Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Metody ukládání uživatelských hesel v operačních systémech

Karel Suchý

Diplomová práce

2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2014/2015**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Karel Suchý**
Osobní číslo:	**I13447**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Metody ukládání uživatelských hesel v operačních systémech**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Cílem práce je zjistit mechanismy využívané pro ukládání uživatelských hesel v moderních operačních systémech. Autor zmapuje a podrobně popíše principy a matematické funkce pro ukládání uživatelských hesel na systémech Windows 7, Windows 8.1, Windows server 2012 (systém AD), Linux (distribuce CentOS, Fedora a Ubuntu). V praktické části nalezne vhodné nástroje pro útoky na autentizační mechanismy v jednotlivých OS, útoky zrealizuje a vyhodnotí.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

RUSSINOVICH, Mark E, David A SOLOMON a Alex IONESCU. Windows internals Part 1. 6th ed. Redmond: Microsoft Press, 2012, xxii, 726 s. ISBN 978-0-7356-4873-9.
RUSSINOVICH, Mark E, David A SOLOMON a Alex IONESCU. Windows internals Part 2. 6th ed. Redmond: Microsoft Press, 2012, xxi, 645 s. ISBN 978-0-7356-6587-3.
JELÍNEK, Lukáš, David A SOLOMON a Alex IONESCU. Jádro systému Linux: kompletní průvodce programátora. Vyd. 1. Brno: Computer Press, 2008, 686 s. ISBN 978-80-251-2084-2.
DRÁB, Martin, David A SOLOMON a Alex IONESCU. Jádro systému Windows: kompletní průvodce programátora. Vyd. 1. Brno: Computer Press, 2011, 472 s. ISBN 978-80-251-2731-5.

Vedoucí diplomové práce: **Mgr. Josef Horálek, Ph.D.**
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2014**
Termín odevzdání diplomové práce: **15. května 2015**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2014

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 8. 2015

Karel Suchý

Děkuji svému vedoucímu práce panu Mgr. Josefu Horálkovi, Ph.D.
za neocenitelné rady a pomoc při tvorbě této diplomové práce.

Dále děkuji svým kamarádům za jejich rady, zkušenosti a konzultace,
které mi taktéž byly velice nápomocny a vedly ke zkvalitnění této práce.

Poděkování také patří mé rodině za podporu při studiu.

Abstrakt

Motivací k vypracování této práce byla snaha změnit nízkou informovanost o možnostech neautorizovaného přístupu do operačních systémů Windows a Linux. Uvedeme konkrétní postupy a aplikace, kterými prezentujeme možné způsoby útoku na tyto operační systémy v praxi. Neméně důležitá část je věnována pro pochopení struktury a fungování operačního systému. Umožní nám to lépe čelit možným útokům a zjistíme, z čeho vychází různá bezpečnostní doporučení a kdy je použít.

Klíčová slova

bezpečnost, ochrana, hacking, Linux, Windows, prolomení hesel, haše

Title

Methods of storing user passwords in operating systems

Abstract

The motivation to elaboration of this thesis was effort to change low knowledge about possibilities of unauthorized access into operating systems Windows and Linux. We will introduce specific processes and applications, which we present possible methods of attack on these operating systems in practice. No less important part is given to understanding of structure and functioning of operating system. It allow us to better to brave to attacks and we will find out from what various security measures comes and when it apply.

Keywords

security, protection, hacking, Linux, Windows, braking the passwords, hash

Obsah

Seznam tabulek a obrázků

Úvod

Abychom mohli efektivně bránit svá data, potřebujeme znát slabiny a vlastnosti našeho systému a možnosti útočníků. Potřebujeme vědět, jak ke škodám dochází, abychom jim mohli zabránit. K tomu nám pomáhá takzvaný etický hacking.

Cílem etického hackera je nalézt seznam slabých míst (všech zneužitelných chyb), případně se pomocí penetračního testování pokusit do konkrétního systému proniknout s cílem získat co nejvíce práv či informací a ukázat tak (většinou nějaké firmě), jak by útočník chyby mohl zneužít. Poté může navrhnout bezpečnostní opatření, která by vedla k eliminaci nalezených slabin. Naše práce se nesnaží toto vše obsáhnout. Zaměříme se na oblast ochrany přístupu do systému heslem, nebo chcete-li na možnosti útoku na hesla operačního systému, případně jak se do systému přihlásit a heslo k tomu nepotřebovat. Jakkoli se to může zdát být složité, existují jednoduché postupy, které k tomu nevyžadují zvláštní znalosti ani prostředky. Možná překvapí, že uvedené postupy jsou v dnešní době stále možné a nejsou automaticky znemožněny už při instalaci operačního systému.

Cílem práce není zmapovat veškeré možnosti hackera, ale spíše prezentace informací, které by zájemce měly vést k uvědomění si, že je potřeba o zabezpečení přemýšlet a že bezpečnost dat není tak automatická, jak by se možná dalo předpokládat. Že dodržování určitých pravidel pro ochranu svých dat (ale i firemních či ostatních uživatelů pracujících na stejném počítači či počítačové síti) je nezbytné. Aby si například uvědomil, že vyžadování určitého formátu hesla správcem sítě (či jiného informačního systému, například internetového bankovnictví) má svá opodstatnění a že to není jen jeho snaha otrávit život běžného uživatele.

Motivace útočníka může být různá. Od zavedení „špehovacího“ programu či různých trojských koní, přes získání soukromých dat a informací až po zamezení přístupu a jistě bychom našli další a další důvody.

Musíme myslet také na to, že dlouhou dobu vůbec nemusíme poznat, že se do našeho systému někdo naboural. Může se totiž v systému chovat nedestruktivně a snažit se po sobě zanechat minimum stop. Je-li jeho cílem získat nějaké informace, dejme tomu k naší diskreditaci (například z našeho emailu nebo z profilu na Facebooku, u kterých máme v internetovém prohlížeči

nastavené automatické přihlášení), tak si je prostě jen zkopíruje a vrátí zpět případné provedené změny. Takto po sobě takzvaně zamete. Pokud se jedná o firemní počítač, může takto získat citlivé firemní údaje a získat konkurenční výhodu.

Pro lepší pochopení bezpečnostního modelu současných operačních systémů si nejprve popíšeme, z čeho se operační systém skládá, co se v systému děje při jeho spuštění, co je potřeba k tomu, abychom se mohli autentizovat a jak je to vše zabezpečeno. Věnovat se budeme hlavně popisu operačního systému Windows 7 a společným vlastnostem současných distribucí operačního systému Linux.

V praktičtější části práce analyzujeme, navrhneme a předvedeme možné postupy a nástroje vedoucí k získání přístupu do operačního systému, aniž bychom znali heslo k jakémukoli z uživatelských účtů zřízených v daném operačním systému. Důležitou částí je pak návrh opatření, která by takovému postupu zamezila.

Tato práce nevznikla jako pomoc začínajícím hackerům (ti pokročilí zde uvedené informace a nástroje už znají), ale naopak pro obranu proti nim. Použijeme-li uvedené či podobné nástroje pro nekalé účely, vystavujeme se nebezpečí trestního stíhání! Je zde, možná více než v jiných situacích, velice snadné se dostat za hranici zákona. Dalo by se říci pouhým kliknutím na tlačítko v programu. Český trestní řád na to pamatuje v zákoníku 40/2009, § 230 Neoprávněný přístup k počítačovému systému a nosiči informací (odnětí svobody až na osm let), § 231 Opatření a přechovávání přístupového zařízení a hesla k počítačovému systému a jiných takových dat (odnětí svobody až na pět let) či § 232 Poškození záznamu v počítačovém systému a na nosiči informací a zásah do vybavení počítače z nedbalosti (odnětí svobody až na dvě léta).

1 Rešerše

Nejprve uvedeme zdroje, které se věnují podobné oblasti jako naše práce a jsou hodnotnými zdroji informací.

1.1 Knihy věnující se tématu zabezpečení operačního systému a hackingu

Uživatelské rady pro zabezpečení počítače jsou obsaženy prakticky v každé příručce věnující se nějakému operačnímu systému. Například [1] [2] [3] [4] [5].

Pro nás zajímavějšími tituly jsou [6] či [7] a jejich anglické originály. Jsou již staršího data, ale dá se očekávat jejich novější vydání, hlavně v anglických originálech. Dávají nám velmi dobrý pohled na postupy hackerů a jejich možnosti, dále jak útok odhalit, jak se mu bránit a jak se zachovat, pokud útok odhalíme.

1.2 Knihy věnující se tématu struktury operačního systému

Hlavním zdrojem informací o vnitřní struktuře operačního systému Windows je v anglickém jazyce psaná kniha [8] a její druhý díl [9], jenž je oficiálním zdrojem přímo od Microsoftu. Jiný zdroj prakticky neexistuje. Pokud se požadované informace v této knize nenajdou, nebo potřebujeme informace aktuálnější, je možné se pokusit je dohledat online na portále [10]. Knihy [8] a [9] jsou velice rozsáhlé a většinou je potřeba nastudovat jejich větší část, protože informace v nich obsažené jsou na sobě vystavěné a bez předchozích znalostí jsou pak používané termíny těžce pochopitelné. Pokud jsme na internetu našli jiné zdroje, informace v nich byly vždy převzaté z uvedené knihy a dost často ještě zkresleně interpretované a to včetně zdrojů z některých univerzit.

Česky psaná kniha [11] je asi jediná, která se tématu struktury operačního systému Windows dotýká. Není ale tak podrobná jak bychom pro naši práci potřebovali, je zaměřená spíše na

základní datové struktury, vývoj ovladačů, procesy, vlákna a podobně, tedy informace potřebné hlavně pro programátora.

Operačnímu systému Linux se věnuje velice kvalitní a podrobná příručka z roku 2007 [12]. Podle autorů se jedná o největší volně dostupný zdroj informací o Linuxu, je zaměřena na uživatele i administrátory, rozhodně doporučujeme jí věnovat pozornost.

Další kvalitní kniha o operačním systému Linux je [13]. Zaměřuje se spíše na programátora, nicméně z podstaty Linuxu jako otevřeného operačního systému a podstatně jednodušší struktuře oproti operačnímu systému Windows, jsou zde nám potřebné informace obsaženy v dostatečné míře. Kniha je ale z roku 2008, takže už nemusí obsahovat všechny aktuální informace. Obecně od doby vydání uvedených dvou knih o operačním systému Linux na podobné téma nové nevycházejí (alespoň v českém jazyce). Ono to ale příliš nevadí, protože existují například precizně zpracované manuálové stránky distribuce Ubuntu dostupné online, kde jsou prakticky všechny aktuální informace a principy fungování dostupné.

1.3 Vysokoškolské práce

Diplomová práce [14] z Vysokého učení technického v Brně z roku 2009 se věnuje stejnému tématu, jako práce naše. Je již ale šest let stará a za tu dobu se přeci jen v oblasti zabezpečení operačních systémů mnohé posunulo, stejně jako možnosti, zdroje a nástroje pro provedení útoku. Obsahuje podstatně méně informací ohledně struktury jádra a procesu spuštění operačního systému, což je dle našeho názoru podstatné k pochopení možností zranitelnosti operačního systému. Autor se v praktické části zaměřil na měření času prolomení několika ilustračních hesel, nicméně aby takovýto výzkum podal relevantní výstup, musel by obsahovat velké množství hesel, které v našich podmínkách není reálné otestovat. Například při použití slovníkového útoku silně záleží na uspořádání a volbě slov ve slovníku. Můžeme tedy něco naměřit, ale pokud útočník použije jiný slovník, nebo třeba jen jinak uspořádaný, výsledek může být fatálně jiný. Nicméně práci považujeme za kvalitní a doporučujeme ji k prostudování.

Jiné vysokoškolské práce se věnují tématům hašování, kryptografie a podobně a jsou hodnotným zdrojem informací i pro naši práci, nicméně se nevěnují našemu hlavnímu tématu. Odkazy na ně jsou případně uvedeny v příslušných kapitolách této práce.

2 Bezpečnost dat

Bezpečnost je jedno z nejdiskutovanějších témat v oblasti informačních technologií. Z důvodu rostoucího počtu citlivých osobních dat uložených v počítačích a počtu v nich realizovaných závažných úkonů se bezpečnost stává stále důležitější a živí se jí bezpočet specialistů. Jak roste množství svěřených dat počítačům, tak roste i potřeba tato data chránit před zneužitím. Je třeba si uvědomit, že je nutné chránit nejen data a komunikaci, ale také identitu. Útočník nám může napáchat mnoho škody, pokud bude vystupovat pod naší identitou. [12 s. 587-588] [15]

S nutností zabezpečení se nutně setkávají i běžní uživatelé, a všeobecná osvěta je tak velmi důležitá. Trendem posledních let je snaha o automatizování bezpečnostních procesů tak, aby byly možné chyby uživatele minimalizované. Zamezit se jim ale zatím nepodařilo. Příkladem může být situace, kdy uživatel nechá své heslo na papírku poblíž svého zařízení. Pak můžeme aplikovat sebelepší zásady hesla, ale zamezit přístupu se nám tím stejně nepodaří. [12 s. 588]

Prevence neautorizovaného přístupu k citlivým datům je nezbytná všude, kde více uživatelů přistupuje ke stejnému zařízení. Operační systém, stejně jako jednotliví uživatelé, musí být schopen ochránit soubory, paměť i konfiguraci nastavení od nežádoucích úprav a prohlížení. Operační systém zahrnuje především mechanismy pro ochranu hesel a souborů. Dále také ochranu před poškozením systému, prováděním určitých akcí a zabránění uživatelským programům ovlivňovat operační systém nebo programy jiných uživatelů. [8 s. 487]

Při ochraně dat musíme také myslet na situaci, kdy máme fyzický přístupu k uložišti dat, a tedy pro jejich čtení původní operační systém vůbec nepotřebujeme, a jehož nastavení tak nemá na přístup vliv. Ochranou proti této formě útoku je nějakým způsobem data šifrovat.

V rámci bezpečnosti rozlišujeme tři základní aspekty:

- Důvěrnost (confidentiality) – data (obecněji služby) jsou dostupná pouze oprávněným osobám. Přístup neoprávněných osob je narušením bezpečnosti.
- Integrita (integrity) – data (obecněji služby) mohou modifikovat pouze oprávněné osoby. Narušení integrity bývá většinou maskované.
- Dostupnost (availability) – data (obecněji služby) jsou přístupná v okamžiku, kdy je (oprávněná) osoba potřebuje. Opakem dostupnosti je odepření (denial of service).

Při různých útocích na informační systémy a ve snaze jim zabránit jsou jednotlivé aspekty zastoupeny v různé míře.

Například při zcizení počítače je narušena primárně dostupnost. Pokud data nejsou šifrována, může dojít k porušení důvěrnosti. Při napadení virem většinou dojde k odepření přístupu (dostupnost) nebo k povolení přístupu neoprávněným osobám (důvěrnost). Může dojít i k narušení integrity, pokud vir modifikuje data.

Pro zkoumání bezpečnosti je dále potřeba definovat oblast IT technologií, k níž se bezpečnost vztahuje a vůči čemu je útok veden. Mezi základní lze zařadit:

- bezpečnost hardwarová – zabezpečení fyzického odcizení hardware;
- bezpečnost na úrovni OS – správná konfigurace OS;
- bezpečnost síťová (především ve WAN sítích) – zabezpečení vzdáleného přístupu;
- bezpečnost aplikační (včetně např. bezpečnosti databázové) – zabezpečení před jejich neautorizovanou modifikací;
- bezpečnost personální (útoky na IT vedené skrze osobu, která oprávnění vlastní);
- bezpečnost organizační (definování přístupu osob k jednotlivým zdrojům v rámci organizace).

Jako se prolínaly tři základní aspekty bezpečnosti, i zde se jednotlivé oblasti zabezpečení navzájem prolínají. Situaci je potřeba vidět komplexně. Celková bezpečnost systému je jak známo dána nejslabším článkem řetězce. Často bývá opomíjena bezpečnost personální a organizační. Většinou z důvodu, že bezpečnost systému navrhuje IT manažer, pro kterého je toto na okraji jeho zájmů. Těmto oblastem se ale dále věnovat nebudeme.

[15]

2.1 Kvantifikace bezpečnosti

Úroveň bezpečnosti se kvantifikuje jen obtížně a porovnat bezpečnost dvou operačních systémů je tedy náročné. Existují jak metody obecnější a méně přesné, tak metody pro přesnější a komplexnější hodnocení. Ty jsou pak ale také časově i personálně náročnější, tedy mnohem dražší.

Množství investovaných prostředků do zabezpečení systému je jednou z měr kvantifikace bezpečnosti a umožňuje nám porovnávání bezpečnosti systémů. Bezpečnost zajištěná určitým množstvím investovaných prostředků by měla odpovídat finančním možnostem potenciálních

útočníků. Nevýhodou peněz jako míry bezpečnosti je jejich velmi relativní hodnota. Mění se nejen s časem a místem, ale i s jinými vlivy. Například pořídíme drahou jednu část, která ale bude jen omezeně spolupracovat se zbytkem našeho systému. Náklady tedy budou velké, výsledná bezpečnost nízká.

Další možností je hodnocení bezpečnosti systému nezávislou firmou, která následně vydá certifikát bezpečnosti s různými úrovněmi kvantifikovaných charakteristik. Problémem pak jsou velké náklady na hodnocení systému jako celku, což si mohou dovolit jen velké firmy či orgány státní správy. Dalším problémem je pak rozdílný přístup různých firem ke stanovení charakteristik a jejich hodnot a tedy mnohdy vzájemná neporovnatelnost hodnocení.

Dosavadní vývoj se tak dostal ke stanovení určitých standardů. Dva nejdůležitějších jsou TCSEC (Trusted Computer System Evaluation Criteria), běžně označovaný jako „Orange Book“ a komplexnější CC (Common Criteria), který je standardem mimo jiné i pro ČR.

U těchto standardů se jedná se o takzvanou předběžnou evaluaci (hodnocení) informačního systému. To znamená, že se nebere ohled na konkrétní nasazení (instalaci) v konkrétní firmě a s konkrétními lidmi. Hodnocení tedy není tak přesné, jako při certifikaci přímo ve firmě, nicméně je nesrovnatelně levnější a umožňuje snadné porovnávání systémů.

[15]

2.1.1 Trusted Computer System Evaluation Criteria (TCSEC)

Vytvořit řadu bezpečnostních hodnocení pro označení stupně ochrany komerčních operačních systémů, síťových prvků a důvěryhodných aplikací, byl jeden z cílů společnosti The National Computer Security Center (NCSC) založené v roce 1981 jako součást Národní bezpečnostní agentury (National Security Agency (NSA)) amerického ministerstva obrany (U.S. Department of Defense’s (DoD)). Tato bezpečnostní hodnocení byla definována v roce 1983. V originále jsou popsaná v dokumentu [16].

Standard je vhodný především pro orientační klasifikaci bezpečnosti infomačních systémů. Zaměřen je na obecné bezpečnostní charakteristiky. Ohodnocení (level of trust) na určitou úroveň znamená, že zároveň systém splňuje i všechny nižší úrovně – přidává přísnější ochranu. Nejnižší stupeň bezpečnosti je označen D (systém byl hodnocen, ale nesplnil žádnou vyšší úroveň),

následují C2, C1, B6, B2, B1, až po stupeň A1, představující nejvyšší bezpečnost. Tento standard je zastaralý a nepoužívá se pro hodnocení nových systémů. Je však možné úroveň současných systémů odhadnout, což nám umožní porovnávání a také nám umožňuje pochopit možnosti zabezpečení.

Nejvyšší stupeň A1 zatím žádný operační systém nesplňuje. Některé sice dosáhly jedné z úrovní B, nicméně úroveň C2 je stále považována za základ pro jakékoli bezpečné operační systémy pro běžné použití.

Mezi Windows jako první (v červenci 1995) byly na úroveň C2 certifikovány Windows NT 3.5 s aktualizací Service Pack 3. Certifikované byly dále i Windows NT 4 s aktualizací Service Pack 3 a i s aktualizací Service Pack 6a. Hodnocením by neprošli DOS a ani Windows 3.1 či 95. Odhaduje se, že úrovně C2 by dosáhly i Windows 2000, XP, Server 2003 a Vista.

UNIX systémy, které nepoužívaly stínová hesla, nebyly ani v souladu s úrovní C1. Modernější UNIX systémy splňovaly požadavky C2. Se změnami ve funkčnosti bylo možno dosáhnout B1, pro B2 by musely být změněny základní struktury a mechanismy a B3 by již vyžadoval konstrukční změny systému. Úrovně B2 dosáhly například Trusted XENIX 3 (1992) či Trusted XENIX 4 (1993). Úrovně B1 pak například HP-UX BLS 9.09+ (1995), Trusted IRIX/B 4.0.5EPL (1995), Trusted Rolaris V1.1 (1994) a další.

[15] [17] [8 s. 487]

Tabulka 1 *Úrovně hodnocení TCSEC [8 s. 488]*

Hodnocení	Popis
D	Minimální ochrana (Minimal Protection)
C1	Volitelná ochrana přístupu (Discretionary Access Protection)
C2	Kontrolovaná ochrana přístupu (Controlled Access Protection)
B1	Povinné řízení přístupu (Labeled Security Protection)
B2	Strukturovaná ochrana (Structured Protection)
B3	Bezpečnostní domény (Security Domains)
A1	Verifikovaný návrh (Verified Design)

2.1.2 Common Criteria (CC)

V lednu 1996 vydaly Spojené státy, Velká Británie, Německo, Francie, Kanada a Nizozemsko společně specifikaci hodnocení bezpečnosti Common Criteria for Information Technology Security Evaluation (CCITSE), obvykle označovanou jako Common Criteria (CC).

Je to v současné době nejpoužívanější mezinárodní bezpečnostní standard. Je novější a komplexnější než TSEC. Označuje se jako ISO/IEC 15408 a je jím evaluována většina současných operačních systémů. Není zde definována žádná výsledná stupnice, jako tomu bylo u TCSEC. Evaluace je vztažena k různým bezpečnostním požadavkům a mechanismům.

Posouzení bezpečnosti, označováno jako TOE (Target Of Evaluation), je provedeno na základě existence několika provázaných dokumentů.

CC zahrnuje koncepci (dokument) Protection Profile (PP), která se používá pro sběr bezpečnostních požadavků sepsaných do snadno specifikovatelných a srovnatelných sad.

CC dále mimo jiné zahrnuje koncepci Security Target (ST) – nejdůležitější dokument při konkrétní evaluaci. Obsahuje sadu bezpečnostních požadavků (tvořící bezpečnostní cíl), které mohou být vytvořené na základě PP. Umožňuje to evaluovat konkrétní produkt na bezpečnostní cíl požadovaný uživatelem. Stejný produkt tak může být evaluován oproti různým ST. Tento dokument musí být vždy zveřejněn.

Hlavní výsledek evaluace obsahuje dokument Evaluation Assurance Level (EAL). Úroveň bezpečnosti daného cíle je ohodnocena jedním ze sedmi stupňů ohodnocení. Nejnižším je EAL1 a nejvyšším EAL7. Hodnocení neudává absolutní bezpečnost, ale udává záruku získanou testováním daného kritéria. Obecně tak podle nich nelze porovnávat. Proto vznikly pro běžně používané operační systémy jisté společné cíle, podle kterých částečně porovnávat můžeme.

Většina operačních systémů dosahuje úrovně EAL4 (MS Windows XP, Suse Linux Enterprise Server 9 a podobně), specializovanější systémy (které v TCSEC dosahovali úrovně B2-B3) pak EAL5.

Konkrétně Windows 2000, Windows XP, Windows Server 2003 a Windows Vista Enterprise získaly certifikaci Common Criteria pod Controlled Access Protection Profile (CAPP) s ratingem EAL4+. To je zhruba ekvivalentem ratingu TCSEC C2.

Windows 7 a Windows Server 2008 R2 byly vyhodnoceny jako splňující požadavky (opět stupněm EAL4+) amerického ministerstva obrany pro všeobecné použití operačních systémů v síťovém prostředí (US Government Protection Profile for General Purpose Operating Systems in a Networked Environment, version 1.0, 30 August 2010 (GPOSPP)).

[8 s. 487-9] [15]

Stupněm EAL4+ byly ohodnoceny také systémy Red Hat Enterprise Linux 5 (2007) a Red Hat Enterprise Linux 6 (2012), Oracle Solaris 11.1 (2014), SUSE Linux Enterprise Server 11 Service Pack 2 (2013). Stupně EAL5+ dosáhl například IBM RACF for z/OS Version 1, Release 12 (2012). RACF (Resource Access Control Facility) je produkt společnosti IBM pro řízení zabezpečení ve svých operačních systémech. [18]

Tabulka 2 *Úrovně hodnocení CC [19]*

Úroveň	**Popis**
EAL1	Funkčně testováno (Functionally Tested)
EAL2	Strukturálně testováno (Structurally Tested)
EAL3	Metodicky testováno a kontrolováno (Methodically Tested and Checked)
EAL4	Metodicky navrženo, testováno a revidováno (Methodically Designed, Tested and Reviewed)
EAL5	Semiformálně navrženo, testováno a testováno (Semiformally Designed and Tested)
EAL6	Semiformálně ověřený návrh a testováno (Semiformally Verified Design and Tested)
EAL7	Formálně ověřený návrh a testováno (Formally Verified Design and Tested)

3 Bezpečnostní doporučení

Základní doporučení pro bezpečné používání počítače jsou uvedena v následujících bodech.

1) Používat správné heslo, případně čtečku otisku prstů (biometriku).

Dostatečně kvalitní heslo je nutné vyžadovat od všech uživatelů, kteří mají do systému přístup. Ne vždy musí být cílem útoku administrátorský účet, útočník může mít za cíl například získání přístupu do databáze. Pak by mu stačilo zaútočit na účet databázového uživatele.

Silné heslo by mělo splňovat následující požadavky [1] [2]:

- Minimální délka 8 znaků.
- Mělo by obsahovat číslice i písmena, pokud je to možné tak i znaky z rozšířené klávesnice (musíme si dávat pozor, aby z míst, kde se přihlašujeme, jsme tyto znaky měli na klávesnici).
- Nesmí se shodovat s uživatelským jménem.
- Heslo by nemělo obsahovat jméno naše / partnera / partnerky / domácího mazlíčka, název ulice, datum narození a podobné odvoditelné řetězce a ani jiná celá slova.
- Heslo by se mělo pravidelně měnit.

