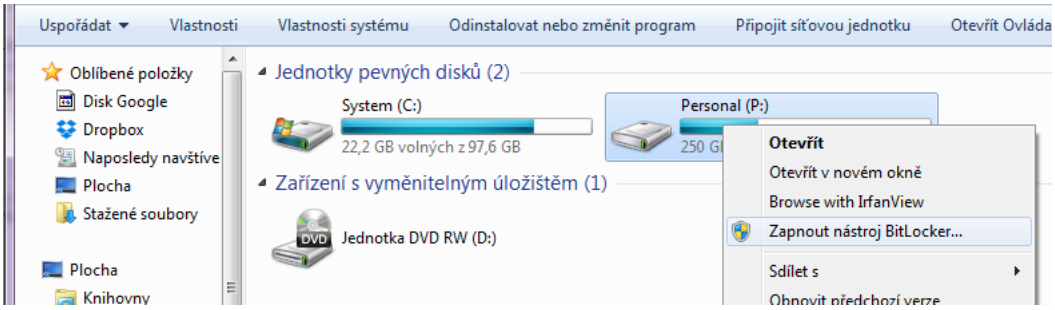
- https://technet.microsoft.com/library/Cc962103
- https://www.winhelp.us/encrypting-file-system-in-windows.html

- https://technet.microsoft.com/en-us/library/dd630631%28v=ws.10%29.aspx

BitLocker Drive Encryption. Tento nástroj je podporován pouze v nejvyšších verzích desktopových Windows (Ultimate a Enterprise) od verze Vista a v serverových verzích od Windows Server 2008 R2. Šifruje vždy celý oddíl nebo disk, varianta BitLocker To Go dokáže pracovat i s výměnnými paměťovými médii (od verze 7). BitLocker není na rozdíl od EFS vázán na oddíly se souborovým systémem NTFS, dokáže pracovat i s oddíly typu FAT a exFAT (ale žádnými non-Win, optickými médii ani síťovými souborovými systémy).

BitLocker používá jako dodatečné zabezpečení buď TPM (Trusted Platform Module) čip nebo jiný způsob většinou hardwarového zabezpečení (USB token, úložiště v Active Directory, Recovery agent nebo něco podobného). Pokud je na základní desce čip TPM, BitLocker tento čip použije k uložení soukromých klíčů pro asymetrické šifrování, v opačném případě musíme zvolit jinou metodu.

Samotný proces šifrování a dešifrování je podobný jako u EFS (jen se týká celého oddílu a máme dodatečné zabezpečení), používá se symetrická šifra AES v módu CBC, asymetrická RSA, hashe SHA a HMAC.



Obrázek 5.4: Aktivace nástroje BitLocker Drive Encryption ve Windows 7

Aktivace se provádí v kontextovém menu příslušného oddílu (viz obrázek 5.4), případná konfigurace také v Zásadách skupiny.

Je zřejmé, že ze zašifrovaného systémového oddílu by nebylo možné bootovat (protože dešifrování se provádí až během bootování), tedy musíme mít k dispozici alespoň malý oddíl, ze kterého by bylo bootování Windows zahájeno. Windows při své instalaci na prázdný disk automaticky vytvářejí malý speciální oddíl sloužící právě pro tento účel.

Další informace:

- http://www.forensicswiki.org/wiki/BitLocker_Disk_Encryption
- https://technet.microsoft.com/en-us/library/ee449438%28v=ws.10%29.aspx
- http://www.howtogeek.com/193649/how-to-make-bitlocker-use-256-bit-aes-encryption-instead-of-128-bit-aes/
- http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1332.pdf
- Zdroj [3] v seznamu literatury na konci kapitoly

Poznámka:

Uvedená řešení jsou „lokální“ – nejsou použitelná tehdy, když chceme soubor bezpečně transportovat mezi dvěma počítači, ať už jakkoliv (snad až na BitLocker To Go). Samotný BitLocker dokonce uzamkne disk v případě, že nedodáme tu správnou hardwarovou komponentu (například pokud disk, při jehož šifrování byl použit TPM čip, vyndáme a zkoumáme na jiném počítači).

5.2.2 Šifrování disků v Linuxu

V Linuxu a jiných UNIXových systémech coby mnohem volnějších a modulárnějších produktech není přímo určité řešení vestavěno, ale uživatel obvykle některé přesto k dispozici má. V různých linuxových distribucích se můžeme setkat s více různými nástroji. Některé z nich se ovládají v textovém režimu, jiné mají i GUI s integrací do prostředí v podobě položky v kontextovém menu.

Co se týče konkrétních algoritmů a módů, šifrovací nástroje využívají to, co je implementováno v CryptoAPI daného systému. Typicky se jedná o AES, Twofish, Serpent apod. pro symetrické šifrování, také je výběr mezi různými módy – jednoduše využijí, co je k dispozici. Tedy v následujících odstavcích se nebudeme bavit o implementovaných algoritmech a módech, v tomto ohledu jsou šifrovací nástroje modulární.

DM-crypt. Tento nástroj pracuje jako transparentní filtr šifrující a dešifrující komunikaci s konkrétním paměťovým médiem (je určen pro disky, oddíly, RAID, LVM, výměnná média) nebo konkrétním souborem obsahujícím obraz disku či oddílu. Jedná se modul běžící v jádře, jako frontend (ovládací program v uživatelském režimu) používáme buď *cryptsetup* nebo *cryptmount*, obojí má textové rozhraní. Pro správu klíčů se typicky používá LUKS (Linux Unified Key Setup).

Pokud chceme grafické rozhraní, i tu možnost máme – záleží, jaké desktopové prostředí je nainstalováno: například v GNOME umí s DM-crypt pracovat například gnome-disk-utility, můžeme ji najít pod názvem GNOME Disks. V takovém nástroji se s šifrovanými oddíly zachází vpodstatě podobně jako ve Windows, tedy přes volbu v kontextovém menu.

Další možnosti pro šifrování disků. Dřív byl i v Linuxu populární TrueCrypt, také je k dispozici Loop-AES a pak další nástroje, které byly portovány i na jiné systémy, a tedy se s nimi seznámíme o pár odstavců dále.

Šifrování souborů. Pokud nechceme šifrovat celý diskový oddíl, můžeme použít nástroj *eCryptfs*. Výhodou eCryptfs je nezávislost na konkrétním paměťovém médiu (veškerá metadata ukládá do záhlaví šifrovaného souboru), a tedy je možné šifrovaný soubor přenášet mezi různými počítači. Najdeme ho na mnoha NASech.

EncFS na rozdíl od jiných šifrovacích nástrojů nemá žádnou svou vlastní část v jádře, místo toho používá modul jádra FUSE (ostatně, přesně tak funguje i podpora NTFS v Linuxu). Také pracuje jako speciální filtr mezi uživatelským procesem a ovladačem skutečného souborového systému na příslušném oddílu disku.

Další informace:

- http://www.cyberciti.biz/hardware/howto-linux-hard-disk-encryption-with-luks-cryptsetup-command/

- http://martin.vancl.eu/sifrovany-disk-kontejner-s-dm-crypt-luks
- http://www.root.cz/clanky/proc-a-jak-na-sifrovani-disku-v-linuxu/
- https://www.digitalocean.com/community/tutorials/how-to-use-dm-crypt-to-create-an-encrypted-volume-on-an-ubuntu-vps
- https://github.com/vgough/encfs
- http://lifehacker.com/five-best-file-encryption-tools-5677725
- https://wiki.gnome.org/Design/Apps/Disks (GNOME Disks)

5.2.3 Nástroje třetích stran

„První“ a „druhá“ strana jsou dotyčný operační systém (jeho producent) a uživatel, tedy třetí strana je prostě někdo mimo tuto dvojici. Kdybychom chtěli být přesní, s nástroji třetích stran jsme se setkali už v předchozí podsekci o nástrojích v Linuxu, nicméně zde se budeme věnovat zejména nástrojům existujícím pro různé operační systémy nebo alespoň pro Windows.

TrueCrypt. Je (vlastně byl) určen pro Windows, Linux i některé další UNIXové systémy, včetně MacOS X. Šifruje jak jednotlivé soubory, tak i celé oddíly. Vše šifrované ukládá do jediného chráněného souboru šifrovaného některým symetrickým algoritmem (případně nekolika různými v kaskádě) v různých módech (často XTS) v kombinaci s hashovacím algoritmem (většinou Whirlpool), přičemž je třeba používat dostatečně silné heslo.

O tomto nástroji zde již bylo psáno dostatečně, v současné době se v souvislosti s ním hovoří již jen o následovnících, jejichž kvalita by byla alespoň srovnatelná.

Další informace:

http://opensourceforu.efytimes.com/2012/02/truecrypt-learn-the-art-of-encryption/

VeraCrypt. Běží v systémech Windows, Linux, MacOS X. Typicky se používá pro šifrování celých oddílů nebo disků (včetně výměnných médií), přičemž vše probíhá na pozadí bez nutnosti asistence uživatele.

Z následovníků TrueCryptu je svému vzoru asi nejpodobnější, dokonce existuje určitá kompatibilita – je možné archivy TrueCryptu buď přímo používat ve VeraCryptu nebo alespoň provést konverzi. Podporuje různé šifrovací algoritmy a hash funkce, výběr máme vpodstatě podobný jako u TrueCrypt. Zapnutí, vypnutí a konfigurace šifrování se provádí v aplikaci s grafickým rozhraním.

VeraCrypt můžeme použít i pro zašifrování systémového disku, na kterém je nainstalován některý operační systém (typicky Windows) – v tomto případě se používá mód XTS. Z takového oddílu však nelze spustit systém přímo, vždy nejdřív musí proběhnout autentizace VeraCryptu (pre-boot authentication). Reálně to znamená, že se nejdřív spustí nikoliv boot-loader Windows, ale VeraCrypt Boot Loader, který zajistí autentizaci a následně zprovozní boot-loader Windows. Takto probíhá nejen skutečný start operačního systému, ale také například probrání z hibernace.

Další informace:

- https://veracrypt.codeplex.com/

- https://veracrypt.codeplex.com/wikipage?title=Beginner%27s%20Tutorial
- https://www.veracrypt.fr/en/System%20Encryption.html

AxCrypt. Tento nástroj je určen pouze pro Windows. Dokáže šifrovat jednotlivé soubory či složky (i rekurzívně), spolupracuje také s většinou cloudových úložišť. Používá symetrický algoritmus AES se 128bitovým klíčem s módu CBC, pro hashe SHA-1, což pro domácí použití jakž takž stačí. Je distribuován pod licencí GNU GPL, tedy jsou k dispozici i zdrojové kódy.

Pro uživatele Windows je velmi intuitivní – po instalaci se v kontextovém menu souborů a složek objeví položka AxCrypt, přes kterou provádíme jak samotné šifrování, tak i konfiguraci.

Další informace:

http://www.axantum.com/AxCrypt/

GnuPG. Tento nástroj běží pod Windows, Linuxem a mnoha UNIXovými systémy včetně MacOS X. Jedná se o free software šířený pod licencí GNU GPL. Používá se velmi často k šifrování e-mailů nebo komunikace mezi IRC klienty, ale také k šifrování souborů nebo celých oddílů a disků včetně výměnných médií. Další informace k GnuPG jsou na straně 149.

Používá se hybridní šifrování – je vygenerován symetrický klíč (session key), tímto klíčem jsou zašifrována data, a následně je symetrický klíč zašifrován veřejným asymetrickým klíčem. Dešifrování dat zase probíhá tak, že nejdřív je soukromým asymetrickým klíčem dešifrován soukromý klíč, a ten je následně použit k dešifrování dat. Jsou podporovány různé symetrické algoritmy včetně AES, jako asymetrický algoritmus je ve výchozím nastavení zvolen DSA.

Instalační program pro Windows je dostupný v rámci projektu *GPG4Win*. V Linuxu je buď už nainstalován nebo stačí nainstalovat balíček `gnupg2` (na webu projektu je v části FAQ popis pro jednotlivé distribuce), na MacOS X to je projekt *GPG Tools*. Nástroj je možné ovládat v textovém režimu nebo pomocí programů s grafickým rozhraním (například součástí GPG4Win je aplikace GPA, na různých platformách funguje GnuPG Shell).

Další informace:

- https://www.gnupg.org/
- http://www.gpg4win.org/ (Windows), http://sourceforge.net/projects/gpgosx/ (MacOS X)

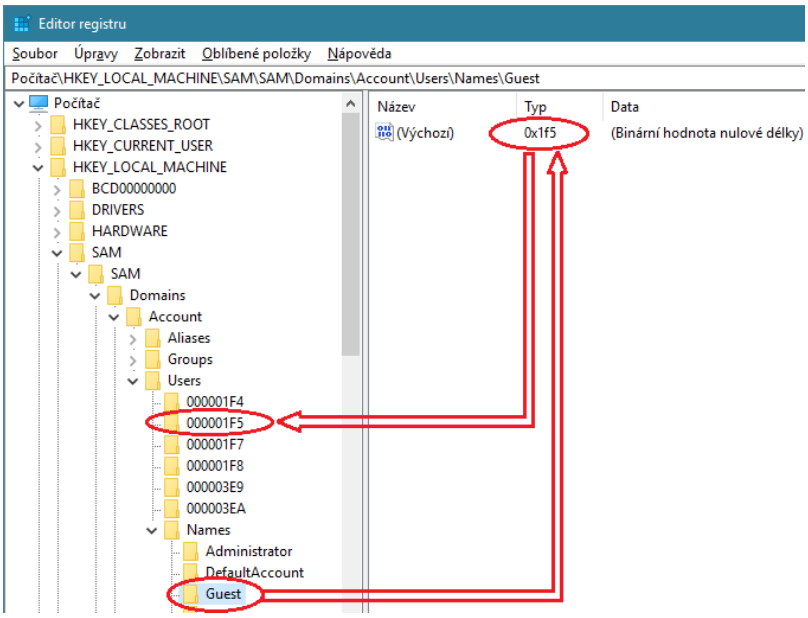
5.3 Autentizace

5.3.1 Autentizace na počítači

Na straně 98 bylo zmíněno, že se hash funkce dnes používají při ukládání autentizačních informací. Operační systém tedy neukládá přímo hesla, ale pouze jejich hashe (digitální otisky).

Hesla ve Windows. V případě Windows jsou hashe hesel zároveň s dalšími informacemi uloženy v registru v klíči `HKEY_LOCAL_MACHINE/SAM` (SAM je Security Accounts Manager), tedy v souboru `.../Windows/System32/config/SAM`.

Řetězec `HKEY_LOCAL_MACHINE` budeme dále zkracovat na `HKLM`.



Obrázek 5.5: Editor registru – zabezpečení uživatelů

Poznámka:

Dostat se do klíče `SAM` není až tak jednoduché, protože vlastníkem celého klíče je systémový účet a oprávnění jsou nastavena dost „napevno“. Přístup tam nemá dokonce ani administrátor (přesněji – „kousek“ dovnitř se sice dostane, ale uvnitř se mu nic nezobrazuje). Řešením je převzít vlastnictví jak klíče `SAM`, tak i jeho stejně pojmenovaného podklíče (pravým tlačítkem myši klepneme na klíč, položka *Oprávnění*). Abychom toto mohli provést, musíme pracovat s oprávněními správce.

Postup (Průzkum bezpečnostních informací o uživatelském účtu)

Pokud se jedná o hesla uživatelů, jsou zakódovaná v podklíči `HKLM/SAM/SAM/Domains/Account/Users` (viz obrázek 5.5). Kam jít:

- v podklíči `Names` najdeme příslušného uživatele,
- zjistíme si jeho RID (Relative ID), což je hexadecimální číslo, které se nám objeví poněkud netypicky ve sloupci *Type* (*Typ*),
- v Users najdeme podklíč pojmenovaný podle tohoto RID,
- v tomto podklíči je hodnota pojmenovaná písmenem „`V`“.

Hodnota „`V`“ je binární řetězec, ze kterého se dají vyextrahovat různé údaje (tj. kombinuje více informací do jednoho výsledného řetězce). Jsou v něm postupně zakódovány především tyto informace:

- přihlašovací jméno k účtu,
- „dlouhé“ jméno,
- komentář,
- domovský adresář uživatele (profil),
- dva hashe hesel,

- další údaje, jako informace, zda jde o správcovský účet nebo naopak hostovský, počet hesel v historii, atd.

Na začátku řetězce jsou uloženy offsety („pointery“ ukazující dál do řetězce na místo, kde se daná informace nachází), za (skoro) každým offsetem je i délka dané informace (takže například za offsetem přihlašovacího jména je délka přihlašovacího jména). Za těmito metainformacemi tedy následují samotné informace.

Například hash přihlašovacího jména je od adresy 0x0C do adresy 0x0F (desítkově 12 až 23), následující údaj (délka přihlašovacího jména) začíná na adrese 0x10. Offsety hashů hesel začínají na adrese 0x9C. Takže pokud se chceme dostat k hashům, musíme si na této adrese přečíst adresu hashů, k ní přičíst ještě hexadecimální číslo 0xCC, tedy adresa pro hashe samotné je V[0x9C]+0xCC.

Adresy jsou ukládány systémem little-endian, na což je třeba myslet, když jednotlivé Byty skládáme za sebe (na nejnižší adresu se ukládá nejméně významný Byte, tj. při čtení obracíme pořadí Bytů).

Další zajímavý klíč je `HKLM/SAM/SAM/Domains/Builtin` obsahující podobné údaje o dalších účtech (skupinových), včetně Backup Operators, Users, Remote Desktop Users, atd., informace jsou zde strukturovány podobně.

Pokud jde o hashe hesel pro doménu (ve firemní síti, na počítači je možné je najít v klíči `HKLM/Security/Cache`, pokud jsme zrovna do domény přihlášeni. Tyto údaje pro lokální i doménové účty dále mohou být v operační paměti.

Z předchozího plyne, že ve skutečnosti nejde jen o jeden hash. Ve Windows se standardně ukládají dva hashe hesla – LMHash a NTLMHash. LMHash je považován za nepříliš bezpečný (nejde jen o algoritmus bez „soli“ a jakékoliv náhodnosti jako je třeba IV vektor, ale například není možné mít delší heslo než 14 znaků, toto heslo se rozdělí na dvě části a obě se hashují zvlášť), jeho jedinou výhodou je široká kompatibilita v různých verzích Windows, dokonce až ke starému MS-DOSu. NTLMHash je mnohem bezpečnější, používá se od Windows Vista. Heslo pro LMHash se před hashováním konvertuje na velká písmena, kdežto NTLMHash se konvertuje do Unicode.