2) Nepracovat s administrátorskými právy.

Pokud se podaří útočníkovi napadnout náš uživatelský účet či námi spuštěnou aplikaci, pak se mu díky tomuto opatření nepovede dostat do důležitých částí systému (nebo to bude mít minimálně o dost složitější).

Ve Windows je k tomuto účelu zaveden nástroj Řízení uživatelských účtů (User Account Control, UAC). Máme-li tento nástroj povolen, pak přestože jsme přihlášení do operačního systému účtem s administrátorskými právy, standardně pracujeme s právy obyčejného uživatele. Pokud nastane situace, kdy budeme chtít přistoupit (my, nebo námi spuštěná aplikace) k zabezpečenému zdroji vyžadující vyšší oprávnění, systém se nás dotáže, zda tuto akci provádíme záměrně a vyžádá si naše potvrzení. [2, s. 201-203]

V Linuxové distribuci Ubuntu je toto implicitně řešeno tak, že účet root (obdoba účtu Administrator ve Windows) je deaktivován a systém je nastaven tak, že první účet vytvořený při instalaci má právo provádět správu systému a to skrze příkaz sudo. Ostatním uživatelům lze toto právo dodatečně udělit. Pokud nepracujeme v terminálu, použije se grafická podoba sudo (zobrazí se formulář pro zadání hesla) [20]. V jiných distribucích se tento přístup může lišit, proto je potřeba to vždy ověřit v příslušných manuálových stránkách.

3) Udržovat operační systém i nainstalované aplikace aktualizované.

Toto je velmi důležité. V počítačových systémech existují zranitelnosti, které mohou útočníci využít. Existují databáze těchto zranitelností, například NVD (National Vulnerability Database) provozovanou společností NIST. Je pak na výrobci software, jestli na zveřejněnou zranitelnost bude reagovat a jak rychle ji případně opraví a vydá opravnou aktualizaci.

Nalezená zranitelnost nemusí být přímo kritická. Problém může nastat především tehdy, pokud je pro zranitelnost známý takzvaný exploit. Jedná se o konkrétní postup či přímo aplikaci, která umožňuje zranitelnost prakticky využít. Existuje i databáze exploitů, například Exploit Database. Existuje mnoho exploitů, které jsou navrženy pro použití člověkem, který má relativně málo znalostí ohledně této problematiky.

[21]

4) Používat bránu firewall, antivirový a antispywarový program.

Tyto programy pomáhají zabránit útočníkovi v napadení počítače, případně napadení odhalit a počítač vyléčit.

5) Šifrovat paměťová zařízení.

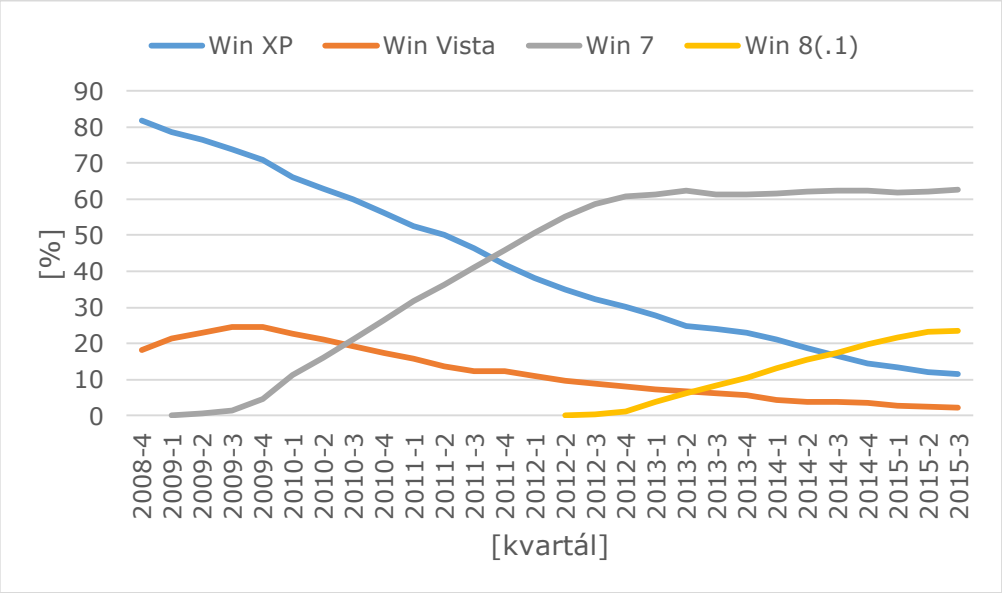
Jedná se o jedinou účinnou ochranu, pokud má útočník k zařízení fyzický přístup. Existuje mnoho programů, které toto zajišťují, někdy bývají přímo součástí operačního systému. Možné je šifrovat jak celé disky, tak i pouze některé jejich části (vybrané složky).

4 Bezpečnost v OS Windows

Hlavní podíl mezi operačními systémy na osobních počítačích má společnost Microsoft. V současné době je od této firmy používáno několik verzí operačního systému [22]. Kromě toho společnost Microsoft vydává i serverové verze odvozené od některých svých operačních systémů – jedná se o rozšíření o určité aplikace, administrátorské nástroje, širší podporu hardware a podobně. Struktura architektury zabezpečení ale zůstává zachována. [8]

4.1 Přehled verzí operačních systémů společnosti Microsoft

V následujícím grafu je zobrazen vývoj zastoupení používání jednotlivých verzí operačního systému Windows. Data pocházejí od společnosti StatCounter [22], která analyzuje data na vzorku přesahující 15 miliard přístupů za měsíc k více než 3 milionům webových stránek. Jedná se tak o systémy uživatelů, kteří využívají internet.



Obrázek 1 *Vývoj zastoupení verzí OS Windows mezi uživateli. Data pocházejí z [22].*

4.1.1 Windows XP

Nejstarší stále používaný operační systém vytvořený společností Microsoft je Windows XP. První verze těchto Windows byla vydána v roce 2001 a poslední velká aktualizace (označená jako Service Pack 3) vyšla 21. dubna 2008. Přestože zastoupení tohoto operačního systému je stále nezanedbatelné, jeho podpora ze strany Microsoftu byla ukončena k 8. dubnu 2014 [23] a neměl by tedy už být používán. K tomuto postoji, který bezesporu podporuje i stálý pokles jeho používání, se připojila i společnost Google. Ta totiž oznámila, že ukončí podporu svého internetového prohlížeče Chrome pro operační systém Windows XP ke konci roku 2015. Ve své tiskové zprávě [24] (16. dubna 2015) mimo jiné uvádí, že není bezpečné tento operační systém používat, protože již přes rok pro něj nejsou vydávány bezpečnostní aktualizace.

Obrázek 2 *Logo Windows XP*[1]

4.1.2 Windows Vista

Dalším operačním systémem byly Windows Vista v roce 2007. Poslední verze (Service Pack 2) byla vydaná v červnu 2009. Mezi klíčové novinky z oblasti zabezpečení určitě patří zavedení nástroje Řízení uživatelských účtů (User Accound Control – UAC). Tento nástroj je od té doby používán i ve všech novějších verzích operačních systémů společnosti Microsoft. Velké oblibě se ale tento systém netěšil, pravděpodobně pro svůj dlouhý start, vyšší požadavky na výkon hardware a pro běžného uživatele málo výrazné vylepšení oproti velice populárním Windows XP. [5]

Obrázek 3 *Logo Windows Vista*[2]

4.1.3 Windows 7

V říjnu 2009 vydala společnost Microsoft další operační systém pod označením Windows 7. Jeho poslední verze (Service Pack 1) byla vydaná 15. března 2011. Oproti nepříliš úspěšným

[1] Zdroj: *Microsoft Windows XP logo and wordmark.svg* [online]. [cit. 2015-08-18]. Dostupné z: https://en.wikipedia.org/wiki/File:Microsoft_Windows_XP_logo_and_wordmark.svg

[2] Zdroj: *Windows Vista.png* [online]. [cit. 2015-08-18]. Dostupné z: https://fr.wikipedia.org/wiki/Fichier:Windows_Vista.png

Windows Vista zde byl kladen důraz kromě jiného na rychlejší start systému a plnou kompatibilitu s existujícími ovladači zařízení, aplikací i hardwarem. Tento operační systém se stal velice populárním a v současné době to je nejpoužívanější systém. To je jeden z důvodů, proč následující popis operačního systému Windows bude stažen právě k Windows 7. Dalším důvodem je to, že pro tyto Windows existuje oficiální publikace od společnosti Microsoft [8], zabývající se podrobněji architekturou systému. Pro pozdější verze tato publikace zatím vydaná není. Nutno podotknout, že bezpečnostní architektura je mezi systémy Windows Vista, Windows 7 a Windows 8 podobná. [3] [5]

Obrázek 4 *Logo Windows 7*[3]

4.1.4 Windows 8

Aktuálním operačním systémem společnosti Microsoft je Windows 8. Vydán byl v říjnu 2012. Hlavní změna oproti předešlým systémům je způsobena příchodem dotykového ovládání na pracovní stanice a snahou o propojení pracovní stanice s mobilními zařízeními. Do tohoto systému byla zavedena podpora procesorů ARM a bylo vytvořeno částečně nové uživatelské prostředí označované jako ModernUI. Na to si mnoho uživatelů stěžovalo, a tak Microsoft provedl několik jeho úprav a v říjnu 2013 vydal zdarma dostupnou aktualizaci na verzi Windows 8.1. Základní principy ModernUI zůstaly zachovány, je ale lépe ovladatelné.

Mezi novinky v zabezpečení tohoto operačního systému patří možnost přihlášení se do systému online přes „Účet Microsoft“. To přináší komfort ve formě přenesení vlastního nastavení systému mezi různými zařízeními, nebo při přeinstalování systému. Přináší to ale také nové možnosti pro útočníka. Pokud nějakým způsobem získá heslo, může jej z jakéhokoli zařízení připojeného na internet změnit (nástrojem pro změnu hesla, pokud ho zapomeneme) a odříznout nás tak od přístupu k počítači, přesněji ke všem zařízením, na kterých tento účet používáme.

Windows 8 také zavedl kromě přihlášení se heslem další dvě nové možnosti přihlášení ve formě obrázkového hesla nebo čtyřmístného PIN kódu. Motivací pro tento krok bylo již zmiňované přizpůsobení operačního systému dotykovým zařízením. Oba typy nového přihlášení do systému ovšem snižují zabezpečení – existuje méně potenciálních variant než u klasického hesla.

[3] Zdroj: Windows 7 logo.svg [online]. [cit. 2015-08-18]. Dostupné z: http://so.wikipedia.org/wiki/File:Windows_7_logo.svg

Je ale lepší alespoň nějaké heslo, než žádné. V případě, že se heslo zadá pětkrát po sobě špatně, vyzve systém uživatele k přihlášení pomocí klasického hesla. To musí existovat vždy.

Obrázkové heslo je založeno na definování určitých gest provedených nad nějakým významným bodem (kvůli zapamatování jeho umístění) v obrázku. Možných gest je pět: ťuknutí, malý nebo velký kroužek ve směru nebo proti směru hodinových ručiček. Každá dvojice významných bodů lze navíc ještě spojit linií a to v obou směrech. Výpočet počtu variant lze odvodit následujícím způsobem. Počet gest je `5*p`. Počet kombinací spojovacích linií je `p!/(p-2)!`. Je vyžadováno použití sekvencí tří akcí. Počet variant takovýchto hesel je tedy dán vzorcem `(5p+(p!/p-2)!))^3`, kde `p` je počet významných bodů v obrázku. Například pro obrázek se třemi významnými body je 9261 variant.

Přihlášení pomocí osobního identifikačního čísla (personal identification number, PIN) vyžaduje zadání čtyř čísel. Zde je výpočet variant jednoduchý, možné jsou čísla od „0000“ do „9999“, tedy existuje 10000 možností.

[3 s. 357-362]

Obrázek 5 *Logo Windows 8*[4]

4.1.5 Windows 10

Novinkou, která se objevila během realizace této práce, bylo vydání nové verze operačního systému Windows 10. Stalo se tak 29. července 2015. Přeskočení číslovky 9 bylo pravděpodobně nutné z toho důvodu, že Windows 95 a Windows 98 se identifikují jako Windows 9 a některé aplikace by pak s tím mohly mít problémy.

Není známo, že by tato verze přinesla nové bezpečnostní prvky či zásadní změnu ve struktuře

Obrázek 6 *Logo Windows 10*[5]

[4] Zdroj: Windows 8 logo and wordmark.svg [online]. [cit. 2015-08-18]. Dostupné z: http://commons.wikimedia.org/wiki/File:Windows_8_logo_and_wordmark.svg

[5] Zdroj: Windows 10 Logo.svg [online]. [cit. 2015-08-18]. Dostupné z: https://commons.wikimedia.org/wiki/File:Windows_10_Logo.svg

jádra. Zdá se, že hlavním důvodem vydání této verze je nepřijetí nového uživatelského rozhraní ModernUI (známého pod označením Metro) většinou uživatelů Windows 8 a to i přes jeho podstatné vylepšení ve Windows 8.1. Nová verze Windows tak spojuje nabídku start z Windows 7 a dlaždice z Windows 8(.1) a dále vylepšuje pracovní prostředí, například přichytávání aplikací na různé části obrazovky či podporu více ploch. Microsoft, stejně jako tomu bylo u předchozí verze, se dále snaží o sjednocení systému pro všechny podporované platformy (stolní počítače, notebooky, tablety, telefony, Xbox) a snadné sdílení dat mezi různými zařízeními.

Majitelům Windows 7 a Windows 8(.1) je nabízen přechod zdarma. Minimální požadavky na výkon nebyly oproti Windows 8 změněny.

[25] [26]

4.2 Architektura systému Windows

Architektura Windows spadá spíše do rodiny operačních systémů s monolitickým jádrem. Znamená to, že jádro obsahuje kromě nejnutnějších součástí, mezi které patří obsluha přerušení, zasílání zpráv mezi procesy, časování a další některé pokročilejší funkce, jako jsou například souborové systémy, síťová komunikace a správa paměti. Operační systém pak pro svůj běh potřebuje jen pár procesů, které zajišťují například přihlášení uživatele přes grafické rozhraní.

Na nejnižší vrstvě je abstrakce hardwaru (Hardware Abstraktion Layer – HAL). Ta odděluje části operačního systému a aplikace od konkrétního hardware. Umožňuje, aby při přechodu na novou hardwarovou architekturu nemusely být přeprogramovány součásti v celém jádře, ale pouze části této vrstvy.

Následuje vrstva takzvaného tvrdého jádra (Kernel). Ta implementuje jednodušší mechanismy, jako je plánování vláken na procesoru, odložené volání procedur, základní synchronizační primitiva, práce s hardwarovými přerušeními a část obsluhy systémových volání.

Na stejné vrstvě se ještě nacházejí ovladače zařízení (Device drivers), umožňující správci ovladače komunikovat s různými typy hardware.

Jedna z nejzajímavějších vrstev je executiva Windows. Nad tvrdým jádrem vytváří složitější struktury a mechanismy, které jsou z uživatelského režimu dostupné přes systémová volání.

[11 s. 46-48]

Exekutiva se skládá z mnoha oddělených částí, mezi zajímavé patří:

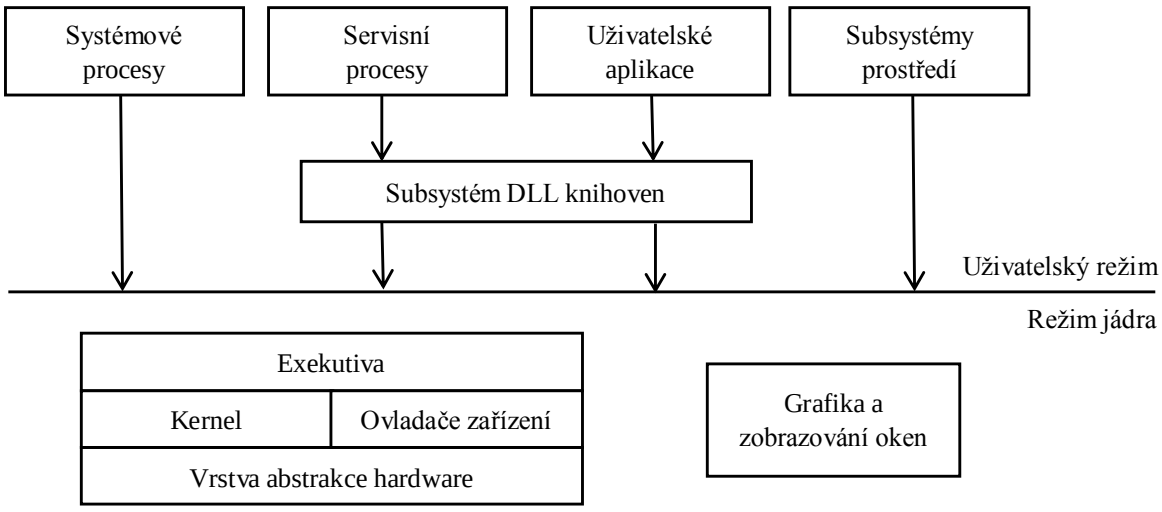
- *Configuration manager* – odpovědný za zavedení a správu systémového registru,
- *Process manager* – spravuje procesy a vlákna (vytváří, řídí a ukončuje) a poskytuje informace o nich,
- *I/O manager* – správa fyzických i virtuálních ovladačů pro vstupně/výstupní zařízení, umožňuje je načítat i za běhu systému,
- *Memory manager* – spravuje činnosti spojené s virtuální i fyzickou pamětí,
- *Security reference monitor* – součást zabezpečení Windows, popsán níže.

Executiva jakožto vrchní část spolu s tvrdým jádrem ve spodní části tvoří soubor Ntoskrnl.exe (`%SystemRoot%\System32\Ntoskrnl.exe`). Tento soubor exportuje množství zdokumentovaných funkcí s pevně danou strukturou názvů pro vyšší vrstvy operačního systému.

Mezi nedílné součásti širšího jádra systému Windows patří ještě Správce grafiky a grafického uživatelského interface (GUI). Podrobnosti těchto součástí, v souvislosti se zadáním této práce, pro nás nejsou důležité.

Popsané komponenty spadaly do režimu jádra. Co se týče součástí uživatelského režimu, budou nás zajímat jen některé systémové procesy. Mezi mě patří například Session Manager (`Smss.exe`), LSASS, Winlogon, wininit, logonUI, a Userinit. Označení „systémové“ znamená, že se jedná o proces důležitý pro běh systému. Pokud se takový proces ukončí, způsobí to restartování počítače.

[8 s. 35, 54-57] [11 s. 49, 51]



Obrázek 7 *Zjednodušená architektura Windows. Překresleno podle [8 s. 35].*

4.3 Registry

Registry slouží pro uchování nastavení Windows či aplikací. Je to jakási malá databáze sloužící pro rychlé hledání a manipulaci se záznamy. Oproti systémům na bázi Unixu, která svá nastavení ukládají v textových souborech, jsou data v registrech Windows uložena binárně. Tato forma by měla zajistit rychlejší zpracování. Jelikož pro člověka binární forma není čitelná, potřebujeme k tomu nějaký program. Typicky se jedná o Editor registru (regedit.exe).

[8 s. 23, 277] [11 s. 393]

4.4 Bezpečnostní komponenty operačního systému

Hlavní systémové komponenty a jimi využívané databáze zajišťující zabezpečení operačního systému Windows jsou popsané v následujících odstavcích.

Security reference monitor (SRM, „Bezpečnostní hlídač“) – služba ve Windows exekutivě, vynucuje zásady zabezpečení v místním počítači, chrání prostředky operačního systému, chrání objekty za běhu (run-time) a vytváří zprávy auditu zabezpečení.

Local security authority subsystem (LSASS, „Úřad podsystému místního zabezpečení“) – proces běžící v uživatelském režimu (`%SystemRoot%\System32\Lsass.exe`), je zodpovědný za lokální bezpečnostní politiku v systému (spadá sem povolení uživateli přihlásit se k počítači, politika hesel, privilegia uživatelů a skupin, nastavení bezpečnostního systémového auditu).

Databáze politiky LSASS – obsahuje nastavení zásad místního zabezpečení systému a je uložena v registru HKLM\SECURITY. Obsahuje informace:

- jaké domény vyžadují autorizaci,
- kdo má oprávnění k přístupu do systému a jak (přihlášení interaktivně, síťově nebo službou),
- kdo je přihlášený a s jakými právy,
- jaký druh bezpečnostního auditu má být proveden,
- přihlašovací informace používané při kešovaném doménovém přihlášení.

Security Accounts Manager (SAM, „Manažer zabezpečených účtů“) – služba spravující SAM databázi, je implementována jako knihovna (`%SystemRoot%\System32\Samsrv.dll`) a je spuštěna v rámci procesu LSASS.

Databáze SAM – obsahuje zavedené místní uživatele a skupiny, jejich hesla a další atributy. Je uložena v registru HKLM\SAM, najít ji lze v `%SystemRoot%\system32\config`.

Active Directory (AD) – adresářová služba, obsahuje databázi informací o objektech domény. Ukládá informace o heslech a právech doménových uživatelských účtů a skupin. Doménu tvoří skupina počítačů a jim přidružené skupiny zabezpečení, které jsou spravovány jako jeden celek (sdílejí společnou adresářovou databázi). Active Directory server (implementovaný v `%SystemRoot%\System32\Ntdsa.dll`) běží v rámci procesu LSASS.

Authentication packages („Autentizační balíčky“) – dynamicky nahrávané knihovny (DLL) zodpovědné za ověření uživatelského jména a hesla. V případě úspěchu pak do LSASS předají podrobné informace o uživatelově bezpečnostní identitě (aby mohl vygenerovat token). Operační systém standardně pro interaktivní přihlášení používá balíčky Kerberos a MSV_0. Výchozím je MSV1_0 (`%SystemRoot%\System32\Msv1_0.dll`), který slouží k přihlášení k místnímu počítači, pro přihlášení k doménám založených na systémech starších než Windows 2000 a v případě, kdy je pro stanici řadič domény nedostupný (například byla stanice odpojená od sítě). Balíček Kerberos (`%SystemRoot%\System32\Kerberos.dll`) slouží pro přihlášení k doméně. Spolupracuje s Kerberos službou běžící na doménovém řadiči podporující Kerberos protokol. Tento síťový autentizační protokol je popsán v RFC 1510 [27].

Winlogon („Manažer interaktivního přihlašování“) – proces běžící v uživatelském režimu (`%SystemRoot%\System32\Winlogon.exe`), je zodpovědný za reakci na SAS (Secure Attention Sequence, klávesová zkratka `CTRL+ALT+DEL`), řídí interakci s uživatelem související se zabezpečením, koordinuje přihlášení, spustí první uživatelský proces, zpracovává odhlášení a řídí různé jiné činnosti týkající se bezpečnosti (včetně spuštění LogonUI). Proces Winlogon zajišťuje nedostupnost uvedených činností pro jiné procesy.

Logon User Interface (LogonUI, „Přihlašovací uživatelské prostředí“) – služba běžící v uživatelském režimu (`%SystemRoot%\System32\LogonUI.exe`), poskytuje uživatelské rozhraní pro přihlášení, zajišťuje změnu hesla a zamknutí a odemknutí pracovní stanice. Pro zjištění uživatelských oprávnění používá CP. Může načíst i síťové poskytovatele pro sekundární ověření. Toto umožní současné ověření více poskytovateli najednou a to v průběhu standartního

přihlášení k systému Windows. Uživatel tak například získá přístup k určitým prostředkům serveru UNIX bez nutnosti dalšího ověřování.

Credential providers (CP, „Pověření poskytovatelé“) – objekt běžící v rámci LogonUI, získává z šifrované databáze uživatelská jména a hesla, PIN k čipovým kartám, nebo volitelně i další identifikační údaje. Standardně je umístěn v `%SystemRoot%\System32\authui.dll` a v `%SystemRoot%\System32\SmartcardCredentialProvider.dll`. Operační systém povoluje instalovat i další poskytovatele zprostředkovávající různé druhy ověření. Příkladem je ověření pomocí biometrických údajů, nejčastěji otiskem prstu.

Netlogon („Služba síťového přihlášení“) – nastaví zabezpečený kanál k doménovému řadiči pro zasílání ověřovacích žádostí. Umístěn v `%SystemRoot%\System32\Netlogon.dll`.

Kernel Security Device Driver (KSecDD, „Ovladače zabezpečení v jádře“) – knihovna funkcí v rámci režimu jádra (`%SystemRoot%\System32\Drivers\Ksecdd.sys`). Implementuje ALPC (Advanced Local Procedure Call – Rozšířené místní volání procedur) rozhraní, které využívají ostatní bezpečnostní složky z režimu jádra, například EFS (Encrypting File System – šifrování systémových souborů) používaném ke komunikaci s LSASS.

AppLocker („Aplikační zámek“) – mechanismus pro definování, které spustitelné soubory, DLL knihovny a skripty, mohou používat určení uživatelé nebo skupiny. Je složen z ovladače (`%SystemRoot%\System32\Drivers\AppId.sys`) a služby běžící v rámci procesu SVCHost (`%SystemRoot%\System32\AppIdSvc.dll`).

[8 s. 490-492, 558] [28]

Obrázek 8 v kapitole 4.6 ukazuje vztahy mezi některými zde uvedenými komponentami.

4.5 Proces spuštění operačního systému

Abychom správně pochopili proces samotného přihlášení se do systému, je vhodné znát kroky, které mu předcházejí. Při spouštění operačního systému se nejprve načte jádro a exekutiva Windows (Ntoskrnl), poté se spustí první proces běžící v uživatelském režimu, kterým je Session Manager Subsystem (`%SystemRoot%\System32\Smss.exe`).

SMSS je spuštěn po celou dobu běhu systému a hlídá kritické systémové procesy Windows. Pokud některý z nich bude ukončen, vyvolá to takzvanou modrou obrazovku smrti s kódem

STATUS_CRITICAL_SYSTEM_PROCESS_DIED. Při svém spuštění provádí důležité inicializační funkce – například vytvoří prostředí systémových proměnných, spustí Win32.sys pro přepnutí z VGA režimu při startu systému na standardní zobrazovací rozlišení, spustí CSRSS.exe zpřístupňující aplikacím Windows API, aplikuje mnoho nastavení z registrů. Hlavním úkolem procesu SMSS je správa relací. Relace představuje jakýsi uzavřený prostor, který neovlivňuje okolí. To umožňuje oddělení jednotlivých uživatelů, kdy například každý může mít namapovaný disk pod jiným písmenem, neovlivňují se instance programů a tak dále. Pro každou relaci spustí jednu kopii procesu winlogon.exe, kromě nulté, kde místo Winlogonu spouští Wininit – Windows Initialization Process. Ten kromě jiného spustí LSASS.

[8 s. 74] [9 s. 522-526] [11 s. 57]

Proces Winlogon bezprostředně po svém spuštění provede několik následujících kroků, aby zajistil kontrolu nad pracovní stanicí. Nejprve pro ni vytvoří bezpečností popis (desktriptor), který má pouze SID systému. Tento unikátní deskriptor zajišťuje, že jiný proces nemůže přistoupit ke stanici, pokud to Winlogon výslovně nepovolí.

Winlogon také vytvoří a otevře dvě pracovní plochy: *aplikační* (známá také jako interaktivní plocha) a *přihlašovací* (známá také jako bezpečnostní plocha). K přihlašovací ploše může pouze proces Winlogon, k aplikační je povolen přístup jak Winlogonu, tak uživateli. Toto uspořádání zajišťuje, že kdykoli je přihlašovací plocha aktivní, žádný jiný proces nemá přístup k aktivnímu kódu nebo datům spojeným s aplikační plochou. Operační systém tímto zajišťuje ochranu autentizačních operací, zahrnující nakládání s heslem a uzamykání a odemykání plochy.

Winlogon následně naváže ALPC spojení s autentizačním portem LSASS pro výměnu informací při přihlášení, odhlášení a operacemi s heslem. Nakonec ještě zaregistruje server zpráv, který čeká na SAS (Secure Attention Sequence, označuje implicitní bezpečnou klávesovou zkratku `Ctrl+Alt+Delete`), odhlášení a zamčení stanice. Toto opatření zabrání trojskému koni získat kontrolu nad obrazovkou při zadání SAS.

Než se někdo přihlásí k počítači, je zobrazena přihlašovací plocha. Pokud po přihlášení uživatel stiskne klávesovou zkratku `Ctrl+Alt+Delete` (tedy SAS), dojde k přepnutí plochy z aplikační na přihlašovací. To znamená, že SAS vždy vyvolá přihlašovací plochu kontrolovanou Winlogonem. Winlogon je také jediným procesm, který zachycuje požadavky na přihlášení.

[8 s. 555-557]

4.6 Postup přihlášení uživatele do operačního systému

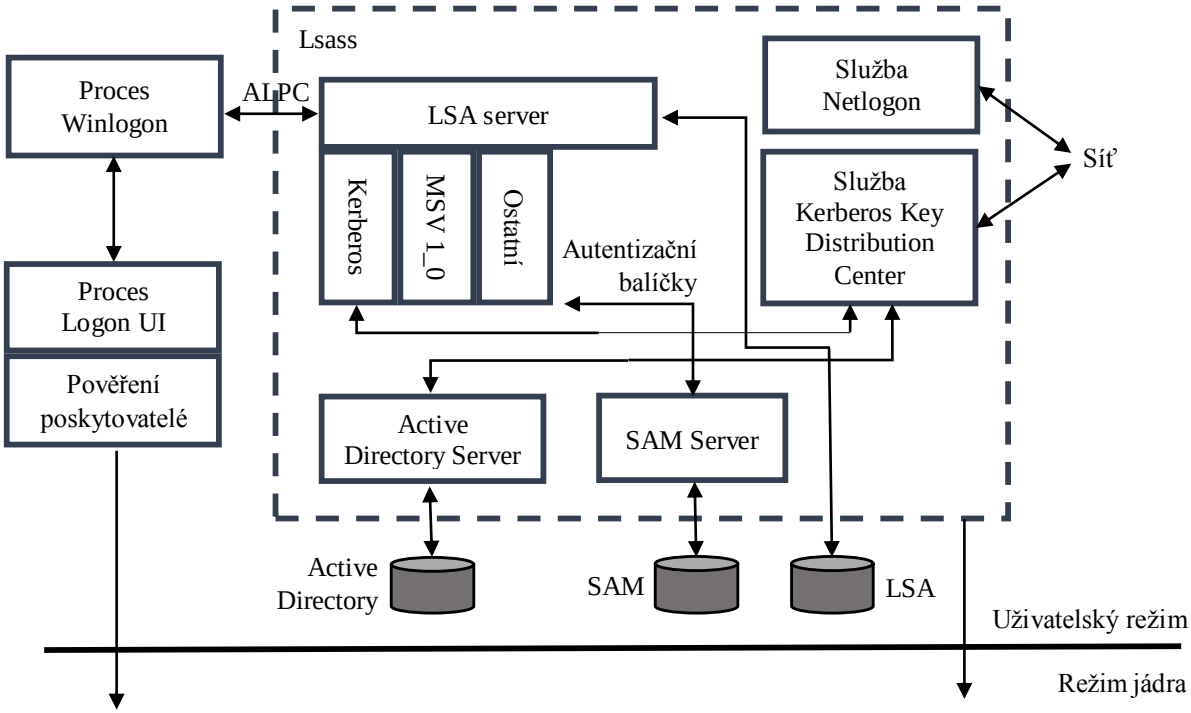
Základní přihlášení ve Windows je takzvaně interaktivní. To znamená, že uživatel aktivně zadává identifikační údaje. Neinteraktivní přihlášení je pak takové, které zajišťuje určitý software za nás, přesněji řečeno udržuje relaci, které předcházelo interaktivní přihlášení. [29]

Interaktivní přihlášení k počítači může být provedeno lokálně (uživatel má přímý fyzický přístup), nebo vzdáleně (prostřednictvím terminálové služby nebo služby vzdálené plochy). [30]

Tato přihlášení probíhají prostřednictvím:

- procesu přihlášení (Winlogon),
- přihlašovacího uživatelského rozhraní (LogonUI) a jím pověřených poskytovatelů,
- LSASS,
- jedním nebo více autentizačních balíčků,
- SAM nebo Active Directory.

[8 s. 555-556]



Obrázek 8 *Schéma komponent potřebných pro přihlášení. Překresleno podle [8 s. 556] a [8 s. 492].*

V okamžiku, kdy proces Winlogon obdrží žádost o přihlášení, okamžitě spustí LogonUI, který vyzve k zadání uživatelského jména a hesla.

Po zadání uživatelského jména a hesla získá Winlogon ukazatel na autentizační balíček zavoláním funkce LsaLookupAuthenticationPackage z LSASS. Seznam balíčků je umístěn v `HKLM\SYSTEM\CurrentControlSet\Control\Lsa`. Winlogon předá balíčku přihlašovací informace funkcí LsaLogonUser. Po ověření pokračuje Winlogon v přihlašovacím procesu daného uživatele. Pokud žádný z balíčků nevrátí úspěch, přihlašovací proces se přeruší.

Primární autentizační balíček MSV1_0 přijme uživatelské jméno a haš hesla a odešle požadavek místnímu SAM na získání informací o účtu, které zahrnují haš hesla, seznam přiřazených skupin a případná omezení účtu. MSV1_0 nejdříve zkontroluje omezení účtu, jako je povolený čas či typ přístupu. V případě, že se uživatel nemůže přihlásit, protože má omezení v databázi SAM, přihlašování selže a MSV1_0 pošle do LSA chybový stav. MSV1_0 pak zkontroluje haš hesla obdrženého ze SAM. Pokud se haše shodují, MSV1_0 vygeneruje LUID (locally unique identifier), vytvoří přihlašovací relaci, přiřadí ji vygenerovaný LUID a předá do LSA informace potřebné k dokončení vytvoření přístupového tokenu uživatele. Připomeňme, že přístupový token obsahuje SID uživatele, SID skupin a přiřazená oprávnění.

Pokud potřebujeme k autentizaci použít vzdálený systém, provedou se navíc následující kroky.

- V případě, že ověřování probíhá pomocí balíčku MSV1_0 (uživatel se hlásí k doméně založené na systému starším než Windows 2000), využije se služba Netlogon ke komunikaci s instancí Netlogonu na vzdáleném systému. Netlogon na vzdáleném systému ve spolupráci s tamním ověřovacím balíčkem MSV1_0 pošle autorizační informace zpět systému, na kterém je přihlašování prováděno.
- Pokud ověřování probíhá pomocí balíčku Kerberos, je základní (nesíťový) průběh stejný jako v MSV1_0. Ve většině případů se ale přihlášení do domény provádí z pracovních stanic, nebo serverů v síti (spíše než přímo na serveru s řadičem domény). Balíček pro komunikaci s doménovým řadičem používá port protokolu TCP/IP (port 88) na němž v rámci procesu LSASS běží služba Kerberos Key Distribution Center s autentizačním protokolem Kerberos (`%SystemRoot%\System32\Kdcsvc.dll`).
- Po ověření hašovaného uživatelského jména a hesla v Active Directory, vrátí proces Kdcsvc informace o pověření do LSASS, a ta vrací výsledek autentizace a uživatelská doménová práva (v případě úspěšného přihlášení) přes síť do systému, kde přihlašování probíhá.

Poté, co je přihlašování autorizováno, se LSASS podívá do databáze přístupových práv lokální stanice. Pokud přístup nebude povolen, proces přihlašování bude zastaven. LSASS odstraní nově vytvořenou přihlašovací relaci a Winlogonu vrátí chybu. Pokud je přístup povolen, LSASS přidá další bezpečnostní identifikátory (Everyone, Interactive, a podobně). To následně umožní zkontrolovat přístupová práva tohoto uživatele pro všechna SID (security identifiers) a přidat tato oprávnění do uživatelova přístupového tokenu.

Když LSASS nashromáždí všechny potřebné informace, zavolá exekutivu pro vytvoření přístupového tokenu. Exekutiva vytvoří primární přístupový token pro interaktivní přihlášení a token pro přihlášení k síti. Poté LSASS tokeny duplikuje, vytvoří ukazatele pro Winlogon a zavře své vlastní ukazatele. Nakonec pošle Winlogonu zprávu o úspěchu obsahující ukazatel na přístupový token, LUID pro přihlašovací relaci a informace o profilu (v případě, že nějaké autentizační balíček vrátil).

Winlogon následně vytvoří proces ke spuštění a to na základě hodnoty v `HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon\Userinit`. Tato hodnota může být několik .EXE souborů oddělených čárkou, přičemž výchozí hodnotou je Userinit.exe. Userinit načte profil uživatele a ve výchozím nastavení spustí Explorer.exe. Userinit se pak ukončí (což je důvod, proč Explorer.exe ve správci zařízení nemá žádného rodiče).

[8 s. 558-562]

4.7 Jak Windows uchovávají hesla

Jak jsme popsali dříve v této kapitole, heslo je lokálně v operačním systému Windows uloženo v databázi SAM, která je uložena v registru `HKLM\SAM` a na disku ji lze najít ve složce `%SystemRoot%\system32\config`. V případě, že využíváme Active Directory, je heslo uloženo zde, v objektu daného uživatele (`%SystemRoot%\NTDS\ntds.dit`). V obou případech je heslo při uložení převedeno z otevřeného textu na haš.

Windows implicitně do verze XP při nastavování či změně hesla vytvoří a uloží dva jeho otisky. Používá k tomu algoritmy pojmenované LM haš a NT haš. LM haš se vytváří z důvodu zpětné kompatibility pro případ, že by bylo potřeba se přihlásit k systémům, které NT haš nepodporují. LM haš byl zaveden v operačním systému Windows NT SP4 (systémy jako Windows 95/98/ME tento protokol nepodporovali). Nepotřebujeme-li zpětnou kompatibilitu, je důrazně

doporučováno ukládání staršího LM haše zakázat. Návod uveden například v [26]. LM haš je totiž i při základním útoku hrubou silou velice slabý.

V systémech Windows Vista a novějších je již naštěstí vytváření LM haše implicitně zakázáno, nicméně v případě potřeby stále zůstala možnost používání LM haše povolit. Současné uložení LM i NT haše útočníkovy dovolovalo se zaměřit pouze na jednodušší LM haš a kvalitnější NT haš tak úplně obejít. Bylo dokázáno, že i na procesoru Pentium II 450 MHz je možné rozluštit všechna alfanumerická hesla do 24 hodin.

[31] [32] [33 s. 150]

Extrahované záznamy z databáze SAM vypadají například takto (záznamy jsou zkrácené):

```
Anonymous:1039:e52cac67419a9a224a3b108f3fa6cb6d?:8846f7eaee8f1l17…

Administrator:500:48b48ef5635d97b6f513f7c84b50c317:8a6a398a2d8c84f…

Guest:501:a0e150c75a17008eaad3b435b51404ee:823893adfad2cda6e1a414f…

IUSR_ACMEPDCl:1001:cabf272ad9e04b24af3f5fe8c0f05078:e6f37a469ca3f8…
```

Tento příklad zobrazuje postupně dvojtečkami oddělené uživatelské jméno, ID, LM haš a část NT haše (u úplného výstupu bude více políček). [33 s. 152]

4.7.1 LM haš (nebo také LanMan)

Algoritmus je sice založen na dnes již zastaralém DES, bohužel je ale v postupu LM haše několik špatných kroků, které jsou podstatně důležitější pro velice špatnou sílu tohoto hašovacího algoritmu. Postup algoritmu je následující:

1) Znaky hesla jsou převedeny na velká písmena – algoritmus je tedy „case nonsensitive“.
2) Heslo je prodlouženo prázdnými znaky na délku 14 znaků (maximální délka hesla).
3) Rozšířené heslo se rozdělí na dvě části po 7 znacích.
4) Obě tyto části zvlášť vytvoří 64bitové šifrovací klíče pro DES algoritmus (ke každé části je přidán paritní bit).
5) Pomocí takto vytvořených klíčů se zašifruje přednastavený řetězec „`KGS!@#$%`“. Výsledkem jsou dva 64bitové šifrové texty (16 znaků).
6) Oba šifrové texty se jednoduše zřetězí do jednoho 128bitového šifrového textu.

Například, heslo „PassWord123“ bude zpracováno takto:

1) PASSWORD123
2) PASSWORD123000
3) PASSWOR a D123000
4) PASSWOR1 a D1230001
5) E52CAC67419A9A22 a 664345140A852F61
6) E52CAC67419A9A22664345140A852F61

Hlavní slabiny jsou zřejmé. Omezila se možná abeceda tím, že se převádějí všechny znaky na velké a maximální délka hesla má vzhledem k rozdělení na dvě části vlastně jen 7 znaků – každá část hesla se dá hledat samostatně. Toto je s dnešním běžně dostupným výkonem počítačů rychle prolomitelné hrubým útokem.

[29] [33 s. 154-5] [34] [35]

4.7.2 NT haš

Vytvořen jako nástupce slabého LM haše. NT haš je podstatně jednoduší algoritmus, který se více spoléhá na zvolené MD4 hašování. Postup NT algoritmu se tak zkrátil na pouhé dva kroky:

1) Znaky hesla jsou převedeny na znaky formátu Unicode.
2) Takto upravené heslo se pomocí algoritmu MD4 zahešuje, vznikne 128bitový haš.

Kromě toho, že algoritmus MD4 byl již také prohlášený za zastaralý a doporučují se delší výsledné haše než jen 128 bitů, směřuje největší kritika NT haše na nepoužití kryptografické soli. Technika takzvaného zasolení vygeneruje náhodný řetězec, který je přidán k původnímu heslu. To má za následek, že dvě stejná hesla budou mít různý haš. Díky tomu je možné k útoku použít takzvané rainbow tables. Jedná se o úsporně uložené předgenerované hodnoty hašů k různým textům, což umožňuje velmi zefektivnit metodu hrubé síly.

[29] [34] [35]

4.8 Řízení přístupu k prostředkům operačního systému

K řízení přístupu k prostředkům operačního systému, které lze zabezpečit (například soubor nebo tiskárna), potřebujeme pracovat na svazku se systémem souborů NTFS. Windows kvůli zpětné kompatibilitě stále podporují i systémy souborů založené na systému FAT, na nichž ale nelze aplikovat oprávnění. Každý prostředek umožňující zabezpečení má vlastníka, který nad ním má hlavní kontrolu – určuje, kdo k prostředku může, nebo nemůže přistupovat.

Jak již bylo řečeno výše, po úspěšném přihlášení do operačního systému má uživatel přiřazen „přístupový token zabezpečení“. Každému programu spuštěnému daným uživatelem se předá kopie jeho tokenu.

Aby operační systém povolil přístup k zabezpečenému prostředku, a to ať uživatelem nebo programem spuštěným uživatelem, provede ověření přístupového tokenu zabezpečení. Ověření provede porovnáním SID se „seznamem řízení přístupu“ (ACL) daného prostředku. Ten obsahuje seznam SID a k nim přiřazená přístupová oprávnění. Pokud takový přístup zamítne, oznámí nám to zprávou.

Pokud se do systému přihlásíme pod administrátorským účtem a používáme-li nástroj „Řízení uživatelských účtů“ (UAC), budou nám přístupové tokeny zabezpečení přiřazeny dva. Jeden obsahuje skutečně přiřazená administrátorská práva a druhý oprávnění standardního uživatele. Při spuštění aplikace je jí předán token se standardním oprávněním. Pokud si aplikace vyžádá přístup k prostředkům povoleným jen administrátorským účtům, pak nástroj UAC vyzve uživatele k souhlasu s předáním tokenu obsahující jeho skutečná administrátorská práva. Obdobná situace nastane v případě, že uživatel pracuje pod standardními právy a pokusí se přistoupit k prostředku vyžadující administrátorská práva. V tom případě nástroj UAC umožní uživateli aplikaci pustit pod právy jiného účtu vyzváním k zadání jména a hesla k němu.

[5 s. 522-525]

5 Bezpečnost v OS Linux

Linux je *moderní* (rychlá implementace nových technologií), *evoluční* (vývoj prováděn na základě požadavků uživatelů a hardware), *svobodný* (zdrojové kódy jsou veřejné, kdokoli si je může upravit a dále šířit) a *flexibilní* (lze nasadit jak na superpočítače, tak na jednoúčelová zařízení jako je síťová karta a podobně) operační systém.

Označení „Linux“ pro operační systém není úplně správné, přestože je tak většinou myšleno. Správněji bychom měli používat název GNU/Linux, jelikož název Linux používá i samotné jádro a většina programů a knihoven je z projektu GNU (GNU's Not Unix). Pro jednoduchost se ale používá pojmenování Linux pro celý operační systém.

Z hlediska většiny vlastností patří Linux mezi unixové systémy. Vznikl právě jako svobodná implementace jádra UNIXu, splňuje standard POSIX, praktické využití je obdobné. Ovšem nemůžeme prohlásit, že Linux je UNIX. Linux nevychází ze zdrojových kódů UNIXu a ani není certifikován sdružením The Open Group.

[13 s. 31-33] [36]

5.1 Přehled nejpoužívanějších distribucí Linux

Distribucí existuje velké množství a vybrat mezi nimi nejpoužívanější je dosti složité.

Podle [37], kde vedou žebříček na základě toho, na jakých distribucích jsou spuštěné webové stránky. Více než jedno procento tohoto trhu zabírají distribuce uvedené v následující tabulce.

Tabulka 3 *Nejpoužívanější distribuce Linux podle [37]*

Debian	32.4%
Ubuntu	28.7%
CentOS	20.5%
Red Hat	4.3%
Gentoo	2.1%
Fedora	1.3%
SuSE	1.0%

Dále existují dva vydavatelé žebříčků využívanosti Linuxových distribucí, kteří své statistiky vytvářejí na základě analýzy přístupu na jejich web. Nemusejí proto odrážet skutečný stav, nicméně pro náš účel je to postačující. Prvních sedm distribucí v žebříčku zahrnujícího přístupy za posledních 12 měsíců je dostupných na [38] a jsou uvedeny v následující tabulce (číslo znamená průměr přístupů k webu za den).

Tabulka 4 *Nejpoužívanější distribuce Linux podle [38]*

Mint	2752
Ubuntu	1751
Debian	1628
openSUSE	1314
Fedora	1236
Mageia	1100
CentOS	1052

Významný podíl podle [39] mají distribuce v následující distribuce (číslo je počet registrovaných zařízení).

Tabulka 5 *Nejpoužívanější distribuce Linux podle [39]*

Ubuntu	39,719
Debian	26,955
Fedora	10,098
Slackware	9,683
SuSE	9,483
Gentoo	7,388
Red Hat	4,720
CentOS	4,643

Nejen z uvedených statistik, ale i z nepřeberného množství „top ten“ internetových článků se dá usuzovat na významné distribuce. V následujících podkapitolách je stručně představíme. Jejich řazení je abecední.

Popisy jednotlivých distribucí jsou zpracované na základě [40], [41], [42] a [43].

5.1.1 CentOS

CentOS (Community Enterprise Operating Systém) je komunitní projekt podporovaný společností Red Hat. Komunita znovu sestavuje Red Hat Enterprise Linux tak, aby odstranila všechny ochranné známky. Je to tedy vhodná distribuce, pokud potřebujeme zdarma stabilní operační systém s dlouhou podporou.

Obrázek 9 *Logo distribuce CentOS*[6]

5.1.2 Debian

Debian je operační systém složený pouze z open-sources software, funguje od roku 1993. Je široce respektovaný, není aktualizován tak často jako například Ubuntu, ale je stabilnější. Původně byl zaměřen na servery, dnes je ale součástí i pracovních stanice. Obsahuje všechna populární prostředí, jako Gnome, KDE, Mate, XFCE a tak dále. Doporučuje se pro domácí servery. Je zaměřen na starší software, který je odladěný a není příliš vhodný pro začátečníky.

Obrázek 10 *Logo distribuce Debian*[7]

5.1.3 Fedora

Fedoru sponzoruje společnost Red Hat, je totiž základem pro komerční Red Hat Enterprise Linux. Oproti Red Hatu má výrazně zkrácenou dobu podpory a nemusí být tolik stabilní. Fedora používá prostředí GNOME ve výchozím nastavení, nemá tedy vlastní „vylepšení“. Je tu ale možnost snadného přechodu na KDE, Xfce, LXDE, MATE a Cinnamon.

Obrázek 11 *Logo distribuce Fedora*[8]

[6] Zdroj: Logo.png [online]. [cit. 2015-08-18]. Dostupné z: https://www.centos.org/images/logo.png
[7] Zdroj: Debian-61_glossy-large.png [online]. [cit. 2015-08-18]. Dostupné z: https://www.debian.org/Pics/debian-61_glossy-large.png
[8] Zdroj: Fedoralogo.png [online]. [cit. 2015-08-18]. Dostupné z: http://fedora.cz/wp-content/themes/fedora-cz/img/fedoralogo.png

5.1.4 Gentoo

Distribuce, která klade důraz na konfigurovatelnost a téměř jakoukoli přizpůsobitelnost. Jeho všestrannost a výkon ho odlišuje od ostatních distribucí. Obsahuje také pokročilý systém správy balíčků nazývaným Portage. Doporučuje se pouze zkušeným uživatelům, nebo těm, kteří se zajímají o vnitřek systému Linux.

Obrázek 12 *Logo distribuce Gentoo*[9]

5.1.5 Mageia

Francouzská Mageia začínala jako komunitní verze Mandriva Linux, která je dnes jen komerční a je založena na kódech distribuce Mageia. Mageia nabízí prostředí KDE a GNOME. Nejvýraznějším rysem distribuce je Mageia Control Centre, kde je možno přizpůsobit mnoho vlastností systému.

Obrázek 13 *Logo distribuce Mageia*[10]

5.1.6 Mint

Mint je založené na Ubuntu, používá stejné repositáře. Oproti Ubuntu zahrnuje v základu audio video kodeky pro různá multimédia a proprietární software. Dále místo Ubuntu Unity obsahuje prostředí Cinnamon nebo Mate. Navržen jako elegantní a moderní distribuce, která je snadno ovladatelná a výkonná.

Obrázek 14 *Logo distribude Mint*[11]

[9] Zdroj: Gentoo-logo.svg [online]. [cit. 2015-08-18]. Dostupné z: https://wiki.gentoo.org/wiki/File:Gentoo-logo.svg

[10] Zdroj: Logo_ln_2.png [online]. [cit. 2015-08-18]. Dostupné z: http://mageia.cz/wp-content/themes/mageia_twenty_1.5/images/logo_ln_2.png

[11] Zdroj: Logo.png [online]. [cit. 2015-08-18]. Dostupné z: http://www.linuxmint.com/img/logo.png

5.1.7 Red Hat

Red Hat je komerční stabilní distribuce Linuxu určená pro servery a pracovní stanice. Je založena na open-source distribuci Fedora. Společně se SUSE Linux Enterprise patří k nejlepším enterprise distribucím s profesionální podporou. Výrazně přispívají do mnoha open source projektů i do jádra samotného.

Obrázek 15 *Logo distribuce Red Hat*[12]

5.1.8 open SUSE / SUSE Linux Enterprise

To, co je pro společnost Red Hat distribuce Fedora, je pro komerční SUSE Linux Enterprise komunitní projekt open SUSE. Je vytvořený a sponzorovaný společností Novell, která v roce 2003 SuSE Linux koupila. Stejně jako Fedora je open SUSE méně stabilní. SUSE, stejně jako Ubuntu, klade důraz na uživatelské prostředí. Open SUSE cílí jak na začátečníky, tak na zkušené uživatele Linuxu. Jeho zajímavou vlastností je možnost po provedení změn nastavení systému tyto změny porovnat a případně změny vrátit zpět.

Obrázek 16 *Logo distribuce SUSE*[13]

5.1.9 Ubuntu

Ubuntu je založené na Debian distribuci, ale má své vlastní repositáře (jedny z největších). Výchozím prostředím je jeho Unity. Velmi populární a moderní distribuce, v posledních verzích probíhá optimalizace i pro dotyková zařízení (tablety, chytré telefony). Zaměřena na líbivý vzhled, jednoduchou a komplexní instalaci, vhodná pro začátečníky.