S využitím LMHashe se dnes setkáváme spíše v případě, kdy je novější systém provozován v lokální síti se staršími verzemi nepodporujícími NTLMHash, ovšem to je čím dál vzácnější situace.

Poznámka:

V případě, že systém generuje a ukládá i LMHash, můžeme vypnout použití LMHashe a vynutit si pouze NTLMHash. Provádí se to v registru v klíči `HKLM/SYSTEM/CurrentControlSet/Control/Lsa`, kde najdeme hodnotu `NoLmHash`. Pokud je tato hodnota nastavena na „1“, pak je to v pořádku (je vypnut LMHash). Pokud je nastavena na „0“ nebo vůbec neexistuje, měli bychom ji vytvořit (pokud tedy neexistuje) a nastavit na jedničku. Tím zakážeme generování a ukládání LMHashe.

Ve Windows 10 od buildu Aniversary Update se používá sice také hash MD4, ale v kombinaci s AES128 (místo DES), což sice taky není úplně „top level“, ale je to lepší než v předchozích buildech/verzích.

Další informace:

V dokumentu http://www.acad.ro/sectii2002/proceedings/doc2013-3s/05-OPREA.pdf najdeme podrobnější informace o struktuře informací o účtech v registru Windows. Ovšem jde o verze do XP (použití MD5, DES apod.).

Novější, ale stručnější informace je na https://security.stackexchange.com/questions/157922/how-are-windows-10-hashes-stored-if-the-account-is-setup-using-a-microsoft-accou.

Další zdroj (ale „useknutý“) je
https://ad-pdf.s3.amazonaws.com/Forensic_Determination_Users_Logon_Status.pdf.

Hesla v Linuxu. Hashe hesel jsou zároveň s jinými souvisejícími informacemi uloženy v souborech `/etc/shadow` (hesla uživatelů) a `/etc/gshadow` (hesla skupin). Jedná se o textové soubory, jejichž vlastníkem je root (to je hlavní administrátor). Uživatelé s vyššími oprávněními mohou tento soubor alespoň číst.

Zaměřme se na soubor `/etc/shadow`. Každý uživatel (včetně systémových uživatelů) zde má jeden řádek, záznam na řádku má tuto strukturu (jednotlivé položky jsou odděleny dvojtečkou):

- přihlašovací jméno,
- hash hesla s údajem o použitém algoritmu,
- čas, kdy bylo heslo naposledy změněno (počet dnů od 1. ledna 1970),
- *min*: nejmenší počet dnů, který musí uplynout od poslední změny hesla, aby bylo možné znovu změnit heslo (často bývá 0), používá se obvykle jen tehdy, když je použito pole max,
- *max*: maximální doba platnosti hesla v počtu dnů, tedy vynucuje na uživateli pravidelně měnit heslo (často bývá 99999, což znamená, že se změna nevynucuje),
- *warn*: pokud je používána hodnota max, pak je zde počet dnů, po které je před uplynutím hodnoty max uživatel varován, že musí změnit heslo (např. když je tu číslo 7, pak během 7 dnů před uplynutím lhůty *max* je uživatel při každém přihlašování varován),
- *inactive*: pokud uživatel promeškal lhůtu max, pak po dobu inactive ještě má možnost při přihlašování změnit heslo, ovšem bez změny hesla se do systému nedostane,
- *expire*: účet může být dočasný (s omezeným časem platnosti), pak je zde čas, kdy přestane platit – počet dnů od 1. ledna 1970 do ukončení platnosti účtu.

Příklad

V souboru `/etc/shadow` můžeme mít třeba takovýto řádek:

```
jannovak:$6$18APx6475Bw....3oTI1:17562:3:90:7:99999
```

Co z toho vyčteme:

- uživatel je jannovak,
- pro tohoto uživatele je hash nastaven na uvedenou hodnotu (není zde celý uveden, je to celkem dlouhý řetězec), přičemž na začátku mezi symboly `$` je identifikátor algoritmu (6 znamená SHA-512), pak může/nemusí následovat „sůl“ oddělená od samotného hashe dalším symbolem `$`,
- čas, kdy tento uživatel naposledy měnil heslo, se dá zjistit z čísla 17562 (to je počet dnů od 1. ledna 1970),

- pokud si uživatel změní heslo, pak minimálně 3 dny nemůže heslo znovu změnit,
- na druhou stranu si musí měnit heslo co čtvrt roku (platnost hesla je max. 90 dnů),
- když se blíží termín nucené změny hesla, je uživatel 7 dnů předem při každém přihlašování varován,
- účet má defacto neomezenou platnost.

Pokud je místo hashe hvězdička, znamená to, že heslo není nastaveno. Pokud je před hashem symbol „!“, je účet uzamčen (vpodstatě to heslo je uzamčeno, nemůže být použito pro přihlášení). Dva vykřičníky za sebou znamenají vpodstatě totéž s tím rozdílem, že jde o nově vytvořený účet, do kterého se ještě nikdo nepřihlásil, a při prvním přihlášení si uživatel bude muset nastavit heslo.

Další informace:

Vysvětlení formátu záznamu v `/etc/shadow` máme například na stránce https://www.cyberciti.biz/faq/understanding-etcshadow-file/.

Co s hashi hesel. Pokud máme dostačující přístupová oprávnění a potřebné znalosti, můžeme si hash z příslušného umístění sami vytáhnout (týká se to Windows i Linuxu), nebo můžeme použít k tomu příslušné nástroje. Tyto nástroje se obvykle spouštějí buď přímo na daném systému, nebo na některém zařízení v místní síti, nebo někde úplně jinde (v tom případě musíme dodat hash, který chceme prověřit).

Například komerční program *Proactive Password Auditor* od firmy Elcomsoft prochází síť, hledá cesty k heslům na zařízeních a prověřuje jejich zabezpečení, včetně pokusu o cracknutí.

Velmi známý je volně šiřitelný software *Cain&Abel* s podobnými vlastnostmi, vlastně i poněkud širšími možnostmi použití. Další oblíbený software je *John the Ripper*, *HashCat*, *THCHydra*, *OphCrack*.

Pokud máme k dispozici hash hledaného hesla, můžeme si vytvořit Rainbow table (seznam hesel a k nim příslušných hashů, takový seznam závisí na použitém hash algoritmu a dalších parametrech).

Další informace:

- http://www.elcomsoft.com/ppa.html
- http://www.oxid.it/cain.html
- https://project-rainbowcrack.com/table.htm
- https://securityxploded.com/rainbowcrack.php
- https://neverendingsecurity.wordpress.com/tag/rainbowcrack/ (přehledová stránka)

5.3.2 Reset hesla do Windows pomocí SystemRescueCD

Stane se, že se potřebujeme dostat do systému Windows, ale zapomněli jsme heslo (případně ho zapomněl ten, kdo nás žádá o pomoc). Předně bychom si měli být jisti, že náš pokus o proniknutí do systému bude legální...

Hesla jsou uložena v klíči registru `cd /mnt/windows/Windows/System32/config/SAM` ve formě hashů, navíc je tento klíč chráněn proti přístupu (ovšem pouze možnostmi běžících Windows). Potřebujeme nástroj, který se do tohoto klíče dokáže dostat a smazat hash pro dané heslo.

Postup (Reset hesla do Windows)

Pořídíme si výměnné médium se systémem *SystemRescueCD* nebo jiným obsahujícím potřebné nástroje. Nabootujeme, a pokračujeme následovně:

- zjistíme (například v souboru `/etc/fstab`, jak je označen oddíl s instalací Windows, zde např. `/dev/sda2`
- připojíme oddíl, na kterém jsou nainstalovány Windows
  `mkdir /mnt/windows`
  `ntfs-3g /dev/sda2 /mnt/windows` (nebo použijeme `mount`)
- přesuneme se do složky, ve které je registr Windows (dosaďte skutečné umístění)
  `cd /mnt/windows/Windows/System32/config`
- spustíme program, který umožní resetovat heslo
  `chntpw -u username SAM`

Existují však i další nástroje, které jsou pro tento účel použitelné, například EBCD nebo Parted Magic. V každém případě je však třeba obejít instalovaný systém.

Další informace:

- http://onubuntu.blogspot.cz/2011/08/reset-your-windows-password-with.html
- http://partedmagic.com/clearing-the-administrator-password-in-windows/
- http://www.prime-expert.com/articles/a09/temporarily-change-windows-password-and-restore-it-back.php

5.3.3 Dvoufaktorová autentizace

V poslední době se čím dál častěji setkáváme s možností dvoufaktorové autentizace – především v internetovém bankovnictví, ale také u internetových služeb (nabízí ji například Google) nebo mobilních zařízení. Účelem je co nejvíc ztížit „cracknutí“ účtu, je to bezpečnější možnost prokázání, že „jsem to já“.

Definice (Metody autentizace, dvoufaktorová autentizace)

Základní metody autentizace stojí na minimálně jedné možnosti z následujících:

- *něco vím a jen já to vím* (bez této znalosti se dál nedostanu, „anything to know“) – heslo, PIN, odpověď na bezpečnostní otázku
- *něco mám a jen já to mám* (bez této věci se dál nedostanu, „anything to have“) – průkaz, klíč (fyzický), karta, hardwarový token, smartphone
- *něčím jsem a jen já tím jsem* (souvisí to s mou fyzickou existencí, „anything to be“) – biometrie (otisk prstu, rohovky oka, identifikace hlasu, chůze, apod.)

Dvoufaktorová autentizace (2FA) kombinuje alespoň dvě z těchto možností. Multifaktorová autentizace více než dvě (tedy z každého typu alespoň jednu možnost).

V internetovém bankovnictví se dnes často používá dvoufaktorová autentizace využívající první dvě možnosti – něco vím (heslo) + něco mám (autentizační řetězec zaslaný v SMS nebo vygenerovaný mobilní aplikací). U některých internetových služeb je volitelná (Google, Facebook, atd.).

Jaké mohou nastat problémy:

- problém mezi židlí a klávesnicí (uživateli se nedaří, uživatel protestuje, uživatel odmítá, uživatel zahlcuje servisní podporu),
- není až tak neprůstřelná, jak by se mohlo zdát,
- mohou nastat technické problémy (např. u biometrie), nebo u podmínky „něco mám“ to „zrovna nemám“,
- může přestat být dvoufaktorovou (např. když přes smartphone přistupujeme tam, kde je druhým faktorem „něco mám“ právě smartphone).

Poznámka:

Ani dvoufaktorová autentizace není plně spolehlivá. Například:

- Pokud v internetovém bankovnictví používáme pro potvrzení platby kombinaci hesla a potvrzovací SMS, měli bychom zkontrolovat obsah zaslané SMS (bývá tam kromě jiného i částka, která má být převedena). Vyskytly se už podvody s navýšením částky.
- Pokud do internetového bankovnictví přistupujeme přes mobilní zařízení a toto mobilní zařízení nám slouží jako „druhý faktor“, není to úplně v pořádku, protože napadnout mobilní zařízení a zmanipulovat jeho komunikaci se sítí není až tak složité. V tomto smyslu jsou nejproblematičtější zařízení s Androidem, ale existují zranitelnosti i pro další mobilní platformy.
- Bezpečnější než mobilní telefon (zvláště s Androidem) je jako druhý faktor mít hardwarový token (obvykle zařízení velikosti přívěšku na klíče).
- Jakékoliv bezpečnostní opatření stojí a padá na způsobu a důkladnosti jeho implementace.

Další informace:

- http://www.systemonline.cz/it-security/dvoufaktorova-autentizace-myty-a-realita.htm
- http://www.cleverandsmart.cz/vicefaktorova-autentizace/

5.4 Bezpečná komunikace se serverem

5.4.1 SSL a TLS

V aplikacích typu webový prohlížeč, e-mailový klient a další se pro zajištění zabezpečené komunikace používají protokoly SSL a TLS.

SSL (Secure Sockets Layer) je síťový protokol sloužící k autentizaci komunikujících stran – buď oboustranně, nebo se autentizuje jen jedna strana.

Účelem je zajistit důvěrnost a integritu dat. Důvěrnost se zajišťuje hybridním postupem – jako symetrický algoritmus lze použít několik různých algoritmů včetně AES, to je doplněno asymetrickým algoritmem (dnes typicky Diffie-Hellman nebo DSA), jako hash je používán MD5 nebo SHA-1. Důvěryhodný web by měl mít svůj veřejný klíč důvěryhodně podepsaný, tedy certifikát.

V současné době existuje několik verzí SSL, z nichž starší (do verze 2.0 včetně) již nejsou považovány za bezpečné.

TLS (Transport Layer Security) je považován za bezpečnějšího nástupce SSL. Zatímco SSL není přímo standardizován, TLS je standardizován v RFC (TLS verze 1.2 je popsán v RFC 5246, verze 1.3 v RFC 8446 ze srpna 2018). TLS verze 1.0 je téměř totožný s SSL verze 3.0, až na drobné odlišnosti, přesto jsou tyto dva protokoly navzájem nekompatibilní (tj. komunikující strany používají obě buď jeden nebo druhý protokol, je třeba se na začátku komunikace domluvit – i včetně verze).

Ať už se jedná o SSL nebo TLS, vždy je třeba nejdřív vytvořit *session* – spojení mezi dvěma uzly v síti (klientem a serverem), tedy něco na způsob komunikační cesty mezi dvěma městy, a poté již *connection* – spojení mezi konkrétními procesy na obou stranách.

SSL/TLS Handshake protocol zajišťuje vytvoření session, což může vypadat takto:

- klient pošle požadavek na spojení (například při komunikaci s webovým serverem se zadá adresa `https://...`),
- server zašle klientovi certifikát svého veřejného klíče,
- klient vygeneruje náhodné číslo, zašifruje podle veřejného klíče serveru, pošle serveru (pre-master secret),
- server dešifruje svým soukromým klíčem, klient i server přidají k těmto datům další informace a vytvoří hlavní šifrovací klíč (master secret).

Bezpečnější alternativou je využití algoritmu Diffie-Hellman, který máme popsán na straně 115. Součástí handshake je samozřejmě také dohoda o použití konkrétních kryptografických algoritmů – takových, kterým rozumí obě strany, přičemž u TLS je finální rozhodnutí na straně serveru.

Další informace:

- https://www.wolfssl.com/wolfSSL/Blog/Entries/2010/10/7_Differences_between_SSL_and_TLS_Protocol_Versions.html
- Odvárka, Petr. SSL protokol a jeho akcelerace [online]. Seriál na *Svět Sítí.cz*, 2002 [cit. 2015-11-19]. Dostupné z: http://www.svetsiti.cz/rubrika.asp?rid=17&tid=171
- https://www.thesslstore.com/blog/tls-1-3-approved/ (je zde i pěkně graficky vyvedený postup navazování TLS spojení)
- https://datatracker.ietf.org/doc/rfc8446/
- https://wiki.openssl.org/index.php/SSL_and_TLS_Protocols

5.4.2 Webový server a webový prohlížeč

V současné době slouží webový prohlížeč nejen ke komunikaci s běžnými webovými servery, ale také k nakupování v internetových obchodech, komunikaci s bankou, firemním souborovým či aplikačním serverem, e-mailovým serverem, placenou databázovou službou atd. Přenášíme často citlivá data typu přihlašovacích a kontaktních údajů, čísla kreditní karty, objednávek, apod. To vše by mělo být přenášeno zabezpečeným kanálem – šifrováno a s ověřením komunikujících stran.

Navázání a zajištění průběhu zabezpečeného připojení k webovému serveru probíhá pomocí protokolu `https` (bez potřeby zabezpečení se jedná o protokol `http`). Samotné šifrování (včetně doprovodných kroků) pak zajišťuje některý z protokolů SSL nebo TLS.

Poznámka:

Zabezpečenou komunikaci poznáme na první pohled tak, že v adresním řádku je před adresou serveru napsáno `https://...` a u adresního řádku je ikona zámku (většinou zelená – konkrétní tvar a umístění se může u různých webových prohlížečů lišit).

Není dobré se orientovat jen podle ikony zámku – není problém tuto ikonu podvrhnout. Větší jistotu získáme, když v adresním řádku vidíme `https://...`, nicméně ani pak to neznamená, že bychom byli kryti před všemi možnými útoky.

Postup (Ověření zabezpečení komunikace s webovým serverem)

Následující postup platí pro Firefox, v jiných prohlížečích to bude podobné.

První pohled by měl vést k ikoně zámku u adresy. Po klepnutí na ikonu a následně na šipku v zobrazeném poli se dostaneme do okna s kompletními informacemi o certifikátu.