Obrázek 17 *Logo distribuce Ubuntu*[14]

[12] Zdroj: 640px-RedHat.svg.png [online]. [cit. 2015-08-18]. Dostupné z: https://upload.wikimedia.org/wikipedia/en/thumb/6/6c/RedHat.svg/640px-RedHat.svg.png

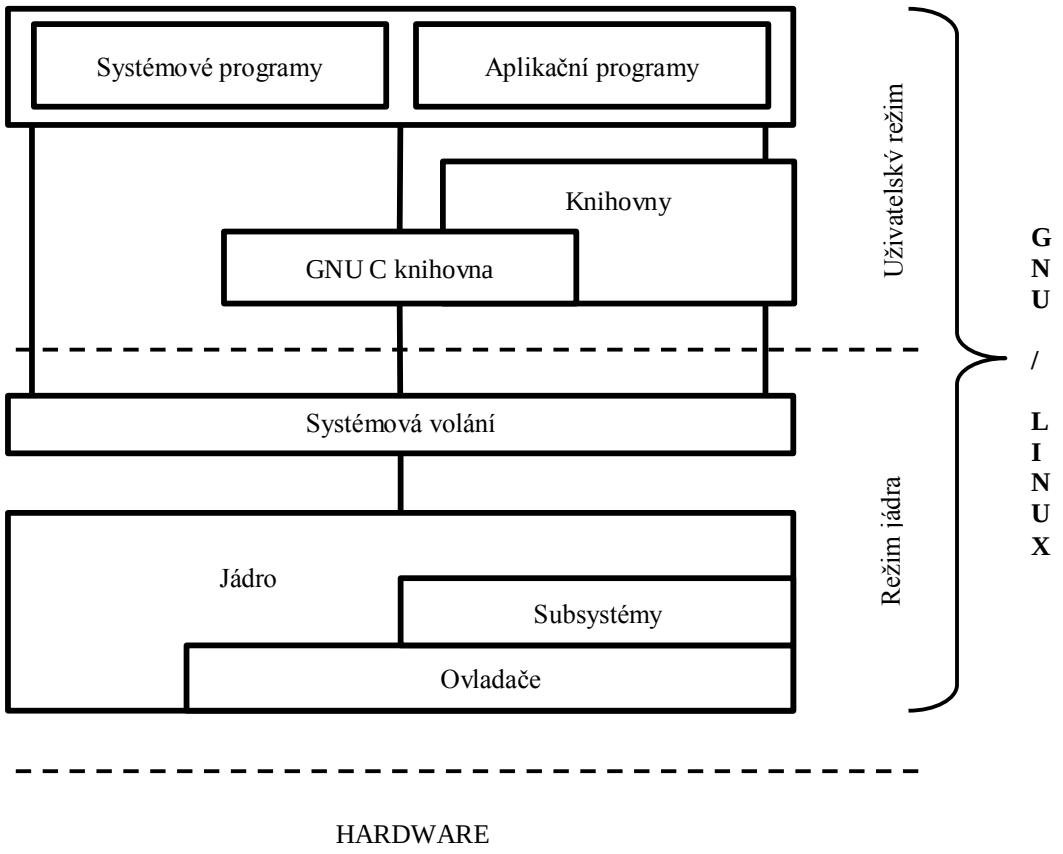
[13] Zdroj: Suse_logo_w-tag_color.png [online]. [cit. 2015-08-18]. Dostupné z: https://upload.wikimedia.org/wikipedia/en/b/b5/Suse_logo_w-tag_color.png

[14] Zdroj: Ubuntu-logo22.png [online]. [cit. 2015-08-18]. Dostupné z: https://design.ubuntu.com/wp-content/uploads/ubuntu-logo22.png

5.2 Architektura systému Linux

Operační systém Linux se skládá z *jádra* a *programů*. Programy bychom mohli rozdělit na *aplikační,* které zajišťují běžnou práci uživatele a na *systémové*, které zajišťují služby vyžadující se od operačního systému (hranice pro rozdělení je někdy nejasná, ale z praktického hlediska je její hledání nedůležité). Programy pro správné fungování a plnění úkolů, které od nich požadujeme, využívají základní nástroje poskytnuté jádrem. Z těchto základních nástrojů (služeb) lze zajistit všechny ostatní služby. Služby jádra jsou aplikacím dostupné prostřednictvím vrstvy systémových volání (též volání jádra). Programátor aplikací je většinou od systémových volání oddělen vrstvou standardní knihovny GNU C či jiných knihoven (nepotřebuje tak využívat přímo systémová volání).

[13 s. 65] [44 s. 149] [45]



Obrázek 18 *Architektura operačního systému Linux. Překresleno podle [45] a [13 s. 49].*

5.3 Jádro systému Linux

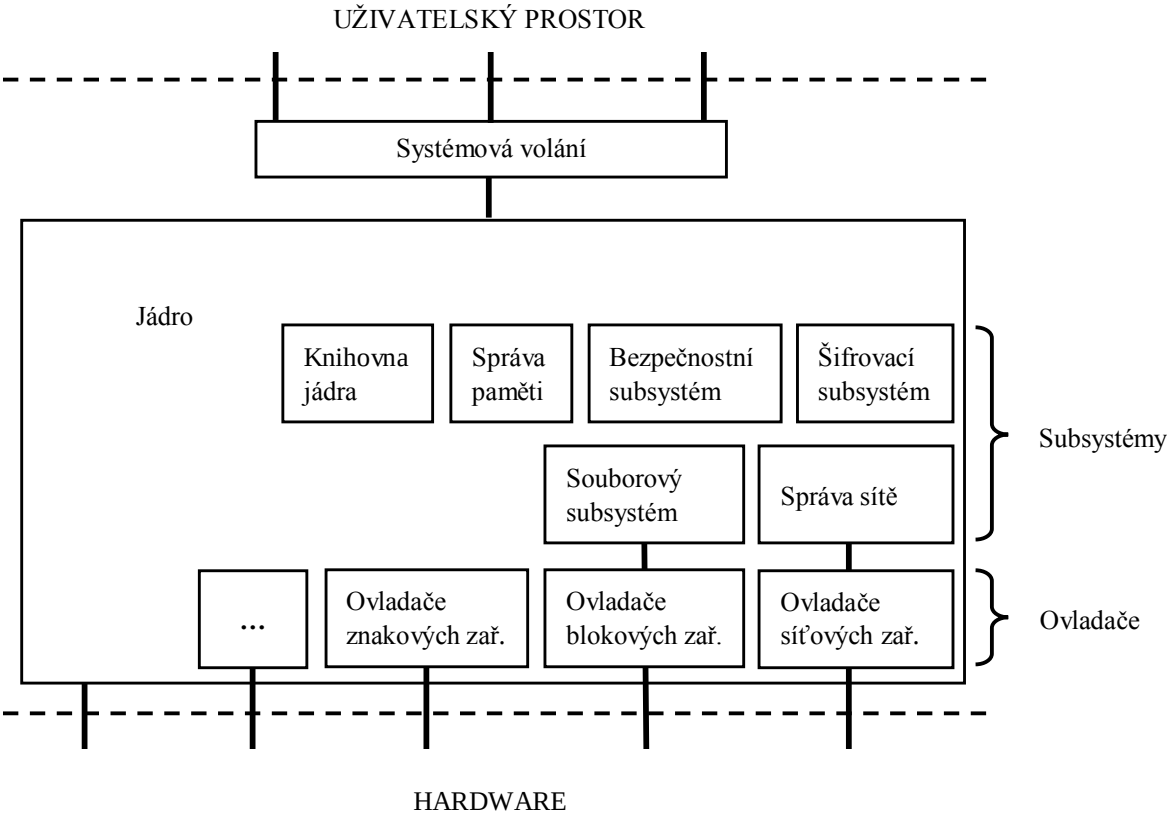
Jádro Linux je navrženo jako monolitické a modulární (stejně jako u Windows NT). Modularity se využívá pro urychlení běhu jádra, zmenšení jeho velikosti, zvýšení stability. Modularita umožňuje základní jádro (base kernel) rozšiřovat o potřebné moduly a to i za běhu systému. Například není problém přidat nový souborový systém, novou podporu určitého hardware, přidat nějaká systémová volání a podobně. Stejně tak je možno modul odebrat – pokud je již nevyužívaný, ušetřit paměťové místo. V případě Linuxu se toto označuje pojmem Loadable kernel module (LKM), Windows NT toto označují jako kernel-mode driver. [13 s. 485-488] [46]

Oproti systému Windows, kde je grafické prostředí součástí jádra (běží tedy v režimu jádra a sdílí s jádrem paměťový prostor), má Linux grafické prostředí mimo jádro. Grafické prostředí Linuxu – Windows X se spouští jako program, což umožňuje prostředí snadno měnit a v případě pádu jednoduše restartovat bez dopadu na běh zbytku systému. Cenou za tyto výhody je nutnost častějšího přepínání paměťového prostoru uživatelského režimu a režimu jádra a v důsledku tedy i větší režie. [13] [36]

Jádro operačního systému Linux tvoří několik důležitých částí. Mezi hlavní mimo jiné patří:

- Základní jádro (datové objekty a operace pro zajištění fungování jádra, zajištění řízení procesů jako je vytváření a přepínání, včetně multitaskingu),
- knihovna jádra (široce používané algoritmy jako řazení, vyhledávání, výpočet haše a podobně),
- správa paměti (přidělování paměťových oblastí a odkládacího prostoru pro různé části jádra, jednotlivé procesy, a pro vyrovnávací paměti),
- správa a ovladače souborových systémů,
- správu sítě,
- zprostředkování jednotného rozhraní programům, které vyžadují přístup k hardware počítače (obsahuje ovladače všech podporovaných technických prostředků),
- šifrovací subsystém (obsahuje různé šifrovací, hašovací a kompresní algoritmy)
- bezpečnostní subsystém (Linux Security Modules, místo standardního unixového řízení přístupu neboli diskrečního, umožňuje používat řízení bezpečnějším mandatorním přístupem, zajišťuje například SELinux nebo AppArmor).

[36] [44 s. 150, 151]



Obrázek 19 *Schéma jádra operačního systému Linux. Překresleno podle [44 s. 150] a [13 s. 487].*

5.4 Proces spuštění operačního systému

Jakmile je úloha zavaděče při spuštění počítače dokončena, předává se řízení procesu spuštění Linuxovému jádru. To se dekomprimuje, načte se do paměti, spustí se, inicializuje ovladače zařízení, datové struktury a tak dále. Inicializace jádra končí tím, že spustí proces init (zkratka z anglického initialization). Standardně se jedná o program `/sbin/init`.

Proces init je prvním spuštěným procesem v systému (má tedy číslo procesu 1) a automaticky se stává předkem (přímým či nepřímým) veškerých dalších procesů v operačním systému (v systému Unix musí mít každý proces nějakého rodiče – procesy vytvářejí jedinou stromovou strukturu). Init je spuštěn po celou dobu běhu systému, pokud se ho nepodaří spustit, nastane takzvaný kernel panic – zastavení činnosti jádra a pád systému. Init nemůže zaniknout (nelze jej zabít ani signálem SIGKILL).

Proces init pokračuje v procesu startu systému. Jeho úkolem je připravit systém pro běžné použití. Obvykle připojuje a kontroluje souborové systémy, síťové služby, spouští různé systémové démony, mimo jiné spouští a kontroluje procesy potřebné pro přihlášení uživatele do

systému a tak dále. Konkrétní seznam prováděných úloh je závislý na dané verzi operačního systému a jeho nastavení.

Procesu init existuje více verzí. Většina distribucí používá program sysvinit, jehož předlohou je proces init pro Unix System V. Alternativou může být init proces Unixu verze BSD. My se budeme věnovat více používanému programu sysvinit. Pro System V je v tomto kontextu důležité, že implementuje takzvané úrovně běhu (runlevel).

Úroveň běhu představuje celkový stav systému v závislosti na poskytovaných službách. Úroveň běhu tedy definuje sadu procesů, které proces init spouští či zastavuje. Možné úrovně jsou 0 až 6, přičemž uživatelsky definovatelné jsou úrovně 2 až 5. Pro uživatelsky definovatelné úrovně nejsou pevně stanovena pravidla pro jejich nastavení – je na správci systému, jestli a jak je nastaví. Úroveň běhu lze měnit za běhu systému. V tom případě proces init ukončí původně definovanou sadu procesů a spustí novou sadu procesů (pro nově zvolenou úroveň).

Tabulka 6 *Tabulka runlevelů [36] [48]*

0	vypnutí systému (Zavolá procesy nezbytné pro zastavení systému)
1	jednouživatelský režim (všechny služby vypnuté, k dispozici pouze jedna virtuální konzole pracující pod superuživatelem
2	normální režim bez grafického prostředí a síťových služeb
3	normální režim bez grafického prostředí
4	rezervován - například pro vytvoření vlastního runlevelu
5	normální režim (víceuživatelský režim s přístupem do sítě a grafickým přihlašovacím rozhranním - KDM (KDE), GDM (Gnome), XDM (X11) nebo jiné)
6	restart systému (Zavolá procesy nezbytné pro restart systému)

Jakmile je proces init spuštěn, jako první začne načítat konfigurační soubor `/etc/inittab`. Standardně je provedeno vše, co je ještě potřeba k inicializaci systému (nastavuje hodiny, inicializuje sériové porty, vyčistí adresář `/tmp`, spouští určité služby a tak dále). Pak init pokračuje ve čtení `/etc/inittab` souboru, kde je definováno nastavení systému pro jednotlivé úrovně běhu a nastaví výchozí úroveň běhu. Pro zvolenou úroveň běhu načte iniciační soubor `/etc/rc<x>.d`, kde `x` je zvolená úroveň (umístění se může lišit dle distribuce, v Gento je to `/sbin/rc`, v Red Hatu `/etc/rc.d/rc.sysinit`) a spustí všechny procesy zde uvedené.

Při přihlašování přes terminál (nebo konzolu) je pro každé takové spojení vytvořený samostatný proces getty. Ten čeká, až uživatel zadá své přihlašovací jméno, které poté předá parametrem programu login a svou činnost ukončuje. Následně si login vyžádá od uživatele heslo, které

zkontroluje podle databáze v `etc/passwd` a `etc/shadow`. Je-li heslo správné, spustí příkazový interpret (podle konfigurace uživatele, uloženo také v `etc/passwd`). V opačném případě se program login ukončí, proces init to rozpozná a pro daný terminál spustí novou instanci programu getty. Stejně tak se stane v případě, že se uživatel odhlásí, čímž je zajištěna možnost nového přihlášení. Soubor `/etc/inittab` dále obsahuje nastavení, které se má provést v případě výpadku proudu (je-li systém připojen k záložnímu napájení), nebo jak reagovat na klávesovou zkratku `Ctrl+Alt+Del`. Verzí programů getty (stejně jako u init) je více, například mingetty a agetty.

Grafické přihlašovací programy fungují podobně, místo příkazového interpreteru ale spustí nějaké grafické rozhraní. Na většině systémů založených na RPM (balíčkovací systém pro správu software v Linuxu) je grafická přihlašovací obrazovka spuštěna na úrovni běhu 5, kde init spustí skript `/etc/X11/prefdm`. Tento skript spustí preferovaného správce X Windows implementujícího přihlášení typicky prostřednictvím procesu gdm pro GNOME nebo kdm pro KDE.

[12] [13 s. 53-57] [36] [46] [48] [49].

5.5 Jak Linux uchovává hesla

Databáze uživatelů je uložena v souboru `/etc/passwd`. Informace o uživatelích, jako například jméno či domovský adresář, je potřeba nechat přístupné všem. Tedy uvedený soubor musí být čitelný pro všechny uživatele systému. Tento soubor mimo jiné ale může obsahovat také heslo. Přestože je zde uchováváno v zašifrovaném tvaru, existují vcelku jednoduché kryptografické metody, které zkouší hesla uhodnout. Zvláště pokud není heslo „silné“, je úspěšnost těchto metod velmi vysoká. Proto byla zavedena možnost heslo přesunout do „stínového“ souboru `/etc/shadow`, který již nemusí být (a není) uživatelem čitelný, a přístup k zašifrovaným heslům je tím výrazně ztížen. Číst ho může pouze superuživatel, nebo programy, které potřebují ověřit totožnost uživatele (jsou jim propůjčena přístupová práva vlastníka pomocí systémového volání setuid). Většina současných distribucí systému stínových hesel využívá.

Databáze skupin uživatelů je uložena v souboru `/etc/group`, případně v systému používajícím stínová hesla v souboru `/etc/gshadow`.

[44 s. 218-219, 158-159, 222] [50]

Použití stínových hesel nám umožní balík „Shadow Suite“. Ten nám kromě toho přináší další užitečné vlastnosti [51]:

- nastavení přihlašování podle konfiguračního souboru (`/etc/login.defs`),
- nástroje pro přidávání, modifikaci a rušení uživatelských účtů a skupin,
- definování platnosti hesla,
- vypršení platnosti účtu a jeho zamykání,
- stínová skupinová hesla (volitelně),
- dvojitá délka hesla (16 znaků) (nedoporučuje se),
- Lepší kontrola uživatele nad výběrem hesla,
- Programy pro sekundární ověřování (nedoporučuje se).

5.5.1 Formát souboru /etc/passwd

Každý řádek přísluší jednomu uživateli. Záznam je ve formátu

```
username:passwd:UID:GID:full_name:directory:shell
```

a jednotlivé položky (oddělené znakem „:“) uvádějí:

- přihlašovací jméno (username),
- heslo, „x“ znamená, že jsou hesla uložené ve stínovém souboru (shadow),
- ID uživatele (UID),
- ID skupiny (GID),
- Celé jméno (full_name),
- Cesta k domovskému souboru (direktory),
- Přihlašovací shell (shell).

Záznam může vypadat například takto:

```
NovakJ:x:561:561:Jan Novák:/home/NovakJ:/bin/bash
```

[50] [52] [53]

5.5.2 Formát souboru /etc/shadow

Každý řádek přísluší jednomu uživateli. Některé distribuce vyžadují, aby pořadí záznamů bylo stejné jako v souboru `/etc/passwd`. Záznam je ve formátu

```
username:passwd:last:may:must:warn:expire:disable:reserved
```

a jednotlivé položky (oddělené znakem „:“) uvádějí:

- přihlašovací jméno (username),
- heslo (passwd). Prázdná položka znamená, že heslo není vyžadováno a „*“ označuje zakázaný účet,
- počet dnů od 1. ledna 1970, kdy bylo heslo naposledy změněno (last),
- počet dnů po jejichž uplynutí může být heslo změněno, číslo 0 znamená, že může být změněno kdykoli (may),
- počet dnů po jejichž uplynutí musí být heslo změněno, číslo 99999 naznačuje, že změna hesla není vyžadována (must),
- počet dní před vypršením platnosti hesla, kdy bude na toto uživatel upozorněn (warn),
- počet dní po vypršení platnosti hesla, kdy bude účet zakázán (expire),
- počet dnů od 1. ledna 1970, kdy byl účet zakázán (disable),
- vyhrazené pole pro případnou budoucí potřebu (reserved).

Záznam může vypadat například takto:

```
NovakJ:$1$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15651:0:99999:7:::
```

[50] [52] [53]

5.5.3 Formát řetězce reprezentující heslo

Heslo v `/etc/shadow` obvykle bývá uloženo ve formátu

```
$id$salt$encrypted
```

A jednotlivé položky oddělené znakem „$“ znamenají:

- identifikátor použitého šifrovacího algoritmu (id),
- kryptografická sůl (salt) – náhodně vygenerovaný řetězec,
- zašifrované heslo (encrypted).

Řetězec reprezentující heslo může vypadat například takto:

```
$1$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1
```

[50] [54]

5.6 Proces ověření hesla

Předpokládejme, že řetězec reprezentující heslo v záznamu v souboru `/etc/shadow` vypadá například takto:

```
$1$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1
```

Vidíme, že je použito MD5 algoritmu (podle `$1$`), že při vytvoření účtu byla vygenerovaná sůl „`Etg2ExUZ`“ (řetězec mez druhým a třetím `$`) a že výsledné zašifrované heslo je „`F9NTP7omafhKIlqaBMqng1`“ (řetězec za třetím `$`).

Přihlašovací program, poté co uživatel zadá heslo, vyhledá podle přihlašovacího jména záznam v `/etc/shadow`, přečte typ použitého šifrovacího algoritmu a použitou sůl. V našem příkladu je to řetězec „`$1$Etg2ExUZ$`“. Nyní přihlašovací program má všechny potřebné informace pro vytvoření haše uživatelem zadaného hesla. Zavolá tedy níže popsaný program crypt, jehož návratovou hodnotou je zašifrovaná podoba hesla, kterou již jen stačí porovnat s hodnotou uloženou v `/etc/shadow`. V našem případě se tedy návratová hodnota programu crypt musí shodovat s řetězcem „`F9NTP7omafhKIlqaBMqng1`“.

Tímto je identita ověřena a program init může pokračovat v procesu přihlášení popsaném výše.

[50]

5.6.1 Šifrovací program crypt

O šifrování hesel (a dat obecně) se v Linuxu standardně stará program crypt. Přijímá nezašifrované heslo a takzvanou kryptografickou sůl. Pokud sůl začíná řetězcem ve tvaru „`$id$`“, pak místo implicitního DES šifrování použije podle id následující:

- 1 – MD5,
- 2 – Blowfish,
- 5 – SHA-256,
- 6 – SHA-512.

Sůl může mít až 16 znaků. Šifrované heslo musí mít pevnou velikost závislou na zvoleném algoritmu:

- MD5 – 22 znaků,
- SHA-256 – 43 znaků,
- SHA-512 – 86 znaků.

Uvedené informace se mohou dle distribuce měnit, proto je nutné je ověřit v příslušných manuálových stránkách.

[55]

5.7 Řízení přístupu k prostředkům operačního systému

Protože je v operačním systému Linux vše považováno za soubor (shellový skript, tiskárna a tak dále), je na vše aplikováno určité oprávnění. Základní oprávnění (označováno jako tradiční unixová přístupová práva) pocházejí z období kolem roku 1970. Tehdy měly počítače velmi málo paměti, a bylo tak nutné aplikovat oprávnění co nejjednodušeji. V rámci základních oprávnění lze nastavit přístupová práva pro vlastníka, pro skupinu, do níž vlastník patří a pro ostatní. Tento model je ale velice hrubý a například neumožňuje nastavit přístupová práva pro daného uživatele či skupinu a přitom nepovolit přístup všem ostatním. Pro situace, kdy tradiční přístupová práva nestačí, bylo do jádra systému zavedeno rozšíření ACL (Access Control List), nabízející jemnější nastavení přístupových práv. Zde můžeme udělit určitá přístupová práva konkrétním uživatelům a skupinám. Toto rozšíření musí podporovat souborový systém, například ReiserFS, Ext2, Ext3, JFS, XFS a další, protože nastavení ACL se ukládá do rozšířeného systémového atributu daného souboru.

Uvedené řízení přístupu se označuje jako takzvané diskreční, tedy volitelné (DAC – Discretionary Access Control). Každý soubor má definována práva, na jejichž základě je přístup povolen či odmítnut. Každý proces běží pod právy uživatele, který jej spustil a daný program má tak plnou kontrolu nad daty, ke kterým má daný uživatel přístup. Toho může útočník zneužít, pokud se mu nad běžícím procesem povede převzít kontrolu. Navíc existují procesy, které jsou spuštěny s právy superuživatele i přes to, že je spustil běžný uživatel.

V rámci operačního systému Linux ale existují i jiné přístupy, například SELinux nebo AppArmor, implementující mandatorní, tedy povinné řízení přístupu (MAC – Mandatory Access Control). To poskytuje plnou kontrolu nad všemi akcemi programů. Programy se spouštějí v takzvaném sandboxu (pískoviště), který omezuje přístup k částem systému ležících mimo prostor sandboxu (a to včetně programů běžících pod právy superuživatele). Povede-li se útočníkovi převzít kontrolu nad běžícím procesem s právy superuživatele, získá přístup jen v rámci sandboxu a ne do celého systému. Povinné řízení přístupu může být již součástí linuxové distribuce (AppArmor je obsažen od Ubuntu 7.10; SELinux je obsažen od Fedora core 2 a od CentOS 4), nebo lze dodatečně doinstalovat (nutné ověřit kompatibilitu s naší verzí jádra).

[13 s. 124-127] [56] [57] [58] [59] [60]

6 Vybrané algoritmy vytvářející haše

Algoritmy, které v této práci zmiňované operační systémy v rámci ochrany hesla využívají, jsou níže krátce popsané. Kromě odkazů na podrobnější informace uvedených přímo u jednotlivých algoritmů bychom pro hlubší porozumění o kryptografických algoritmech doporučili práci [61], ve které jsou, kromě některých dalších, rozebírány všechny zde uvedené algoritmy. Velice informačně obsáhlý dokument je také tento [62]. Zabývá se hašovacími funkcemi, jejich principy, příklady, kolizemi a bezpečností. Hašovacími funkcemi a některými útoky na ně se také zabývá další kvalitní práce [63].

6.1 DES (Data Encryption Standard)

Jedná se o blokovou symetrickou šifru. Symetrické šifrování znamená, že pro šifrováni i dešifrování zprávy je používán stejný klíč. Blokové šifrování spočívá v tom, že data se nejprve rozdělí na bloky pevně stanovené délky (většinou 64 nebo 128 bitů) a pokud poslední blok není zcela vyplněn, doplní se určitou výplní a teprve tyto bloky se jeden po druhém šifrují případně dešifrují vybraným algoritmem.

Jedná se o šifru publikovanou v roce 1975. která byla o dva roky později zvolena vládou USA za standard pro šifrování dat v civilních státních organizacích. V současné době je šifra považována za nedostatečnou, používá pouze 56bitový klíč a pomocí útoku hrubou silou je na běžném počítači rozluštitelná do 24 hodin. Za vznikem této šifry stojí firma IBM. Oficiální dokument popisující algoritmus DES je známý pod označením FIPS 46-3 a dostupný z [64].

Des používá 16 rund a 64b bloky. Šifrovací klíč má přitom délku jen 56 bitů, protože každý 8 bit je bitem parity, někdy je uváděno, že se jedná o takzvaný klamný bit.

Kromě oficiálního dokumentu jsou další podrobnosti uvedeny zde [69] a zde (včetně příkladu výpočtu) [68].

[65] [66] [67] [68]

6.2 Blowfish

Jedná se jako u DES o blokovou symetrickou šifru, jejímž autorem je Bruce Schneier. Poprvé byla zveřejněna v roce 1994, autor ji šířil jako neplacenou alternativu k DES a dodnes nebyla prolomena.

Blowfish také jako DES používá 16 rund a blok velikosti 64 bitů, ale klíč má proměnlivou délku 32 až 448 bitů. Vytvoření klíče pro jednotlivé rundy vyžaduje preprocessing ekvivalentní zašifrování asi 4 kilobajtů textu, což znamená oproti jiným algoritmům značnou režii. Pokud ale tento algoritmus používáme pro vytvoření haše, bere se tato režie jako výhoda při útoku hrubou silou, ke kterému je tak potřeba další výkon.

Kromě prvního zdroje v úvodu této kapitoly je více o algoritmu k nalezení zde [70].

[67] [70]

6.3 MD (Message Digest)

Jedná se o skupinu hašovacích algoritmů navržených profesorem Ronaldem R. Rivestnem. Hašovací algoritmus znamená, že jde o jednosměrné funkce (oproti například DES a Blowfish). To znamená, že je jednoduché spočítat haš zprávy, ale velice obtížné podle haše nalézt původní zprávu. Formálně je hašovací funkce taková funkce, která převádí vstupní posloupnost bitů na výstupní posloupnost pevné délky n bitů. Algoritmy nejsou využívány pouze k uchovávání hesel, používají se také pro kontrolu integrity dat. Uvedené hašovací algoritmy jsou založeny na podobných principech jako blokové šifry. Zajímají nás dvě varianty, MD4 a MD5.

MD4 byl vydaný v roce 1990. Algoritmus používá 64 rund a výsledný haš má 128 bitů a je většinou vyjádřen jako 32 znakové číslo. Oficiální dokument popisující algoritmus MD4 je pod označením RFC 1320 dostupný z [71].

Před vytvořením haše je zpráva prodloužena na délku dělitelnou 512 s posledními 64 bity, ve kterých je zakódována původní délka zprávy. Dále je zpráva rozdělena na bloky právě velikosti 512 a teprve tyto bloky jsou jednotlivě zpracovávány. Tento princip je shodný s MD5 i SHA-1 a SHA-256.

MD5 byl vydaný v roce 1991, jako náhrada za MD4, u které byla zpochybněna její dostatečná bezpečnost. V roce 2005 byly zveřejněny postupy k nalezení kolizního páru zpráv – a tak ani MD5 se již pro kryptografické účely nedoporučuje a jeho náhradou je SHA. Oficiální dokument popisující algoritmus MD5 je pod označením RFC 1321 dostupný z [72].

Kromě oficiálního dokumentu a zdrojů v úvodu této kapitoly jsou další podrobnosti uvedeny zde [74].

[50] [73] [74]

6.4 SHA (Secure Hash Algorithm)

Skupina hašovacích funkcí od americké NSA (Národní bezpečnostní agentura v USA). Jedná se o SHA-1 z roku 1995 s 80 rundami a délkou výstupu (haše) 160 bitů, což odpovídá 40 znakům a skupiny SHA-2 z roku 2001 obsahující SHA-224 (64 rund, 56 znaků), SHA-256 (64 rund, 64 znaků), SHA-384 (80 rund, 96 znaků) a SHA-512 (80 rund, 128 znaků), kde číslo za pomlčkou vyjadřuje délku haše. Algoritmy jsou založeny na podobném principu jako fungují algoritmy MD. Oficiální dokument popisující algoritmy SHA je pod označením FIPS 180-4 dostupný z [75].

Na skupinu SHA-2 zatím nebyly oznámeny žádné úspěšné útoky, nicméně se odhaduje, že SHA-1 je možno prolomit s následujícími náklady:

- v roce 2012 $2770 tis.,
- v roce 2015 $700K tis.,
- v roce 2018 $173 tis.,
- v roce 2021 $43 tis.

Dá se tedy předpokládat, že kolem roku 2018 bude slabá bezpečnost využívána organizovaným zločinem.

Kromě oficiálního dokumentu a zdrojů v úvodu této kapitoly jsou další podrobnosti uvedeny zde [77].

[73] [74] [76]

6.5 Bezpečnost hašovacích algoritmů

Hašovací algoritmy jsou považovány za bezpečné, pokud je současnými prostředky nemožné

1. najít zprávu, která odpovídá svému otisku – jednosměrnost,
2. najít dvě rozdílné zprávy, které mají stejný otisk – bezkoliznost.

K porušení prvního kritéria může být použit „vzorový útok“ – k nalezení zprávy korespondující s hašem je potřeba útok hrubou silou s 2^L výpočtů, kde L je velikost bitů v otisku zprávy.

K porušení druhého kritéria může být použit „narozeninový útok“ – k nalezení kolize je zapotřebí útok hrubou silou s $2^{L/2}$ výpočtů, kde L je velikost bitů v otisku zprávy.

Pokud uvažujeme problematiku prolomení hesla, pak nás zajímá hlavně porušení prvního pravidla. Kromě toho potřebujeme nějakým způsobem získat haš hledaného hesla (což je další část ochrany proti útoku na heslo). Proto nám příliš nevadí, že u algoritmů MD5 a SHA-1 byly nalezeny kolizní zprávy.

Pokud bychom hovořili například o digitálním podpisu, byla by situace jiná. Potřebovali bychom mít jistotu i v druhém pravidle. To se týká například certifikátů webových serverů, kde v lednu tohoto roku jich 82 % stále používalo SHA-1 a to i přes to, že je certifikačními autoritami k 1. lednu 2016 oznámen konec vydávání těchto certifikátů.

[73] [74] [76]

7 Analýza možností získání přístupu

Cílem našeho snažení je získání přístupu do operačního systému, ke kterému doposud přístup nemáme. Samozřejmě nejzajímavější pro útočníka je zjištění hesla správce systému, které mu poskytne nejvíce možností, jak systém využít pro svou potřebu či ho poškodit. Někdy je ale jednodušší se do systému přihlásit jinou cestou, než zjištěním hesla existujícího uživatele. Existují způsoby, jimiž do systému vložíme nový účet, případně heslo existujícího uživatele změníme, aniž bychom potřebovali znát jeho původní.

Proces, jehož cílem je získání přístupu do operačního systému, by se dal rozdělit do třech fází:

1) Získání přístupu k počítači,
2) Získání přístupu k lokální databázi s hesly,
3) Lámání nebo podvržení hesla.

Podvržením hesla rozumíme situaci, kdy se nám povede heslo odstranit, nebo nastavit vlastní, aniž bychom k tomu potřebovali zjistit heslo původní. Kroky dva a tři jsme si dovolili spojit do podkapitoly s názvem „Získání přístupu do operačního systému“.

7.1 Získání přístupu k počítači

Ideální stav (z pohledu hackera) je takový, kdy máme k zařízení fyzický přístup. Pak se můžeme začít snažit získat přístup do operačního systému.

Pokud tuto „výhodu“ nemáme, pokusíme si přístup získat zaútočením na některou ze síťových služeb. Tento proces zahrnuje úkony pro zjištění, s jakým operačním systémem a v jaké jeho verzi máme tu „čest vejít ve styk“ s následnou analýzou již odhalených slabin dané verze operačního systému (například hledáním na internetu). Pokračovat pak můžeme analýzou spuštěných síťových služeb a aplikací a opět hledáním jejich známých slabin. Naštěstí takových slabin není mnoho a správný administrátor o nich ví a v rámci možností buďto úplně zamezí jejich zneužití, či alespoň minimalizuje jejich dopad izolováním kritických oblastí. Za tímto účelem existuje mnoho nástrojů, některé dokonce dovedou pracovat „tiše“ a zanechat na síti po sobě minimum stop. Tato oblast je vzhledem k zadání této práce mimo oblast našeho hlubšího zájmu.

7.2 Získání přístupu do operačního systému

V zásadě existují dvě možnosti, jak přístup do operačního systému získat.

1) Heslo určitého uživatele smažeme či nahradíme, což nám následně umožní se přihlásit bez hesla, respektive s naším heslem,
2) heslo chceme rozluštit.

Za předpokladu, že k počítači máme fyzický přístup, je první varianta podstatně jednodušší.

- V případě Windows můžeme použít program „Offline Windows Password & Registry Editor“, se kterým zvládneme heslo vymazat asi za 1 minutu (je vyžadován restart systému). Jedná se o jakousi minidistribuci Linux s připraveným průvodcem.
- V Linuxu máme dvě možnosti, které nám umožní změnit heslo standardním příkazem passwd (původní heslo po nás vyžadováno nebude). Jedná se o úpravu programu odpovědného za zavedení systému GRUB (starší verze distribucí), respektive GRUB2 (od verzí Centos 7, Ubuntu 9.1, Fedora 16).
    - Systém spustíme v single user mode (úroveň běhu 1),
    - místo úrovně běhu spustíme bin/bash (spustí se přímo místo procesu init).

  Jinou možností je pak použít nějakou live distribuci Linux k nabootování systému (například z USB flash paměti či CD) a k následné editaci souboru /etc/shadow. Buďto haš hesla prostě smažeme, nebo ho nahradíme za vlastní, který jsme si předem připravili na jiném systému (například jednoduše zkopírujeme haš svého hesla námi používaného systému).

Aby se napadený o útoku nedozvěděl, můžeme nejprve soubor s hesly zálohovat a teprve poté provést reset hesla. Nakonec zálohu nahrajeme zpět.

- Toto ve Windows můžeme provést stejným programem jako jsme použili pro reset hesla, nebo pokud toto nechceme provádět pomocí příkazové řádky v Linux, použijeme například Windows PE. Jedná se o takzvanou live verzi operačního systému, která je veřejně dostupná od Windows Vista. V případě Windows ve spojení s programem „Offline Windows Password & Registry Editor“ můžeme využít ještě jeden jednodušší a dle našeho názoru velmi elegantní postup. Pomocí zmíněného programu aktivujeme implicitní, skrytý účet s administrátorskými právy se jménem „Administrator“, který po útoku opět deaktivujeme. Aktivovat účet „Administrator“ lze také úpravou jisté hodnoty

v registru nebo záměnou určitého procesu za příkazový řádek, který se spustí s administrátorskými právy ještě předtím, než se kdokoli přihlásí do systému. K tomu můžeme použít zmíněný Windows PE nebo máme-li instalační médium k dané verzi Windows, pak můžeme použít opravný mód, ve kterém po jeho načtení klávesovou zkratkou `shift+F10` spustíme příkazovou řádku a použijeme stejné postupy popsané v následující kapitole pro Windows PE.

- Jelikož se v Linuxu z předchozího kroku nacházíme ve standardní příkazové řádce, využijeme běžného příkazu pro kopírování „`cp`“.

Jen toto by většinou pro zahlazení stop nestačilo, záleží na tom, na jaký systém se útočí, respektive jak pečlivý správce ho spravuje a sleduje. Existuje mnoho možností co auditovat a následně upozorňovat na možné napadení. Například v systému Linux může útočník věnovat pozornost souboru `.bash_histori`, kde se ukládá historie příkazů. Může ho buďto smazat celý (což může být podezřelé), případně smazat jen záznamy o naší práci nebo jednodušeji na začátku ho zálohovat a na konci nahrát zpět.

Potřebujeme-li heslo rozluštit, situace je malinko zdlouhavější.

- Ve Windows si pro lámání hesla představíme programy Ophcrack a Cain.

  Pokud pracujeme pod administrátorskými právy v systému, který chceme napadnout (mohou nás zajímat hesla k jiným účtům, nebo jsme si výše zmíněným postupem aktivovali účet „Admnistrator“), pak zmíněné programy haše hesel sami načtou. Je na nás (nebo spíše na dané situaci), jestli rovnou provedeme lámání, nebo zda extrahované haše exportujeme do textové podoby a přeneseme je na jiný počítač, kde teprve provedeme lámání. Haše je nutné extrahovat pomocí uvedených programů, protože pokud systém běží, nelze soubor SAM (a ani jeho zálohu RegBack/SAM) přímo zkopírovat.

  Pokud administrátorský přístup nemáme, musíme si databázi hesel zkopírovat přes jiný systém, než který chceme napadnout. Tento krok je vlastně shodný s krokem, kde jsme soubory s hesly zálohovali. Soubory přeneseme na jiný počítač, kde máme nainstalovaný například Cain a teprve zde haše hesel extrahujeme a provedeme jejich lámání.

- V Linuxu máme výhodu, že není potřeba haše hesel z databáze extrahovat a máme je rovnou v textové podobě. Další výhodou je, že databáze jako taková není šifrována. Pro lámání představíme nejpoužívanější nástroj John the ripper a Hashcat.

  Pokud máme administrátorský přístup, můžeme soubory přímo přečíst a provést lámání, nebo soubory přenést na jiný systém.

  Pokud administrátorský přístup nemáme, kromě dvou možných metod získání přístupu do systému uvedených na začátku této kapitoly (spuštění systému v single user mode nebo přímé spuštění bin/bash), lze použít jakoukoli live distribuci Linux k nabootování a ke zkopírování souboru `/etc/shadow`.

7.3 Ochrana před uvedenými metodami

Ochranou ne zcela v pravém slova smyslu je už to, že známe uvedené slabiny. Největší nebezpečí tkví v tom, že útočník může mít k systému fyzický přístup. Pak má velmi účinné, rychlé a levné prostředky k napáchání škod.

Základním nástrojem ochrany našeho systému by tedy měla být ochrana zamezením přístupu. Někdy je to automatické (nečekáme-li od svých rodinných příslušníků útok na náš domácí počítač, pak k němu útočník nemá přístup, protože máme zavřené vchodové dveře bytu), jindy záměrně vynucené (do místnosti se servery je vpuštěna pouze ověřená osoba, u které je na recepci zaznamenaná její totožnost a účel návštěvy, ostatním je vstup zakázán, prostor je sledován kamerami, dveře můžou být vysoce bezpečnostní, v místnosti bude alarm a podobně). Bohužel toto nejde aplikovat vždy, všude a proti všem uživatelům.

Určitou ochranou ve smyslu zamezení přístupu může být to, že počítač samotný je uložen v zamčené skříňce, a uživatel tak nemá možnost do něj vložit CD či USB disk nebo alespoň má nějakým fyzickým způsobem zabráněno dostat se k základní desce. V tom případě má smysl zaheslovat nastavení, kde zakážeme bootování z jiného než systémového disku (v opačném případě by mu stačilo na základní desce toto nastavení resetovat). Ani toto ale nejde samozřejmě aplikovat vždy.

Ochranou (velmi efektivní) proti útokům při fyzickém přístupu je šifrování souborového systému. V systémech Linux nám toto umožňují například programy LUKS nebo dmcrypt,

ve Windows můžeme použít například TrueCrypt nebo pokud vlastníme Windows 7 v edicích Professional, Enterprise a Ultimate, nebo Windows 8 v edicích Pro a Enterprise, pak máme k dispozici Technologie EFS (Encrypting File System). Takto zaheslovaný disk je z jiného systému nečitelný a útočník nemůže „domovský“ operační systém obejít.

Ochranou proti úpravě programu odpovědného za zavedení systému GRUB (starší verze distribucí), respektive GRUB2 (od verzí Centos 7, Ubuntu 9.1, Fedora 16), jak jsme si uváděli u útoku změnou hesla v systému Linux, je jeho zaheslování. Toto ale stále útočníkovi nezabrání, aby tento zavaděč přehrál svým. Pak nám nezbývá jiná možnost, než šifrovat celý disk.

Problémem u šifrování souborových systémů a ještě výrazněji u celých disků je v tom, že kromě snížení výkonu máme velmi omezené možnosti obnovy dat při jejich rozbití. Pak je o to důležitější potřebná data pravidelně zálohovat.

8 Metody lámání hašů hesel

Existuje několik způsobů, jak k lámání hesla přistoupit. Jednotlivé způsoby si stručně představíme a zároveň doporučíme, jak se proti daným způsobům útoku bránit. Při lámání hesla nám může velmi pomoci znalost cíle. Užitečnými informacemi jsou například jazyk, jímž vlastník účtu mluví, jaké má v oblasti IT vzdělání (dá se očekávat kvalitní heslo?), zdali jsme ho třeba neviděli zadávat heslo, takže dovedeme odhadnout, jakou asi může mít délku, zda použil jen písmena, nebo i čísla a podobně. Pak lze efektivněji volit metodu pro lámání hesla. Popis jednotlivých metod je brán ve vztahu k lámání haše hesla.

8.1 Útok hrubou silou (Brute-Force Attack)

Jedná se o postupné zkoušení všech možných kombinací znaků zvolené abecedy, které hašujeme algoritmem, jímž si myslíme, že je cílové heslo hašováno. Většinou se začíná od minimálního počtu znaků, o němž víme, že systém na který útočíme povoluje a postupně přidáváme další, dokud nenarazíme na shodu nebo nevyprší námi stanovený čas, který chceme útoku hrubou silou věnovat. Výhodou tohoto útoku je, že za předpokladu zvolení dostatečně velké abecedy, vždy najdeme výsledek. Nevýhodou je pak to, že se tohoto výsledku při současném výkonu počítačových systémů nemusíme dožít.

Ochranou proti tomuto útoku je dlouhé heslo s využitím rozšířené abecedy (použití zvláštních znaků) a číslic a volbou algoritmu vytvářející dlouhý haš.

V následující tabulce jsou uvedeny odhady (měřeno programem Cain na Intel Core i5 2,6 GHz s 4 GB RAM) doby trvání vyzkoušení všech možností dané abecedy v závislosti na délce hesla pro NT haš. (Přibližně se provede 10 milionů porovnání za sekundu)

Z výsledků vidíme, že útok hrubou silou je použitelný pro slova do přibližně pěti znaků. Pokud selžou jiné metody a výsledek nepotřebujeme ihned, může být použitelný ještě přibližně na šesti až sedmi znaková hesla.

Tabulka 7 *Maximální doba trvání útoku hrubou silou na NT haš v závislosti na délce hesla a zvolené abecedě*

Délka hesla / Abeceda	1 až 4	5	6	7	8	9
[0-9]	ihned	ihned	ihned	ihned	< 1 min	1,5 min
[a-z]	ihned	ihned	< 1 min	13 min	6 hodin	6,5 dne
[0-9][a-z]	ihned	ihned	3,5 min	2,5 h	3,5 dnů	120 dnů
[0-9][a-z][A-Z]	ihned	1,5 min	1,5 h	4 dny	258 dnů	43 let
[0-9][a-z][A-Z] !@#%^&*()-_+=~	< 1 min	4 min	5,5 h	17 dnů	3,5 roku	270 let

8.2 Slovníkový útok

Vylepšení útoku hrubou silou je možné slovníkovou metodou. Místo generování všech možných kombinací se použije předgenerovaný slovník. Na internetu jich lze najít několik, pro různé jazyky či obory (představit si můžeme například slova z překladového slovníku či slovníku spisovné češtiny). Výhodné je, pokud víme něco o subjektu, na který útočíme. Můžeme tomu přizpůsobit volbu slovník. Výhodné je, pokud aplikace pro slovníkový útok umožňuje rozšiřující volby pro úpravu slov, například změnu malých a velkých písmen, doplňování čísel a podobně. Stačí nám pak jednodušší a menší slovník.

Ochrana proti slovníkovému útoku je asi jen jedna a to vyhnout se heslům obsahující celá slova, souslovím, či jinak odhadnutelným heslům (typu „heslo“, „passwd“, jméno uživatele, název ulice, a podobně).

Odhadnutí doby trvání útoku je velice složité, je to silně závislé na velikosti slovníku, jedná se ale o řádově rychlejší útok. Nevýhodou nicméně je, že slovník hledané heslo samozřejmě obsahovat nemusí.

8.3 Útok pomocí rainbow tables

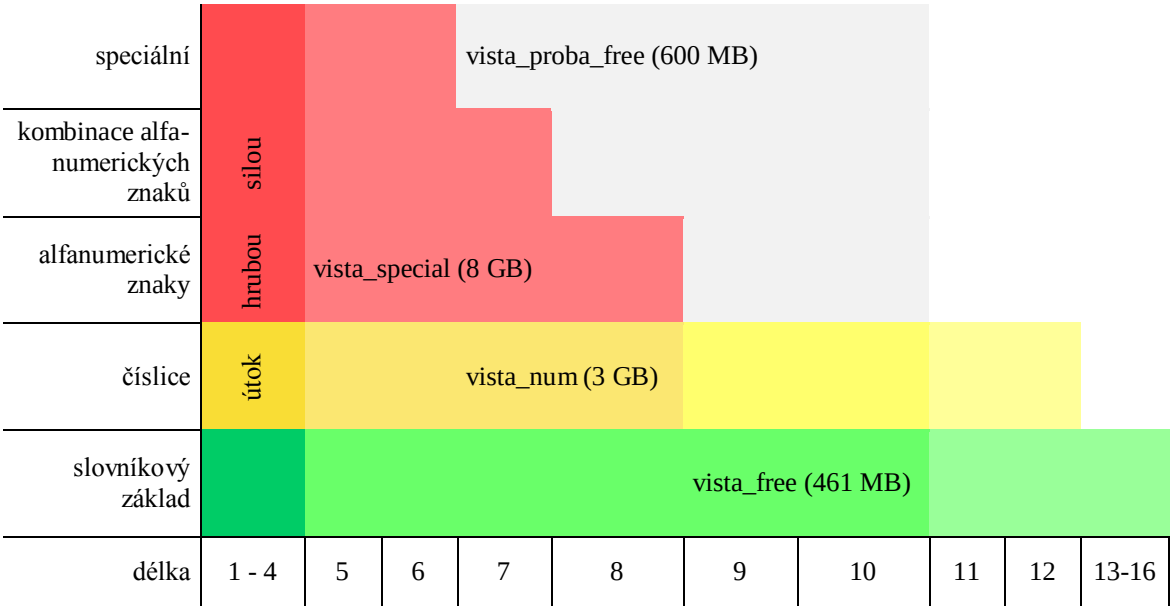
Tabulky vytvořené za účelem zjištění zahašovaných hesel. Zjednodušeně řečeno se jedná o tabulku (obecně může být více dílčích tabulek), která obsahuje již zahašované kombinace posloupnosti znaků. Tabulky by byly datově neúnosně velké, proto se ukládají v redukované podobě. Pro více informací bych odkázal na diplomovou práci kolegy Josefa Šenfelda na téma Rainbow tables a jejich využití k prolomení hašovací funkce [78].

Ochrana proti tomuto útoku je použití soli. Útoku nezamezí, nicméně útočník by musel vygenerovat unikátní tabulky pro konkrétní zasolovací řetězec.

Jedná se o vcelku rychlou metodu, opět silně závislou na její velikosti, a tím na oblasti zvolené abecedy a délce slov z ní vytvořené. Při jejím používání je rychlost výpočtu silně ovlivněna tím, zdali se celá nachází v operační paměti, nebo ne. Pokud ano, je metoda velmi rychlá, proto je vhodné pro větší tabulky vlastnit stroj s dostatkem operační paměti.

Na následují grafice vidíme příklad velikosti tabulek a jimi pokryté oblasti pro NT haš, které jsou poskytovány k programu Ophcrack (ten si představíme později).

Tabulka 8 *Příklad velikosti rainbow tables v závislosti na délce hesla a zvolené sadě znaků. Přeloženo z [79].*



8.4 Útok s využitím kryptoanalýzy

Kryptografie je nauka o metodách šifrování, zatímco kryptoanalýza hledá metody, jak šifry luštit. Útok opírající se o poznatky z kryptoanalýzy se zaměřuje na známé slabiny použitých algoritmů. Jedná se o takzvané prolomené algoritmy, někdy také nazývané jako kompromitované. Tyto slabiny se pak pokusíme využít pro rychlé získání hesla.

Kryptoanalytické metody lze rozdělit zhruba na tři typy útoků:

- útok se známou šifrou, ale neznámým původním textem (Ciphertext Only Atack) – rozborem pravidelností v šifrovém textu se snažíme uhodnout původní text,
- útok se známým původním textem i šifrovým textem, ale neznámým klíčem a šifrovacím algoritmem (Known Plaintext Atack) – rozborem se snažíme odvodit klíč a šifrovací algoritmus,
- útok s vybraným otevřeným textem a přístupným šifrovačem (Chosen Plaintext Atack) – vstupní text vložíme do šifrovače a získáme jeho šifru. Vhodným výběrem vstupního textu mohou být odhalena slabá místa šifrovače.

[80]

Často jsou kryptografické algoritmy známé, takže pomocí matematických metod můžeme hledat jejich slabiny.

Další variantou jsou útoky postranními kanály, které se pokoušejí zneužít informace získané přímo z fyzické implementace. Sledujeme, zda se ve výstupu neobjevuje nějaké nenáhodné chování. Často mají za úkol jen zjistit, jaký typ algoritmu je používán. Využít se dají například časové nebo odběrové analýzy (měří se spotřeba energie či vyzářená energie ze zařízení).

[81] [82]

Ochrana spočívá v použití algoritmů, které doposud nebyly kompromitované. Je tak nutné sledovat všechny nové informace z tohoto odvětví.

Více se tímto tématem zabývá například práce [83].

9 Aplikace pro ověření různých metod útoku

V této části práce představím programy, které jsem používal během procesu, jehož cílem bylo získat přístup do operačního systému, aniž bych znal přístupové údaje. Pokud v popisu není uvedeno, že by daný program na určité verzi operačního systému nefungoval, pak s funkčností nebyly problémy. Zkoumanými operačními systémy byly Windows 7, Windows 8, Windows server 2012, Linux v distribucích CentOS 7, Fedora 22 a Ubuntu 14. Operační systémy měly vždy nainstalovány veškeré dostupné aktualizace a nebyl použit žádný bezpečnostní software třetích stran.

Pro testování jsem měl k dispozici dva notebooky:

1) Lenovo ThinkPad X230, procesor Intel Core i5-3230M 2,6 GHz, 4 GB operační paměti,
2) Acer Aspire 1410, procesor Intel Celeron 743 1,3 GHz, 3GB operační paměti.