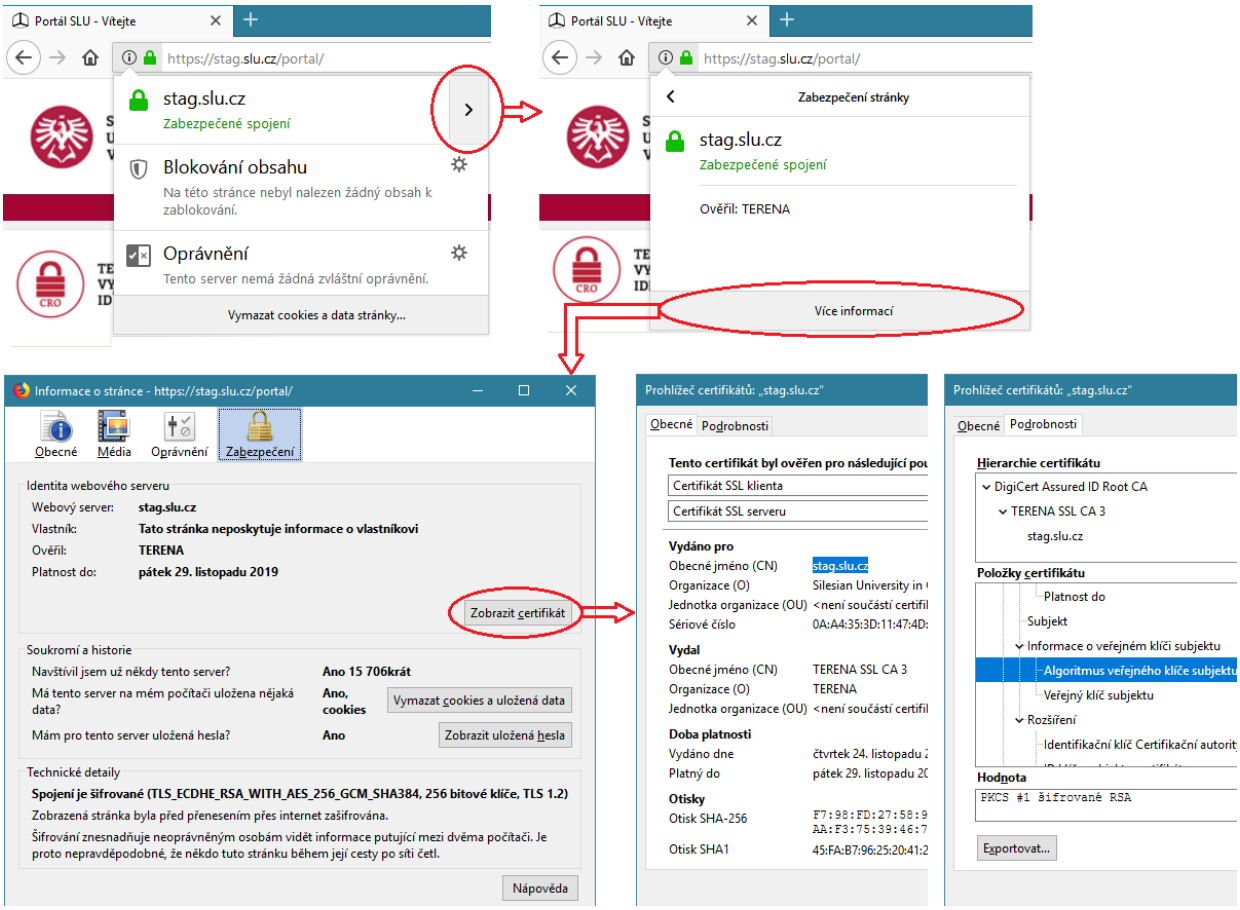
Na obrázcích 5.6 a 5.7 je ukázán jak postup, tak i (částečně) výsledek. V okně s údaji o certifikátu jsou dvě záložky – na první najdeme nejdůležitější informace, na druhé pak podrobnosti včetně sériového čísla certifikátu, vydavatele certifikátu, algoritmu podpisu certifikátu a algoritmu veřejného klíče (asymetrický – většinou RSA nebo eliptické křivky, jako hash obvykle SHA-256), samotného veřejného klíče, platnosti, atd.

Postup (Dostupné certifikáty ve Firefoxu)

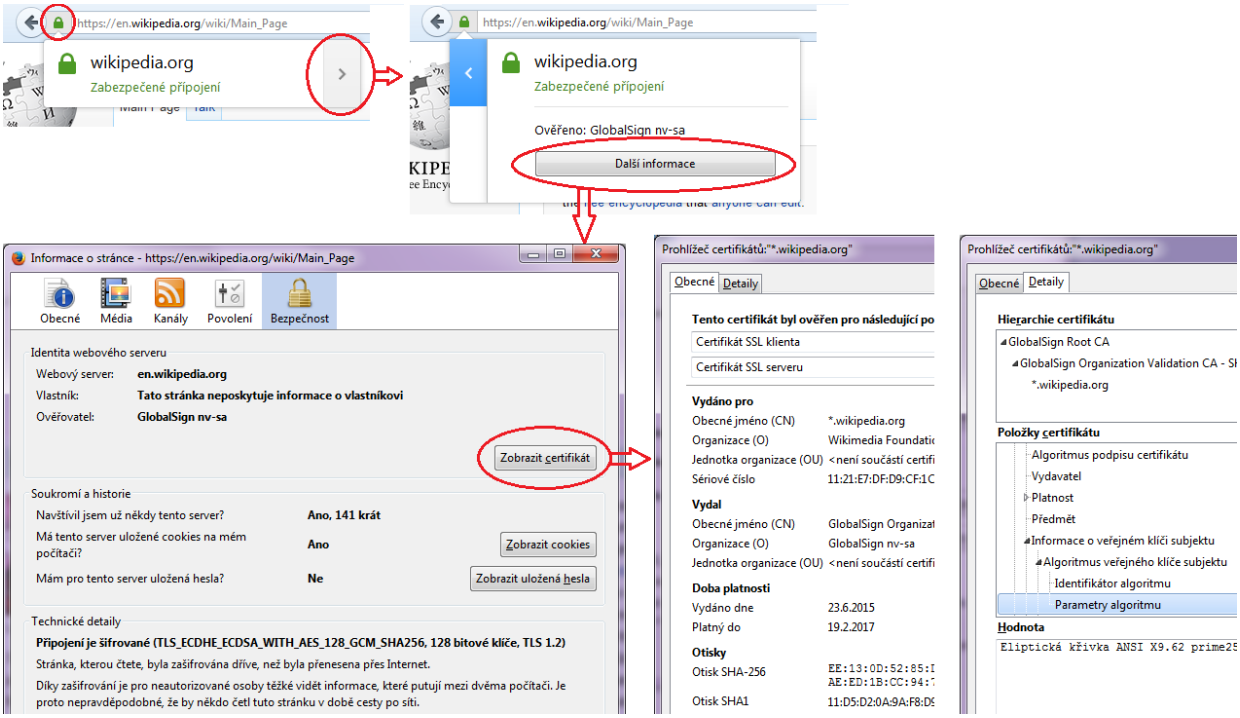
V prohlížeči Firefox se k instalovaným certifikátům dostaneme následovně: v *Možnostech* zvolíme *Rozšířené*, část *Certifikáty*. Dole jsou dvě tlačítka – *Certifikáty* a *Bezpečnostní zařízení*. Druhé z nich použijeme, pokud nás zajímá, jak konkrétně jsou certifikáty uloženy – obvykle jde o softwarové moduly (knihovny), ale může zde být uvedeno i hardwarové bezpečnostní zařízení (třeba hardwarový token).

Klepneme na tlačítko *Certifikáty*, jak je ukázáno na obrázku 5.8. Zobrazí se dialogové okno *Správce certifikátů*. V něm je několik záložek, z nichž nás zajímají následující:

- *Osobní* – pokud máme vlastní certifikát, který nás v komunikaci identifikuje, zde by měl být importován.
- *Servery* – seznam certifikátů serverů komunikujících přes SSL/TLS uspořádaný podle jejich certifikačních autorit. Po poklepání na server se zobrazí podrobná informace o certifikátu.
- *Autority* – seznam certifikačních autorit (jejich certifikátů) s určením bezpečnostního zařízení

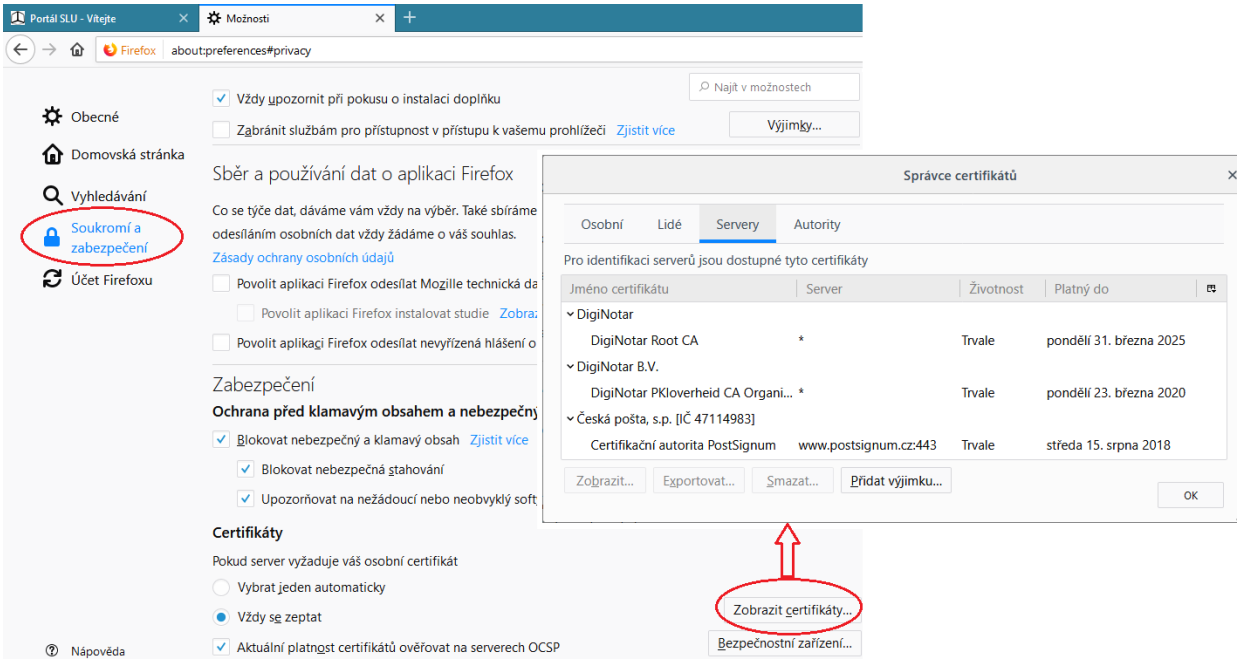


Obrázek 5.6: Údaje o certifikátu pro stránku https://stag.slu.cz/portal/



Obrázek 5.7: Údaje o certifikátu pro stránku https://en.wikipedia.org/wiki/Main_Page

zajišťujícího jejich uložení. Po poklepání se zobrazí informace o certifikátu, přes tlačítko *Upravit důvěru* můžeme zjistit, k jakým účelům může být daný certifikát používán, případně tento údaj upravit. Tlačítko *Smazat nebo nedůvěřovat* použijeme, pokud některý certifikát považujeme za nedůvěryhodný.



Obrázek 5.8: Certifikáty ve Firefoxu

Postup (Externí ověření bezpečnosti komunikace)

Existují stránky, na kterých můžeme otestovat zabezpečení konkrétního webového serveru. Zde se podíváme na dvě takové adresy:

- https://www.ssls.cz/ssltest.html
- https://www.ssllabs.com/ (odkaz „Test your server“)

Na první z těchto adres se testují pouze údaje v certifikátu, zadáváme testovanou adresu. Výstup vidíme na obrázku 5.9 (před testováním je nutno prokázat, že „nejsme roboti“):

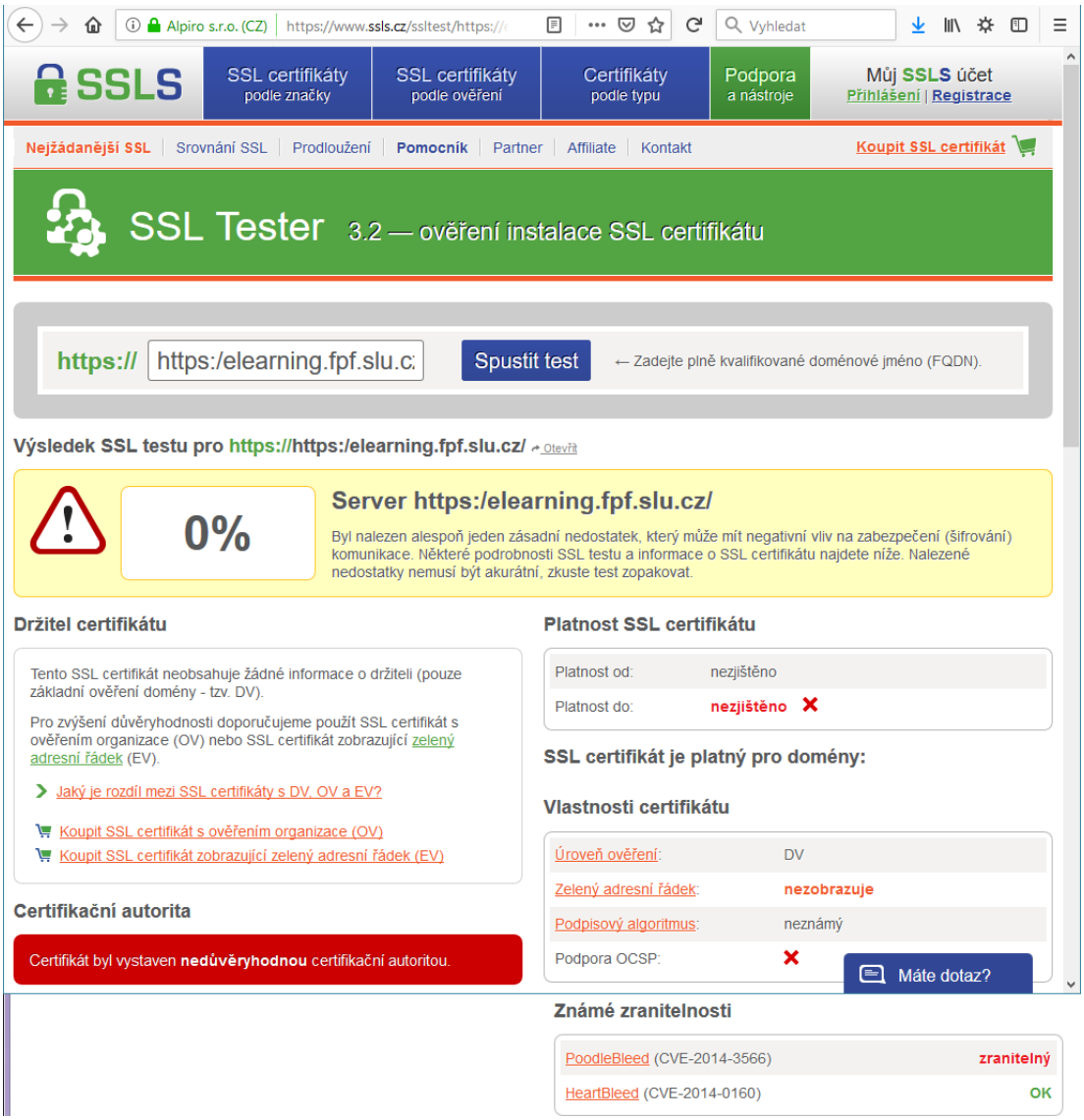
Test dopadl špatně, protože certifikační autorita, od které má web certifikát, zřejmě „vypadla“ z PKI z důvodu nedůvěryhodnosti.

Služba na druhé uvedené adrese je již důkladnější – otestování chvíli trvá, výstupem je podrobný report nejen rozebírající informace v certifikátu, ale komentující bezpečnost podporovaných protokolů, sad kryptovacích algoritmů, zranitelnost vzhledem ke známým útokům, apod. Záhlaví reportu je na obrázku 5.10, ovšem celá stránka je velmi dlouhá – radši vyzkoušejte sami.

Jak vidíme, na první adrese se testuje pouze samotný certifikát, na druhé pak kompletně vlastnosti SSL/TLS zabezpečení.

Poznámka:

Měli bychom si uvědomit, že samotný certifikát neslouží k šifrování ani podepisování. Abychom mohli zprovoznit SSL/TLS, potřebujeme celkem tyto tři položky:



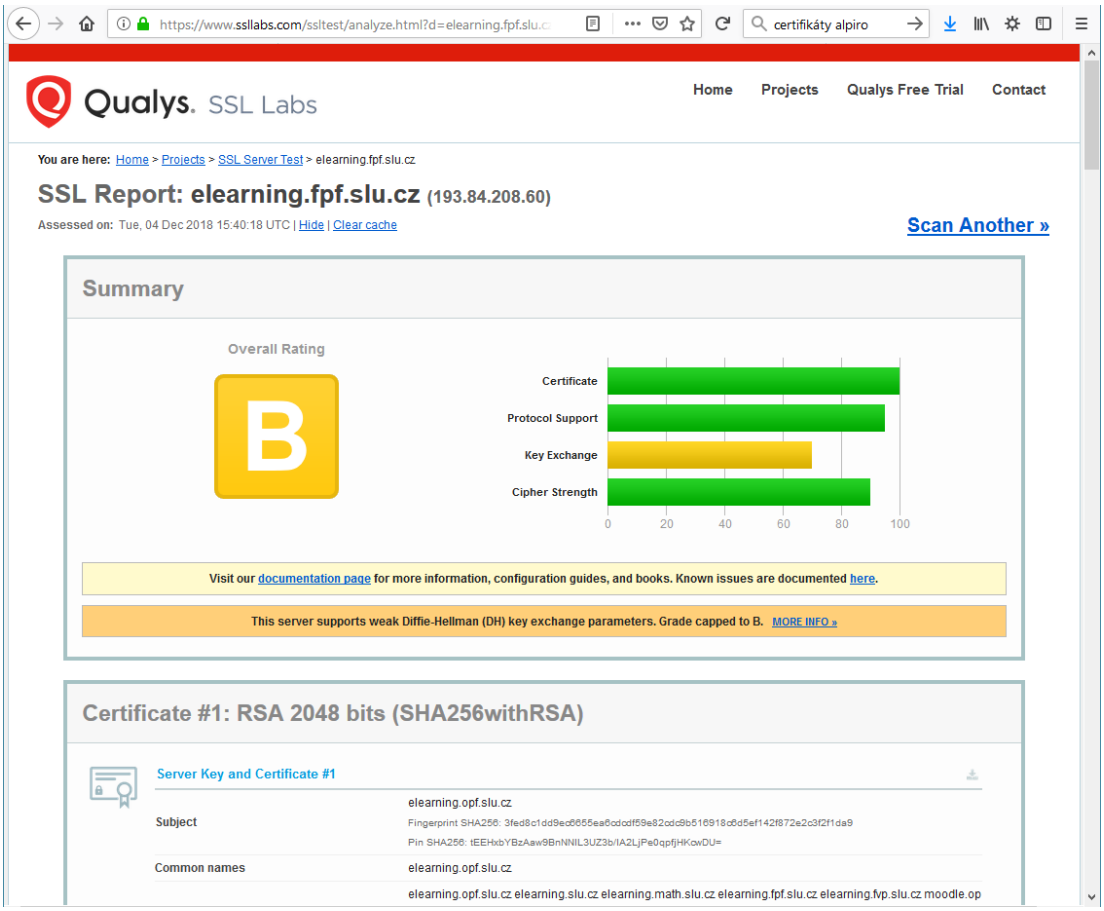
Obrázek 5.9: Testování certifikátu na https://www.ssls.cz/ssltest.html

- soubor s vlastním soukromým klíčem,
- soubor s vlastním certifikátem (v něm je již obsažen i veřejný klíč),
- soubor s certifikátem naší certifikační autority.