Ukázalo se, že mezi verzemi Windows je jen jedna možná odlišnost. Používáme-li online účet Microsoft, pak nefunguje smazání hesla a to ani v případě, že počítač k internetu není připojen. Jak je ale níže popsáno, dá se tento problém elegantně obejít aktivací skrytého účtu „Administrator“ a skrze něj získat přístup do systému.

Během testování se ukázalo, že pokud byl napadaný operační systém v hibernovaném stavu, pak systémy nabootované z USB flash paměti nemohly přistupovat k jeho souborům.

Přestože máme v běžícím operačním systému Windows administrátorská práva, nelze soubor SAM ani SYSTEM kopírovat, je proti tomu chráněn systémovým procesem. Nicméně některé programy umí potřebné informace extrahovat.

Microsoft v rámci zabezpečení SAM souboru zavedl jeho šifrování. Nicméně příliš účinný tento krok nebyl, protože informace potřebné k sestavení dešifrovacího klíče jsou obsaženy v souboru SYSTEM a současné programy pro útoky na Windows hesla se s tímto úspěšně vypořádaly. Jediný důsledek, který z toho vyplynul je ten, že při kopírování souboru SAM je potřeba společně s ním také zkopírovat soubor SYSTEM.

Přístup k souborům `etc/passwd` a `etc/shadow` je u všech uvedených distribucí stejný. Jelikož se jedná o textové a nešifrované soubory, nemusíme hledat speciální nástroje, které by

nám soubory dešifrovaly a záznamy extrahovaly, jako tomu bylo u Windows. Odpadá nám tak část práce. Jelikož Linux spoléhá na standartní hašovací algoritmy, hledáme nástroje vytvořené pro prolomení hašů standartních hašovacích algoritmů. Jak zjistit, jaký algoritmus je použit, jsme si vysvětlovali v části věnované bezpečnosti operačního systému Linux, konkrétně v kapitole 5.6 Proces ověření hesla.

Všechny námi zkoumané distribuce v základu používají hašovací algoritmus SHA-512, což je v současné době nejsilnější běžně dostupný algoritmus. Jeho síla se potvrdila i v praktickém testu, kdy se nám nepodařilo heslo pomocí uvedených aplikací nepodařilo prolomit.

Pro útoky na Windows jsme použili následující programy:

- Offline Windows Password & Registry Editor – dle našeho názoru nejlepší řešení, rychlé a jednoduché resetování hesla i s povolením skrytého účtu „Administrator“,
- Cain – extrahuje záznamy ze SAM databáze, umožňuje volbu metody lámání (podporuje útok hrubou silou, slovníkový útok i útok pomocí rainbow tables),
- Ophcrack – výhodné při lámání více hesel najednou s využitím rainbow tables,
- WinPE – lze využít k získání souborů SAM a SYSTÉM,
- Kali Linux – lze využít k získání souborů SAM a SYSTEM, obsahuje nástroje Ophcrack a chntpw (odpovídá Offline Windows Password & Registry Editor) a další,
- OnlineHashCrack – jednoduchá online aplikace pro lámání hesla.

Pro útoky na Linux jsme použili následující programy:

- Úprava GRUB,
- Kali Linux,
- John the Ripper,
- Hashcat .

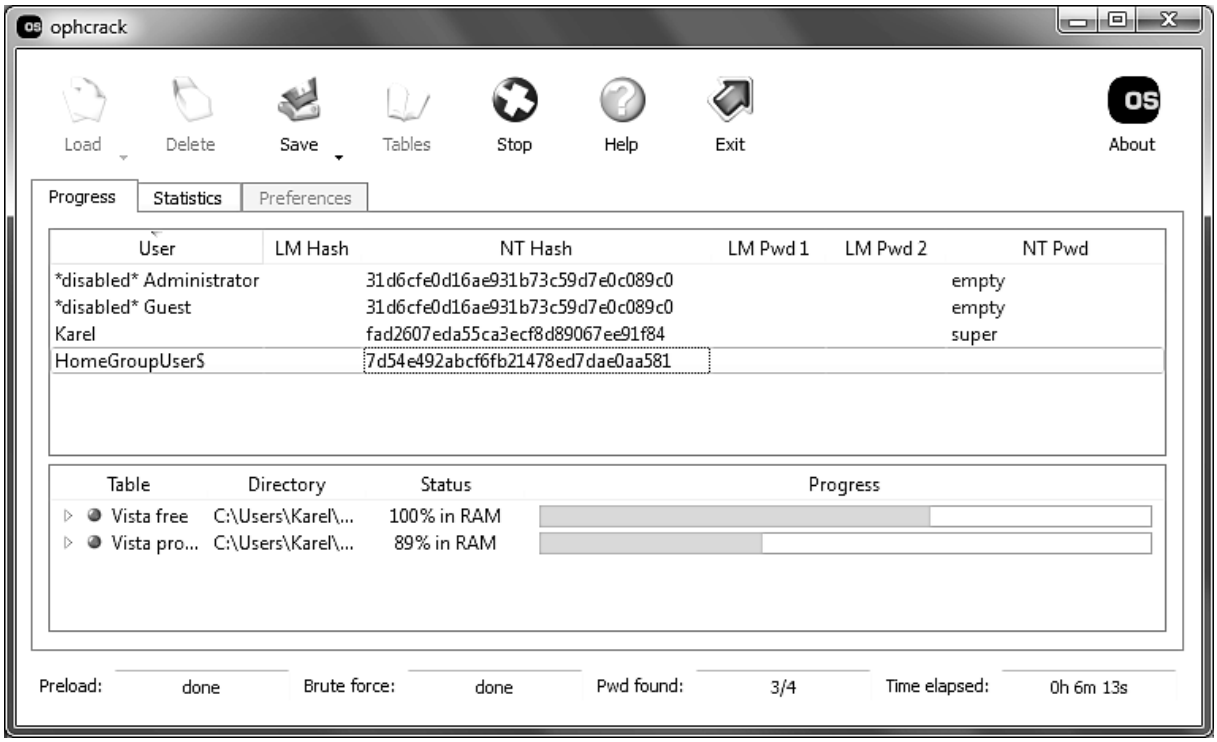
Omluvte, prosím, kvalitu některých obrázků. Jelikož testování probíhalo na skutečném počítači (ne virtualizovaně), tak jsme nemohli udělat snímky obrazovky softwarově, ale museli jsme použít fotoaparát.

9.1 Nástroje použité pro získání přístupu k operačnímu systému Windows

V této části se budeme věnovat jednotlivým nástrojům, které jsme používali během procesu získání přístupu do operačního systému Windows. U každého uvedeme jeho stručnou charakteristiku, jak lze nástroj získat a jak ho použít.

9.1.1 Ophcrack

Program pro lámání LM a NT hašů, pro krátká hesla (do 4 znaků) používá útok hrubou silou, pro delší hesla pak využívá rainbow tables. V případě, že je program spuštěn ve Windows a máme-li administrátorská práva, umí přímo extrahovat záznamy s hesly ze SAM databáze.



Obrázek 20 *Ukázka programu Ophcrack (snímek obrazovky)*

Získání programu a instalace

Program dostupný pro Windows, pro Linux a i jako Live CD.

Pro Windows lze program stáhnout z oficiálních webových stránek http://ophcrack.sourceforge.net/. Jedná se o klasickou instalaci, jak ji z Windows známe. Distribuce Kain Linux program obsahuje v základu, do jiných distribucí ho lze doinstalovat standartní cestou.

Pro použití tohoto programu je nezbytné z uvedených stránek stáhnout soubory potřebné pro útok založený na rainbow tables. Tabulky určené pro lámání LM haše jsou označeny jako „XP“ a pro NT haše jsou označeny jako „Vista“. Základní jsou k dispozici zdarma (zabírají asi 12 GB místa na disku), rozšířené (zabírají asi 2,3 TB místa na disku) vyjdou přibližně za 1000$. U souborů je vždy zobrazeno přehledné schéma, které zobrazuje jaký velký prostor pokrývají. Schéma pro soubory s NT haši dostupné zdarma bylo uvedeno v kapitole 8.3 Útok pomocí Rainbow Tables.

Použití

Ovládání je jednoduché a intuitivní. V nabídce „Load“ vybereme zdroj hašů. Načtené záznamy s haši program umožňuje uložit pro pozdější lámání či pro lámání pomocí jinémho programu. V případě, že proces lámání hesel již proběhl, případná nalezená hesla při ukládání připojí k danému záznamu.

Dalším nutným krokem je načtení souborů s rainbow tables, které jsme si předem stáhli. Načtení provedeme v dialogu vyvolaném kliknutím na ikonu „Tables“. Po úspěšném nahrání již můžeme spustit lámání. Program ve spodní části přehledně informuje o průběhu lámání a o tom, kolik procent tabulky se podařilo nahrát do operační paměti. Toto je důležité pro rychlost lámání. Nalezená hesla se zobrazí ve sloupečku LM Pwd respektive NT Pwd.

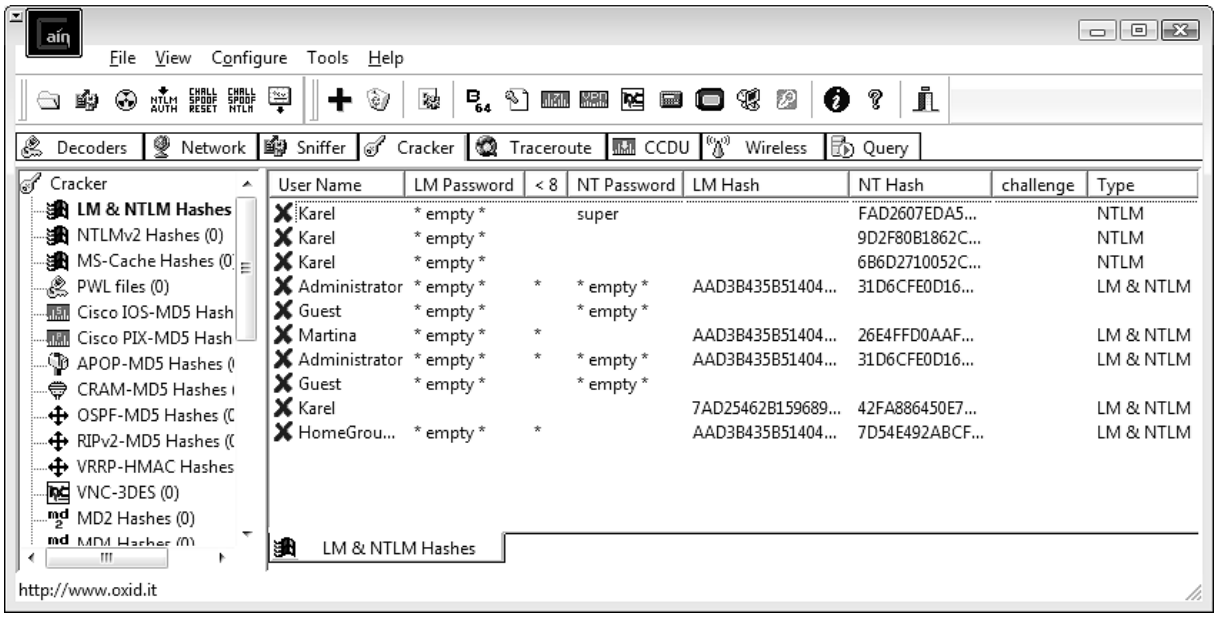
Výhodou programu je, že láme všechna vložená hesla současně.

9.1.2 Cain

Jedná se o velice obsáhlý nástroj na obnovu různých hesel pro systémy Windows. Nás však zajímá to, že umí extrahovat LM a NT haše ze SAM databáze. Umí načíst jak přenesený SAM soubor, tak i z aktuálně běžícího systému Windows (za předpokladu, že máme administrátorská hesla). Druhá, pro náš účel podstatná funkčnost, je podpora útoků na haše hesel pomocí hrubé síly, slovníkového útoku i útoku využívající rainbow tables. U prvních dvou zmíněných útoků poskytuje užitečné nastavení, které může hledání podstatně zefektivnit.

Získání programu a instalace

Program lze stáhnout z oficiálních webových stránek http://www.oxid.it/cain.html. Jedná se o klasickou instalaci, jak ji z Windows známe.



Obrázek 21 *Ukázka programu Cain (snímek obrazovky)*

Použití

Nejprve potřebujeme překliknout na záložku „Cracker“ a v levé části zvolit první položku „LM & NTLM Hashes“. Následně v hlavním prostoru aplikace klikneme pravým tlačítkem a zvolíme možnost „Add to list“. Máme na výběr tři možnosti. Extrahovat haše z lokálního systému (máme-li administrátorská práva), import hašů z textového souboru a konečně import hašů z dříve zkopírované SAM databáze. V tomto případě potřebujeme nejen soubor SAM, ale také soubor SYSTEM. Windows totiž soubor SAM šifrují kvůli snaze zabránit jeho přečtení, ale klíč je extrahovatelný právě ze souboru SYSTEM. Proto se souborem SAM vždy kopírujeme i soubor SYSTEM.

Nyní můžeme zvolit záznam, jenž chceme podrobit útoku. Klikneme na něj pravým tlačítkem a vybereme jednu z prvních třech položek, podle toho, jaký útok chceme provést. V podmenu pak vždy budeme volit „LM hashes“, respektive „NTLM hashes“. První položka je „Dictionary atack“, tedy slovníkový útok. V otevřeném formuláři musíme nejprve vložit slova námi vybraného slovníku (ve formátu txt). Dále si zvolíme, jak chceme slova automaticky upravovat. Je zde několik možností, například otočení písmen ve slově, doplňování slov čísly, převod malých písmen na velká a podobně. Nyní stačí kliknout na tlačítko „start“. Druhou možností je „Brute-force attack“, tedy útok hrubou silou. Zde nejprve vybereme používanou abecedu. Buďto zvolíme jednu z předdefinovaných, nebo si můžeme vyplnit vlastní. Dále volíme mini-

mální a maximální délku hesla. Nyní stačí opět jen kliknout na „start“ a počkat, zdali se výsledek dostaví. Formulář zobrazuje i odhadovaný čas, jaký by byl potřeba k vyzkoušení všech možných kombinací. Poslední možností útoku je „cryptanalysis attack“, ten využívá rainbow tables. Můžeme zde použít stejných souborů, jaké používá program Ophcrack. Zde musíme pouze vybrat umístění souboru s rainbow tables a kliknout na „start“.

9.1.3 Program Offline Windows Password & Registry Editor

Jedná se o jakousi minidistribuci Linuxu určenou pro procházení a editaci registrů Windows a pro práci s existujícími uživatelskými účty – umí smazat heslo u zvoleného účtu, přidávat a odebírat uživatele ze skupin, aktivovat zakázaný účet a běžný účet převést na administrátorský. Tento proces nám může zabrat pouze jednu minutu!

Testována byla verze programu 140201 určtená pro USB falsh disk, dále je dostupná verze také pro instalaci na CD. Velikost programu je 18MB. Dle autora jsou podporovány operační systémy NT 3.51, NT 4, Windows 2000, Windows XP, Windows 2003 Server, Vista, Windows 7, Server 2008, Windows 8, Windows 8.1, Server 2012.

Obrázek 22 *Ukázka programu Offline Windows Password & Recavery Editor (fotografie obrazovky)*

Získání programu a instalace

Program je ke stažení je zde: http://pogostick.net/~pnh/ntpasswd/

Abychom vytvořili bootovací USB flash disk, je postup následující:

Obsah staženého archivu nakopírovat na flash disk (nepoužívat žádné podsložky). Pokud na paměti máme uložené i jiné soubory, tak to nevadí, nebudou smazány.

Spustíme příkazovou řádku Windows (cmd.exe) jako administrátor a zadáme příkaz

```
X:syslinux.exe -ma X:
```

kde za „X“ nahradíme písmeno, které má flash paměť v systému Windows přiřazené. Tímto máme instalaci hotovou.

Použití

Po nabootování a spuštění programu nás navádí intuitivní průvodce, nemusíme tedy umět ovládat systém Linux. Program umožňuje volit různé pevné disky a následně i případně zvolit oddíl, kde se nachází operační systém, na který útočíme. Kromě editace registrů umožňuje zobrazit existující uživatelské účty. Pro tyto účty můžeme provádět mimo jiné následující operace:

1) smazat heslo,
2) aktivovat účet,
3) přiřadit administrátorská práva.

Tyto operace jsou provedeny okamžitě a pokud jsme s naším nastavením spokojeni, změny uložíme a restartujeme počítač.

Tímto programem se mi nepodařilo odebrat heslo u takzvaného účtu Microsoft ve Windows 8. Ačkoli změny program provedl, při spuštění Windows přihlášení bez hesla nefungovalo. Pokud na počítači není jiný místní uživatelský účet (nezáleží, zda je či není administrátorský – pokud není, administrátorský z něj program umí vytvořit, jediné na co nás upozorní, že tato operace může být zalogována a správce systému tak může útok rozpoznat), pravděpodobně tam bude zakázaný místní účet „administrator“. Tento účet je implicitně vytvořen při instalaci Windows a standardně je zakázán. Je možné ho povolit a zase zakázat, nicméně jsem nenašel možnost, jak ho ze systému smazat v rámci ochrany proti uvedenému postupu.

Asi nejjednodušší cestou bude programem účet „administrator“ povolit, smazat případné heslo, uložit změny, restartovat počítač a jen počkat, než se Windows spustí. Účet „administrator“ nemá heslo, takže ho pravděpodobně systém po svém startu automaticky přihlásí.

Rychlost programu jsem měřil na dvou počítačích:

1) Lenovo ThinkPad X230, procesor Intel Core i5-3230M 2,6 GHz, 4 GB operační paměti. Operační systém Windows 7 Professional SP1 s nainstalovanými všemi dostupnými aktualizacemi.
2) Acer Aspire 1410, procesor Intel Celeron 743 1,3 GHz, 3GB operační paměti. Operační systém Windows 8.1 Pro s nainstalovanými všemi dostupnými aktualizacemi.

Na prvním počítači trval start programu 27 sekund a na druhém 33 sekund. Vidíme tedy, že ačkoli je výkon počítačů dosti rozdílný, nemusíme mít obavu, že by program na starých zařízeních potřeboval zásadně více času. Samotné nastavení (povolení účtu, smazání hesla, uložení změn a vyvolání restartu) trvalo asi 32 sekund. Celý tento proces nám tedy zabere pouze přibližně jednu minutu!

Chceme-li program použít pro zálohování původního souboru před úpravami, případně pro vytvoření kopie SAM databáze, počítač po ukončení průvodce nerestartujeme a využijeme k tomu klasickou příkazovou řádku Linux, kterou nyní máme k dispozici.

9.1.4 Windows Preinstallation Environment

Používá se zkrácený název WinPE, přičemž jde o takzvanou Live verzi operačního systému Windows, která je schopná běhu z USB či CD. Od verze 2.0, jež je založena na Windows Vista, je veřejně k dispozici. Microsoft povolil používání WinPE pro diagnostiku či obnovu klasické verze Windows.

Získání programu a instalace

WinPE si můžeme stáhnout v rámci balíčku Windows Assessment and Deployment Kit (Windows ADK). Nejlépe je zadat uvedený balíček do internetového vyhledávače a vybrat požadovanou verzi. My jsme si stáhli verzi založenou na Windows 8.1, tedy Windows PE 5.1 z prosince roku 2013.

Jakmile balíček stáhneme a nainstalujeme, potřebujeme vytvořit bootovatelný disk. My si vytvoříme USB flash disk. Nejprve spustíme nástroj Deployment and Imaging Tools (spustit jako správce), do příkazového řádku zadáme příkaz

```
Copype x86 C:\WinPE
```

Kde „x86“ vyjadřuje architekturu počítače, pro který WinPE vytváříme a „C:\WinPe“ je cesta, kam se zkopírují instalační soubory. Nyní připojíme USB flash disk a zjistíme, pod jakým písmenkem se připojil do systému. V našem případě „D“. Použijeme příkaz

```
MakeWinPEMedia /UFD C:\WinPE D:
```

Následně se program dotáže, zda může USB disk naformátovat. Toto potvrdíme a počkáme, než se disk vytvoří (nakopírují se na něj potřebné soubory). Tímto máme USB flash disk s WinPE připravený k použití.

Použití

Nejprve si ukážeme, jak WinPE využít k vytvoření kopie souborů SAM a SYSTEM.

Po spuštění operačního systému WinPE se zobrazí standartní příkazová řádka operačního systému Windows. Ke zkopírování uvedených souborů využijeme následující příkazy:

- „E:“ pro změnu aktivního disku (v našem případě je disk původního systému pod písmenem E),
- „cd windows\system32\config“ pro přepnutí do složky, ze které budeme kopírovat soubory,
- „copy SAM D:“ a
- „copy SYSTEM D:“ pro zkopírování souborů (v našem případě je USB disk s WinPE připojen pod písmenem D).

Tímto práce s WinPE končí, můžeme zadat příkaz „exit“, jenž vyvolá restartu počítače. Na flash paměti nyní máme soubory, které potřebujeme k lámání hesla.

WinPE můžeme také použít k aktivování lokálního účtu „Administrator“. Představíme dvě metody, jak toho dosáhnout. První spočívá v úpravě registru. Zde je postup následující:

- Příkazem „regedit“ spustíme editor registru.
- Klikneme na položku „HKEY_LOCAL_MACHINE“.
- V menu „Soubor“ vybereme „Načíst podregistr…“.

- Vyhledáme soubor SAM lokálního operačního systému a necháme ho načíst. Program se nás zeptá, pod jakým názvem chceme podregistr připojit. Zvolíme například „SAM_“.
- Vyhledáme položku HKEY_LOCAL_MACHINE\SAM_\SAM\Domains\Account\Users\000001F4.
- V pravém okně klikneme pravým tlačítkem na položku „F“ a vybereme „Změnit“. Na řádku 0038 bude v prvním hloupci uvedena hodnota 11. To znamená, že účet „Administrator“ je zakázaný. Změníme-li hodnotu na 10, pak se tento účet aktivuje. Klepneme na tlačítko „OK“ a restartujeme počítač.

Druhý postup jak aktivovat účet „Administrator“ pomocí WinPE je následující:

- Vyhledáme si pod jakým písmenem je připojen systémový disk, v našem případě je to „E:“.
- Zkopírujeme program „sethc.exe“ pro pozdější navrácení systému do původního stavu: „copy e:\windows\system32\sethc.exe e:\“.
- Nahradíme „sethc.exe“ za „cmd.exe“:
  „copy /y e:\windows\system32\cmd.exe
  e:\windows\system32\sethc.exe“.
- Restartujeme systém. Nyní, jakmile se objeví přihlašovací obrazovka, stiskneme na klávesnici pětkrát „shift“ a tím se nám spustí příkazová řádka.
- Zadáme příkaz: „net user administrator /active:yes“.

Tímto jsme aktivovali lokální účet „Administrator“. V tomto kroku můžeme místo příkazů pro aktivování skrytého účtu požít i příkazy

- „net user“ pro výpis místních uživatlů,
- „net user UŽIVATEL“ pro vypsání informací o uživateli,
- „net user UŽIVATEL HESLO“ pro změnu hesla vybraného uživatele, například „net user administrator noveHeslo“.

Na WinPe je založen i jeden komerční projekt Acrive@ Boot Disk Suite (aktuální verze 10 je založena na WinPe 5.1, tedy na Windows 8.1). Balík obsahuje několik nástrojů užitečných k obnově dat, bezpečnému mazání dat, práci s diskovými oddíly a mimo dalších obsahuje také nástroj pro resetování hesla. Více se dozvíte zde: http://www.softfully.com/system-tools/back-up-recovery/active-boot-disk-suite/.

9.1.5 Kali Linux

Tato distribuce operačního systému Linux je plnohodnotným systémem a je zaměřena na penetrační testování. Za tímto účelem má v sobě předinstalováno mnoho nástrojů, mimo jiné pro offline útoky na hesla. Některé nástroje zde obsažené jsme již popsali. Kali Linux je nástupcem distribuce BackTrak, která již není dále vyvíjena.

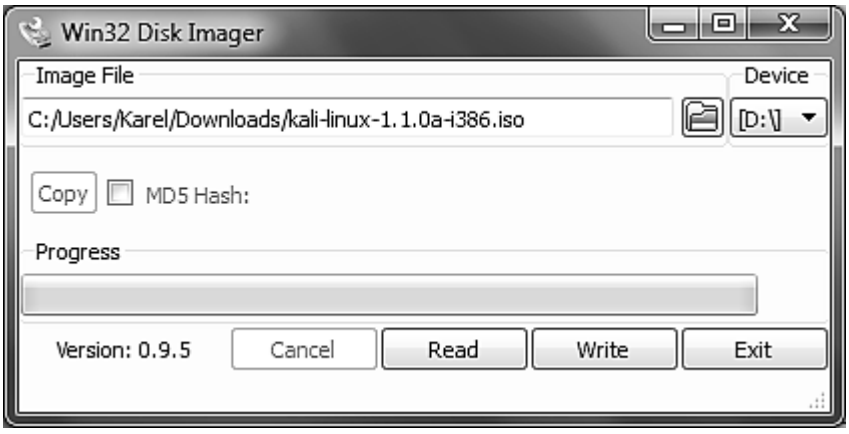
Výhodou Kali distribuce je, že po spuštění její Live verze je automaticky připojen disk, na kterém jsou nainstalovány Windows. Je zde funkční správce oken a i uživatelé, kteří s Linuxem nemají zkušenosti, se dokáží zorientovat.

Obrázek 23 *Ukáza distribuce Kali Linux (fotografie obrazovky)*

Získání programu a instalace

Získání a instalace systému na USB flash disk je o něco jednodušší, než tomu bylo u WinPE, kde cesta byla přece jen trochu krkolomná. Nejprve si stáhneme Kali Linux z oficiálních stránek https://www.kali.org/downloads/ (my jme použili Kali Linux 32 bit 1.1.0a) a zde doporučovaný program pro vytvoření bootovatelného USB disku Win32 Disk Imager stáhneme zde http://sourceforge.net/projects/win32diskimager/.