První položka slouží k šifrování, další dvě pak zveřejňujeme pro toho, kdo s námi komunikuje.

5.4.3 Vzdálená konfigurace serveru

V dávných dobách jsme se serverem komunikovali pomocí *telnetu*. Jednalo se o komunikaci typu klient-server, tedy uživatel seděl u „běžného“ počítače se spuštěným *telnet klientem*, příkazy se prováděly na serveru (nebo jiném počítači přijímajícím tento typ připojení), na kterém běžel *telnet server*. Uživatel se nejdřív přihlásil, pak zadával příkazy na klientovi, tyto příkazy se prováděly na serveru.



Obrázek 5.10: Testování certifikátu na https://www.ssllabs.com/

Poznámka:

Všimněte si, že celý předchozí odstavec je psán v minulém čase. Telnet má totiž jednu nepříjemnou vlastnost – nic nešifruje, dokonce i heslo při přihlašování je přenášeno nešifrovaně. Dnes se telnet sice v některých firmách ještě používá, ale pouze v rámci interní sítě, a to jen tehdy, když si je administrátor naprosto jist, že je v síti bezpečno. Ruku na srdce – kdo může něco takového tvrdit na 100%?

SSH (Secure SHell) je protokol pro vzdálený přístup na server či obecně na jiný počítač, také se tak může označovat aplikace, která tento protokol používá. V současné době používáme verzi 2. Je považován za bezpečnější náhradu telnetu – zajišťuje totiž šifrovaný přenos (důvěrnost) a integritu, dokáže zprostředkovat bezpečnou autentizaci (přihlášení). SSH je popsán v několika standardech: RFC 4250 (komunikační kódy), RFC 4251 (architektura SSH), RFC 4252 (autentizace), RFC 4253 (komunikace s transportní vrstvou) a RFC 4254 (pro různé druhy spojení).

K čemu se SSH používá? Například:

- potřebujeme na serveru na dálku spustit nějaký program/příkaz,
- chceme editovat některý konfigurační soubor,
- kopírujeme soubor na server nebo naopak soubor ze serveru stahujeme (používáme SFTP),
- chceme vytvořit šifrovaný „tunel“,

- přes SSH můžeme přistupovat nejen k serverům, ale také k jakýmkoliv zařízením, se kterými nelze pracovat přímo pomocí klávesnice (nemají klávesnici, ale běží na nich systém, který „umí“ SSH server),
- atd.

V třetím bodu je zmíněna komunikace se souborovým serverem, se kterým bychom jinak (nezabezpečeně) komunikovali pomocí protokolu FTP. SFTP (tedy zabezpečená varianta FTP) je vlastně kombinací SSL/TLS a FTP.

Základem zabezpečení je asymetrické šifrování kombinované se symetrickým. Může být používán mechanismus Diffie-Hellman (pro bezpečnou výměnu symetrického klíče – session key) nebo jiný. Jako symetrické šifrování pro přenos dat se v SSH 2 obvykle používá AES nebo 3DES.

Důležitým bezpečnostním prvkem je autentizace. SSH umožňuje vybrat si mezi různými způsoby autentizace (záleží taky na konkrétní implementaci, podrobnosti bychom se dozvěděli na stránkách příslušného projektu), obecně máme tyto možnosti:

- zadání hesla, které se v otevřené formě přenáší k serveru (to radši ne),
- nejdřív navážeme šifrované spojení jiným způsobem (například Diffie-Hellman) a pak se přenáší jméno a heslo,
- autentizace na základě IP adresy uživatele,
- Diffie-Hellman, přičemž při požadavku na oboustrannou autentizaci musíme zajistit transport veřejného klíče uživatele na server,
- externí mechanismus, např. GSSAPI využívající Kerberos (v komerční sféře).

Postup (Výběr SSH klienta a SSH serveru)

V SSH klientech máme celkem hodně na výběr. Předně záleží, jaký systém běží na počítači, ze kterého chceme někam vzdáleně přistupovat. Obecně jsou oblíbenější open-source projekty.

V Linuxu či jiném UNIXovém systému (včetně MacOS X) je už většinou předinstalován *OpenSSH*, případně není problém instalovat něco jiného. Solaris má svého vlastního SSH klienta. Ve Windows si musíme klienta doinstalovat, většinou se volí *PuTTY*, *DropBear*, *OpenSSH* nebo na něm založený jednodušší *WinSCP* (mimochodem – produkt českého vývojáře).

Co se serveru týče, v UNIXových systémech už obvykle bývá instalován OpenSSH. Pokud hledáme SSH server, který má běžet na Windows, asi budeme mít trochu problém (obvykle se nepočítá s tím, že bychom s Win serveru přistupovali přes SSH), nicméně taky se najdou projekty, které nabízejí alespoň částečnou SSH funkcionalitu, například komerční Tectia.

Další informace:

- http://www.openssh.com/
- http://www.putty.org/ (přes odkaz dále, stáhněte si celý instalační soubor)
- https://matt.ucc.asn.au/dropbear/dropbear.html
- https://winscp.net/

5.5 E-mail

K zajištění integrity, nepopiratelnosti a důvěrnosti e-mailové komunikace existují dva přístupy. První z nich je použití S/MIME s certifikačními autoritami (X.509), druhý přístup je PGP či GPG, kdy certifikační autority nepotřebujeme. Tyto dva přístupy jsou vzájemně nekompatibilní. Zatímco S/MIME je typičtější pro svět Windows, řešení PGP/GPG je hodně využíváno v UNIXových systémech.

Poznámka:

Je třeba upozornit, že pokud ke své e-mailové schránce přistupujeme přes webové rozhraní poskytovatele služby, velice pravděpodobně nebudeme moci tímto způsobem svou komunikaci chránit. Obvykle potřebujeme nějakého e-mailového klienta – Mozilla Thunderbird, Outlook, Apple Mail, apod., případně mobilní aplikaci ve svém mobilním zařízení.

5.5.1 S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions) zavádí do e-mailové komunikace zabezpečení podle protokolu X.509 a PKI. Obvykle jde o certifikát využívající algoritmus RSA s vhodnou délkou klíče (většinou 2048 bitů).

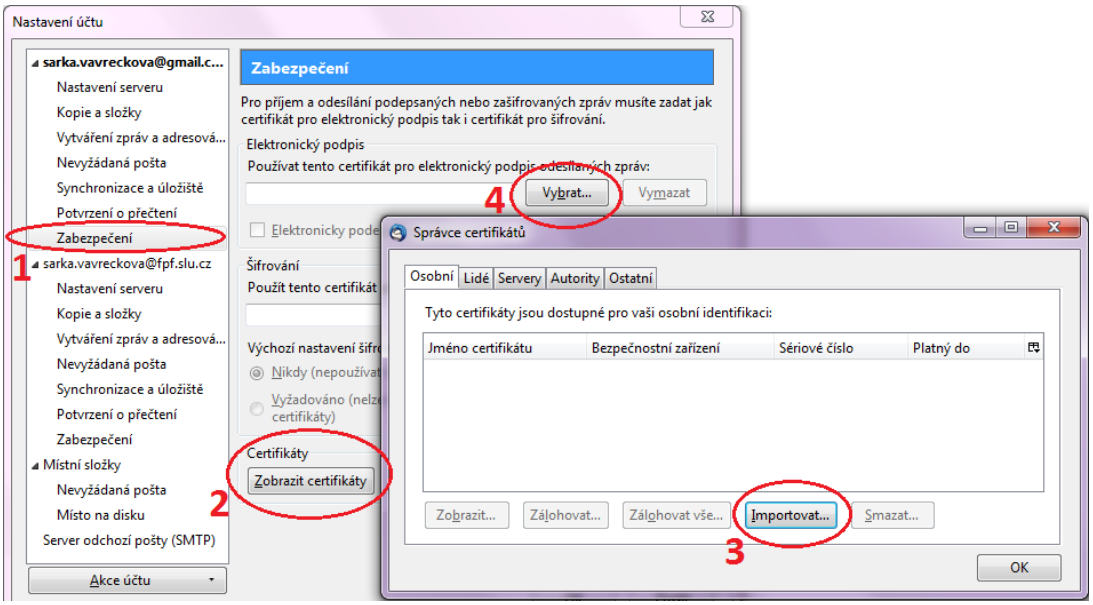
Pokud chceme pouze podepisovat své e-maily, potřebujeme (ideálně důvěryhodný) certifikát, který si nainstalujeme do svého e-mailového klienta (Outlook, Thunderbird apod.), a při psaní zprávy jednoduše v menu zvolíme podepsání zprávy. Jestliže chceme navíc i šifrovat, potřebujeme certifikát s veřejným klíčem té osoby, jíž chceme zprávu adresovat.

Jak bylo výše řečeno, pro podepisování e-mailů nejsou vhodné certifikáty vydané lokálními certifikačními autoritami, protože certifikáty těchto CA typicky nebývají v klientech předinstalovány, takže by byly považovány za nedůvěryhodné. Tedy bychom měli zvolit světovou CA.

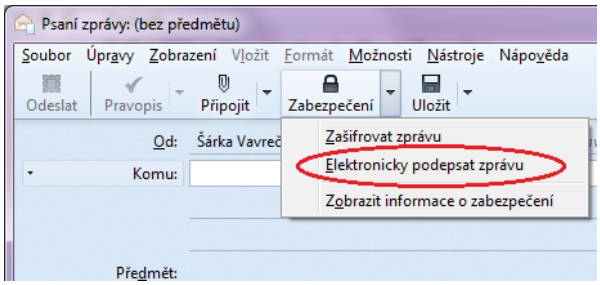
Postup (Získání certifikátu pro podpis e-mailů)

Pro běžné použití nám stačí například certifikát od společnosti Comodo, který lze získat on-line, a to velmi rychle, dokonce zdarma, na adrese
https://www.comodo.com/home/email-security/free-email-certificate.php nebo
https://www.instantssl.com/ssl-certificate-products/free-email-certificate.html.

Výhodou je rychlost a snadnost, nevýhodou je, že takový certifikát zní na konkrétní e-mailovou adresu, nikoliv na jméno (jméno není ověřováno). Nicméně – certifikáty Comodo jsou bez problémů přijímány jako důvěryhodné prakticky ve všech programech a zařízeních pracujících s S/MIME.

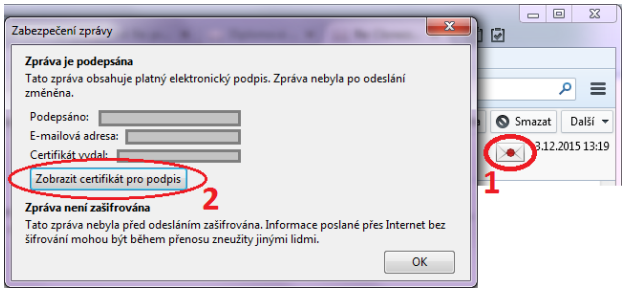


Obrázek 5.11: Instalace vlastního certifikátu v Thunderbirdu



Když už máme svůj certifikát, musíme ho importovat do e-mailového klienta. Na obrázku 5.11 je naznačen postup pro Mozilla Thunderbird (v jiných klientech je to podobné).

Zbývá poslední věc – certifikát použít, tedy odesílaný e-mail podepsat. Na obrázku vpravo vidíme, jak se to provede v aplikaci Mozilla Thunderbird, podobný postup by byl i pro jiné klienty – v okně pro vytvoření nové zprávy jednoduše v menu najdeme příslušnou položku. Pokud máme nainstalováno víc certifikátů, vybereme ten, který je druhou stranou považován za důvěryhodnější. Adresát e-mailu pak má možnost



zkontrolovat si důvěryhodnost odesílatele – postup pro Thunderbird najdeme vlevo: klepneme na ikonu obálky, objeví se okno s informací o certifikátu.

V závislosti na tom, jakou certifikační autoritu jsme ve skutečnosti zvolili (nemusí to být jen Comodo), je taky dobré si pohlídat, aby ve správě certifikátů na záložce Autority byla tato CA uvedena, a ve vlastnostech certifikátu aby bylo uvedeno, že může být používán pro podpis e-mailu.

Výše uvedený postup zajistí integritu a nepopiratelnost (co se týče adresy – nikoliv co se týče osoby). Ale jak to udělat, když chceme zajistit i důvěrnost, tedy zprávu zašifrovat?

Postup (Šifrování e-mailu)

Abychom mohli odesílanou zprávu zašifrovat, potřebujeme především veřejný klíč (resp. certifikát) toho, komu chceme šifrovaný e-mail poslat. Pokud nám dotyčný posílal digitálně podepsaný

e-mail, pak velmi pravděpodobně už jeho certifikát máme k dispozici – tedy exportujeme z onoho podepsaného e-mailu a importujeme do svého účtu (ve správě certifikátů na záložce *Lidé*).

Pokud tedy chceme zprávu šifrovat a máme k dispozici certifikát příjemce, zvolíme v menu *Zabezpečení – Zašifrovat zprávu*, podobně jako u podepisování.

Další informace:

- http://www.dfens-cz.com/view.php?cisloclanku=2012020508
- https://www.souki.cz/jak-na-digitalne-podepisovane-emaily-behem-dvou-minut
- https://www.youtube.com/watch?v=ZzWn5cq0Ky8
- https://www.sslshopper.com/email-certificates-smime-certificates.html

5.5.2 PGP, GPG

Alternativou je použití PGP/GPG. Něco jsme se o GPG dozvěděli na straně 133 (použití pro šifrování souborů a oddílů disku), nyní trochu podrobněji.

Program PGP (Pretty Good Privacy – „dost dobré soukromí“) vznikl roku 1991 a jeho různé verze byly pod názvem OpenPGP postupně standardizovány, nejnověji jde o RFC 4880, dále je důležitý RFC 3156 (MIME Security with OpenPGP). Jedná se o volně dostupný nástroj pro šifrování a digitální podepisování (e-mailů, souborů apod.), případně ověřování identity při komunikaci.

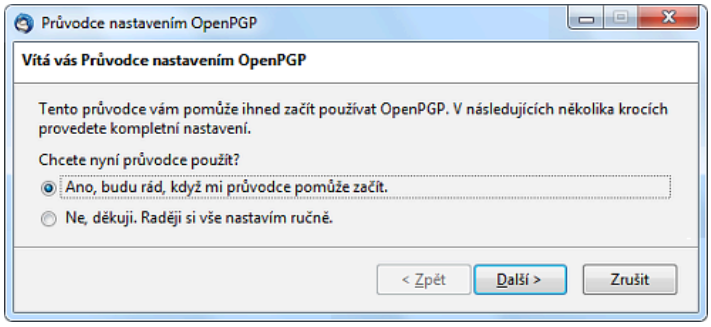
V současné době existuje kromě původního PGP i několik dalších implementací standardu OpenPGP, z nichž nejznámější je *GNU Privacy Guard* (GnuPG, GPG) vytvořený v rámci projektu GNU, který je s PGP kompatibilní.

PGP/GPG je podporován většinou e-mailových klientů, dokonce ho lze použít i v Outlooku. K jeho použití potřebujeme mít nainstalovanou některou implementaci OpenPGP (například GPG) a dále v e-mailovém klientovi příslušný doplněk.

Postup (Zprovoznění a používání GPG v klientovi Mozilla Thunderbird)

Instalaci GPG můžeme provést ve Windows například pomocí *GPG4Win* (zahrnuje v sobě jak samotný šifrovací nástroj, tak i správce certifikátů a pluginy do některých aplikací), v Linuxu už bývá nainstalován.

V Thunderbirdu dále přidáme doplněk *Enigmail* (naprosto stejně jako jakýkoliv jiný doplněk). Po restartu programu Thunderbird se v liště nástrojů nebo v menu objeví položka *OpenPGP*, po jejímž rozbalení najdeme položku pro spuštění průvodce (obrázek vpravo), případně přes *Preferences*.



Průvodce nám pomůže s vygenerováním páru klíčů pro asymetrickou kryptografii, jejich zabezpečením heslem, vytvořením revokačního certifikátu (ten v budoucnu použijeme v případě, že

náš certifikát bude kompromitován a my budeme potřebovat bezpečně o tom informovat všechny komunikační partnery.

Pokud chceme poslat *digitálně podepsanou zprávu*, v okně pro vytvoření nové zprávy klepneme v menu na položku *OpenPGP* a vybereme *Podepsat zprávu*. Případně se dá nastavit, aby byly automaticky podepisovány všechny odesílané zprávy.

Pokud chceme, aby nám mohli ostatní posílat *šifrované zprávy*, musíme takovému odesílateli poslat svůj veřejný klíč. To provedeme tak, že v menu okna pro vytvoření nové zprávy zvolíme *OpenPGP*, a pak *Připojit můj vlastní veřejný klíč* (a taky *Podepsat zprávu*). Pokud druhá strana používá klienta „rozumícího“ PGP, pak by mělo být jednoduché v tom klientovi importovat veřejný klíč z takto odeslaného e-mailu.

Další informace:

- https://tools.ietf.org/html/rfc4880
- https://tools.ietf.org/html/rfc3156
- https://support.mozilla.org/cs/kb/sifrovani-digitalni-podepisovani-zprav
- http://www.root.cz/serialy/gnupg-komunikace-po-internetu-bezpecne/
- http://superuser.com/questions/274169/how-does-s-mime-differ-from-pgp-gpg-for-the-purpose-of-signing-and-or-encryptin
- http://security.stackexchange.com/questions/7874/how-does-pgp-differ-from-s-mime

Literatura

[1] Doseděl, Tomáš. *Počítačová bezpečnost a ochrana dat.* Vyd. 1. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.

[2] Gutmann, Peter. *X.509 Style Guide* [online]. University of Auckland, New Zealand, 2000 [cit. 2015-11-19]. Dostupné z: https://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[3] Hester, Matthew – Chris Henley. *Microsoft Windows Server 2012 administration instant reference.* Indiana, Ind: J. Wiley, 2013. ISBN 11-187-5381-X. Část knihy dostupná na: https://books.google.cz/books?id=RCnRnochyREC&printsec=frontcover

[4] *Magazín o SSL certifikátech a bezpečnosti* [online]. Dostupné na: http://www.blog.sslmarket.cz/

[5] Pinkava, Jaroslav. Moderní kryptografické algoritmy pro elektronický podpis [online]. *Seminář ČAČK*, 2000 [cit. 2015-11-18].
Dostupné z: http://crypto-world.info/pinkava/konference/cack.pdf

[6] Předpis č. 227/2000 Sb.: Zákon o elektronickém podpisu [online]. *Zákony pro lidi.cz.* 2000 [cit. 2015-11-19]. Dostupné z: http://www.zakonyprolidi.cz/cs/2000-227

[7] RFC 5246. *The Transport Layer Security (TLS) Protocol Version 1.2.* RSA Laboratories, 2008. Dostupné také z: https://tools.ietf.org/html/rfc5246

Kapitola 6

Bezpečnost síťové komunikace

V této kapitole budeme předpokládat, že čtenář již má určité základy z oblasti počítačových sítí. Podíváme se na některé úlohy, které souvisejí právě s bezpečností v oblasti počítačových sítí.

6.1 E-mail

6.1.1 Cesta e-mailu

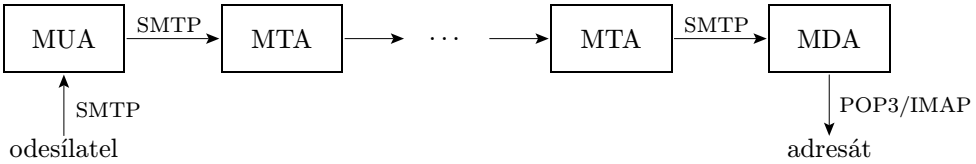
Z hlediska práce s e-maily jsou důležité tyto síťové protokoly:

- *SMTP* (Simple Mail Transfer Protocol) – potřebujeme, když e-mail odesíláme,
- *POP3*, *IMAP* – některý z nich používáme, když přistupujeme do své e-mailové schránky (tj. například e-mail přijímáme).

E-mail se po odeslání nemusí nutně dostat přímo do schránky adresáta, většinou jeho cesta vede přes různé servery:

- *MUA* (Mail User Agent) je SMTP klient (stroj/program, ze kterého odesíláme e-mail), je tedy první na řadě na celé cestě e-mailu,
- *MTA* (Mail Transfer Agent) je SMTP server, který dokáže přijmout e-mail, zkontrolovat doménu, do které směřuje a podle potřeby ho poslat na cestě dál,
- *MDA* (Mail Delivery Agent) je SMTP server spravující schránku adresáta e-mailu, tedy cílový MTA.

Protokol SMTP umožňuje postupně *poskládat* celý e-mail, tedy poskytuje možnost určit, co má být v jednotlivých položkách záhlaví a co má být konkrétně posláno. Když odesíláme e-mail, komunikujeme právě podle protokolu SMTP s „prvním MTA“ na cestě. Ten po zkompletování celé zprávy tuto zprávu odesílá dalšímu MTA na cestě, atd.



Obrázek 6.1: Cesta e-mailu

6.1.2 Z čeho se e-mail skládá

Struktura e-mailu je následující:

- *záhlaví* (hlavička) obsahuje informace o odesílateli, příjemci, předmět e-mailu, v čem byl e-mail vytvořen, případné informace od antiviru či antispamu a další metainformace,
- *tělo zprávy* se skládá z posloupnosti balíků odesílaných informací (ve formě textu, HTML, odesílaného souboru apod.), minimálně jednoho.

Přílohy můžeme počítat do těla zprávy, protože je to vlastně jeden z typů posílaného obsahu.

Co se samotného obsahu týče, před každým blokem dat určitého typu je záhlaví tohoto bloku, ve kterém najdeme informaci o typu obsahu a podle potřeby i další informace (například kódování textu, u přílohy název souboru, apod.). Typ obsahu je určován pomocí *MIME* (Multipurpose Internet Mail Extensions) a je standardizován v dokumentech RFC RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 a RFC 2049.

V záhlaví je stanovena verze MIME (většinou 1.0), dále v záhlaví bloku dat najdeme typ těchto dat, například:

- text/plain – čistý text,
- text/HTML – text v HTML formátu jako na webové stránce,
- image/jpeg, image/gif, image/png, image/tiff, atd. – obrázek,
- video/mpeg, video/h264 audio/wav, audio/mp4, apod. – video a audio soubor,
- application/pdf, application/msword, application/zip, atd. – binární soubor v daném formátu,
- application/octet-stream – binární soubor bez udání typu, ten si určí klient adresáta podle přípony posílaného souboru,
- zadání multibloku – e-mail obsahuje více datových bloků různého typu; rozlišujeme
  - multipart/mixed – jednotlivé části obsahují různé informace, ale je potřeba je uspořádat do určité posloupnosti,
  - multipart/alternative – jednotlivé části obsahují vpodstatě tytéž informace, ale v různých formátech, tedy různé verze téhož,
  - multipart/x-zip – sada ZIP souborů,
  - multipart/signed – najdeme ji v digitálně podepsaných e-mailech (první blok je podepsaná zpráva, druhý obsahuje informace potřebné ke kontrole podpisu),
  - multipart/encrypted – používá se v šifrovaných e-mailech (je zvlášť blok s informacemi potřebnými k dešifrování a zvlášť blok s šifrovanými daty, obvykle typu application/pgp-encrypted nebo application/pkcd7-mime),

  v záhlaví multibloku je stanovena oddělující sekvence pro jednotlivé bloky (aby bylo zřejmé, kde jeden blok končí a kde začíná jiný), tato sekvence musí být zvolena tak, aby se „nepomíchala“ s obsahem bloků.

Pro každý typ dat jsou pak v záhlaví dodatečné informace. Například pro text jde o určení znakové sady (například utf-8) a zvolené kódování (kolik bitů zabírá jeden znak, 7bit, 8bit), u binární přílohy název posílaného souboru, apod.

Poznámka:

Pokud se jedná o zprávu s obsahem multipart/alternetive, kde jedna z částí je text/plain, mnohé antispamy procházejí právě pouze tuto část a u zbytku předpokládají, že je „vpodstatě stejný“. Toho zneužívají spammeři – do textové části dají něco nevinného a text obsahující „podezřelá“ klíčová slova umístí až do text/HTML, kterou antispam neprojde.

Další informace:

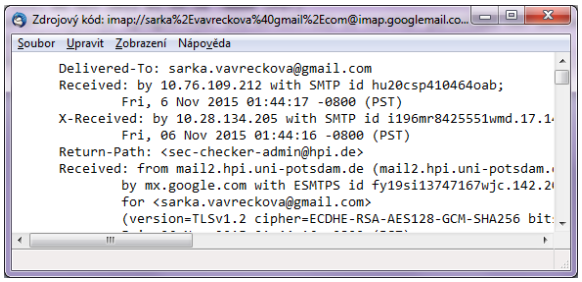
- http://ccm.net/contents/120-mime-multipurpose-internet-mail-extensions (jde o seriál s více díly)
- https://en.wikipedia.org/wiki/MIME

Speciální znaky: http://homoglyphs.net/ (zaškrtnout display Unicode position of characters)

6.1.3 Analýza e-mailu

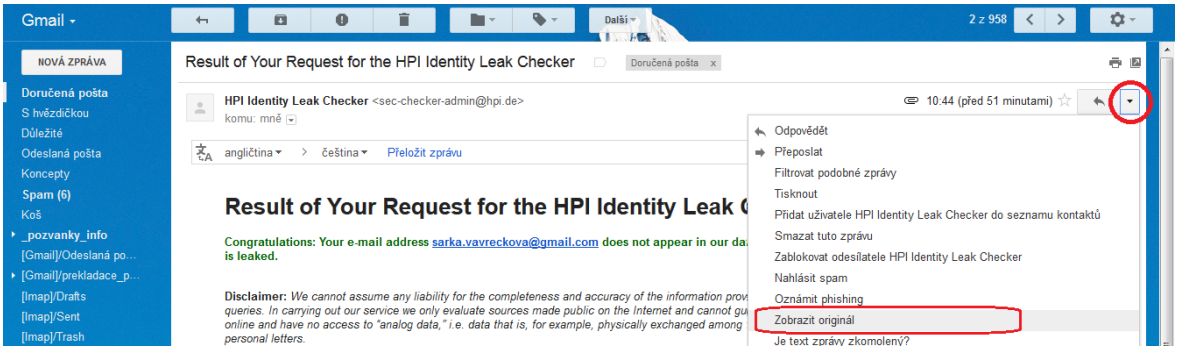
E-mail není nic jiného než souhrn dat a metadat posílaných ve formě paketu počítačovou sítí podle určitých síťových protokolů. To, co vidí odesílatel nebo adresát, je jen vizualizace těchto dat provedená buď e-mailovým klientem (Outlook, Thunderbird apod.) nebo rozhraním webové aplikace. Ovšem informatik by měl být schopen dostat se i k něčemu dalšímu.

Postup (Jak se dostat ke zdrojovému kódu e-mailu)



Pokud používáme e-mailového klienta, je obvykle někde v menu položka *Zobrazit zdrojový kód zprávy* nebo podobná. Například v aplikaci Mozilla Thunderbird ji najdeme v menu *Zobrazení*. Tedy nejdřív se přesuneme na ten e-mail, jehož zdrojový kód nás zajímá (nemusíme zprávu otevírat) a pak v menu najdeme dotyčnou volbu. Můžeme také použít klávesovou zkratku Ctrl+U. Otevře se okno se zdrojem, jak vidíme na obrázku vpravo.

Jestliže nevyužíváme klienta a e-maily čteme ve webovém rozhraní, je situace o něco horší, protože jsem odkázáni na možnosti poskytnuté programátory tohoto rozhraní. Nicméně – u mnoha poskytovatelů této služby tu možnost máme. Například u GMailu je po otevření zprávy vpravo



Obrázek 6.2: Cesta ke zdrojovému kódu e-mailu ve webovém rozhraní GMailu

dostupné rozbalovací menu s volbou *Zobrazit originál* (viz obrázek 6.2), která nás zavede ke zdrojovému kódu otevřené zprávy. Ovšem nejdřív je nutné zprávu otevřít, což například v případě podezření na infikovaný e-mail není ideální.

Příklad

Dále postupně rozpitváme zdrojový kód e-mailu na jeho jednotlivé součásti. Abychom si dokázali udělat základní představu, vezměme si následující úsek e-mailu (jeho zdrojový kód):

```
Message-ID: <545B3395.5000105@fpf.slu.cz>
Date: Thu, 06 Nov 2014 09:38:45 +0100
From: =?ISO-8859-2?Q?=A9=E1rka_Vavre=E8kov=E1?= <sarka.vavreckova@fpf.slu.cz>
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko/20120428 Thunderbird/12.0.1
MIME-Version: 1.0
To: 'jindra' <jindra.plankova@fpf.slu.cz>
Subject: Ucebna na pondeli
Content-Type: text/plain; charset=ISO-8859-2; format=flowed
Content-Transfer-Encoding: 8bit

Ahoj, xxxxx
```

Na tento e-mail byla zaslána následující odpověď (mírně promazaná):

```
Return-Path: <jindra.plankova@fpf.slu.cz>
Delivered-To: Vavreckovas@maia.slu.cz
Received: from smtp.slu.cz by maia.slu.cz (Postfix)
    for <sarka.vavreckova@fpf.slu.cz>; Thu, 6 Nov 2014 10:35:30 +0100 (CET)
Received: from localhost (localhost [127.0.0.1]) by smtp.slu.cz (Postfix)
    for <sarka.vavreckova@fpf.slu.cz>; Thu, 6 Nov 2014 10:35:30 +0100 (CET)
X-Virus-Scanned: amavisd-new at smtp.slu.cz                    Začíná "X": přidává mail systém
Received: from smtp.slu.cz ([127.0.0.1])                      (klient nebo některý mail server)
    by localhost (smtp.slu.cz [127.0.0.1]) (amavisd-new, port )
    for <sarka.vavreckova@fpf.slu.cz>; Thu, 6 Nov 2014 10:35:28 +0100 (CET)
Received: from uib2221 (unknown [10.6.13.183]) by smtp.slu.cz (Postfix)
    for <sarka.vavreckova@fpf.slu.cz>; Thu, 6 Nov 2014 10:35:28 +0100 (CET)

From: "Jindra Plankova«jindra.plankova@fpf.slu.cz>
To: =?iso-8859-2?B?J6nhcmthIFZhdnJl6GtvduEn?= <sarka.vavreckova@fpf.slu.cz>
References: <545B3395.5000105@fpf.slu.cz>
In-Reply-To: <545B3395.5000105@fpf.slu.cz>
Subject: RE: Ucebna na pondeli
Date: Thu, 6 Nov 2014 10:35:27 +0100
Message-ID: <004901cff9a4$ff8060f0$fe8122d0$@plankova@fpf.slu.cz>
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable
X-Mailer: Microsoft Office Outlook 12.0
Thread-Index: Ac/5nRS6CImpKcHnSdKyIkxRxswswAAB8ALA
Content-Language: cs
X-Antivirus: avast! (VPS 141105-1, 05.11.2014), Outbound message
X-Antivirus-Status: Clean

Ahoj, xxxxx
```

Obě zprávy byly zachyceny na tomtéž zařízení (které odeslalo první zprávu a obdrželo druhou), proto se druhá zdá „upovídanější“ – záhlaví cestou postupně kyne.

Projdeme si postupně nejdůležitější součásti záhlaví. Některé jsou generovány už při vytváření e-mailu, další se přidávají na cestě do schránky adresáta. Přidává se vždy na začátek záhlaví, tedy to, co vidíme na začátku záhlaví, bylo přidáno jako poslední.

Message-ID, In-Reply-To. Tento řetězec slouží jako identifikátor, přes který se párují dvojice dotaz-odpověď. Pokud je určitý e-mail odpovědí na jiný e-mail, uloží do pole `In-Reply-To` ta hodnota, která byla v původním e-mailu v poli `Message-ID`. V příkladu výše jsme viděli v poli `Message-ID` určitý řetězec, a tentýž řetězec byl v odpovědi v poli `In-Reply-To` (zeleně zatrženo).

From, To, Delivered-To. `From` a `To` jsou adresa odesílatele a příjemce tak, jak je zadal odesílatel (u spamu může být adresa odesílatele podvržená). SMTP server, který přijme zprávu k odeslání a neověří zadavatele nebo umožňuje odeslat zprávu i tomu, kdo není registrovaným uživatelem, se nazývá „open relay“, a takové servery (pokud vysloveně nejsou používány pro komunikaci jen v rámci organizace) obvykle bývají na black listech MTA serverů.

`Delivered-To` (pokud je použito) pomáhá zamezit případným smyčkám mezi SMTP servery při přeposílání.

Pokud je v poli `To` řetězec „undisclosed recipients:“, znamená to, že zpráva byla poslána více různým osobám, které o sobě navzájem nemají vědět (skrytí příjemci). První MTA na cestě za dvojtečkou najde seznam adres a záhlaví upraví následovně:

- vytvoří tolik kopií e-mailu, kolik je skrytých adresátů,
- v příslušné kopii dá do pole `Received...for` jednu ze „skrytých adres“,
- seznam za dvojtečkou nechá prázdný.

Takže příjemce se může dozvědět, že e-mail byl doručen i někomu jinému, ale nedostane dotyčný seznam (ten nedostane vlastně ani druhý MTA na cestě).

Received. Těchto částí záhlaví může být více za sebou (nebo žádné). Jsou přidávány na cestě e-mailu do schránky adresáta – e-mail postupně přechází přes MTA servery, které postupně přeposílají e-mail k cíli. Může být užitečné, když chceme sledovat, přes které servery se komunikuje. Může se stát, že mezi dvěma adresami jsou v různých směrech tyto údaje naprosto odlišné (často za to mohou e-mailoví klienti).

cc, bcc. Adresy, na které je odeslána kopie e-mailu (tj. až tak nutně se od nich neočekává reakce, když nejsou v poli `To`). První typ pole označuje adresáty kopie, kteří jsou ostatním adresátům viditelní, druhý typ pole obsahuje skryté adresáty kopie.

Return-Path. Adresa, na kterou má směřovat odpověď. Pokud je jiná než adresa odesílatele, může to být v pořádku (odesílatel prostě pro další komunikaci odkazuje na někoho jiného), nebo může jít o trik škodlivého softwaru ukrývající skutečnou adresu, ze které byl e-mail odeslán (pak bude zřejmě adresa v poli `From` podvržená).

Poznámka:

U podezřelého e-mailu je dobré projít všechny části záhlaví obsahující adresy a tyto adresy srovnat. Není nutné, aby byly stejné, ale měla by tu být určitá konzistence. Může se stát, že odesílatel nebo příjemce má pro svou adresu několik aliasů, pak na začátku nebo konci cesty bude patrné

přesměrování, nicméně tyto aliasy (pokud je vše v pořádku) by pravděpodobně byly v rámci téže domény. Také adresy serverů MTA by měly být „uvěřitelné“.

Subject, Date. Předmět zprávy a datum odeslání jsou nepovinné položky. K času se přidává i informace o časovém posunu vzhledem k nultému poledníku (u nás je to jedna hodina, proto je na konci řetězec `+0100`).

Content-type, Content-Transfer-Encoding, atd. Tyto položky souvisí s typem posílaných dat. Položky typu Content byly diskutovány v předchozí sekci, jsou obvykle určeny některým MIME typem.

User-Agent, Mailer, X-Mailer. Sem se obvykle uloží identifikace klientského softwaru, který zajišťuje sestavení e-mailu (v předchozím příkladu to je Mozilla Thunderbird, v druhém MS Outlook).

Položky začínající písmenem X. Jde o položky, které se přímo nevztahují ke komunikaci zajišťované protokolem SMTP, například to může být informace od antiviru či antispamu, nebo položky navíc přidávané některým MTA či jiným serverem.

Postup (Služby na analýzu záhlaví e-mailu)

Na webu existují také služby, které mohou pomoci analyzovat záhlaví e-mailu. Například:

- *Message Header Analyzer* (https://toolbox.googleapps.com/apps/messageheader/) provádí jednoduchou rychlou analýzu záhlaví,
- *Email Header Analyzer* (http://mxtoolbox.com/EmailHeaders.aspx) provádí podrobnější analýzu záhlaví.