Po spuštění Win32 Disk Imager se zobrazí jednoduché prostředí, kde stačí vybrat stažený soubor s Kali Linux, vybrat na jaké zařízení chceme operační systém nahrát a kliknout na Write.



Obrázek 24 *Ukázka programu Win32 Disk Imager (snímek obrazovky)*

Po zkopírování souborů máme USB falsh disk s Kali Linux připraven k použití.

Pokud pracujeme v operačním systému Linux, je postup obdobný, jen pro vytvoření bootovacího média použijeme jiný nástro. Kali Linux doporučuje využití příkazu „dd“. Podrobný návod je uveden zde [84].

Chystáme-li se použít Ophcrack, měli bychom si ještě předem stáhnout a na USB flash paměť uložit soubory s rainbow tables, ty v distribuci vzhledem ke své velikosti zahrnuty nejsou. Návod ke stažení je uveden u popisu programu Ophcrack.

Použití

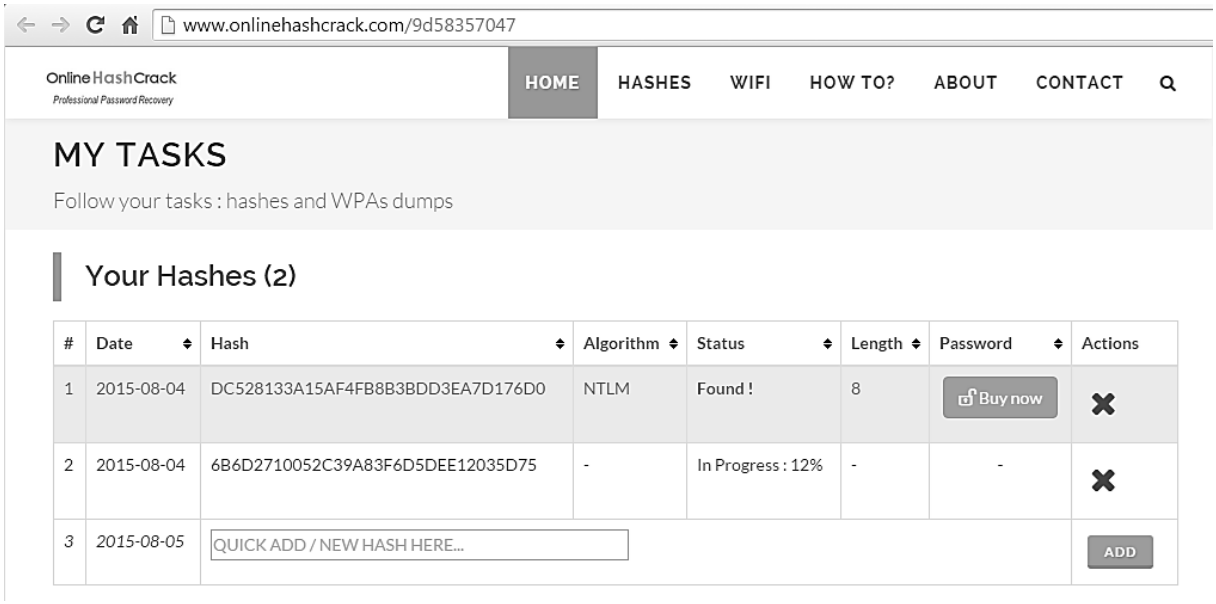
Kali Linux je asi nejuniverzálnější nástroj. Použijeme ho jak k získání Windows souborů SAM a SYSTÉM, tak i Linuxových souborů passwd a shadow. Pomocí předinstalovaných programů můžeme rovnou přejít k lámání hesel.

Jelikož se jedná o plnohodnotný operační systém, máme zde spuštěn správce oken. Mezi zařízeními se zobrazují všechny připojené disky (včetně těch s operačním systémem Windows), a stačí nám tedy vědět, kde soubory hledat a kam je chceme zkopírovat.

Pokud chceme lámat hesla některým z programů běžících pod Windows, nejjednodušší cestou je zřejmě připojení druhého USB disku se souborovým systémem FAT31 nebo NTFS, na který soubory zkopírujeme (soubory samozřejmě můžeme kopírovat i na flash disk s Kali Linux, ten nám ale Windows nenačtou a budou jej chtít naformátovat).

9.1.6 Online Hash Crack

Jedná se o webovou službu, která podporuje rozluštění mimo jiných také LM a NT haše. Stačí haš vložit do formuláře a odeslat. Na svých stránkách uvádějí, že doba trvání luštění je od několika sekund až po 4 dny a informaci o dokončení pošlou na email. Zdarma je luštění hesel kratších než osm znaků, delší jsou za poplatek (cena se pohybuje od 3 do 5 €).



Obrázek 25 *Ukázka webové aplikace OnlineHashCrack.com [85]*

Službu jsem testoval dvěma haši, které se mi nepodařilo jinými cestami prolomit. První bylo osmiznakové heslo, které bylo rozluštěné za 11 hodin a 20 minut. Druhé, desetiznakové, bylo rozluštěné za 5 dnů, 23 hodin a 35 minut, což tedy vyvrací jejich informaci, že je heslo rozluštěné do čtyř dnů.

9.2 Nástroje použité pro získání přístupu k operačnímu systému Linux

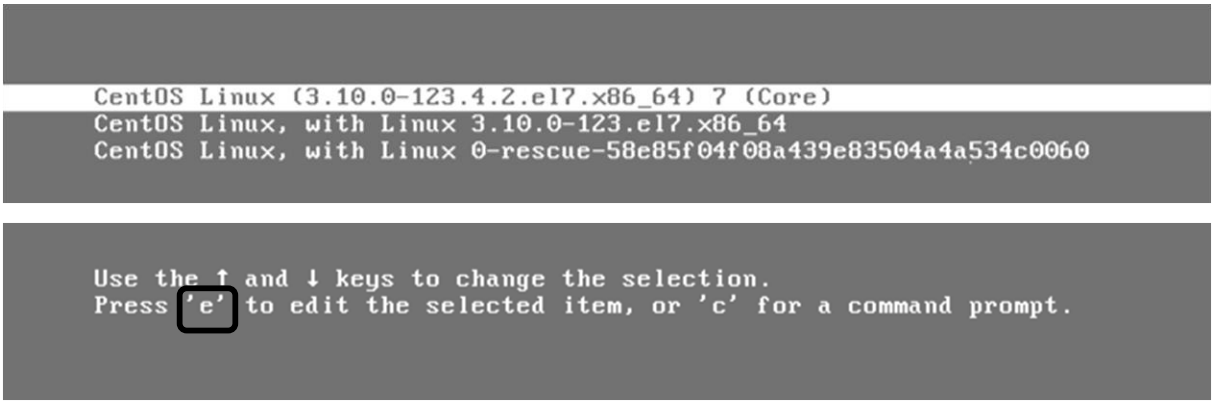
Stejně jako v předchozí části se i zde budeme věnovat jednotlivým nástrojům, které jsme používali během procesu získání přístupu, tentokráte ale do operačního systému Linux. Opět pro každý nástroj uvedeme jeho stručnou charakteristiku, jak ho lze získat a použít.

9.2.1 Modifikace zavaděče systému GRUB2

V tomto případě se nejedná o útok s využitím nějaké aplikace. Jde o metodu, která při vhodné modifikaci zavaděče operačního systému změní jeho způsob spuštění. Místo toho, aby naběhl standartní režim (většinou s nějakým okenním systémem), vyvolá se spuštění operačního systému v takzvaném single user mode, ve kterém je přístupná konsola s rootovskými právy bez jakéhokoli přihlášení. Místo single user mode lze nastavit, aby se místo procesu init (zajišťující spuštění systému ve zvolené úrovni běhu) spustil pouze bin/bash. Postup pro resetování hesla je ale shodný. Jedná se o základní a známý způsob, jak se dostat do operačního systému Linux.

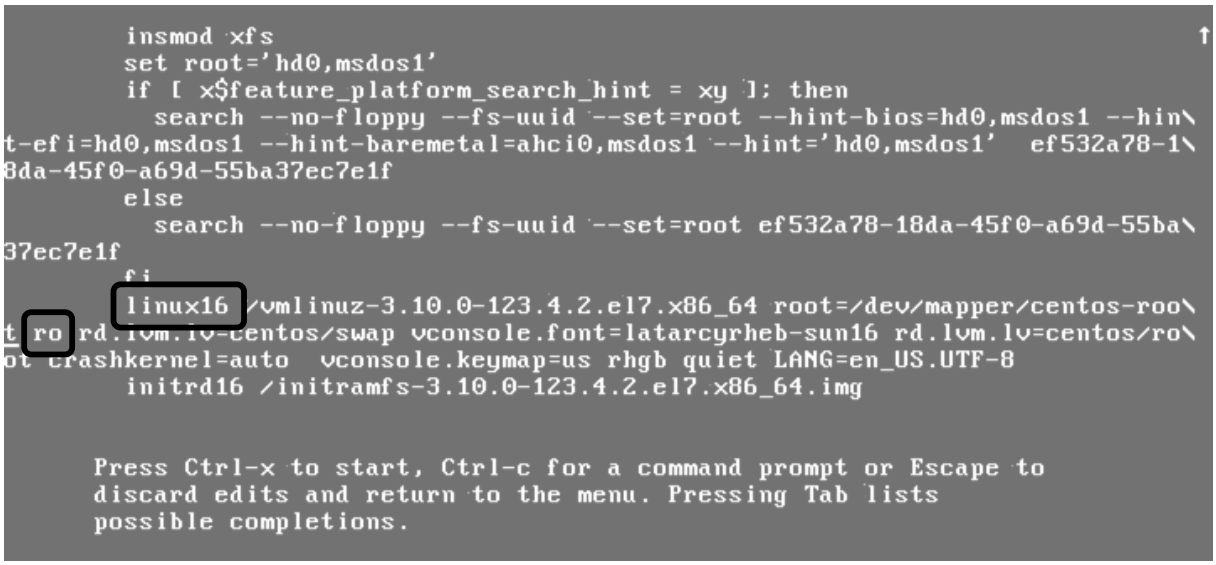
Z důvodu lepší kvality ilustrací jsme si dovolili použít snímky obrazovky uveřejněné na internetu, namísto fotografií obrazovky pořízených fotoaparátem. Níže uvedený postup platí pro Ubuntu 14, Fedora 22, CentOS 7 a další.

Nejprve se potřebujeme dostat na obrazovku zavaděče systému, která se zobrazí krátce po zapnutí počítače. Pokud se nezobrazí automaticky, vyvoláme ji standardně klávesou „ESC“. Jedná se o takzvané boot menu, kde bývá několik implicitních variant pro pokračování startu systému. Necháme zvýrazněnou základní položku a stiskneme klávesu „e“, která vyvolá její editaci.



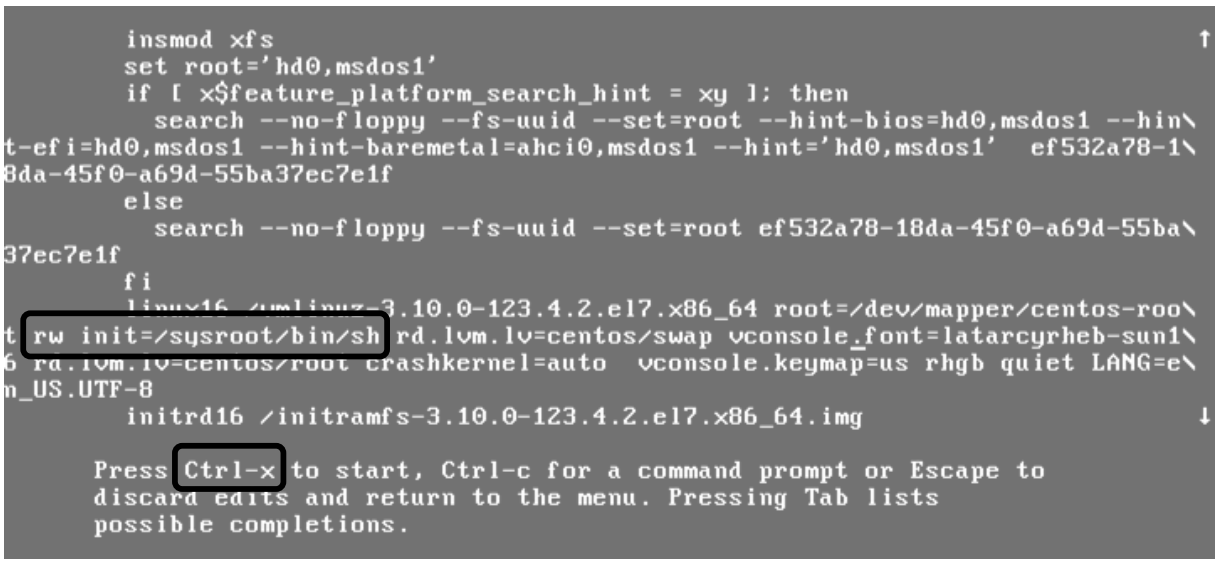
Obrázek 26 *Zavaděč GRUB2 [86]*

Nalezneme řádek, který začíná textem „`linux`“.



Obrázek 27 *Zavaděč GRUB2 – editace [86]*

Na tomto řádku zaměníme řetězec „`ro`“ (read only) za „`rw`“ (read write) a připíšeme ještě „`init=/sysroot/bin/sh`“ pro CentOS a Fedoru, „`init=/bin/bash`“ pro Ubuntu.



Obrázek 28 *Zavaděč GRUB2 – výsledek editace [86]*

Stiskneme klávesovou zkratku „`CTRL+X`“ a počkáme, než se systém spustí v single user mode. Poté zadáme příkaz „`chroot /sysroot`“ (v Ubuntu není třeba).

Nyní použijeme příkaz „`passwd`“ pro nastavení hesla účtu root, nebo příkaz „`passwd jmeno`“ pro nastavení nového hesla konkrétního uživatelského účtu. Staré heslo ke změně vyžadováno není, jedná se tak opravdu o nastavení hesla a ne o jeho změnu. Jestli-že je v systému používané řízení přístupu SELinux, zadáme příkaz „`touch /.autorelabel`“ pro aktualizaci jeho informací. Tímto máme hotovo a stačí restartovat systém. Ten naběhne běžným způsobem, modifikace, které jsme provedli na začátku v GRUB2, se použili jen pro jeden start sytému, neuložily se nastálo.

Uvedený postup je většinou uveden přímo v manuálových stránkách konkrétní distribuce. Je-li tedy postup v jistých distribucích v něčem specifický, nemělo by to představovat problém.

9.2.2 John the Ripper

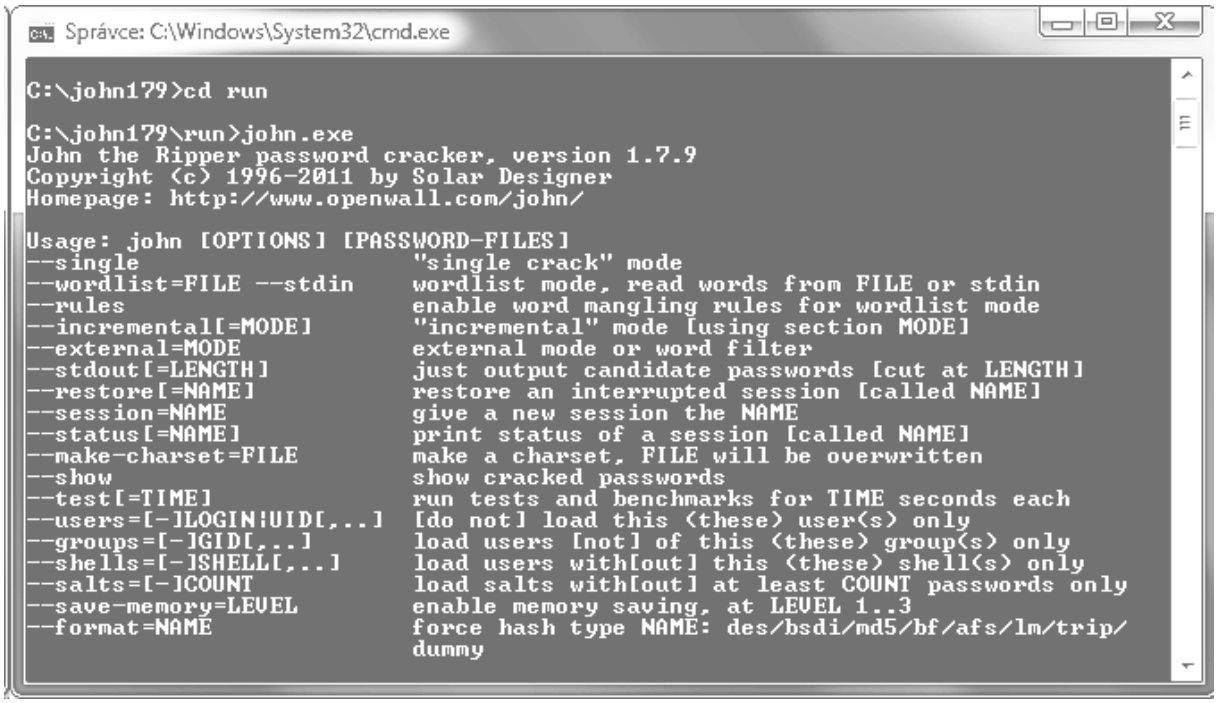
Pravděpodobně nejpoužívanější nástroj pro prolamování hašů hesel operačního systému Linux. Dále umí lámat NT haše (Windows) a další. Je k dispozici jak pro Linux, tak pro Windows. Podporuje jak útok hrubou silou (Incremental mode), tak slovníkový útok (Wordlist mode). Obsahuje ještě jeden speciální mód "Single crack". Právě tento mód představuje podle tvůrců programu jeho hlavní poslání – odhalit slabá hesla v operačním systému Linux. Tímto módem bychom měli začít, je nejrychlejší. Zkouší hesla uhodnout na základě přihlašovacího jména, plného jména, názvu domácího adresáře a dalších podobných informacích, které dovede ze systému získat.

Získání programu a instalace

Soubor s programem získáme z oficiálních stránek http://www.openwall.com/john/. Pokud ho chceme používat ve Windows, musíme stažený archiv rozbalit a spustit klasickou příkazovou řádku Windows (cmd.exe) a z té teprve spustit `john.exe`.

Tento program je již obsažen v distribuci Kali Linux. Do ostatních distribucí ho lze doinstalovat standardní cestou (například příkazem „`yum install john`“ ve Fedoře).

Pokud budeme sledovat jednotlivé hlášky vypisované na obrazovku, je ovládání intuitivní. Zezačátku je ale potřeba větší pozornosti, jelikož různých voleb je větší množství.



Obrázek 29 *Ukázka programu John the Ripper (snímek obrazovky)*

Použití

Příkazem „`john -test`“ můžeme provést malý test výkonu našeho systému. Výsledkem je počet provedených pokusů za sekundu pro různé hašovací algoritmy. Umožní nám to odhadnout dobu potřebnou k provedení útoku.

Nyní musíme zkopírovat soubory `/etc/shadow` a `/etc/passwd` do aktuálního adresáře. Použijeme příkazy „`cp /etc/shadow ./`“ a „`cp /etc/passwd ./`“ (v případě, že se nacházíme na počítači, pro který chceme hesla lámat).

Program John the Ripper potřebuje informace z těchto souborů spojit. K tomu zadáme příkaz: „`./unshadow passwd shadow > passwords`“.

Nyní již můžeme spustit lámání hesla příkazem „`john passwords`“. Pravděpodobně bude heslo hašováno algoritmem SHA512 a proto se nám na obrazovce objeví hláška, že musíme změnit formát. Proto zadáme příkaz „`--format sha512`“.

Tímto jsme provedli takzvaný single crack. Pro slovníkový útok použijeme příkaz „`john -wordlist=/umístění_slovníku passwords`“ (místo „umístění_slovníku“ zadáme cestu k souboru se slovníkem, který jsme si dříve stáhli.)

Příkaz pro útok hrubou silou je „`john --incremental=alpha passwords`“. Místo „alpha“ můžeme zadávat jiná klíčová slova, kterými definujeme používanou abecedu. Jejich seznam nalezneme například v nápovědě programu.

Chceme-li zobrazit nalezená hesla, zadáme příkaz „`john -show passwords`“

Pravděpodobně bude program John the Ripper využívat jen jedno jádro procesoru. Je vhodné toto ověřit a případně přenastavit. Pracujeme-li pod Windows, v příkazech souborům přiřadíme koncovku, takže budeme používat `passwords.txt` (pojmenování souboru je libovolné, passwords je příklad)

9.2.3 Hashcat

Tento program umí získat heslo pro více než sto variant zůzných hašovacích algoritmů a programů. Mimo tradiční haše (včetně NT haše z Windows) je v seznamu uveden například Android pin, GRUB 2, Skype či OS X. Lze používat jak v Linux, tak i ve Windows a OS X. Existují dvě základní verze tohoto programu. Jedna využívá klasický procesor a to včetně hardwarové akcelerace pomocí SSE2, AVX a XOP. Druhá verze provádí výpočty na grafické kartě od AMD (s ovladačem Catalyst 14.9 nebo novějším) nebo nVidia (s ovladačem ForceWare 346.59 nebo novějším). Tvůrci programu uvádějí, že toto je nejvýkonnější program pro lámání hesel a jediný, který je založen na výpočtech na grafické kartě. Hashcat podporuje několik způsobů útoku, mimo jiné útok hrubou silou a slovníkový útok (dovede i slova ze slovníku různě skládat, přidávat k nim jiné znaky, či znaky slova různě prohazovat).

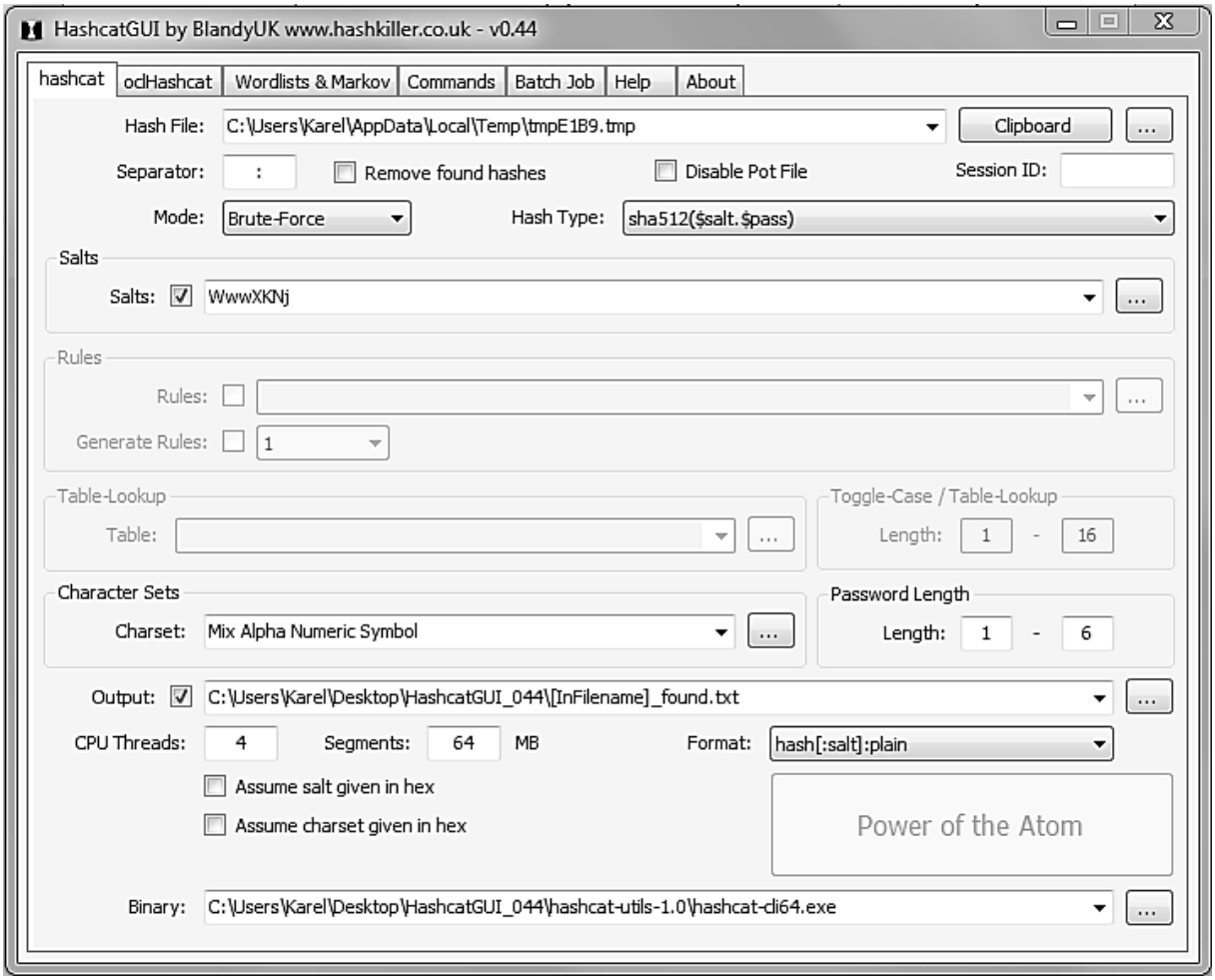
Získání programu a instalace

Program je primárně určen pro příkazovou řádku a lze ho získat z oficiálních stránek programu http://hashcat.net/. Na těchto stránkách je také velice rozsáhlá a užitečná dokumentace. Existuje i grafická nadstavba určená pro Windows a tu také použijeme. Zatím ji nevyvíjí přímo Hashcat, ale odkazuje na ni. Získat ji můžeme ze stránek http://www.hashkiller.co.uk/hashcat-gui.aspx. Program se neinstaluje, stačí stažený archiv rozbalit a spustit „App.HashcatGUI.exe“.

Tento program je již obsažen v distribuci Kali Linux. Do ostatních distribucí ho lze doinstalovat standardní cestou.

Chceme-li použít slovníkový útok, musíme si slovníky nejdříve stáhnout. Stačí do internetového vyhledávače zadat výraz „mega wordlist download“, „cz wordlist“ a podobně. Existuje

jich velké množství a je potřeba počítat s větší datovou náročností v řádově jednotek či desítek GB dat. Těžko se doporučuje konkrétní slovník, jednou se nám může osvědčit a jindy naopak selže. Je tedy potřeba zkoušet slovníků více a testovat, který pro daný účel bude vhodný.