V obou případech je třeba do příslušného pole zkopírovat zdrojový kód e-mailu nebo alespoň jeho záhlaví.

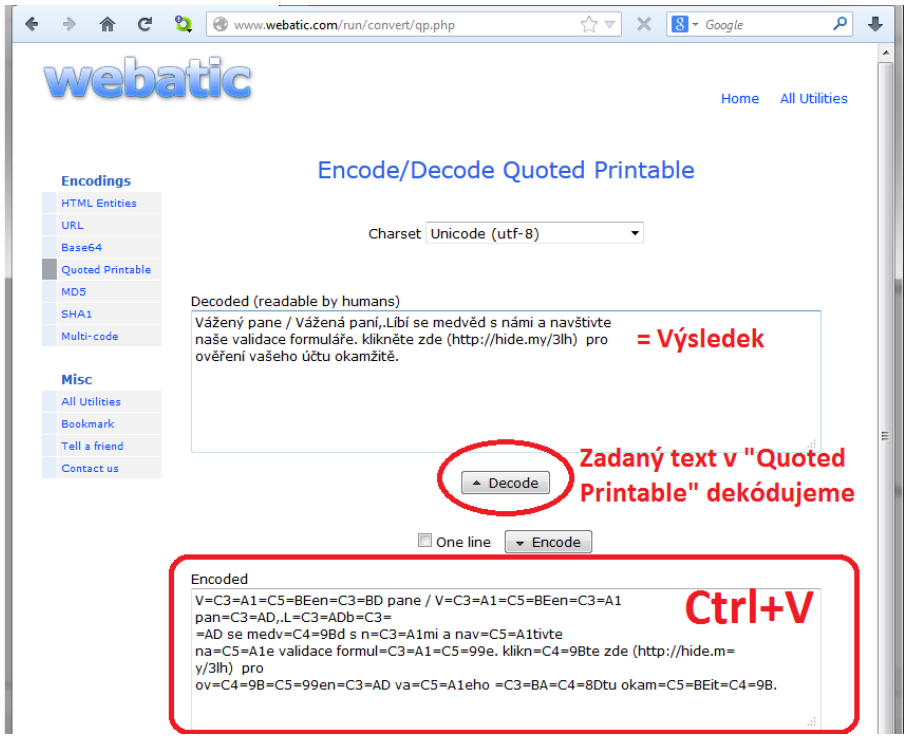
Dalším možným problémem je kódování zprávy. Pokud si e-mail čteme v klientovi nebo webovém rozhraní, obvykle není problém, ale ve zdrojovém kódu se v případě češtiny můžeme v bloku text/plain nebo text/html setkat s kódováním *Quoted Printable*.

Účelem použití tohoto kódování je snadnější převod textu na sekvenci 7bitových znaků (do kódování 7bit). Místo znaků, jejichž ASCII kód je vyšší než 127 (tj. nevejde se do 7 bitů), se ve zprávě objeví speciální kódy začínající symbolem rovnítka, za kterým je dvouciferné hexadecimální číslo. Například místo znaku „á“ tam najdeme „=C3“ (hexadecimální číslo C3 je dekadicky 195). Řetězec „Vážený pane“ bude zakódován do posloupnosti `V=C3=A1=C5=BEen=C3=BD pane`.

Postup (Překlad z kódování „Quoted Printable“)

Na stránce http://www.webatic.com/run/convert/qp.php je možno provést překlad z Quoted Printable do některého čitelnějšího kódování. Stránku vidíme na obrázku níže.

Text k dekódování je třeba vložit do druhého textového pole (ne do prvního, tam se objeví výsledek). Nad prvním polem zvolíme v rozbalovacím poli cílový kód (většinou bude vyhovovat utf-8). Klepneme na *Decode* a zobrazí se výsledek. Kdybychom chtěli provést opačný překlad, vložíme text do prvního textového pole a klepneme na *Encode*.



Obrázek 6.3: Dekódování textu ve formě Quoted Printable

Další informace:

- https://www.cs.tut.fi/~jkorpela/headers.html
- http://www.computerhope.com/issues/ch000918.htm
- https://www.uic.edu/depts/accc/newsletter/adn29/headers.html
- http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1048&context=adf
- http://www.computerhope.com/issues/ch000918.htm

Poznámka:

Další ukázky e-mailů včetně spamu jsou v samostatném souboru na stránce tohoto předmětu.

6.1.4 Konverzace s SMTP serverem

Co vše se děje, když chceme odeslat e-mail?

- klient se připojí k SMTP serveru na portu 25 (třeba přes telnet nebo ssh),
- sdělí SMTP serveru, že chce poslat e-mail, dále zadá svou adresu a adresu příjemce,
- zadá případně další části záhlaví, dále text e-mailu.

Následuje ukázka jednoduché komunikace s SMTP serverem (na kterém běží Sendmail), kdy chceme odeslat e-mail obsahující předmět a tělo zprávy ve formě čistého textu.

Příklad

Navážeme spojení se serverem (zde to je pomocí telnetu), a to na portu 25, čímž se napojíme na SMTP. Červeně jsou zbarveny odpovědi serveru.

- `telnet adresa.smtp.serveru.com 25`
  `Trying ip.adresa.serveru.`
  `Escape character is '^]'.`
- `220 adresa.smtp.serveru.com ESMTP Sendmail 8.10.0/8.10.0 ready; datum čas`
- `helo moje.identifikace`
  `250 adresa.smtp.serveru Hello moje.identifikace [moje.ip.adresa], pleased to meet you`
- `mail from: moje@adresa`
  `250 2.1.0 moje@adresa... Sender ok`
- `rcpt to: adresa@prijemce`
  `250 2.1.5 adresa@prijemce... Recipient ok`
- `data`
  `354 Enter mail, end with "."on a line by itself`
- ```
  From: Moje Jmeno <moje@adresa>
  To: Jmeno Adresata <adresa@prijemce>
  Subject: Predmet e-mailu

  Tady napisu telo zpravy, prazdny radek pred textem zpravy je nutny.
  xxxxxx

  .
  ```
  `250 Message accepted for delivery`
- `quit`
  `221 2.0.0 staff.uiuc.edu closing connection`

Jak vidíme, pro odeslání e-mailu vlastně ani není třeba žádný klient nebo webové rozhraní, pokud ovšem dokážeme přímo komunikovat s SMTP serverem. Taky záleží na tom, do jaké míry server ověřuje odesílatele (resp. uživatele, od kterého přijímá zprávu k odeslání).

Další informace:

- http://www.earthinfo.org/example-smtp-conversation/
- http://www.anta.net/misc/telnet-troubleshooting/smtp.shtml
- https://www.cs.cf.ac.uk/Dave/PERL/node175.html

6.2 Údaje o doméně a jejím vlastníkovi

Někdy potřebujeme u určité domény zjistit jejího vlastníka, například v těchto případech:
  ```

- když chceme registrovat vlastní doménu a ověřujeme, jestli už není registrovaná,
- když z určité adresy přichází spam či malware a chceme zjistit odpovědnost, upozornit.

Registrátoři domén evidují vlastníky svých domén v databázi, a my potřebujeme mechanismus, který nám dovolí do takové databáze nahlížet. K tomuto účelu slouží mechanismus WHOIS.

Postup (Zjišťování informací o doméně)

Kromě jiných nástrojů máme k dispozici webové aplikace na Internetu. Například:

- zeptáme se Googlu – do vyhledávacího pole zadáme například `whois seznam.cz`,
- http://www.nic.cz/whois/ je rozhraní k databázi českého registrátora domén CZ.NIC,
- http://whois.com/whois/ nebo http://whois.net/ jsou rozhraní ke všem whois databázím, tedy funguje pro jakoukoliv doménu,
- https://apps.db.ripe.net/search/query.html nám zprostředkuje přístup do databáze RIPE, tedy pro domény v rámci Evropy (moc nefunguje).

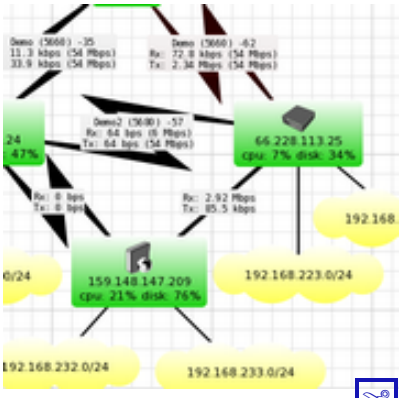
V databázi se dozvíme, kdo je držitelem domény, kdy byla registrována, přes kterého lokálního registrátora, dokdy registrace platí, adresy DNS serverů a kontakty na správce domény.

6.3 Struktura sítě

Každý administrátor by měl mít předhled o tom, co všechno se v jeho síti nachází. Existují nástroje, které mu tuto úlohu mohou zjednodušit.

Postup (Dynamické zjišťování a zobrazování struktury sítě)

Volně šiřitelný nástroj *The Dude* od MikroTiku zobrazuje seznam a mapu všech zařízení v síti, která jsou dostupná, monitoruje běžící služby. U některých zařízení poskytuje možnost vzdálené správy. Monitoring je řešen automatickým procházením adres v zadaných subnetech.



Ukázka takové vytvořené struktury je na obrázku vpravo.

The Dude existuje ve variantě pro Windows, přes Wine i pro Linux a přes Darwin i v MacOS X, je dostupný samozřejmě i v RouterOS (ten je právě od MikroTiku).

Dalším podobným nástrojem je například *NetToolset* nebo *Connecter*.

Další informace:

- http://www.mikrotik.com/thedude
- http://www.zbynek.adamh.cz/cs/061762/Pocitac/Clanky/Connecter-jednoduchy-prehled-o-pocitacich-na-siti/

6.4 Vyhledávání bezpečnostních problémů v síti

Je celkem dobře známo, že mnohé společnosti mají svůj *Bug Bounty Program*, v rámci kterého může kdokoliv nahlásit nalezenou zranitelnost v produktu dané společnosti a za to získat odměnu. Bug Bounty Program mají například Google, Facebook, Apache, Intel, Uber,...

Bug Bounty Program – seznam zapojených organizací: https://www.bugcrowd.com/bug-bounty-list/ (jednotlivé organizace pak na svých webech uvádějí informace o tom, jak konkrétně postupovat při nahlašování chyby).

6.4.1 Databáze zranitelností

Informace o zranitelnostech jsou vedeny ve *formátu CVE* (Common Vulnerabilities and Exposures). Vytvoření formátu CVE bylo velkým přínosem, protože do té doby každá organizace popisovala a ukládala nalezené zranitelnosti v jiném formátu – jeden formát na druhý se překlápěl jen problematicky, záznamy se těžko srovnávaly, byla vyšší pravděpodobnost, že tatáž zranitelnost bude zaznamenána ve dvou různých podobách a nepozná se, že jde o totéž, navíc se těžko programovaly nástroje, které by s tolika formáty dokázaly pracovat.

Formát CVE je založen ja JSON (JavaScript Object Notation), takže vlastně jde o textový záznam s položkami typu "vlastnost": "hodnota" oddělenými čárkami, případně strukturovanými do bloků. Informace o zranitelnosti se dá poslat přímo ve formátu JSON, ale může být akceptován například csv nebo čistý text.

Každý záznam má jednoznačný identifikátor ve formátu CVE-rok-číslo, takže snadno zjistíme, ve kterém roce byla zranitelnost objevena. Součástí záznamu je pak popis chyby a odkazy vedoucí na další informace.

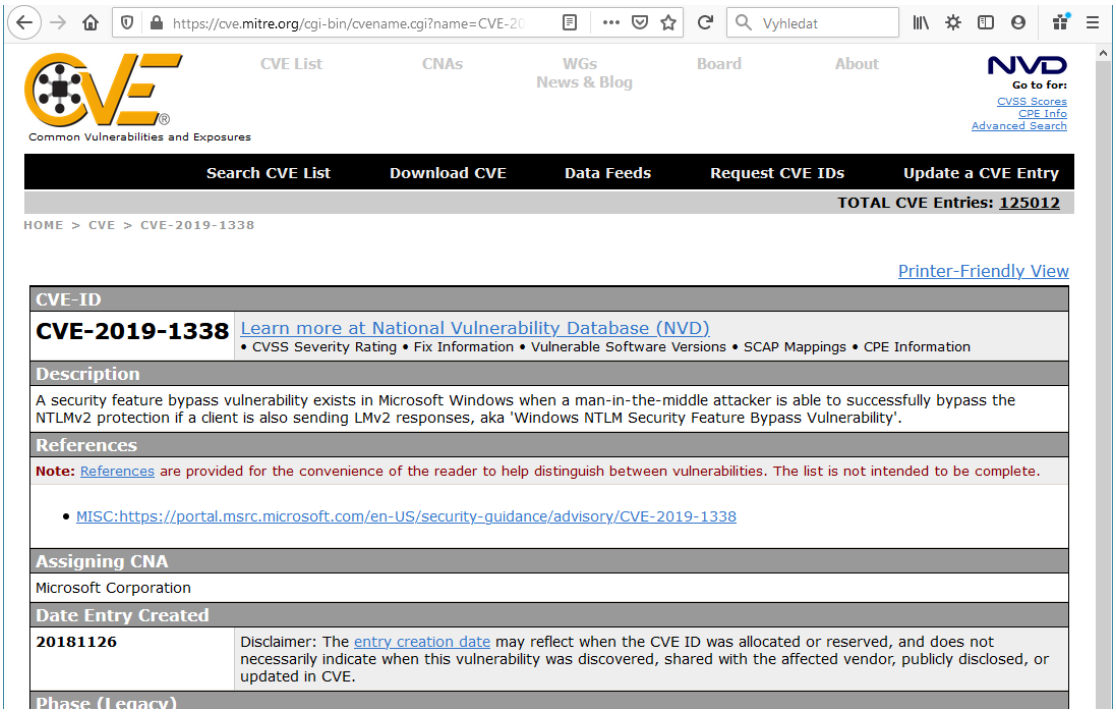
Hlavním správcem databáze CVE záznamů je MITRE Corporation, databáze je dostupná na https://cve.mitre.org. Každé nahlášené zranitelnosti se přiděluje identifikátor (CVE Number), což provádějí *CNA* (CVE Numbering Authorities). CNA jsou nejen společnosti provozující celý systém, ale patří zde také organizace, které mají svůj Bug Bounty Program, a mnohé další, které přímo Bug Bounty neprovozují.

Další informace:

- Prezentace *CVE Submission Process*: https://slideplayer.com/slide/14301166/
- Podrobný popis JSON CVE: https://github.com/CVEProject/automation-working-group/blob/master/cve_json_schema/DRAFT-JSON-file-format-v4.md
- Informace o CNA jsou na https://cve.mitre.org/cve/cna.html, odtud také vede odkaz na aktuální seznam CNA

Na webu https://cve.mitre.org se k databázi CVE dá dostat (na obrázku vidíme odkazy „Search CVE List", „Download CVE"). V části „Download CVE" jsou odkazy trochu problematické, některé vedou na staré soubory, tj. je dobré volit odkazy pro jednotlivé roky.

K položkám databáze CVE vedené organizací MITRE se dostaneme i na https://www.cvedetails.com. Je to jen jiná cesta k téže databázi, ale pro někoho může být zajímavější – nabízí i statistiky a grafy.



Obrázek 6.4: CVE záznam

Na obrázku 6.5 je naznačen přístup k zranitelnostem systému Windows ve verzi 10 (pozor, u produktů se zvlášť vedou zranitelnosti pro různé verze).

V menu vlevo zvolíme „Browse“ – „Products“, dále počáteční písmeno hledaného produktu (zde W), a protože takových produktů je hodně, o řádek výše klikneme na „Operating Systems“, čímž omezíme počet nalezených stránek. Hledaný produkt je na stránce 3 (říjen 2019, může se změnit), kde už najdeme Windows 10. Výsledek hledání je na obrázku 6.6. V tabulce se dostaneme k jednotlivým typům zranitelností, pod tabulkou jsou grafy.

Existují i další databáze, obvykle napojené na MITRE, například americká standardizační organizace NIST vede databázi NVD (National Vulnerability Database, https://nvd.nist.gov).

Existují různé typy zranitelností, a CVE záznamy se pomocí těchto typů klasifikují. Také pro typy zranitelností máme formát: *CWE* (Common Weakness Enumeration). Ten má taky na starosti MITRE. Formáty zranitelností jsou ve stromové struktuře (obecnější jsou nadřízené, jejich potomci jsou konkrétnější).

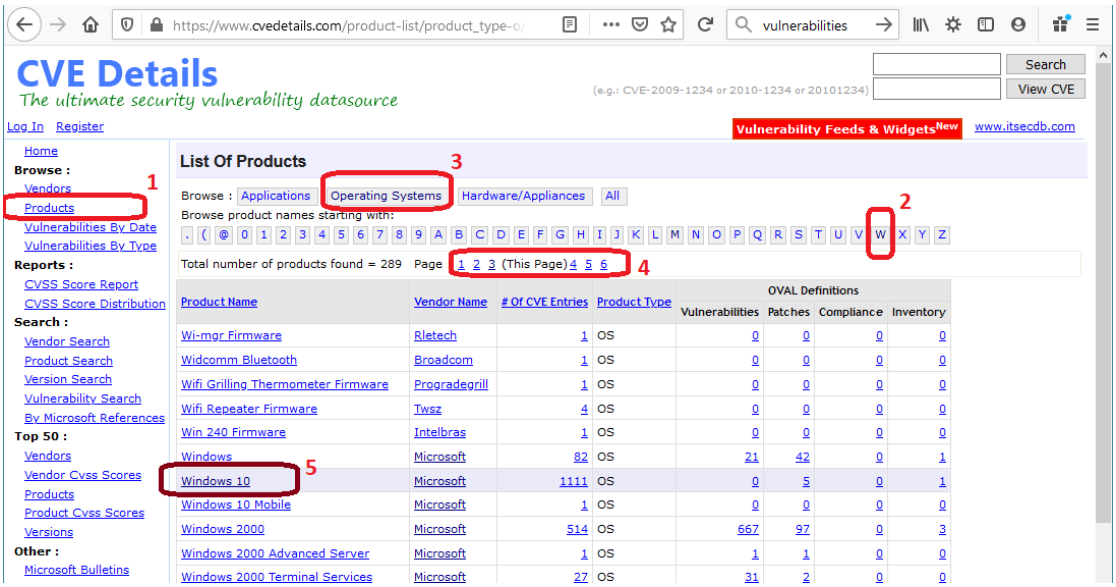
Další informace:

Seznam CWE najdeme například na

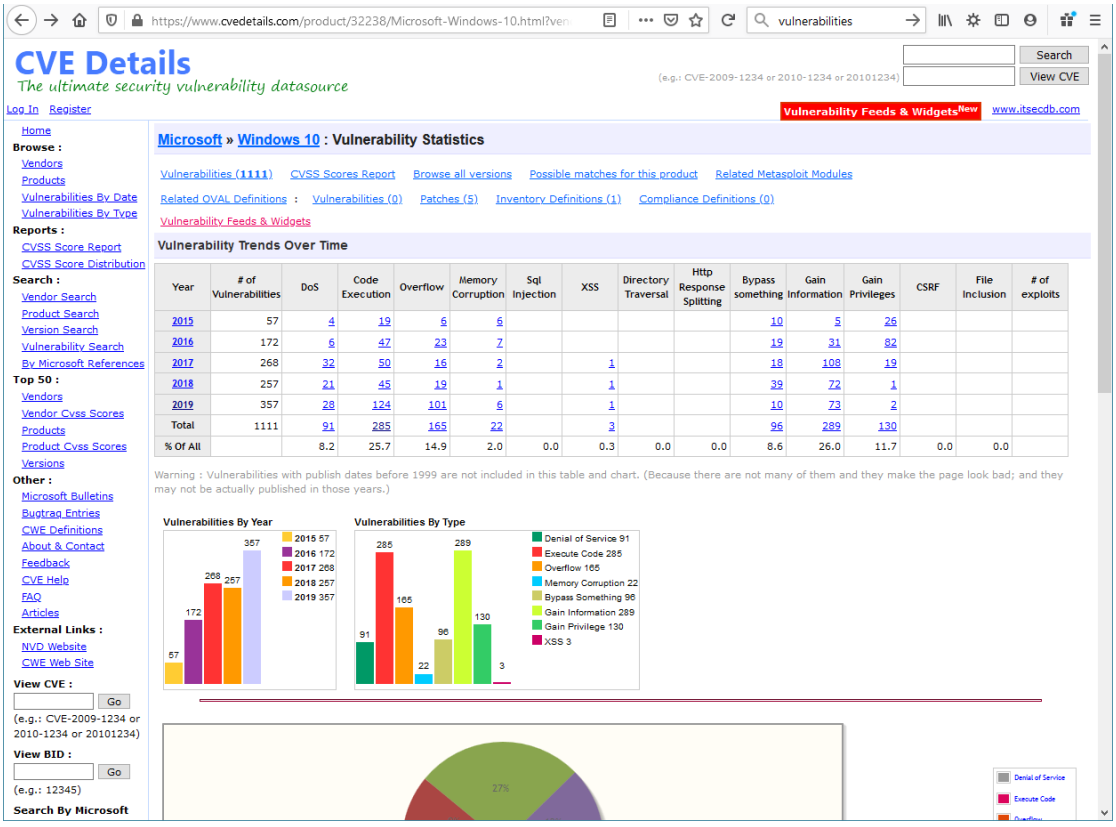
- https://cwe.mitre.org/data/
- https://www.cvedetails.com/cwe-definitions.php

Soubory s různými vizualizacemi CWE včetně barevného vyznačení závažnosti problému jsou na https://cwe.mitre.org/data/pdfs.html.

Příklad



Obrázek 6.5: Přístup k databázi CVE Details



Obrázek 6.6: CVE Details, údaje o Windows 10

Například populární typ zranitelnosti „Cross-site Scripting“ neboli XSS, oficiálně „Improper Neutralization of Input During Web Page Generation“, má číslo CWE-79. Jeho rodičem ve stromě typů zranitelností je CWE-74 s názvem „Improper Neutralization of Special Elements in Output Used by a Downstream Component (’Injection’)“, má několik potomků upřesňujících typ XSS.

Velmi podrobné informace o XSS včetně příkladů najdeme na
http://cwe.mitre.org/data/definitions/79.html.

Úkoly

1. Projděte si různé části webů https://cve.mitre.org a https://www.cvedetails.com. Najděte seznam CNA a zjistěte, jak se dá u vybraných organizací reportovat bezpečnostní problém.
2. Zjistěte podrobnější informace o zranitelnosti CVE-2019-0543.
3. Najděte informace o typu zranitelnosti Cross-site Scripting zmíněné výše a projděte si informace o souvisejících typech.

6.4.2 OVAL

OVAL (Open Vulnerability and Assessment Language, tj. otevřený jazyk pro popis zranitelností a jejich vyhodnocování) je jazyk pro popis zranitelnosti z pohledu napadnutelného systému založený na XML (tj. také strojově čitelný). Zatímco CVE popisuje samotnou zranitelnost, OVAL stanovuje, jakým způsobem je tato zranitelnost nebezpečná a jak systém může být touto zranitelností napaden. Stejně jako CVE a CWE, i OVAL si pod svá křídla vzala společnost MITRE.

Záznamy OVAL jsou používány především různými nástroji pro hledání zranitelností v síti (Nessus, Retina, OpenVAS apod.), OVAL jim poskytuje „informační servis“ v těchto třech hlavních oblastech a šablonách (schématech):

1. sběr informací o konfiguraci testovaných systémů (šablona OVAL System Characteristics Schema),
2. sběr informací o problémech na testovaném systému – testování zranitelností, instalovaných/chybějících aktualizací apod. (šablona OVAL Definition Schema),
3. vyhodnocení a prezentace výsledků (šablona OVAL Results Schema).

První dvě schémata jsou určena pro definici vstupů, třetí definuje výstup.

Příklad

Hlavní schémata jsou kořenem struktury odvozených schémat. Následuje úsek struktury schémat odvozených ze schématu pro druhé hlavní schéma (převzato z webu
http://oval.mitre.org/language/about/structure.html).