Obrázek 30 *Ukázka grafické nadstavby programu Hashcat (snímek obrazovky)*

Použití

Grafická nadstavba nám umožní lépe se zorientovat v možnostech programu. V záložce „Wordlist & Markov“ můžeme vložit slovníky. První záložka „hashcat“ se týká nastavení programu pro výpočty na CPU, druhá záložka „odhashcat“ obsahuje nastavení pro výpočty na GPU. Jakmile provedeme všechna potřebná nastavení v uvedených záložkách, přepneme se do záložky Commands a klikneme na Generate. Zobrazí se nám textový výstup obsahující veškeré nastavení programu. Nakonec se přepneme do Batch Job, kam vložíme vygenerovaný řetězec. V případě, že chceme provést útok s více nastaveními, můžeme je sem postupně vložit všechna.

Jakmile jsme s příkazy spokojeni, klikneme na „run“. Spustí se standartní verze programu v příkazové řádce, ale již s nastavením, které jsme provedli v grafické nadstavbě. Jakmile se v nastavení začneme orientovat, může být klasická verze programu rychleji nastavitelná.

V jedné z prvních dvou záložek je potřeba zvolit vstupní soubor s hesly (případně kliknout na tlačítko „Clipboard“ a vložit haš v textové podobě), dále zvolit typ hašovacího algoritmu, způsob útoku a soubor, do kterého se uloží výsledek. V závislosti na zvoleném útoku se zobrazí možnosti pro jeho bližší specifikaci (například slovník pro slovníkový útok, abecedu pro útok hrubou silou a podobně.

9.2.4 Kali Linux

Zde bychom si dovolili odkázat na popis této distribuce operačního systému Linux v předchozí části, která se věnovala operačnímu systému Windows. Získání a použití je stejné, pouze se využívá jiných nástrojů. Vybrané nástroje jsme výše popsali samostatně, protože jsou dostupné i pro jiné distribuce a netýkají se přímo Kali Linuxu. Výhoda Kali Linux je v tom, že je obsahuje již v základním instalačním balíčku.

10 Závěr

Pro někoho, kdo se pokouší proniknout do operačního systému, je prolomení hesla základní dovedností. Proto je důležité, aby správci systému pochopili, jak jsou hesla uložena, jak se dají odcizit a lámat. Pro získání přístupu do operačního systému ale někdy heslo vůbec nepotřebujeme. Ověřili jsme si, že tyto techniky jsou navíc i podstatně rychlejší. Znalost uvedené problematiky je nutným základem pro správné nastavení bezpečnostní politiky a ochrany dat.

Zjistili jsme, že operační systémy Windows i Linux uchovávají hesla v hašované podobě. Linux používá obyčejný textový soubor, ve kterém jsou haše hesel uloženy. V současné době je ve většině distribucí v základním nastavení zvolen zatím nejsilnější, běžně dostupný algoritmus SHA-512 se solí. Haše jsou tak poměrně odolné vůči útoku hrubou silou i útoku s pomocí rainbow tables.

Windows používá podstatně slabší hašovací algoritmus – NT haš. Jedná se víceméně o MD4 algoritmus. Výsledný haš má 128 bitů, takže je v porovnání s SHA-512 čtyřikrát kratší a tedy podstatně méně odolný vůči útoku hrubou silou. Možná důležitějším nedostatekm NT haše je to, že nepoužívá sůl, takže je možné použít velmi výkonný útok s pomocí rainbow tables. Proto je opravdu potřeba dodržovat zásady tvorby silného hesla, které tyto slabiny částečně eliminuje. Microsoft ve snaze zabránit přečtení souboru s haši, zavedl jeho šifrování. Výsledkem mělo být zabránění extrakce hašů, čímž by eliminoval standardní způsob útoku, při kterém se získaný haš podrobuje lámání. Pak by nevadilo, že používá poměrně slabý hašovací algoritmus. Nicméně ze své podstaty je nutné do souboru s haši přístup určitým procesům zachovat, a tak bylo nutné i šifrovací klíč v systému někam uložit. Netrvalo dlouho než byly nalezeny postupy, jakými je šifrovací klíč vytvořen. Pak nebyl problém do aplikací pro extrahování hesel (soubor obsahující hesla ve Windows je binární, ne textový jako v Linux) vložit i algoritmus pro jeho dešifrování.

Kromě způsobu uchování hesla v systému, jsme také kompletně zmapovali proces spuštění systému, jak systém ověřuje uživatele a jakým způsobem chrání své zdroje. Tyto informace jsou důležité pro pochopení fungování mechanismů získání přístupu do operačního systému, které nejsou založeny na uhodnutí hesla. Ověřili jsme a popsali jednoduché a rychlé postupy, jak se do systému přihlásit, aniž bychom potřebovali znát heslo k jakémukoli účtu.

V práci uvedené mechanismy a postupy, stejně jako praktické testy, byly vztaženy na systémy Windows 7, Windows 8.1, Windows server 2012 (systém active directory), Linux distribuce CentOS, Fedora a Ubuntu. Co se týká útoku na Windows server 2012 s AD, tak přestože samozřejmě databáze AD v sobě hesla uchovává, útoky na ní se spíše neprovádějí. Je to z toho důvodu, že vždy existuje i místní účet, který využívá lokálního ověření (mimo jiné z důvodu, když není dostupná síť pro ověření skrze AD), a tak útočit na databázi AD není potřeba. Během tvorby této práce jsme nenarazili na odlišnosti v zabezpečení jak v rámci uvedených verzí Windows, tak ani v rámci různých distribucí systému Linux.

Během zpracovávání této práce bylo vyzkoušeno více postupů, než je zde uvedeno. Pokud měly stejné požadavky na přístup k systému a vedly ke stejnému výsledku, pak byly uvedeny jen ty, které se nám zdáli nejjednodušší či nejrychlejší. Vysvětlili jsme základní principy a postupy pro získání přístupu do operačního systému Windows či Linux, aniž bychom znali uživatelské, respektive administrátorské heslo. Pokud by na konkrétním systému uvedené postupy nefungovali, pak by po přečtení této práce neměl být problém na internetu nalézt postup obdobný, který již fungovat bude.

Důležitou součástí práce jsou navržené mechanismy, kterými se můžeme bránit proti zjištěným bezpečnostním hrozbám.

Na základě námi provedených útoků na haše hesel jsme zjistili, že je důležité volit hesla o minimální délce osmi znaků, ideálně navíc obsahující znaky z rozšířené abecedy a to z důvodu ochrany proti útoku hrubou silou. Stejně tak je velmi důležité mít heslo odolné vůči slovníkovému útoku. To znamená, že heslo nesmí být slovem a slovo by nemělo ani obsahovat. Aby se heslo dalo snadno zapamatovat, je vhodné si ho vytvořit na základě nějaké věty. Například heslo „hKSj100%b“ se dá považovat za vcelku bezpečné a zapamatujeme si ho větou: „heslo Karla Suchého je stoprocentně bezpečné“.

Nejúčinnější a mnohdy jedinou možnou ochranou jak proti zcizení hašů hesel, tak proti jiným přístupům do operačního sytému (nevyžadující znalost hesla) je šifrování celého disku. V tomto případě ale bude představovat fatální problém zapomenutí hesla či potřeba obnovy dat po havárii paměťového zařízení. O to důležitější je věnovat se zálohování a samozřejmě nezapomenout, že zálohy jsou potřeba proti útočníkům chránit také.

Seznam použité literatury

[1] ONDŘEJ, Bitto. *1001 tipů a triků pro Microsoft Windows 8*. Brno: Computer Press, 2013. ISBN 978-80-251-3806-9.

[2] ROMAN, Kučera. *Bible Microsoft Windows 8: Nejlepší tipy & triky*. Brno: Extra publishing, 2012. ISBN 978-80-741-3228-5.

[3] ANTHONY, Northrup. *Mistrovství v Microsoft Windows 8: kompletní průvodce do posledního detailu*. Brno: Computer Press, 2013, 615 s. Mistrovství. ISBN 978-80-251-4111-3.

[4] BOHDAN, Cafourek. *Windows 7: kompletní příručka*. 1. vyd. Praha: Grada, 2010, 326 s. Profesionál. ISBN 978-80-247-3209-1.

[5] BOTT, Ed, Carl SIECHERT a Craig STINSON. *Mistrovství v Microsoft Windows 7: kompletní průvodce do posledního detailu*. Brno: Computer Press, 2010, 936 s. Mistrovství. ISBN 978-80-251-2817-6.

[6] HARRIS, Shon. *Hacking: manuál hackera*. Praha: Grada, 2008, 399 s. ISBN 978-80-247-1346-5.

[7] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking bez záhad*. 5. vyd. Praha: Grada, 2007, 520 s. ISBN 978-80-247-1502-5.

[8] MARK Russinovich, Solomon DAVID A a Ionescu ALEX. *Windows internals: Part 1*. 6th ed. Redmond, Wash: Microsoft Press, 2012. ISBN 978-0-7356-4873-9.

[9] MARK Russinovich, Solomon DAVID A a Ionescu ALEX. *Windows internals: Part 2*. 6th ed. Redmond, Wash: Microsoft Press, 2012. ISBN 978-0-7356-6587-3.

[10] MICROSOFT. *TechNet* [online]. 2015 [cit. 2015-08-18]. Dostupné z: https://technet.microsoft.com/cs-cz/

[11] MARTIN, Dráb. *Jádro systému Windows: kompletní průvodce programátora*. Vyd. 1. Brno: Computer Press, 2011, 472 s. Programování (Computer Press). ISBN 978-80-251-2731-5.

[12] *Linux: dokumentační projekt*. 4. aktualiz. vyd. Brno: Computer Press, 2007, 1334 s. ISBN 978-80-251-1525-1.

[13] LUKÁŠ, Jelínek. *Jádro systému Linux: kompletní průvodce programátora*. Brno: Computer Press, 2008, 686 s. Programování (Computer Press). ISBN 978-80-251-2084-2.

[14] PAVLÍK, Martin. *Metody ukládání uživatelských hesel v operačních systémech*. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2009. 67 s. Dostupné také z: https://dspace.vutbr.cz/handle/11012/10269. Diplomová práce. Vedoucí práce Ing. Jan Hajný.

[15] FIŠER, Jiří. *Principy operačních systémů II* [online]. Univerzita Jana Evangelisty Purkyně. 2007 [cit. 2015-08-18]. Dostupné z: http://ithil.ujep.cz/workspace/OS-2.pdf

[16] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGI. *Department od defensetrusted computer system evaluation criteria: DoD 5200.28-STD*. [online]. 1985 [cit. 2015-08-18]. Dostupné z: http://csrc.nist.gov/publications/history/dod85.pdf

[17] SEAN, Boran. *The IT Security Cookbook - Operating System (OS) Overview*. WindowsSecurity.com [online]. 2002, 23.1.2013 [cit. 2015-08-18]. Dostupné z: http://www.windowsecurity.com/whitepapers/misc/The_IT_Security_Cookbook/The_IT_Security_Cookbook__Operating_System_OS_Overview_.html

[18] Certified Products. *Common Criteria* [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.commoncriteriaportal.org/products/

[19] COMMON CRITERIA. *Common Criteria for Information Technology Security Evaluation: Part 3: Security assurance components* [online]. 2012 [cit. 2015-08-18]. CCMB-2012-09-003. Dostupné z: http://www.commoncriteriaportal.org/files/ccfiles/ccpart3v3.1r4.pdf

[20] ŠÁCHA, Martin. *Root sudo*. Ubuntu.cz [online]. 2014 [cit. 2015-08-18]. Dostupné z: http://wiki.ubuntu.cz/root_sudo

[21] MICHAL, Valášek. *Základy počítačové bezpečnosti 1: kdo a co nás ohrožuje*. Hospodářské noviny [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://tech.ihned.cz/geekosfera/c1-64468960-slovnik-zaklady-pocitacove-bezpecnosti-prvni-dil?utm_source=mediafed&utm_medium=rss&utm_campaign=mediafed

[22] STATCOUNTER. *GlobalStats* [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://gs.statcounter.com

[23] MICROSOFT. *Ukončení podpory Windows XP* [online]. 2014 [cit. 2015-08-18]. Dostupné z: http://www.microsoft.com/cze/ukoncenipodpory/vyhody-noveho-pocitace.aspx

[24] Google Chrom Blog. *Providing updates for Chrome for XP through 2015* [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://chrome.blogspot.cz/2015/04/providing-updates-for-chrome-for-xp.html

[25] JAN, Stach. *Proč se WINDOWS 10 nejmenuje 9? Už to víme. Marketing za to nemůže. Je to kvůli Win 95 a 98*. DD World [online]. 2014 [cit. 2015-08-18]. Dostupné z: http://www.ddworld.cz/aktuality/software/proc-se-windows-10-nejmenuje-9-uz-to-vime.-marketing-za-to-nemuze.-je-to-kvuli-win-95-a-98-2.html

[26] MICROSOFT. *Funkce Windows 10* [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.microsoft.com/cs-cz/windows/features

[27] KOHL, J. a C. NEUMAN. *The Kerberos Network Authentication Service (V5): RFC 1510*. IETF Tools [online]. 1993 [cit. 2015-08-18]. Dostupné z: http://tools.ietf.org/html/rfc1510

[28] JIRSA, Milan. *Schéma operačního systému a bezpečnostního podsystému.* Moodle: Informační systém vojenských škol [online]. 2012 [cit. 2015-08-18]. Dostupné z: https://moodle.unob.cz/mod/page/view.php?id=8767

[29] MENŠÍK, Miroslav. *Bezpečnost operačního sytému a síťové komunikace: Informatika modul 4*. Vysoké učení technické v Brně, 2004 [cit. 2015-08-18]. Dostupné z: http://lences.cz/domains/lences.cz/skola/subory/Skripta/BU01-Informatika/M04-Bezpecnost%20operacniho%20systemu%20a%20sitove%20komunikace.pdf

[30] *Windows Authentication*. MICROSOFT. TechNet [online]. 2011 [cit. 2015-08-18]. Dostupné z: https://technet.microsoft.com/en-us/library/cc755284(v=ws.10).aspx

[31] *Jak zabránit systému Windows v uložení hodnoty hash systému LAN Manager hesla v adresáři Active Directory a místních databázích SAM*. MICROSOFT. Podpora Microsoft [online]. 2007 [cit. 2015-08-18]. Dostupné z: https://support.microsoft.com/cs-cz/kb/299656

[32] JOHANSSON, Jesper. *The Most Misunderstood Windows Security Setting of All Time: Security Watch*. MICROSOFT. TechNet: Magazine [online]. 2008 [cit. 2015-08-18]. Dostupné z: https://technet.microsoft.com/cs-cz/magazine/2006.08.securitywatch(en-us).aspx

[33] SCAMBRAY, Joel, George KURTZ a Stuart MCCLURE. *Hacking bez tajemství*. 2. aktualiz. vyd. Praha: Computer Press, 2002, xxviii, 625 s. Komunikace a sítě. ISBN 80-722-6644-6.

[34] SANDERS, Chris. *How I Cracked your Windows Password (Part 1)*. WindowSecurity.com [online]. 2010 [cit. 2015-08-18]. Dostupné z: http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/How-Cracked-Windows-Password-Part1.html

[35] *Passwords Technical Overview*. MICROSOFT. TechNet: Library [online]. 2012 [cit. 2015-08-18]. Dostupné z: https://technet.microsoft.com/en-us/library/hh994558(v=ws.10).aspx

[36] BAF, *Jak startuje systém?*. Linux expres [online]. 2007 [cit. 2015-08-18]. Dostupné z: http://www.linuxexpres.cz/praxe/jak-startuje-system

[37] *Usage statistics and market share of Linux for websites: Subcategories of Linux*. W3Techs [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://w3techs.com/technologies/details/os-linux/all/all

[38] *DistroWatch Page Hit Ranking*. DistroWatch.com [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://distrowatch.com/dwres.php?resource=popularity

[39] *Statistics about the Linux distributions*. LinuxCounter.net [online]. 2015 [cit. 2015-08-18]. Dostupné z: https://www.linuxcounter.net/statistics/distributions

[40] BHARTIYA, Swapnil. *10 of the Most Popular Linux Distributions Compared*. How-to geek [online]. [cit. 2015-08-18]. Dostupné z: http://www.howtogeek.com/191207/10-of-the-most-popular-linux-distributions-compared/

[41] SHARMA, Shashank. *10 best Linux distros: which one is right for you?* Techradar [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.techradar.com/news/software/operating-systems/best-linux-distro-five-we-recommend-1090058

[42] *The Top 11 Best Linux Distros for 2015*. Linux [online]. 2015 [cit. 2015-08-18]. Dostupné z: https://www.linux.com/news/software/applications/810295-the-top-11-best-linux-distros-for-2015

[43] ANGELA, Alcron. *The Best Linux Distributions*. Make use of [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.makeuseof.com/tag/best-linux-distributions/

[44] *Linux: dokumentační projekt*. 3. aktualiz. vyd. Brno: Computer press, 2003, 1001 s. ISBN 80-7226-761-2.

[45] *Linux kernel and its architecture*. Knowstuffs: Gain knowledge is all life about [online]. [cit. 2015-08-18]. Dostupné z: https://knowstuffs.wordpress.com/tag/kernel-architecture/

[46] *Linux*. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2015 [cit. 2015-08-18]. Dostupné z: https://en.wikipedia.org/wiki/Linux

[47] JELÍNEK, Lukáš. *Vývoj jádra I*. Linux Expres [online]. 2006 [cit. 2015-08-18]. Dostupné z: http://www.linuxexpres.cz/praxe/vyvoj-jadra-i

[48] *Start Linuxu*. Wikiknihy [online]. 2015 [cit. 2015-08-18]. Dostupné z: https://cs.wikibooks.org/wiki/Start_Linuxu

[49] *Introduction to Linux: Boot process, Init and shutdown*. The Linux Documentation project [online]. [cit. 2015-08-18]. Dostupné z: http://www.tldp.org/LDP/intro-linux/html/sect_04_02.html

[50] PILLAI, Sarath. *How are passwords stored in Linux*. ROOT.IN [online]. 2013 [cit. 2015-08-18]. Dostupné z: http://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils

[51] JACKSON, Michael H. *Why shadow your passwd file?* The Linux Documentation Project [online]. 1996 [cit. 2015-08-18]. Dostupné z: http://www.tldp.org/HOWTO/Shadow-Password-HOWTO-2.html

[52] FRAMPTON, Steve. *Linux Administration Made Easy: Linux Password & Shadow File Formats*. The Linux Documentation Project [online]. [cit. 2015-08-18]. Dostupné z: http://www.tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html

[53] *Shadow - soubor se zašifrovanými hesly*. Ubuntu manuals [online]. 2010 [cit. 2015-08-18]. Dostupné z: http://manpages.ubuntu.com/manpages/hardy/cs/man5/shadow.5.html

[54] VIVEK, Gite. *Understanding /etc/shadow file*. NixCraft: Linux and Unix tutorials for new and seasoned sysadmin. [online]. 2006 [cit. 2015-08-18]. Dostupné z: http://www.cyberciti.biz/faq/understanding-etcshadow-file/

[55] KERRISK, Michael. *CRYPT(3)*. Linux Programmer's Manual [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://man7.org/linux/man-pages/man3/crypt.3.html

[56] DOČEKAL, Michal. *Správa linuxového serveru: Přístupová práva a ACL*. LinuxExpres [online]. 2012 [cit. 2015-08-18]. Dostupné z: http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-pristupova-prava-a-acl

[57] HORÁK, Jan. *Jak správně na SELinux: Proč byste měli chtít SELinux?* Root.cz [online]. 2007 [cit. 2015-08-18]. Dostupné z: http://www.root.cz/clanky/proc-byste-meli-chtit-selinux/

[58] *SELinux*. Wiki.CenOS: HowTos [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://wiki.centos.org/HowTos/SELinux

[59] *SELinux*. Fedora: Wiki [online]. 2014 [cit. 2015-08-18]. Dostupné z: http://fedoraproject.org/wiki/SELinux

[60] AppArmor. Ubuntu wiki [online]. 2015 [cit. 2015-08-18]. Dostupné z: https://wiki.ubuntu.com/AppArmor

[61] MÜLLER, Zdeněk. *Bezpečné kryptografické algoritmy* [online]. Olomouc, 2012 [cit. 2015-08-18]. Diplomová práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta. Vedoucí práce RNDr. Miroslav Kolařík, Ph.D.. Dostupné z: http://theses.cz/id/b638au/

[62] VLASTIMIL, Klíma. *Hašovací funkce, principy, příklady a kolize*. Crypto - World [online]. 2005 [cit. 2015-08-18]. Dostupné z: http://crypto-world.info/klima/2005/cryptofest_2005.htm

[63] KUMPOŠT, Marek. Hašovací funkce [online]. Brno, 2003 [cit. 2015-08-18]. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Václav Matyáš. Dostupné z: http://is.muni.cz/th/44545/fi_b/

[64] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Data Encryption Standard (DES): FIPS PUB 46-3* [online]. 1995 [cit. 2015-08-18]. Dostupné z: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[65] DES. Kryptografie [online]. [cit. 2015-08-18]. Dostupné z: http://www.kryptografie.wz.cz/data/des.html

[66] RÓBERT, CSc. doc. Ing. *Pokrocilá kryptologie DES a AES*. Edux.fit.cvut.cz [online]. 2011 [cit. 2015-08-18]. Dostupné z: https://edux.fit.cvut.cz/oppa/MI-KRY/prednasky/prednaska3_4.pdf

[67] *Šifrování , Symetrické šifrování*. O webu [online]. 2007 [cit. 2015-08-18]. Dostupné z: http://owebu.bloger.cz/Bezpecnost/Sifrovani-Symetricke-sifrovani

[68] VOJTĚCH, Brtník. *DES online - Popis*. Brtnik [online]. 2007 [cit. 2015-08-18]. Dostupné z: http://www.brtnik.eu/des.php

[69] PŘIKRYL, Jan. *Blokové šifry* [online]. České vysoké učení technické, 2013, 26 s. [cit. 2015-08-18]. Dostupné z: http://euler.fd.cvut.cz/predmety/y2kk/kzk-krypto-blok.pdf

[70] SCHNEIER, Bruce. *Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)*. Schneier on Security [online]. Fast Software Encryption, pages 191–204. Springer, 1994 [cit. 2015-08-18]. Dostupné z: https://www.schneier.com/paper-blowfish-fse.html

[71] MIT LABORATORY FOR COMPUTER SCIENCE. *The MD4 Message-Digest Algorithm: RFC 1320* [online]. 1992 [cit. 2015-08-18]. Dostupné z: http://www.ietf.org/rfc/rfc1320.txt

[72] MIT LABORATORY FOR COMPUTER SCIENCE. *The MD5 Message-Digest Algorithm: RFC 1321* [online]. 1992 [cit. 2015-08-18]. Dostupné z: http://www.ietf.org/rfc/rfc1321.txt

[73] *Hašovací funkce*. Mendelova univeržita v Brně: Studijní opora [online]. [cit. 2015-08-18]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=7029

[74] SOTIROV, Alexander a kolektiv. *MD5 considered harmful today: Creating a rogue CA certificate*. University of Technology Eindhoven [online]. 2008 [cit. 2015-08-18]. Dostupné z: http://www.win.tue.nl/hashclash/rogue-ca/

[75] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Secure Hash Standard (SHS): FIPS PUB 180-4* [online]. 2012 [cit. 2015-08-18]. Dostupné z: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[76] MIROSLAV, Knotek. *PKI – Přecházíme na SHA-2*. KPCS [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.kpcs.cz/a/489/PKI-Prechazime-na-SHA-2

[77] LOCKTYUKHIN, Max. *Improving the Performance of the Secure Hash Algorithm (SHA-1)*. Intel Developer Zone [online]. 2010 [cit. 2015-08-18]. Dostupné z: https://software.intel.com/en-us/articles/improving-the-performance-of-the-secure-hash-algorithm-1/

[78] ŠENFELD, Josef. *Rainbow tables a jejich využití k prolomení hašovací funkce*. Pardubice, 2015, 77 s. Diplomová práce. Univerzita Pardubice. Vedoucí práce Mgr. Josef Horálek, Ph.D.

[79] *Rainbow tables*. OBJECTIV SÉCURITÉ. Ophcrack [online]. [cit. 2015-08-18]. Dostupné z: http://ophcrack.sourceforge.net/tables.php

[80] Bezpečnost IS/IT: Kryptografické systémy. KUNDEROVÁ, Ludmila. *Akela.mendelu.cz: Studentský server* [online]. Mendeleova Univerzita v Brně, 2014 [cit. 2015-08-18]. Dostupné z: https://akela.mendelu.cz/~lidak/bis/8kryp.htm

[81] *Cryptoanalysis*. Počítačový Bezpečnostní Web [online]. [cit. 2015-08-18]. Dostupné z: http://pbweb.cz/Pocitacovy%20utok/Kryptograficke/cryptoanalysis.html

[82] *Differential cryptanalysis*. Počítačový Bezpečnostní Web [online]. [cit. 2015-08-18]. Dostupné z: http://pbweb.cz/Pocitacovy%20utok/Kryptograficke/differential_cryptanalysis.html

[83] PETŘÍK, Tomáš. *Modrení Kryptoanalýza*. [online] Brno, 2011, 59 s. [cit. 2015-08-18]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=38343. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Zdeněk Martinásek.

[84] *Making a Kali Bootable USB Drive*. KALI LINUX. Kali Linux: Official Documentation [online]. [cit. 2015-08-18]. Dostupné z: http://docs.kali.org/downloading/kali-linux-live-usb-install

[85] *Online Hash Crack: Professional Password Recovery* [online]. 2008, 2015 [cit. 2015-08-18]. Dostupné z: http://www.onlinehashcrack.com/

[86] *How To Reset Your Forgotten Root Password On CentOS 7 Servers*. Liberian Geek: Tutorials for newbies! [online]. 2015 [cit. 2015-08-18]. Dostupné z: http://www.liberiangeek.net/2014/09/reset-forgotten-root-password-centos-7-servers/