```
Conceptual Breakdown of the OVAL Language
---------------------
OVAL Definition Schema
|
|-> Core Schema
      |
      |-> Independent Schema  (family_test, variable_test, xmlfilecontent_test, etc.)
      |
      |-> UNIX Schema  (file_test, process_test, uname_test, etc.)
      |    |
      |    |-> Solaris Schema
```

```
|     |-> HP-UX Schema
|     |-> MacOS Schema
|     |-> Linux Schema  (dpkg_test, rpminfo_test, etc.)
|
|-> Windows Schema  (file_test, wmi_test, etc.)
|
|-> Apache Schema
```

Jak vidíme, Core Schema se člení na několik podřízených, z nichž první obsahuje testy nezávislé na platformě, druhé je pro UNIXy a systémy UNIXového typu (dále se člení podle různých systémů), třetí pro Windows, další pro webový server Apache,...

Testy určené v OVAL Definition Schema mohou být různého druhu, především:

- OVAL Vulnerability Definitions – kompatibilní s CVE a využívající CVE databázi, pro testování zranitelností,
- OVAL Compliance Definitons – pro testování, zda konfigurace systému odpovídá nastaveným bezpečnostním politikám,
- OVAL Inventory Definitons – testuje se, zda určitý software je v systému instalován,
- OVAL Patch Definitions – zjišťuje se, zda jsou instalovány určité bezpečnostní záplaty,
- Miscellaneous – nezařazený zbytek.

Každá entita v daných XML souborech (schématech) má svůj identifikátor – OVAL-ID. Je to řetězec ve tvaru *oval:organizace:typ:hodnota*, který zachycuje „odpovědnou *organizaci*“ ve tvaru obrácené doménové adresy (například `org.mitre.oval` nebo `com.redhat`), dále *typ* určuje, na co má být daná entita použita (definici, objekt, stav, test, proměnnou). Poslední prvek (*hodnota*) je číslo jednoznačné pro předchozí dvojici údajů.

Ne každý může ve svém produktu přistupovat k databázi OVAL a využívat ji k testování. Zájemce musí projít schvalovacím procesem, ve kterém prokáže, že má celý mechanismus správně implementován, a až po odsouhlasení MITRE dostane své OVAL-ID, přístup do databáze a je uveden v seznamu na webu MITRE.

Další informace:

Informace o OVAL včetně seznamu produktů, které mají přístup k informacím v tomto jazyce, je na http://oval.mitre.org/.

Úkol

Projděte si web http://oval.mitre.org/, včetně seznamu produktů (v menu vlevo *Products*).

6.4.3 Nástroje pro skenování sítě

Existuje celá škála nástrojů, které dokážou prohledat místní síť, najít připojená zařízení a prověřit jejich zabezpečení. Tyto nástroje jsou používány oběma stranami – hackery i administrátory sítí.

Tyto nástroje obvykle fungují tak, že zadáme počáteční podmínky (obvykle rozsah IP adres, které mají být proskenovány, případně další upřesnění, přihlašovací údaje pro testovaná zařízení,

apod.), následně nástroj proskenuje síť (například s použitím protokolu ICMP, ale jsou i jiné možnosti, ICMP může být totiž na cílových zařízeních ignorován). Pak se nástroj zaměří na jednotlivá nalezená zařízení a podle zadání je otestuje.

Každý nástroj je v něčem specifický. Zatímco NMap je orientován spíše na hledání bezpečnostních problémů z pohledu hackera, který se dostává do sítě (otevřené porty, zadní vrátka apod.), jiné se primárně orientují na testování konfigurace, bezpečnostních záplat, některé tyto nástroje využívají OVAL (například OpenVAS nebo Retina).

Postup

NMap (Network Mapper) je program pro sledování stavu, služeb a prostředků sítě. Jde o nástroj pracující v textovém režimu, ale existují i jeho grafické nástavby – například známým GUI frontendem pro NMap je Zenmap. Původně byl NMap dostupný jen v UNIXových systémech, dnes je i pro Windows.

Postup (Základní průzkum sítě pomocí nástroje NMap)

Pokud máme NMap nainstalován, můžeme postupovat takto:

- `nmap názevPC`
  - skenování spuštěných služeb (můžeme zadat i název našeho počítače)
- `nmap -sS -O názevPC`
  - (nutná vyšší oprávnění) aktivní skenování portů, zjištění informací o OS

NMap toho umí víc, například lze používat i skripty (s využitím NSE – NMap Scripting Engine), hodně skriptů už je předdefinovaných.

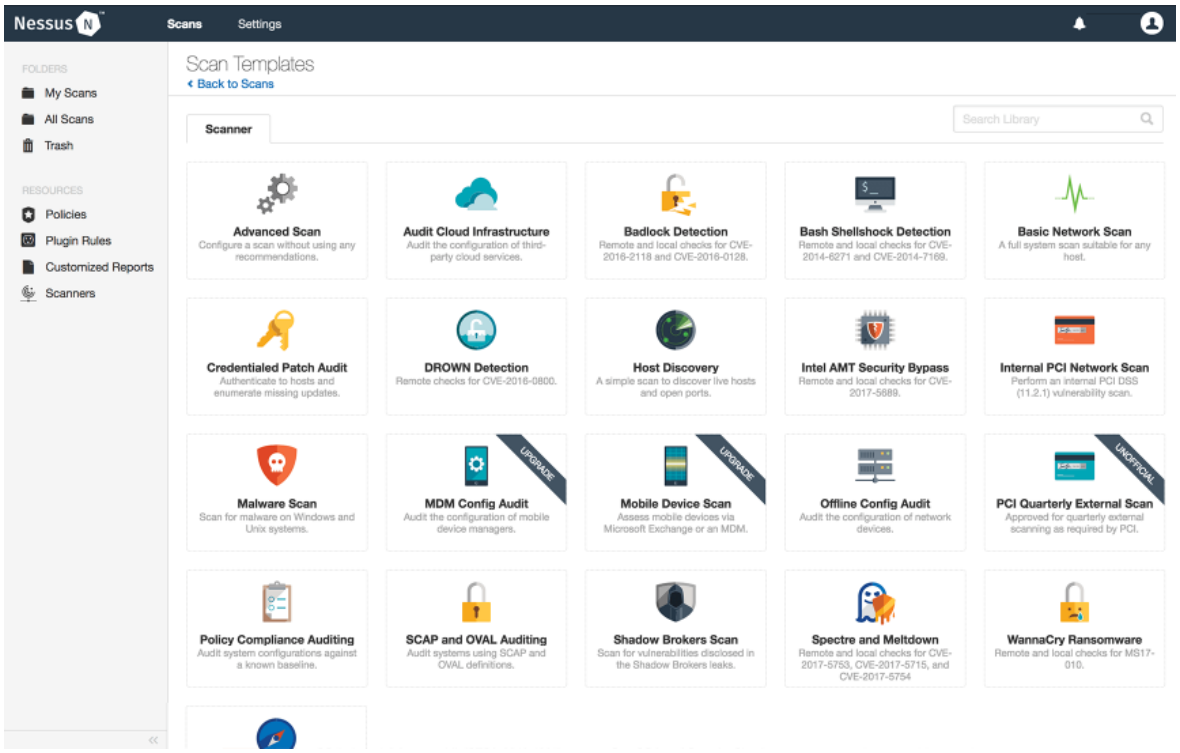
Dalším užitečným nástrojem je například *Nessus* sloužící pro aktivní skenování zranitelností systému. Administrátor si tak může ověřit bezpečnostní stav sítě. Nessus existuje pro různé operační systémy, je k dispozici jak komerční, tak i volná varianta (přičemž jsme omezeni na 16 IP adres). Na obrázku 6.7 je obrazovka s výběrem testů. Test „Advanced Scan“ se dá nejrozsáhleji nakonfigurovat. Všimněte si, že zde existuje i test „Spectre and Meltdown“ na zranitelnosti ve firmwaru procesorů.

Další informace:

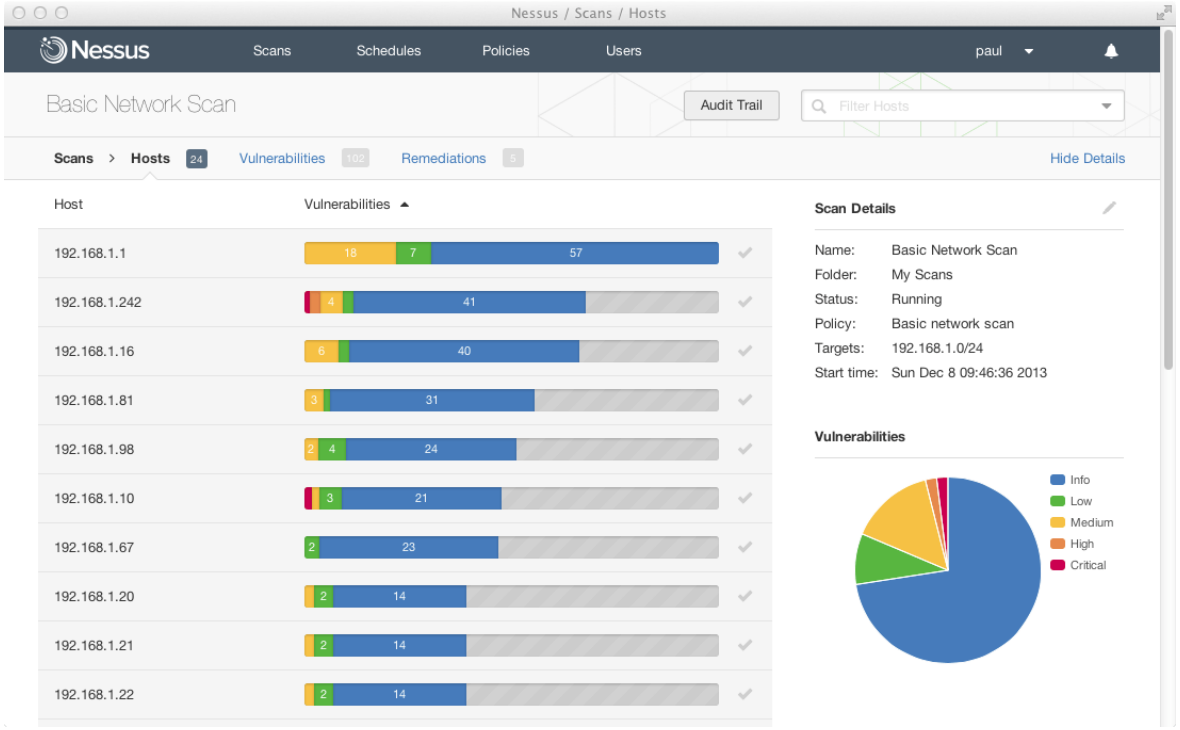
- http://nmap.org/
- *Easily Detect CVEs with Nmap Scripts*: https://null-byte.wonderhowto.com/how-to/easily-detect-cves-with-nmap-scripts-0181925/
- http://www.tenable.com/products/nessus
- *How to Find Exploits Using the Exploit Database in Kali*: https://null-byte.wonderhowto.com/how-to/hack-like-pro-find-exploits-using-exploit-database-kali-0156399/

[1] Zdroj: https://www.tenable.com/products/nessus

[2] Zdroj: https://seniordba.wordpress.com/2017/06/26/best-hacking-tools-of-2017-nessus-vulnerability-scanner/



Obrázek 6.7: Prostředí aplikace Nessus[1]



Obrázek 6.8: Výsledek testu v nástroji Nessus[2]

6.5 Monitoring sítě

Dohledový systém (Network Management Software, Monitoring System) je pokročilejší systém, který monitoruje stav sítě, sbírá informace z různých uzlů sítě, generuje reporty, generuje grafické výstupy v případě potřeby vhodně reaguje.

Existuje hodně dohledových systémů, z nichž mnohé jsou open-source: Nagios, Zabbix, OpenNMS, Zenoss, Cacti, atd.

Dohledový systém především dává dohromady různá data, kombinuje je, zobrazuje a informuje, ale sám data nevytváří – k tomu potřebujeme jiný nástroj. Dohledový systém tedy musí být napojen na některou databázi, kterou naplňuje vhodný systém pro řízení sítě (SNMP, Snort, NetFlow, WMI). Samotný dohledový systém nad těmito daty provádí různé operace, například

- monitoruje různé síťové služby (protokoly HTTP, SMTP, ICMP, atd.) včetně šifrovaných, využívá prostředky na uzlech sítě (Windows/Linux/UNIX),
- umí vizualizovat stav sítě, apod.
- dokáže reagovat v případě problémů:
  - okamžitý report (e-mail, SMS, pager, VoIP)
  - proaktivní ochrana (některé záchranné operace je schopen provést automaticky sám)

Dohledové systémy se dnes většinou konfigurují přes webové rozhraní.

SIEM (Security Information and Event Management) může svou definicí připomínat dohledový systém, je však více orientován na bezpečnost. Tyto systémy slouží jako nástroje pro shromažďování, analýzu a prezentaci informací získaných z různých napojených zařízení, ať už běžných součástí sítě nebo zařízení bezpečnostního charakteru (kamery, nejrůznější senzory, IDS/IPS). Plní obvykle tyto úlohy:

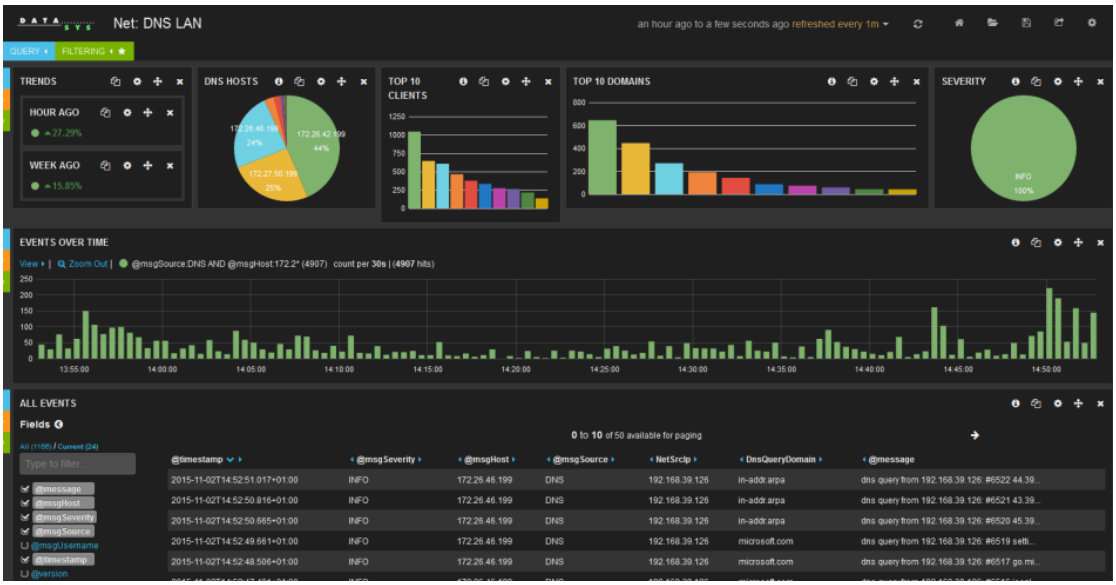
- shromažďování dat z různých zdrojů, práce s logy, normalizace údajů (převod do jednotného formátu),
- korelace – hledání a vyhodnocování vztahů mezi daty, např. mezi různými událostmi v síti,
- okamžité varování při výskytu problému,
- prezentace informací – vytváření přehledů, sestav, grafů, statistik.

Účelem je mít na jednom místě všechny informace potřebné při řešení bezpečnostních incidentů a mít možnost co nejrychleji na ně reagovat.

Můžeme najít buď kompletní SIEM řešení, nebo zvlášť SIM (Security Information Management, tj. práce s logy, třídění informací, reportování, vizualizace apod.) a SEM (Security Event Management, tj. především práce s logy, analýza v reálném čase, hledání závislosti mezi událostmi a rychlá reakce na události). SIEM používají při analýze jak běžné metody z dohledových systémů (jako je práce s logy), tak i metody podobné těm, které používají antivirové programy (práce se signaturami, heuristická analýza apod.).

Nabídka SIEM je poměrně rozsáhlá. Můžeme si vybrat open-source řešení nebo komerční produkt, ovšem vybírat bychom měli především podle vlastních potřeb. Důležitým kritériem je počet zařízení, která v síti máme a na jejichž logy bude SIEM napojen, mělo by nás zajímat, s kterými typy zařízení dokáže SIEM nativně pracovat a naopak která zařízení bude problém napojit, s velikostí sítě souvisí údaj EPS (Events per Second), tedy kolik událostí za sekundu bude muset SIEM zvládat, atd.

Vlastní SIEM nabízí Cisco, z dalších je oblíbený Splunk (má více variant), Helix Security Platform (od FireEye), Qradar SIEM (od IBM), AlienVault OSSIM (od AT&T Cybersecurity), AlienVault USM (od téhož výrobce), FortiSIEM (od Fortinetu), McAfee Enterprise Security Manager, LogRhythm NextGen SIEM Platform, Trustwave SIEM, Rapid7 InsightIDR, Juniper JSA SIEM, nebo český produkt ELISA (od DataSys, staví na Zabbixu, Xlogu a NetFlow).



Obrázek 6.9: Kibana – rozhraní používané v SIEM Elisa[3]

Další informace:

- https://www.nagios.com/products/nagios-xi/ (informace o dohledovém systému Nagios a screenshoty)
- http://www.hw-group.com/software/pd_snmp_cz.html (přehled dohledových systémů, hodně komerčních, pár volně šiřitelných, příp. některé komerční, ale s malým počtem sledovaných uzlů zdarma)
- http://www.monitortools.com/
- Kříž, Lukáš. *SIEM: Náročná cesta k pokročilému zajištění kybernetické bezpečnosti* [online], 2019. Dostupné na: https://ictrevue.ihned.cz/c3-66593380-0ICT00_d-66593380-siem-narocna-cesta-k-pokrocilemu-zajisteni-kyberneticke-bezpecnosti

[3] Zdroj: https://docplayer.cz/36323951-Datasys-elisa-log-management-rizeny-zabbixem-lukas-maly-dis-it-konzultant-bezpecnost-a-monitoring.html

Přílohy

Příloha A

Nástroje a příklady

A.1 Autopsy

Autopsy je aplikace, se kterou jsme se seznámili na straně 48 v kapitole o základním průzkumu systému. Zde si ukážeme postup pro jednoduchou úlohu – průzkum vytvořeného obrazu disku.

Předně potřebujeme jakýkoliv obraz. Pro studijní účely si můžeme vytvořit obraz nějakého (pokud možno malého) USB flash disku – pozor, je jedno, kolik na disku máme dat, obraz se vytvoří z celého prostoru včetně části bez dat. Využijeme některý z postupů popsaných v kapitole o základním průzkumu systému, podle toho, který operační systém používáme.

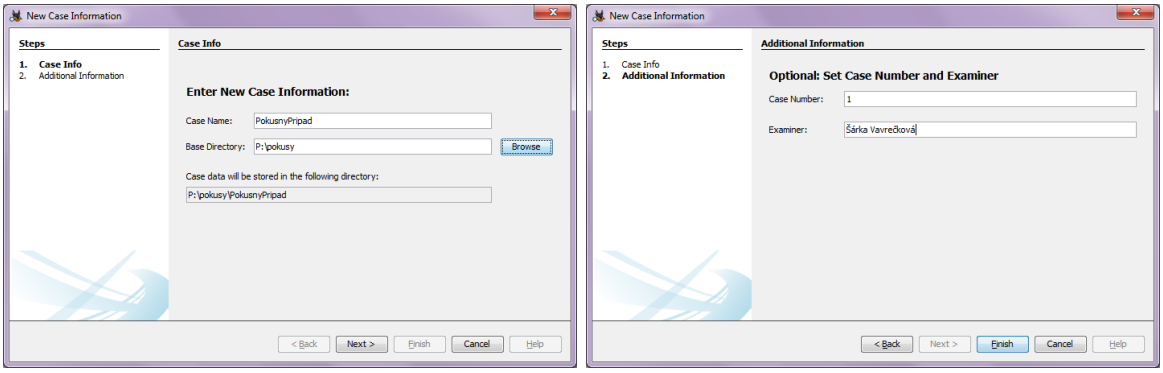
Autopsy dokáže načíst obrazy s příponou .IMG, .DD a .BIN, nikoliv .ISO. To však vůbec nevadí – pokud jsme vytvořili obraz disku (či oddílu) s příponou .ISO, jednoduše tuto příponu přejmenujeme na .DD. Vnitřně jsou totiž tyto dva formáty shodné, v obou případech jde o přesnou bitovou kopii daného média.

Postup (Průzkum obrazu datového média pomocí Autopsy)

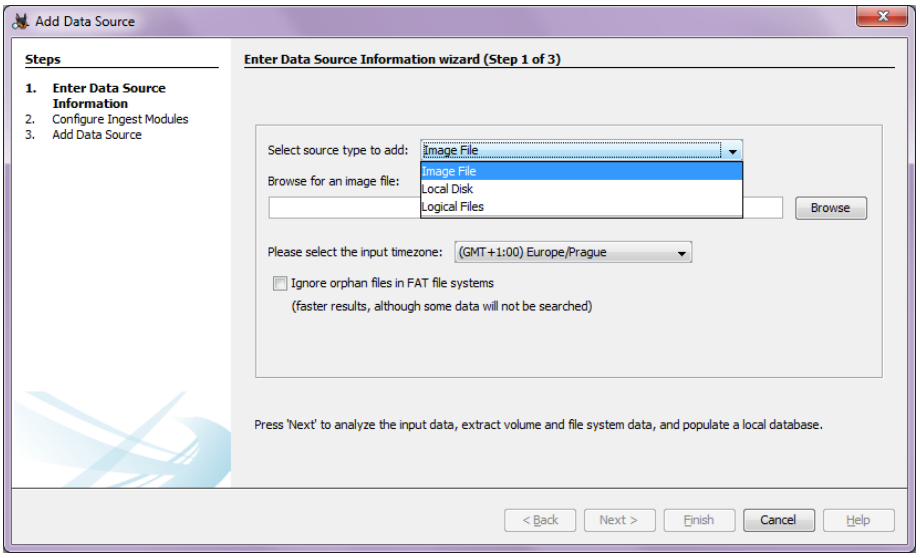
Projdeme si použití Autopsy krok za krokem.

1. Připravíme si obraz datového média k analýze, případně změníme příponu.
2. Pokud nemáme aplikaci nainstalovanou, stáhneme ji z webu http://www.sleuthkit.org/autopsy/ a spustíme instalaci. Instalace probíhá formou průvodce, nic složitého.
3. Po spuštění se objeví okno, ve kterém vybereme vytvoření nového projektu – *Create New Case*. Pokud už máme projekt vytvořený, zvolíme položku *Open Existing Case*.

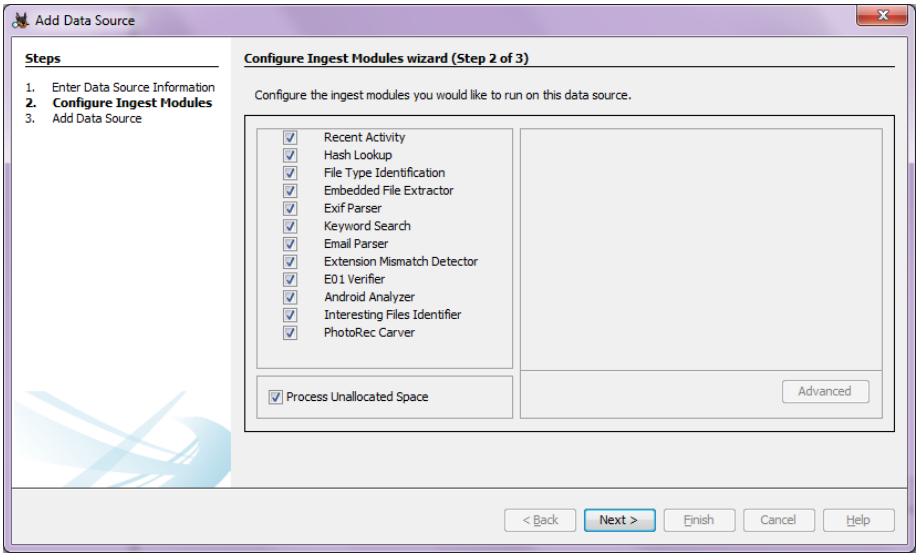
4. Zadáme název projektu a adresář, do kterého chceme ukládat data související s případem. Na další obrazovce průvodce přidáme číslo případu a jméno znalce, resp. zkoumajícího.



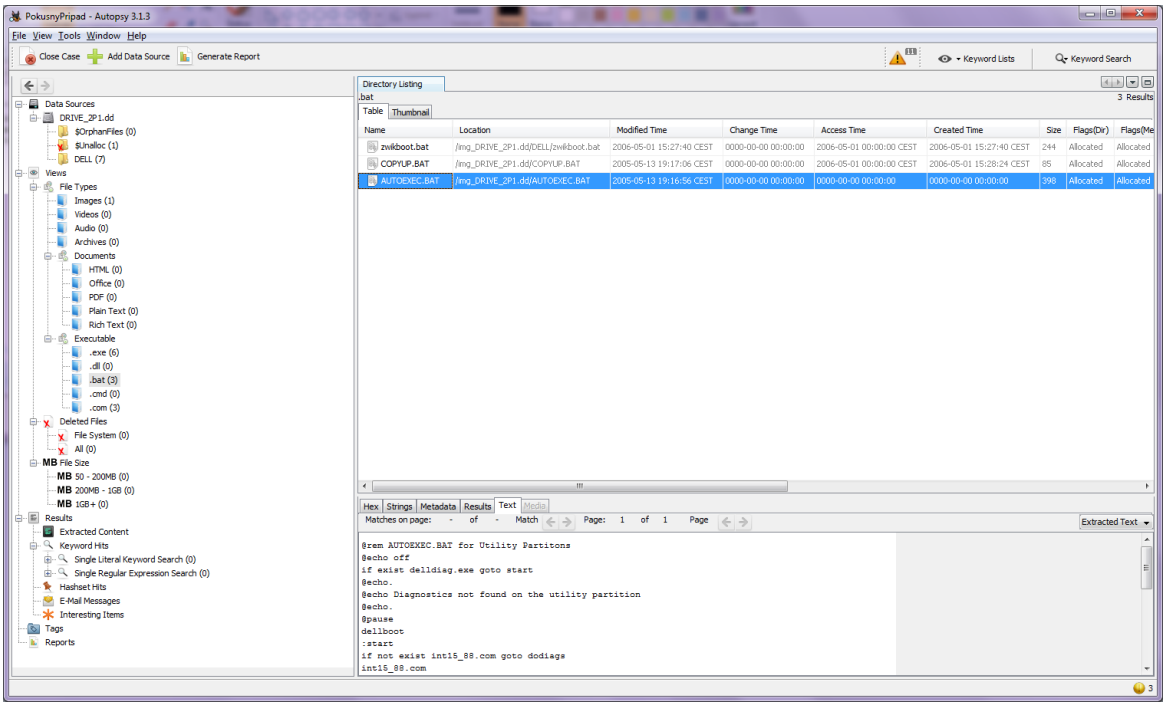
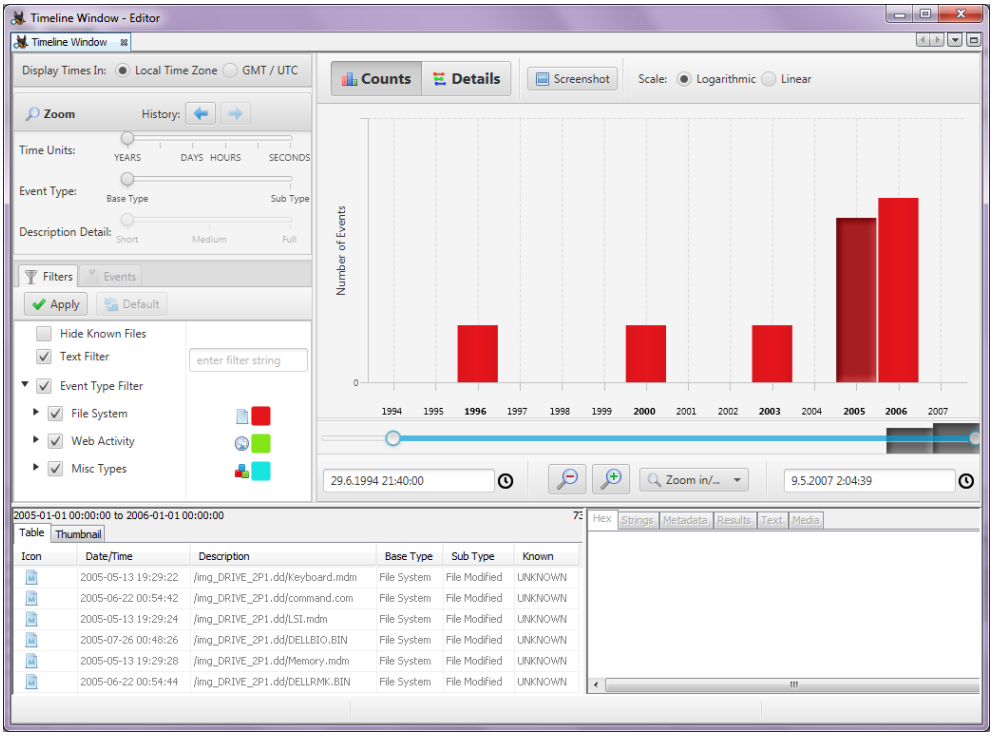
5. Dále vybereme, co chceme analyzovat (v našem případě půjde o vytvořený obraz, tedy *Image File*), pak klepneme na tlačítko *Browse* a najdeme soubor s obrazem.



6. Na další obrazovce máme možnost vybrat, co vše má být vyhledáno, můžeme nechat všechno zatrženo.



7. Po potvrzení se dostáváme k datům, která jsou v obrazu přítomna. Typ dat vybíráme ve stromě vlevo, vpravo nahoře je seznam, po klepnutí na konkrétní položku se vpravo dole objeví relevantní data.



8. V menu *Tools* najdeme dvě zajímavé položky – *Generate Report* (průvodce pro vygenerování výsledné zprávy) a *Timeline*. Druhá zmíněná položka umožňuje vytvořit časovou osu se soubory v obrazu obsaženými: