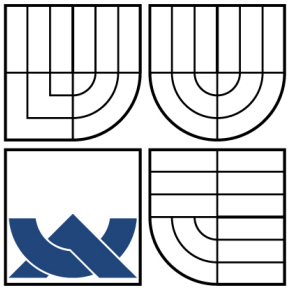
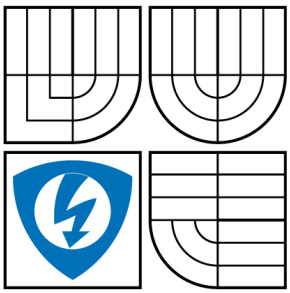


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

METODY UKLÁDÁNÍ UŽIVATELSKÝCH HESEL V OPERAČNÍCH SYSTÉMECH

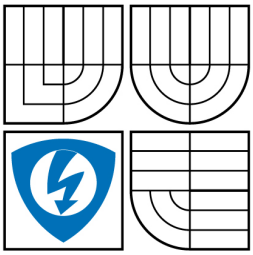
PASSWORD DEPOSITION TECHNIQUES IN OPERATING SYSTEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. MARTIN PAVLÍK
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN HAJNÝ
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Martin Pavlík	***ID:***	83379
Ročník:	2	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Metody ukládání uživatelských hesel v operačních systémech

POKYNY PRO VYPRACOVÁNÍ:

Zjistěte mechanizmy ukládání uživatelských jmen a hesel v moderních operačních systémech. Zaměřte se především na MS Windows, Linux, FreeBSD (OpenBSD) a Mac OS. Porovnejte jednotlivé metody uložení hesla v systému a mechanizmy nutné pro ověření uživatele. Najděte a popište vhodné nástroje pro útoky na autentizační mechanizmy v jednotlivých OS (především na použité hashovací funkce). Prakticky zrealizujte útoky na vybrané OS, srovnejte účinnost zvolených technik a nástrojů, popř. navrhněte další postupy pro prolomení bezpečnosti.

DOPORUČENÁ LITERATURA:

[1] HONTANóN, Ramón. Linux praktická bezpečnost. [s.l.] : [s.n.], 2003. 438 s.
[2] SMITH, Richard E. Authentication: From Passwords to Public Keys. [s.l.] : [s.n.], 2001. 576 s.
[3] KOMAR, Brian, SMITH, Ben. Microsoft Windows Security Resource Kit. [s.l.] : [s.n.], 2003. 720 s.

Termín zadání:	9.2.2009	***Termín odevzdání:***	26.5.2009

Vedoucí práce: Ing. Jan Hajný

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práve třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ANOTACE:

Tato diplomová práce se zabývá způsoby ukládání hesel v současných operačních systémech. Konkrétně se práce zaměřuje na operační systémy Windows, Linux, BSD a OS X. U těchto systémů jsou zkoumány způsoby hašování hesel a odolnost výsledných hašů proti různým útokům.

V první (teoretické) části jsou popsány postupy a algoritmy, které jsou potřebné k autentizaci uživatele. Dále je zde popsán způsob uložení hašů. Na konci teoretické části jsou obecně popsány některé útoky, které je možné vést proti hašovacím funkcím.

V druhé (praktické) části jsou popsány a otestovány nástroje pro získání hašů ze zkoumaných operačních systémů. Na získané haše jsou pomocí vhodných nástrojů vedeny praktické útoky. Dále jsou zde uvedeny výsledky útoků.

Závěrem práce je srovnání metod a nástrojů použitých pro získání otevřených hesel z operačních systémů.

Klíčová slova: haš, hašovací funkce, heslo, prolomení hesla, útok hrubou silou, slovníkový útok, rainbow tables

ABSTRACT:

This master thesis deals with ways to store passwords in current operating systems. Specifically, this work focuses on Windows, Linux, BSD and OS X. These systems are examined for ways of hashing passwords and on resistance of resulting hashes against various attacks.

First (theoretical) section describes the procedures and algorithms that are needed for user authentication. This part also describes methods of hash storing. At the end of the theoretical part are generally described some possible attacks against hash functions.

In second (practical) part is described and tested tools for obtaining hashes of the investigated operating systems. Subsequently practical attacks were conducted against obtained hashes by using appropriate tools. Furthermore there are presented results of the attacks.

In the conclusion of the work there is a comparison of tools and methods which were used to obtain plaintext passwords from operating systems.

Keywords: hash, hash function, password, password break, brute force attack, dictionary attack, rainbow tables

PAVLÍK, M. *Metody ukládání uživatelských hesel v operačních systémech.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 67 s. Vedoucí diplomové práce Ing. Jan Hajný.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma **Metody ukládání uživatelských hesel v operačních systémech** jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne …… ……

(podpis autora)

PODĚKOVÁNÍ:

Děkuji vedoucímu diplomové práce **Ing. Janu Hajnému** za metodické a cíleně orientované vedení při plnění úkolů realizovaných v návaznosti na diplomovou práci.

OBSAH

1 ÚVOD A ZÁKLADNÍ POJMY

Tato diplomová práce se zabývá problematikou ukládání hesel v uživatelských operačních systémech. Hesla jsou základním a nejrozšířenějším prostředkem pro autentizaci uživatelů.

Autentizace
je proces *ověření proklamované identity* subjektu. Autentizace patří k bezpečnostním opatřením a zajišťuje ochranu před falšováním identity, kdy se subjekt vydává za někoho, kým není. Pokud dojde k úspěšné autentizaci uživatele, je mu zpravidla umožněn přístup k určitým neveřejným prostředkům (emailové účty, soukromý prostor na sdílených discích, soukromé soubory atd.). Pokud je autentizace neúspěšná operační systém usoudí, že uživatel není oprávněn k přístupu do systému a chráněné prostředky systému jsou nadále v bezpečí.

Používají se tyto základní metody pro ověření identity:

- podle toho, co uživatel zná (zná správnou kombinaci uživatelského označení a hesla)
- podle toho, co uživatel má (nějaký technický prostředek, který uživatel vlastní USB dongle, smart card, privátní klíč apod.)
- podle toho, čím uživatel je (uživatel má vlastnosti, které lze prověřit – otisk prstu, snímek oční duhovky či sítnice apod.)
- podle toho, co uživatel umí (umí správně odpovědět na náhodně vygenerovaný kontrolní dotaz)

Nejjednodušší autentizace probíhá prostřednictvím uživatelského účtu a hesla. Tato metoda je však nejslabší a nejvíce zranitelná. Jejím největším problémem je bezpečné a utajené sdělení hesla. Protože se při ní používá jediný tajný parametr, označuje se také jako jednofaktorová. [1]

Autorizace
označuje proces ověřování oprávnění subjektu k provedení určité akce. Tento proces obvykle navazuje na autentizaci, např. po úspěšné autentizaci přidělí souborový server uživateli specifická oprávnění ke sdíleným sítovým prostředkům. V moderních operačních systémech není jednoduché přesně určit hranici mezi autentizací a autorizací, protože tyto procesy úzce souvisí. [1]

Haš (anglicky hash)
Hašem se v této práci rozumí výstup z kryptografické hašovací funkce, do které bylo jako vstupní zpráva vloženo heslo.

Hašovací funkce
je algoritmus, jehož účelem je vytvořit jedinečný digitální otisk určité délky (obvykle vyjadřovaný v bitech), který zcela závisí na vstupních datech. Hašovací algoritmus se

považuje za kryptograficky bezpečný pokud je bezkolizní a má dobrý lavinový efekt. Velmi důležitou vlastností hašovací funkce je jednocestnost. Jednocestnost znamená to, že z výstupního řetězce funkce nelze získat zpět původní informaci.

Lavinový efekt (avalanche effect)
Změnou jediného bitu ve vstupní posloupnosti se s velkou pravděpodobností značně (nebo úplně) změní výstupní posloupnost.

Kolize
je označením jevu při kterém hašovací funkce ze dvou různých vstupů vytvoří identické výstupy.

Sůl (salt)
Solí se v kryptografii rozumí jedinečná hodnota (systémová nebo uživatelsky definovaná) přidávaná k heslu před hašováním. Typicky se jedná o náhodnou hodnotu, ale může se také získávat jiným způsobem (používá se například identifikační číslo uživatele (UID)). Sůl může být utajovaná nebo uložena spolu s hašem. Použití soli vede k tomu že stejná hesla nemají stejné haše, což buď zkomplikuje a nebo zcela znemožní útoky pomocí předem vypočtených slovníků.

Heslo
se obvykle používá při autentizaci uživatelů. Obecně se jedná o jediné slovo, alfanumerické a nejlépe více než 8 znaků dlouhé (například banan3759).

Prolomení hašovací funkce
Prolomením hašovací funkce se v kryptologii rozumí případ, kdy dokážeme generovat kolize rychleji, než teoreticky v návrhu.

2 MS WINDOWS

Operační systémy společnosti MS tvoří majoritu uživatelských operačních systémů v celosvětovém rozsahu. K dnešnímu dni již společnost Microsoft vydala celou řadu operačních systémů a jejich různých variant. Z důvodů omezeného rozsahu této práce budou popsány pouze nejpoužívanější uživatelské operační systémy.

Windows 98
je operační systém vydaný společností Microsoft 25. června 1998. Systém Windows 98 je velice podobný o něco staršímu systému Windows 95, mezi novinky patří lepší podpora sběrnic AGP a USB, podpora pro více monitorů. Na rozdíl od řady Windows NT se stále jedná o hybridní 16/32-bitový systém, který měl časté problémy s nestabilitou a pády do modré obrazovky s chybovým hlášením, které se přezdívalo Modrá obrazovka smrti (BSOD). Dědí se ještě podpora DOSu. Veškerá oficiální podpora toho operačního systému byla ukončena ke dni 07.07.2007.

Windows XP
Jedná se o komerční, víceuživatelský (multiuser) operační systém založený na architektuře mikrojádra, systém je víceúlohový (multitask), primárně určený pro 32 bitové procesory. Navazuje na předcházející systémy *Windows 2000* a *Windows NT*, většina programů z dřívějších verzí systémů Windows je s Windows XP kompatibilní. Systém byl naprogramován zejména v jazyce C a C++. Systém existuje ve čtyřech variantách: Professional, Home Edition, 64 bit Edition, Tablet PC Edition.

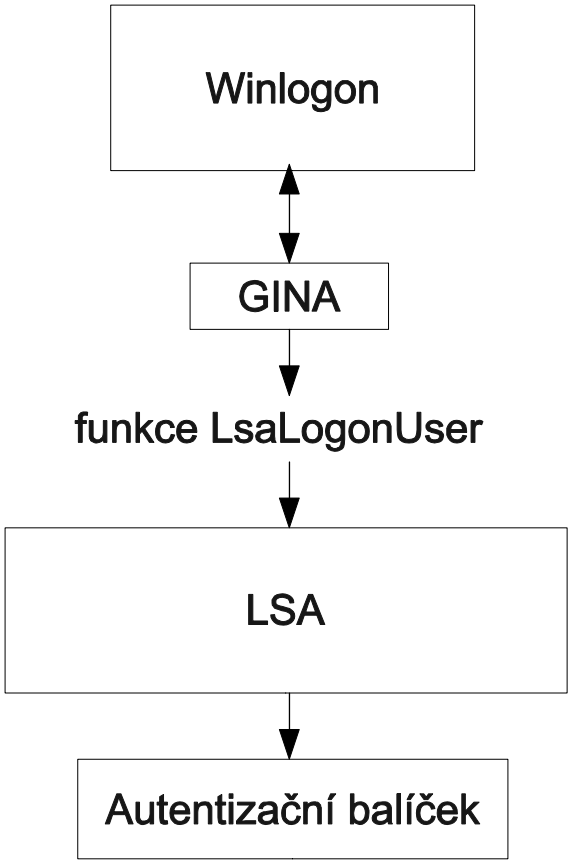
Windows Vista
je nejnovější oficiální verze operačního systému Windows od společnosti Microsoft. Celosvětový prodej Visty v hlavních světových jazycích byl oficiálně zahájen 30. ledna 2007. Společnost Microsoft vydala celkem 6 různých edicí Windows Vista: Starter, Home Basic, Home Premium, Business, Enterprise, Ultimate. Všechny edice jsou k dostání ve verzi jak pro 32 bitové procesory, tak pro 64 bitové procesory.

Windows 7
bude další verze operačního systému Microsoft Windows, jako nástupce Windows Vista. Podle společnosti Microsoft by systém mohl oficiálně vyjít do konce roku 2009. Bude k dostání 64bitová i 32bitová verze. Na rozdíl od svého předchůdce, mají být Windows 7 výrazně modernizovány a cílem je jejich plná kompatibilita s existujícími ovladači zařízení, aplikacemi a hardwarem.

2.1 Interaktivní přihlašování

Interaktivní přihlášení představuje ve Windows základní autentizační úroveň. Po zapnutí počítače dojde k inicializaci operačního systému obvykle následované vyzváním uživatele k autentizaci uživatelským jménem a heslem. Jakmile uživatel tyto údaje zadá, dojde k jejich porovnání s parametry místního nebo doménového účtu. Samotné přihlašování je zajišťováno procesem *Winlogon*.

V jeho bezpečnostním kontextu je zavedena knihovna *ms-gina.dll*, která představuje systémový modul označovaný jako *Microsoft Graphical Identification and Authentication* (MS GINA). Jeho úkolem je zachytit vznik událostí třídy *Secure Attention Sequence* (SAS), do které mimo jiné náleží žádost o: přihlášení, odhlášení, uzamčení sezení. Pokud MS GINA takovou událost zachytí, informuje řídící proces *Winlogon*. Při interaktivním přihlašování využívá MS GINA k autentizaci uživatele lokálního podsystému zabezpečení *Local Security Autority* (LSA), konkrétně se volá funkce LSA s názvem *LsaLogonUser*. Parametry funkce jsou zadané autentizační údaje a požadavek na konkrétní autentizační mechanismus, který je realizován formou *autentizačního balíčku* (authenticaton package). Zvolený autentizační balíček ověří identitu uživatele a o výsledku informuje podsystém LSA, který informaci předává zpět modulu MS GINA. [1]



Obrázek 2.1: Autentizace uživatele Windows

Přihlašování do Windows

Účelem autentizačního balíčku je provést ověření autentizačních údajů pomocí určitého protokolu. Pokud je autentizace úspěšná, vzniká nová relace přihlášení reprezentovaná jedinečným *identifikátorem relace*. Tuto informaci předává autentizační balíček zpět podsystému LSA, který ji používá při sestavování datové struktury označované jako *bezpečnostní záznam* (security token). Základní autentizační balíček ve Windows má název MSV1_0 a je určen pro autentizaci pomocí protokolu *New Technology (NT) LAN Manager* (NTLM). Autentizaci provádí pomocí účtu a otisku hesla (haš) uživatele. Zadané údaje ověřuje přístupem do lokální databáze účtů SAM (Security Accounts Manager) nebo se obrací na řadič domény. V případě úspěšné autentizace se účet a haš hesla stávají součástí relace přihlášení a operační systém je používá při neinteraktivní autentizaci. Modularita operačního systému Windows umožňuje operačnímu systému poskytovat aplikacím různé autentizační mechanismy zavedením více autentizačních balíčků. Standardně systém

Windows zavádí zmíněný balíček MSV1 0 a od verze Windows 2000 také balíček Kerberos Security Service Provider (SSP)/Authentication Package (AP), který se uplatňuje při síťové autentizaci prostřednictvím protokolu Kerberos. [1]

2.2 Ukládání hesel ve Windows

Součástí bezpečnostní politiky operačního systému je správa uživatelských účtů, ke které náleží zejména bezpečné uložení kopií hesel. Hesla zpravidla nejsou uložena v otevřené podobě, ale uchovávají se jejich šifrované obrazy. V současných systémech Windows je heslo z formy otevřeného textu (clear text) převedeno na haš a to jak pomocí algoritmu vytvářejícího LM haš, tak pomocí algoritmu vytvářejícího NT haš. Standartně jsou oba tyto haše uloženy v systémových registrech systému Windows. Na pevném disku počítače je tato část registů reprezentována souborem SAM (Security Account Manger), který je standartně uložen ve složce C:\WINDOWS\system32\config\sam.

LM haš se v systémech Windows používal k ukládání hesel do verze Windows NT SP3. Ve vyšších verzích systému Windows (NT SP4 a další, XP ,Vista) byl zachován z důvodu zpětné kompatibility systémů Windows. Z bezpečnostních důvodů se doporučuje potlačit ukládání hesla pomocí LM haš.

Windows Vista má ukládání hesla pomocí LM haš standartně vypnuto, avšak je možné jej aktivovat. [2]

2.2.1 LM haš

umožňuje sestavit heslo s maximální délkou 14 znaků. I když tato délka teoreticky umožňuje sestavit dostatečně složité heslo, operační systém Windows provádí určitá zjednodušení, která tento prostor omezují. Zpracování uživatelského hesla na haš, který je uložen do systémových registrů, probíhá následujícím způsobem: [3]

1) Uživatelské heslo je převedeno na velké znaky.
2) Heslo je doplněno znaky NUL na délku 14 znaků nebo je na tuto délku zkráceno.
3) Heslo s pevnou délkou je rozděleno na dvě poloviny.
4) Tyto poloviny se použijí jako klíče pro algoritmus DES. Klíče pro DES jsou vytvořeny tak, že znaky hesla jsou převedeny na řetězec bitů kde každý osmý bit je nulový. Takto získáme potřebné dva vstupní klíče o délce 64 bitů.
5) Pomocí DES algoritmu se pak zakóduje řetězec ASCII znaků KGS!@#$%. Toto se provede pro každou polovinu původního hesla zvlášť.

Výstupy z DES jsou 2 řetězece o délce 64bitů. Tyto jsou uloženy do systémových registrů Windows ve formě řetězce šestnácti ASCII znaků. Jedná se o tzv. LM haš.

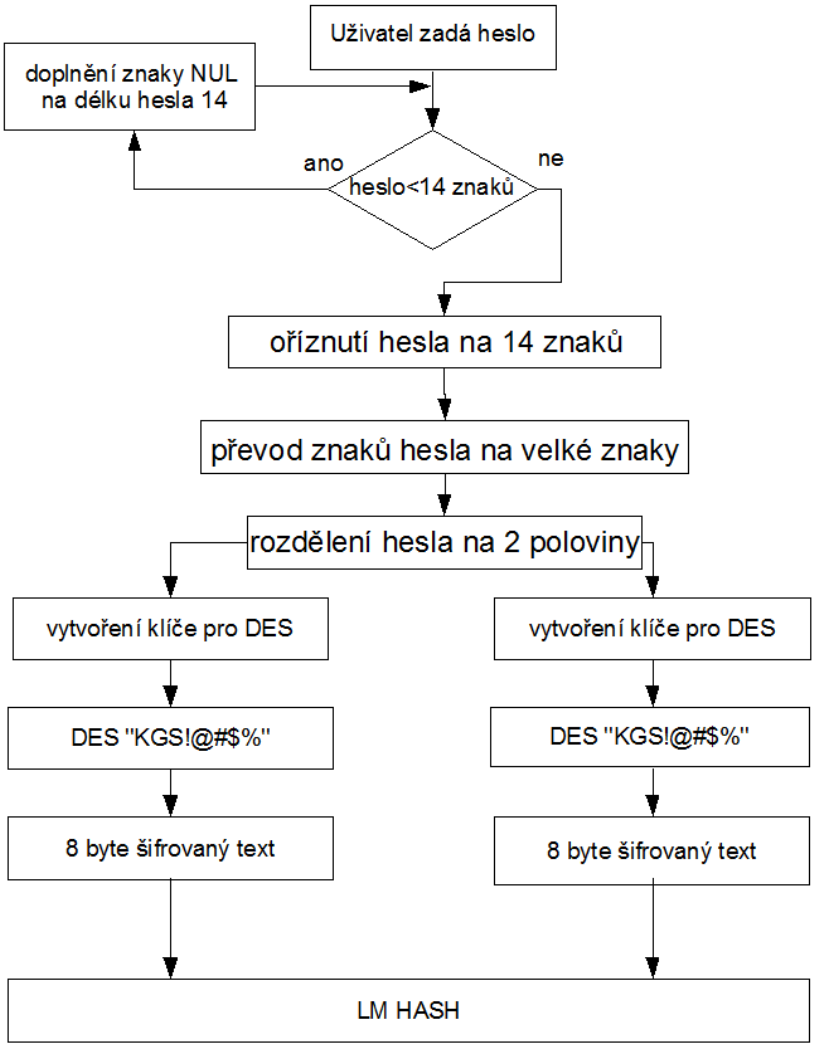
Slabiny LM haš

Prvním problémem LAN Manager (LM) haše je to, že se heslo před kryptografickým zpracováním rozdělí na 2 poloviny. Rozdělení textu hesla na dvě části před kryptografickým

zpracováním umožňuje zaútočit na každou polovinu hesla zvlášť. Pokud se heslo skládá ze čtrnácti ASCII znaků existuje 95^{15} možných kombinací. Pokud heslo bude složeno ze sedmi ASCII znaků existuje pouze 95^{7} kombinací. I přes to, že máme 2 řetězce o délce 7 znaků je, počet možných kombinací mnohem menší než u jednoho řetězce o délce 14 znaků.

Druhým problémem LM haš je zpracování hesla, které nezohledňuje velikost jednotlivých písmen hesla. Operační systém převede všechny znaky hesla na velké. Tento krok negativně omezuje konečný tvar hesla a umožňuje při útoku významně redukovat množinu testovaných znaků. Pokud omezíme ASCII abecedu pouze na velké znaky snižuje se počet kombinací pro jednu polovinu hesla na 69^{7}. [1]

LM haš představuje nejslabší formu zabezpečení hesel v operačních systémech Windows. Při výpočetním výkonu dnešních osobních počítačů je možné alfanumerický LM haš prolomit metodou hrubé síly v řádu hodin. Nicméně je také možné použít tabulky předpočítané tabulky hašů a potom prolomení hesla trvá řádově minuty. [4]



Obrázek 2.2: Blokové schéma vzniku LM Haš

2.2.2 NT haš

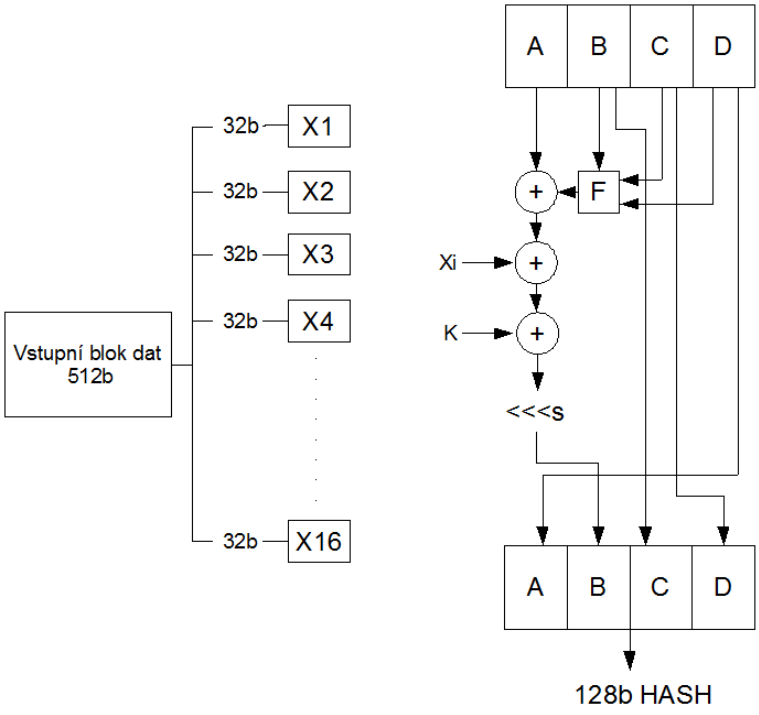
je generován pomocí rozšířeného a veřejně dostupného algoritmu Message Digest 4 (MD4). Heslo může mít délku až 128 Unicode znaků. Tento haš se používá k bezpečnému uložení otisku hesla u systémů Windows od verze Windows NT SP4. Výpočet NT haš probíhá následovně:

1) Heslo je převedeno do Unicode, bez ohledu na to, zda obsahuje i jiné znaky než ASCII. Proto je každý znak reprezentován šestnácti bity. Unicode řetězec má na konci znak NULL (0x00).
2) Pomocí algoritmu MD4 je z Unicode hesla vytvořen haš, který má délku 128 bitů. Jedná se o tzv. NT haš.

Message Digest 4 (MD4)

je typ hašovací funkce, který je v dnešní době již zastaralý. Jedná se o předchůdce algoritmu MD5, se kterým má podobnou strukturu. Zpracovává bloky o délce 512b, přičemž pokud není původní zpráva dělitelná 512, tak dochází k doplnění výplňovými bity. Každý blok se rozdělí na 16 32-bitových částí, které jsou vstupem do jednotlivých operací. Celkem je provedeno 48 operací, struktura operace je uvedena na obrázku 2.3.

ABCD je obsah registru, který je na počátku naplněn inicializačním vektorem. Velikost jednoho pole je 32b. F jsou různé funkce polí B C D, jejíž výstup je XOR přičten k hodnotě A. Následuje přičtení hodnot vstupních částí bloků a konstanty. Na závěr se provede rotace a promíchání pozic v registru. Po provedení všech 48 operací je výsledný haš roven hodnotám v registru, tj. délka haše je 4 x 32b = 128b.



Obrázek 2.3: Blokové schéma MD4

Slabiny NT haše

Protože je MD4 veřejně dostupným algoritmem, byl podroben důkladnému zkoumání a byla u něj nalezena řada bezpečnostních slabin. Přesto, že je NT haš silnější než LM haš, je náchylný na různé útoky včetně útoků slovníkových a útoků hrubou silou. MD4 není bezkolizním algoritmem a existují také další útoky [5] specifické pro MD4. Mimo jiné k snadnějšímu prolomení NT haš připívá absence soli v tomto algoritmu.

3 LINUX

Obecně se jedná o operační systém šířený volně vycházející z koncepce operačních systémů *UNIX*. Jádro systém LINUX vytvořil Linus Torvalds v roce 1991 pro procesor Intel 80386. Postupem času začaly systém rozvíjet skupiny programátorů z celého světa. V roce 1996 bylo jádro systému přepracováno do plnohodnotné podoby. Postupem času došlo k vývoji několika různých distribucí. V této práci byli zkoumány dvě distribuce a to Debian 5 a Ubuntu 9.04, která z Debianu vychází.

Debian 5

Debian je svobodný operační systém určený k provozu na mnoha různých typech počítačů. Operační systém se skládá ze základního programového vybavení a dalších nástrojů, kterých je k provozu počítače třeba. Vlastním základem OS je jádro. Jelikož Debian používá jádro Linux a většina základních systémových programů byla vytvořena v rámci projektu GNU, nese systém označení GNU/Linux.

Debian GNU/Linux je však více než jen samotný operační systém. Obsahuje přes 25113 balíčků s (předkompilovanými) programy a dokumentací, připravených pro snadnou instalaci. [6]

Ubuntu 9.04

Ubuntu je linuxová distribuce pro pracovní stanice, založená na Debian GNU/Linux. Ubuntu je projekt sponzorovaný společností Canonical Ltd (vlastněnou Markem Shuttleworthem) a název distribuce je odvozený z jihoafrického pojmu Ubuntu znamenajícího přibližně „lidskost ostatním“. Na rozdíl od Debianu má pravidelné zveřejňování nových verzí, každých 6 měsíců, s podporou na dalších 18 měsíců; tímto způsobem se Ubuntu snaží poskytnout aktualizovaný a rozumně stabilní operační systém pro běžného uživatele s použitím Svobodného softwaru. [7]

3.1 Ukládání hesla v LINUXu

Na začátku systémy Unixového typu používaly k zašifrování hesla funkci *crypt()*. Funkce crypt akceptuje 2 vstupní parametry: klíč a sůl. V tomto případě je klíč reprezentován uživatelským heslem. Sůl vzniká jako náhodně generovaný řetězec typu string o délce 2 znaky, který se využívá k pozměnění DES algoritmu. Ze soli se používá pouze prvních 12 bitů, což umožňuje 4096 různých variací DES algoritmu. Šifrování hesla pomocí funkce crypt vypadá následovně:

1) Vytvoří se prázdný řetězec obsahující pouze znaky NUL.
2) Vybere se prvních sedm bitů z prvních osmi znaků uživatelského hesla.
3) Z bitů vybraných v předchozím bodě se vytvoří 56 bitový klíč pro DES šifrování.
4) Prázdný řetězec se zašifruje pomocí DES s 56 bitovým klíčem; použije se sůl k pozměnění DES algoritmu.
5) Krok 4 se opakuje 25 krát.

Výsledkem crypt() je řetězec pevné délky, ve kterém je zašifrované uživatelské heslo. Tento řetězec se skládá z následujících znaků: [a-z][A-Z][0-9]./. Nakonec je uložen do souboru **/etc/passwd** nebo **/etc/group** (v novějších verzích OS LINUX se používá **/etc/shadow** nebo **/etc/gshadow**. První dva znaky řetězce se šifrovaným heslem obsahují původní sůl (nešifrovanou), takže je možné při pozdější autentizaci přesně určit způsob jakým byl DES algoritmus pozměněn při šifrování hesla. Použití soli zajistí to, že pokud si dva uživatelé zvolí identické heslo, šifrovací funkce vytvoří dva různé výstupy. [3]

Obrázek 3.1: Výstup funkce crypt s použitím DES

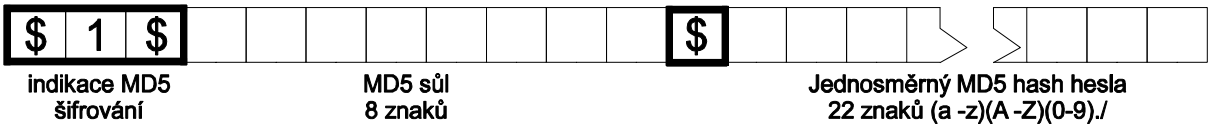
Nevýhodou této původní implementace funkce crypt je to, že k vytvoření klíče pro DES algoritmus se používá pouze prvních osm znaků hesla. Toto představuje zjevné bezpečnostní omezení. Novější implementace funkce crypt () k původnímu DES přidávají další dodatečné zabezpečení. Většina dnešních implementací používá MD5 jako doplněk k původnímu DES. Na rozdíl od DES je MD5 jednosměrnou funkcí. Pokud nějaká aplikace (například příkaz **passwd**) vytvoří osm znaků dlouhou sůl (použitou jako inicializaci pro MD5 algoritmus) a poskytne ji jako parametr příkazu crypt v následujícím formátu **1<osm znaků soli>[$]**, potom funkce crypt() použije k vytvoření zašifrovaného hesla MD5 algoritmus místo DES. Vytvoření hesla s využitím MD5 probíhá následovně:

1) Heslo je doplněno tak, aby odpovídalo délce 448 modulo 512. Je přidán nejméně jeden bit s hodnotou 1 a žádný nebo více bitů s hodnotou 0.
2) K heslu je přidáno 64 bitové slovo, které představuje originální délku hesla
3) Aplikací MD5 na sůl a uživatelské heslo je vygenerován haš

Je důležité si uvědomit, že při použití MD5 se použije heslo v celé délce a nikoliv jen prvních 8 znaků jako je tomu při použití DES. Proto jsou hesla šifrovaná pomocí MD5 silnější.
Existují i další implementace umožňující k šifrování hesel použít i jiné algoritmy, jako například: Blowfish, SHA512. Nicméně nejrozšířenějším algoritmem používaným k šifrování hesel v Linuxech je MD5. [3]

Jak je uvedeno výše Linux ukládá hesla v šifrované formě. V dnešní době se jako další přídavné zabezpečení používají tzv. stíněná (shadowed) hesla. Informace o uživatelských účtech a jejich heslech obvykle uloženy v souboru **/etc/passwd** čitelném pro všechny uživatele. Pokud je aktivováno stínění (shadowing) je šifrované heslo v souboru **/etc/passwd** nahrazeno speciálním symbolem a šifrované heslo je uloženo v odděleném souboru, který není čitelný pro běžné uživatele. Tímto odděleným souborem je obvykle **/etc/shadow**, ke kterému má přístup pouze správce systému (root).

Při základním nastavení Debian 5 jsou údaje o uživatelských účtech uloženy v **/etc/passwd** a haše hesel získané použitím algoritmu MD5 s 8 byte soli jsou uloženy souboru **/etc/shadow**.



Obrázek 3.2: Výstup funkce crypt s použitím MD5

4 BSD SYSTÉMY

4.1 FreeBSD

je svobodný operační systém, který vznikl z BSD Unixu na Univerzitě v Berkley v roce 1993. FreeBSD je vyvíjen jako kompletní operační systém, jádro, ovladače zařízení a všechny uživatelské nástroje jsou vyvíjeny ve stejném stromu systému pro správu verzí zdrojových kódů (CVS). V tomto se FreeBSD odlišuje od Linuxu, kde je typicky vyvíjen každý program jinou skupinou vývojářů a ty jsou poté vydány jako kompletní operační systém (Distribuce). Aktuální verze je FreeBSD 7. [8]

Implementace hesel ve FreeBSD
je shodná s implementací popsanou v kapitole o Linuxových systémech.

4.2 OpenBSD

je svobodný operační systém z rodiny BSD, který vznikl jako tzv. fork projektu NetBSD v roce 1995. Jeho autoři kladou *maximální důraz na bezpečnost* a na programy bez děr. Cílem projektu je vytvořit svobodný a extrémně bezpečný operační systém pro nejrůznější platformy. OpenBSD podporuje emulaci většiny binárních programů ze Solarisu, FreeBSD, Linuxu, BSD/OS a SunOS. Aktuální verze OpenBSD je 4.4. [8]

Implementace hesel v OpenBSD
Již od verze 2.1 OpenBSD používá funkci **bcrypt()**, která je založena na šifrovacím algoritmu Blowfish. Bcrypt používá 128 bitů soli. Sůl se generuje pomocí funkce *arc4random(3)* z klíčového toku (key stream) vycházejícího z náhodných dat sbíraných jádrem systému a šifruje 192 bitů dlouhou magickou hodnotu. Algoritmus využívá výhod náročného ustanovení klíčů (Expensive Key Setup)*EksBlowfish*.

Algoritmus bcrypt je zde dvoufázový. V první fázi se jako inicializace *eksblowfish* stavu volá *EksBlowfishSetup* s cenou (cost), solí, a heslem. Většinu času algoritmus stráví řazením náročných klíčů (expensive key). Je to v důsledku toho, že 192 bitů dlouhá hodnota OrpheanBeholderScryDoubt je 64 krát šifrována pomocí EksBlowfish v režimu ECB se stavem z předchozí fáze. Výstupem funkce je cena a 128 bitová sůl zřetězená s výsledkem šifrovací smyčky. [9]

V současné době se šifrovací algoritmy založené na Blowfish šifrování považují za velmi bezpečné. Jednou z výhod implementace bcrypt je vzájemná kompatibilita souborů hesel z minulých i budoucích verzí (do budoucna je možné zvyšovat náročnost výpočtu hesla).

5 MAC OS X

Mac OS X je v informatice aktuální operační systém pro počítače Macintosh. První Mac OS X v10.0 byla vydána 24. března 2001. Vznikl jako kombinace několika různých technologií. Základ systému se jmenuje Darwin a je složen z hybridního unixového Jádro XNU (anglicky XNU's Not Unix) spolu s množstvím BSD, GNU a dalších open source nástrojů. Nad jádrem je množina knihoven, služeb a technologií, které jsou přejaty většinou z NeXTSTEPu a předchozího operačního systému Mac OS. Aktuální verze OS X je 10.5.

5.1 Implementace hesel v OS X

Nejzřetelnějším rozdílem oproti Unixovým implementacím (který je společný pro všechny verze OS X) je to, že informace o uživatelích nejsou uloženy v souboru jako například /etc/password, ale v databázi *NetInfo*. Historie databáze NetInfo začíná až u samotného *NEXSTEP*. Její koncepce je podobná NIS (Network Information Service) od SUNu a také LDAP (Lightweight Directory Access Protocol) protokolu. Všechny tyto vzájemně nekompatibilní řešení ukládají informace do databáze místo do souborů, čímž dochází ke zlepšení výkonu a škálovatelnosti. V současnosti již je nejlepším systémem LDAP, ale na deskopových verzích *OS X* stále zůstává NetInfo. NetInfo poskytuje určité UNIX kompatibilní funkce. Například příkaz na vypsání uživatelské databáze NetInfo do UNIX formátu souborů [10]

```
% nidump passwd
```

5.2 OS X 10 až 10.2

V *OS X 10* až *10.2*, bylo hašované heslo uloženo přímo do NetInfo. Také zde byl využíván standardní UNIX DES haš. Protože všichni uživatelé mají přístup ke kompletní databázi NetInfo, každý uživatel může spustit příkaz nidump zmiňovaný výše a tím získat DES haše všech hesel uživatelů systému. Řekněme, že máme fiktivního uživatele s přihlašovacím jménem sjobs a heslem macintosh: [10]

```
% nidump passwd . | grep sjobs
sjobs:3dI880QaIz.Wk:501:501::0:0:sjobs:/Users/jobs:/bin/bash
```

Jak je vidět, hesla nejsou stínována (shadowed) a využívá se hašovací algoritmus jako u AT&T Unixu verze 7, který byl vydán v roce 1979. Do algoritmu nebylo zahrnuto žádné zlepšení, se kterým Unixový svět přišel během následujících dvacetipěti let. Verze OS X 10.2 byla vydána v roce 2002. Stíněná hesla byla poprvé představena firmou AT&T v roce 1987. A v roce 1997 bylo na RSA's DES Challenge spolehlivě prokázáno, že DES algoritmus je slabý. DES také zkracuje hesla na 8 znaků, což znamená, že je téměř nemožné používat kódovou

větu (passphrase) místo hesla. Můžeme si to ověřit například použitím nástroje openssl pro příkazovou řádku: [10]

```
% openssl passwd -crypt -salt 3d macintosh
Warning: truncating password to 8 characters
3dI880QaIz.Wk
```

5.3 OS X 10.3

V roce 2003 společnost Apple vydala *OS X 10.3*, který se pokusil tyto nedostatky odstranit. Nejvýznamnější změnou bylo to, že hesla jsou ukládána jako stíněná a jsou uložena úplně mimo NetInfo. Když se uživatel, který nemá patřičná oprávnění, pokusí vypsat databázi uživatelů NetInfo už nezíská heslo: [10]

```
% nidump passwd . | grep sjobs
sjobs:********:501:501::0:0:sjobs:/Users/jobs:/bin/bash
```

Místo standardního souboru /etc/shadow se pro uložení hašů používá adresář **/var/db/shadow/hash/** . Stejně jako v případě unixového /etc/shadow je tento soubor v OS X 10.3 čitelný pouze pro správce (root). Nicméně heslo každého uživatele je uloženo ve zvláštním souboru. Jméno souboru není stejné jako jméno uživatele, ale je shodné s polem *uid* databáze NetInfo. Toto pole z NetInfo je možné zobrazit nástrojem *niutil*:

```
% niutil -readprop . /users/sjobs generateduid
70902C33-AC79-11DA-AFDF-000A95CD9AF8
```

Obsahem tohoto souboru je:

```
% hash_dir=/var/db/shadow/hash
% uid=70902C33-AC79-11DA-AFDF-000A95CD9AF8
% hash_filehash_dir/$uid
% sudo more $hash_file
D47F3AF827A48F7DFA4F2C1F12D68CD608460EB13C5CA0C4CA9516712F7FED9501424F955
C11F92EFEF0B79D7FA3FB6BE56A9F99
% sudo cat $hash_file | wc -c
104
```

Je zřejmé, že už se nejedná o standardní UNIX DES. Ukazuje se, že string o délce 104 znaků je hašované dvakát: jednou pomocí SHA1 a poté ještě jednou pomocí Windows LM haš. [10]

Prvních 64 znaků LM haš:

```
% sudo cat $hash_file | cut -c1-64
D47F3AF827A48F7DFA4F2C1F12D68CD608460EB13C5CA0C4CA9516712F7FED95
```

A posledních 40 znaků SHA1 haše:

```
% sudo cat$hash_file | cut -c65-104
01424F955C11F92EFEF0B79D7FA3FB6BE56A9F99
```

SHA haš je možné ověřit pomocí nástroje pro příkazový řádek openssl:

```
% echo -n macintosh | openssl dgst -sha1
01424f955c11f92efef0b79d7fa3fb6be56a9f99
```

Jak využití stínování, tak využití SHA1 jsou krokem kupředu. SHA1 je opravdovým hašem a proto zde není žádné omezení délky. SHA1 na rozdíl od DES není jednoduché spočítat dokonce ani s využitím hardwaru. Naneštěstí se Apple rozhodlo nepoužít zasolení (salt). Zasolení sice nezkomplikuje uhodnutí hesla, ale podstatně ztíží provedení slovníkového útoku nebo útoku pomocí hrubé síly. [10] Každopádně absence zasolování není nejhorším problémem verze 10.3. Největší problém spočívá ve využití LM haše pro hašování hesel. Dnes již je známo, že LM haš je velmi slabý algoritmus. Stručné připomenutí jeho nedostatků:

- Nezohledňuje velikost písmen.
- Omezení hesla na 14 znaků, které je efektivně zkráceno na 7.
- Jednoduchý na výpočet.
- Absence zasolení.

Jaký byl vlastně důvod k použití LM haš? Důvodem byla interoperabilita s Windows systémy. Použití Windows kompatibilního hesla umožňuje například namapovat přes síť domovskou složku ze stroje běžícího na Windows. Bohužel OS X 10.3 vytváří Windows kompatibilní hesla nezávisle na tom, jestli je sdílení Windows zapnuto nebo vypnuto. Toto chování bohužel zcela eliminuje výhody, které přináší hašování pomocí silnějšího algoritmu SHA1. Chytrému útočníkovi stačí zaútočit na LM haš namísto SHA1. LM haš byl u Windows používán jen do verze Windows ME, poté byl nahrazen silnějším algoritmem NT haš. OS X ho poměrně nelogicky používal i nadále.

5.4 OS X 10.4

Další zlepšení přišlo až s verzí OS X 10.4 vydanou v roce 2005. Zdá se, že všechny změny ve verzi 10.4 byly pozitivní. Hesla jsou stále stíněná a uložena ve stejném souboru jako ve verzi OS X 10.3 . LM haš se zde používá jen pokud je zapnuto sdílení Windows. Pokud dojde k aktivaci sdílení Windows, uživatel je systémem upozorněn na to, že jeho heslo bude uloženo méně bezpečným způsobem. Apple navíc přidal sůl (salt) do SHA1 haše. Formát haš souboru se také změnil: [10]

```
% sudo more $hash_file
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000E6A48F765D0FFFFF6247FA80D748E615F91DD0C7431E4D9
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000

% sudo cat $hash_file | wc -c
1240
```

Zasolený SHA1 haš začíná na znaku 169 a je 48 znaků dlouhý:

```
% salted_hash=`sudo cat $hash_file | cut -c169-216`
% echo $salted_hash
0E6A48F765D0FFFFF6247FA80D748E615F91DD0C7431E4D9
```

Prvních 8 znaků je hexa hodnota 4 byte soli. K ověření tohoto hesla je nejdříve nutné převést sůl na hodnotu typu integer:

```
% hex_salt=`echo $salted_hash | cut -c1-8`
% salt=`echo -n $hex_salt | xxd -r -p`
```

Nyní je potřeba vzít SHA1 haš soli a otevřenou podobu hesla:

```
% sha=`printf "%s%s" $salt macintosh | openss l dgst -sha1`
% printf "%s%sn" $hex_salt $sha
0E6A48F765d0fffff6247fa80d748e615f91dd0c7431e4d9
```

Jak je vidět, toto je stejná hexa hodnota jako $salted_hash, výše. Kde se tedy vzaly všechny další nuly ve stínovaném souboru verze 10.4? Prvních 64 znaků je použito pro LM haš hesla, pokud je zapnuto sdílení souborů Windows. Dalších 40 znaků se využívá pro nezasolený

SHA1 haš, pokud se jedná o účet převedený z verze OS X 10.3. Další nuly jsou pravděpodobně využity pro kompatibilitu s budoucími verzemi OS X. [10]

6 METODY ÚTOKŮ NA HESLA

Existuje celá řada různých technik útoků na hesla. Pokusím se zde vyjmenovat a stručně popsat několik nejpoužívanějších útoků na haše hesel. Útoky na hašovaná hesla lze v podstatě rozdělit na 4 typy: útoky hrubou silou (brute force attack), slovníkové útoky (dictionary attack), útoky s využitím předvypočtených hašů (rainbow tables) [4] a útoky pomocí kryptoanalýzy.

Útok hrubou silou
V principu program vezme všechny možné kombinace znaků dané abecedy a zkouší je hašovat, pokud dostane stejný haš jako hledaný, pravděpodobně bylo nalezeno heslo. Tento útok sice časem určitě najde správné heslo, ale čas potřebný k nalezení hesla může představovat i tisíce let. Teoreticky program postupuje od jednoho znaku a přidává další znaky až do nekonečna. Právě proto, že čas pro hledání hesla může být velmi dlouhý, snaží se různé programy určené pro hledání hesla touto metodou o zefektivnění hledání a proto skutečné implementace toků hrubou silou nepostupují přesně podle teoretického modelu. Účinnost a délka trvání je ovlivněna znalostmi, které má útočník o hesle. Pokud útočník zná například přesný počet znaků hesla, nebo ví, že v hesle nejsou speciální znaky. Může se doba hledání hesla významně zkrátit.

Slovníkový útok
vychází z určitého předem daného slovníku (například jmenný kalendář, překladový slovník, atd.). Algoritmus hledající hesla vezme jednotlivá slova a zkouší je hašovat, pokud najde shodu s hašem, který útočník zná, pravděpodobně bylo nalezeno heslo. Pokročilejší nástroje jednotlivá slova ze slovníku kombinují popřípadě přidají několik znaků před nebo za slovo => *hybridní útok* (kombinace slovníkového útoku a brute force). Ať už je použit slovníkový nebo hybridní útok, zkouší mnohokrát méně různých kombinací vstupního řetězce než při útoku hrubou silou a proto bývá většinou rychlý. Při slovníkovém útoku, je prolomení haše závislé na tom zda slovník obsahuje hledané heslo, nebo alespoň vhodný základ pro mutaci hybridním útokem.

Útoky s využitím rainbow tables
Jedná se o analytickou techniku, která slouží k určení hesla z haše. Princip spočívá v tom, že útočník má k dispozici předvypočtené tabulky hašů z jakéhokoliv myslitelného hesla (v praxi je omezen kapacitou úložiště pro haše a schopností hardware počítat haše). Poté, co útočník získá haše hesel z napadeného stroje, pouze porovná předvypočtené haše se získanými. V případě nalezení shody hašů získá útočník heslo, protože ve svých předvypočtených tabulkách má uloženo slovo ze kterého haš vznikl. [4]

Výhodou této techniky je obrovská rychlost (maximálně desítky minut) a vysoká úspěšnost útoku (rychlost a úspěch závisí na kvalitě lámaného hesla, kvalitě použitých tabulek hašů a výkonu použitého počítače).

Obvykle se tato technika využívá k útoku na systémy Windows a jejich LM haš a NT haš. V principu je možné ji použít i na jiné haše, ale pokud haše obsahují zasolení počet předvypočtených hašů začne enormně růst a vytvoření tabulek přestává být realizovatelné.

Útoky s využitím kryptoanalýzy

Princip útoku s využitím kryptoanalýzy spočívá v nalezení slabiny konkrétního použitého šifrovacího algoritmu a následném využití této slabiny k získání či pozměnění obsahu utajované zprávy.

7 ÚTOKY NA HESLA V PRAXI

Praktickou částí této diplomové práce je útok na hesla uložená v současně používaných operačních systémech. Útoky byly zaměřeny zejména na hašovací funkce hesel. Hlavním cílem útoků bylo zjistit, z jakého řetězce znaků haše vznikly.

Zkoumané systémy

Windows: Microsoft Windows 98, Windows XP SP3, Windows Vista Ultimate, Windows Seven Beta Build 7077

Linux: Debian 5, Ubuntu 9.04 Beta

BSD: FreeBSD 7.1, OpenBSD 4.4

OS X: 10.5 (Leopard)

Vzhledem k tomu, že testovaných systémů je větší množství a tudíž by nebylo možné mít potřebné množství skutečných počítačů, byly instlace jednotlivých operačních systémů provedeny na virtuální stroje. Jako virtualizační nástroj byla využita zkušební verze software firmy VMware, VMware Workstation verze 6.5.2 pro Windows.

7.1 Extrakce hašů hesel ze systémů

K tomu, aby bylo možné prolamovat haše vytvořené operačními systémy, bylo nejprve zapotřebí vytvořit na virtuálních strojích uživatelské účty a hesla (viz příloha A). Následně byly uložené haše ze systémů získány a uloženy pro další zkoumání.

7.1.1 Získávání hašů hesel z Windows 98

Windows 98 ukládá hesla do souborů s příponou PWL (**P**ass **W**ord **L**ist). Obsahem těchto souborů jsou cenné informace, jako například hesla k vytáčenému připojení, síťová hesla atp. Oficiální dokumentace tohoto souboru od firmy Microsoft není dostupná.

Jinými slovy je PWL soubor zabezpečená databáze, která obsahuje tři pole:

1) Typ prostředku (0...255)
2) Jméno prostředku
3) Heslo prostředku

Jak jméno, zdroje tak i heslo zdroje mohou být binární. Jeden PWL soubor může obsahovat maximálně 255 záznamů. Každý uživatel má svůj vlastní soubor. Všechny záznamy spolu

s uživatelským jménem a kontrolním součtem jsou zašifrovány algoritmem RC4. Klíč pro šifrování je odvozen z přihlašovacího hesla. Nicméně přihlašovací heslo není v PWL souboru uloženo. Windows dešifruje PWL soubor pomocí zadaného hesla a porovná kontrolní součet z PWL souboru proti kontrolnímu součtu, který sám vypočte. Pokud se kontrolní součty shodují, je zadané heslo správné. Je také nezbytné znát uživatelské jméno k účtu, protože ovlivňuje ověřování kontrolního součtu PWL souboru. Obvykle je uživatelské jméno stejné jako název PWL souboru. Neplatí to ovšem vždy. Název PWL souboru nesmí být delší než 8 znaků, na rozdíl od jména uživatelského účtu. Navíc Windows nikdy nepřepisuje PWL soubory, takže pokud máme například název účtu mixednumaspec03 je PWL soubor pojmenován mixed000.pwl.

Windows 98 měl na svou dobu z kryptografického hlediska poměrně dobře zvládnuté a spolehlivé ukládání hesel, přesto obsahuje některé prvky, které jejich bezpečnost snižují:

1. Všechna hesla jsou převedena na velké znaky, což významně snižuje celkový počet možných hesel a zvyšuje rychlost jejich hledání.
2. Algoritmy MD5 a RC4 jsou profesionální k dešifrování odolné rychlé algoritmy, ale právě jejich rychlost je to, co umožňuje rychlé hledání hesel hrubou silou.

Samotná extrakce PWL souborů je velmi snadná, stačí zkopírovat sobory s maskou *.pwl ze složky Windows (obvyklá cesta k souborům je C:\WINDOWS\) [11]

7.1.2 Získávání hašů hesel z Windows SAM

Databáze Windows SAM (Security Account Manger) je využívána všemi současnými verzemi Windows. Jmenovitě Windows XP, Windows Vista a Windows 7.

Windows SAM slouží pro uložení informací o lokálních uživatelských účtech. Jak již bylo zmíněno dříve, hesla jsou v této databázi uložena ve formě hašů. Fyzicky lze soubor s databází nalézt obvykle v adresáři C:\WINDOWS\system32\config\sam.

Dalším stupněm ochrany hesel uživatelských účtů šifrování databáze SAM nástrojem SYSKEY. SYSKEY byl přidán jako doplňkový prvek do Windows NT 4.0 SP3. Jeho původním účelem bylo ochrana hesel před offline útoky. Hesla by měla být v bezpečí, i když má útočník kopii databáze SAM. SYSKEY využívá šifrovací klíč o délce 128bitů, který je odvozen ze čtyř hodnot (JD, Skew1, GBG a Data) obsažených v registrech sytému v položce HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\LSA. [12] [3]

Chceme-li tedy zjistit uživatelská hesla do systému Windows, je nutné získat kopii SAM a zjistit hodnotu SYSKEY, aby bylo možné SAM dešifrovat.

LiveCD BackTrack 4

BackTrack je populární linuxová distribuce zaměřená na penetrační testování operačních systémů. Systém je vydáván jako LiveCD či LiveUSB a proto uživateli stačí jen vložit příslušné médium a zavést systém do BackTrack bez jakékoliv instalace do počítače.

BackTrack je možné získat na jeho domovských stránkách: http://www.remote-exploit.org/backtrack_download.html.

bkhive a samdump2 v1.1.1
Beta verze BackTrack 4 obsahuje přímo v sobě nástroj bkhive, sloužící pro extrakci klíče SYSKEY a samdump2 verze 1.1.1, který na základě získaného SYSKEY extrahuje ze SAM souboru názvy uživatelských účtů a hašů jejich hesel.

1. Pokud je SYSKEY uložen lokálně je potřeba získat jej z registrů, aby na jeho základě bylo možné dešifrovat SAM soubor.

   ```
   # bkhive /mnt/XXX/WINDOWS/system32/config/system syskey.txt
   ```

   XXX v příkazu je třeba nahradit aktuální cestou k připojenému oddílu Windows (BackTrack 4 Beta připojí oddíl Windows automaticky). Výsledkem příkazu je textový soubor obsahující hodnotu SYSKEY.

2. Program samdump2 vypíše obsah SAM na obrazovku a pomocí znaku > je jeho výstup přesměrován do souboru haš.txt.

   ```
   # samdump2 /mnt/XXX/WINDOWS/system32/config/sam syskey.txt > hash.txt
   ```

   Obsahem souboru hash.txt jsou názvy uživatelských účtů a hašů k nim patřících hesel.

Software byl testován na Windows XP SP3, Windows Vista SP1 a Windows Seven Beta build 7077. Ve všech třech případech byly haše vyextrahovány. Při zadávání cesty do příkazů je potřeba správně zadat malé a velké znaky v cestě k souborům.

samdump2 v2.0
Aktuální verze aplikace samdump2 je 2.0.1. Jejím přínosem je schopnost vyextrahovat si SYSKEY sama. To znamená zkrácení procesu získávání hašů hesel na jediný příkaz. Verze 2.0.1 bohužel zatím není součástí BackTrack, ale je možné ji snadno použít i na LiveCD. Upgrade na verzi samdump2 na verzi 2.0.1

1. Stáhneme samdump2 z http://sourceforge.net/project/showfiles.php?group_id=133599

2. Rozbalíme archiv s programem.

   ```
   # tar -xjvf samdump2-2.0.1.tar.bz2
   ```

3. Přejdeme do nově vzniklého adresáře s programem.

   ```
   # cd samdump2-2.0.1
   ```

4. Zkompilujeme program.

   ```
   # make
   ```

5. Zkopírujeme nově zkompilovanou aplikaci do složek určených v linuxu pro spustitelné aplikace.

   ```
   # cp samdump2 /usr/local/bin/
   ```

6. Pokud je klíč SYSKEY uloženo lokálně samdump2 verze 2 jej sám načte z registrů Windows a dešifruje SAM, jehož obsah vypíše. V případě níže uvedeného příkazu je výstup programu samdump2 verze 2 přesměrován do souboru hash.txt pomocí znaku >.

   ```
   # samdump2 /mnt/XXX/WINDOWS/system32/config/systém
   /mnt/XXX/WINDOWS/system32/config/sam > hash.txt
   ```

Obrázek 7.1: Konzole distribuce BackTrack 4

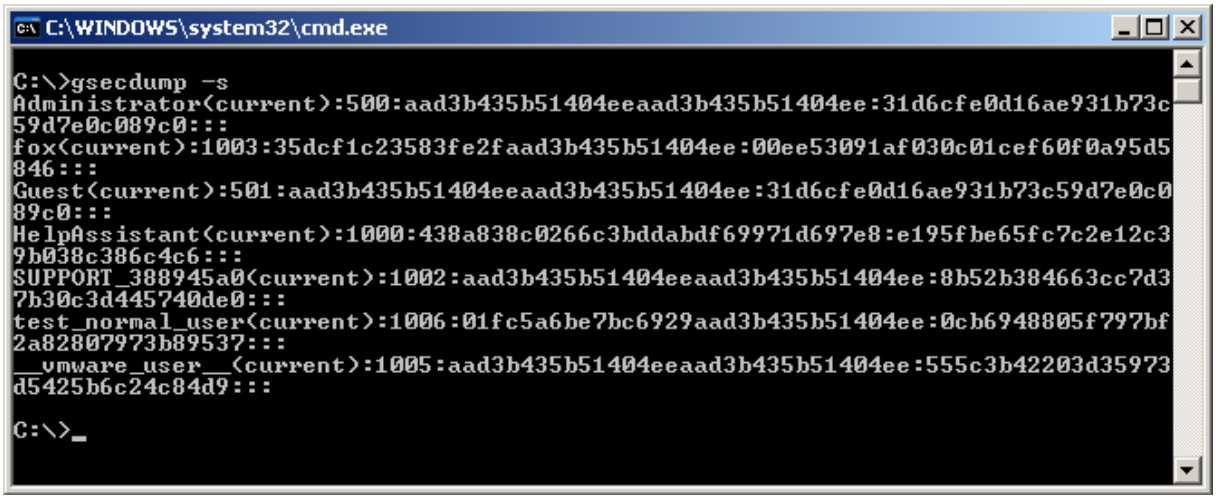
Výhodou distribuce BackTrack je absence nutnosti mít administrátorská práva ke zkoumanému systému.

Nevýhodou je nezbytnost restartu zkoumaného počítače a možnosti spustit tento počítač optického disku či USB flash disku. Tato metoda je také náročnější na čas.

Nástroje běžící na platformě Windows

pwdump

je názvem pro celou skupinu nástrojů, jejichž účelem je získání LM a NTLM hašů hesel ze SAM. Tyto nástroje potřebují pro získání hašů administrátorská práva a proto nejsou samy o sobě vhodné k útoku na systém jako takový.



Obrázek 7.2: Program gsecdump

fgdump

je nástrojem pro získávání hašů hesel ze systému Windows NT/2000/XP/2003. Vznikl jako vylepšení programu pwdump6. Software byl testován na Windows XP SP3, Windows Vista SP1 a Windows Seven Beta build 7077. Ve všech třech případech byly haše vyextrahovány. Hodnotu SYSKEY si program zjistí automaticky. fgdump je možné získat na jeho domovských stránkách: http://swamp.foofus.net/fizzgig/fgdump/downloads.htm

Použití fgdump:

1. Uživatel se přihlásí k systému jako administrátor a spustí příkazovou řádku (Start, Spustit, cmd).
2. Spustit fgdump z místa kam byl stažen (parametr –v zobrazí podrobnější výpis činnosti programu).

   ```
   C:\> fgdump -v
   ```

3. Nakonec si uživatel zkopíruje nově vytvořený soubor 127.0.0.1.pwdump na USB klíč nebo někam jinam pro pozdější prolomení.

pwdump7
představuje nejnovější nástroj z rodiny pwdump, používá vlastní systémový ovladač, což umožňuje uživatelům s administrátorskými právy extrahovat přímo ze systému a SAM. Software byl testován na Windows XP SP3, Windows Vista SP1 a Windows Seven Beta build 7077. Ve všech třech případech byly haše vyextrahovány. Hodnotu SYSKEY si program zjistí automaticky. Program je k dispozici na adrese: http://passwords.openwall.net/dl/pwdump/Pwdump7.zip

1. Uživatel se přihlásí k systému jako administrátor a spustí příkazovou řádku (Start, Spustit, cmd).
2. Spustí pwdump7 z místa kam byl stažen (znak > přesměruje výstup z programu do soubor pwdump7.txt).

   ```
   C:\> pwdump7  > pwdump7.txt
   ```

3. Nakonec si uživatel zkopíruje nově vytvořený soubor pwdump7.txt na USB klíč nebo někam jinam pro pozdější prolomení.

gsecdump
program gsecdump dokáže obdobně jako fgdump získávat haše hesel uživatelských účtů z napadeného stoje lokálně. Oproti fgdump, který fungoval i na novějších systémech jako je Windows Vista a Windows Seven Beta, gsecdump dokáže extrahovat hesla pouze z Windows XP SP3. Hlavní výhodou toho programu by měla být možnost v kombinaci s nástrojem firmy Microsoft psexec získat hesla přes lokální síť. Při testování se tato možnost ukázala jako nefunkční. Program gsecdump je možné získat na adrese: http://www.truesec.com/PublicStore/catalog/categoryinfo.aspx?cid=223

1. Uživatel se přihlásí k systému jako administrátor a spustí příkazovou řádku (Start, Spustit, cmd).
2. Spustit gsecdump z místa kam byl stažen (parametr –s vypíše haše ze SAM, znak > přesměruje výstup z programu do soubor gsecdump.txt).

   ```
   C:\> gsecdump –s  > gsecdump.txt
   ```

3. Nakonec si uživatel zkopíruje nově vytvořený soubor gsecdump.txt na USB klíč nebo někam jinam pro pozdější prolomení.

gsecdump a také ostatní „dump“ programy vytvoří soubor s obsahem v následujícím tvaru:

```
01Num:1004:09752A3293831D17AAD3B435B51404EE:90AD6AB281C4AE016E5A7564C307A7E8:::
01SmallA:1011:7584248B8D2C9F9EAAD3B435B51404EE:186CB09181E2C2ECAAC768C47C729904:::
01Spec:1018:C9E04F959D38CCB0AAD3B435B51404EE:9B0E9CD1456256B6C6A3664BE284E517:::
```

Každý řádek přestavuje jeden uživatelský účet. Položky mají následující význam:

1) Název uživatelského účtu.
2) Identifikační číslo uživatelského účtu.
3) Třetí položkou je LM haš uživatelského hesla.
4) NT haš uživatelského hesla.

Výhodou u těchto programů je možnost rychlého získání hašů ze systému. Nevýhodou je nutnost mít na zkoumaném stroji práva správce. Další nevýhodou těchto programů mohou být jejich kolize s antivirovým software.

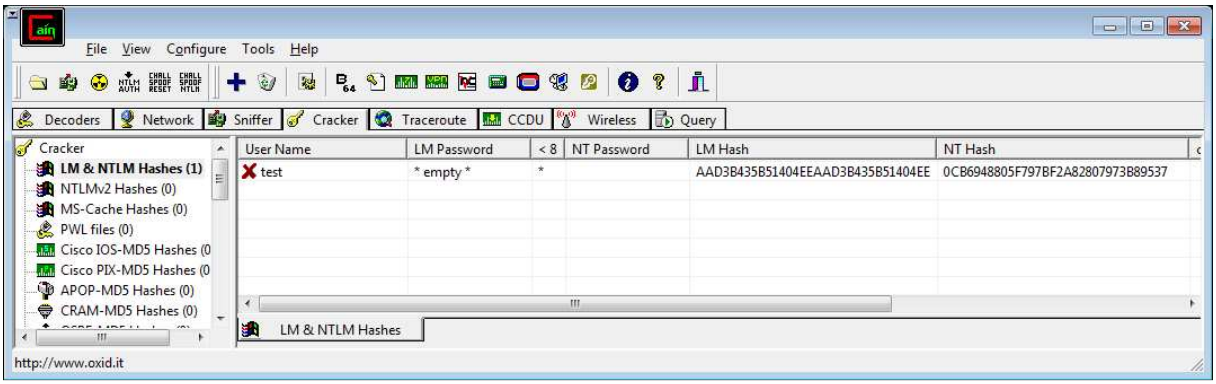
Cain & Abel

Jedná se o velice výkonný nástroj pro odhalení zapomenutých hesel a testování síťové bezpečnosti. Tento nástroj je velice komplexní a obsahuje celou řadu funkcí pro testování bezpečnosti systému. Získání hašů ze systému je pomocí tohoto programu s administrátorskými právy velmi snadné.

Postup extrakce hašů:

1) Spustit cain.exe.
2) Vybrat záložku Cracker.
3) Ve stromu Cracker vybrat položku LM&NTLM Hashes.
4) Přejít do pravé časti okna aplikace a zmáčknout klávesu Insert.
5) Zmáčknout tlačítko next.

Tímto má program načteny haše ze systému, o postupech jejich prolomení bude řeč v následujícím textu. Hodnotu SYSKEY si program zjistí automaticky, v případě potřeby je možné ji i zobrazit. Program je k dispozici na adrese: http://www.oxid.it/cain.html.



Obrázek 7.3: Program Cain & Abel

7.1.3 Získávání hašů hesel z Linuxových systému

Pokud má uživatel práva správce systému (root), je získání hašů hesel snadnou záležitostí. Stačí pouze zkopírovat soubor, kde má systém haše uloženy. V současných distribucích to je obvykle soubor **/etc/shadow**. Pokud uživatel nemá práva super uživatele, je jednou z možností, jak získat haše ze systému nabootování stroje do záchranného režimu (zde je uživatel automaticky root a nemusí znát heslo). Další možností je použít nějaké LiveCD (např. BackTrack 4) nabootovat stroj z něj a haše si zkopírovat. Následuje ukázka části obsahu souboru /etc/shadow z distribuce Debian 5:

```
num01:$1$V1.A70Us$BOeUQY3FiVxhMZT6jhqfU0:14333:0:99999:7:::
num03:$1$eg.3G08u$OU/N4wvAt13zfiqSrc92.1:14333:0:99999:7:::
num05:$1$dGwzPzvN$kTPbg8MSoEBrNIIiKOupl.:14333:0:99999:7:::
```

Jednotlivé řádky představují uživatele a základní údaje o jejich účtech, jednotlivé údaje v řádcích jsou odděleny dvojtečkami. Vezměme například první řádek:

```
num01:$1$V1.A70Us$BOeUQY3FiVxhMZT6jhqfU0:14333:0:99999:7:::
```

1) num01 je názvem uživatelského účtu
2) 1V1.A70Us$BOeUQY3FiVxhMZT6jhqfU0 je hašem uživatelova hesla kde:
    a. 1 udává typ použité hašovací funkce – v tomto případě jde o MD5
    b. V1.A70Us - představuje sůl, která byla použita při hašování hesla (8 byte)
    c. BOeUQY3FiVxhMZT6jhqfU0 – samotný výstup hašovací funkce
3) 14333 – poslední změna hesla (udává se v počtu dní uplynulých od 1. 1. 1970
4) 0 - počet dní, který uplyne než si uživatel může změnit heslo
5) 99999 – počet dní po kterém je uživatel donucen změnit si heslo
6) 7 – počet dní pro upozornění uživatele před vypršením hesla

Sedmá položka (zde prázdná) představuje počet dní po vypršení hesla kdy je účet zablokován
Osmá položka (zde prázdná) představuje datum, po kterém bude účet zablokován (bráno od 1. 1. 1970)

7.1.4 Získávání hašů hesel z BSD systémů

Za předpokladu, že uživatel má práva správce systému (root), je získání hašů hesel bezproblémovým úkonem. Stačí pouze zkopírovat soubor, kde má systém haše uloženy. V současných BSD distribucích to je obvykle soubor **/etc/master.passwd**. Pokud uživatel nemá práva super uživatele, je jednou z možností, jak získat haše ze systému nabootování stroje do záchranného režimu (zde je uživatel automaticky root a nemusí znát heslo). Další možností je použít nějaké LiveCD (např. BackTrack 4) nabootovat stroj z něj a haše si zkopírovat. Soubor master.passwd je podobný jako soubor shadow v Linuxových systémech, ale v některých položkách se liší. Následuje ukázka obsahu souboru master.passwd z distribuce FreeBSD 7.1.

Každý řádek souboru představuje jednoho uživatele. V každém řádku je 10 polí oddělených znakem „:“ (dvojtečka)

```
spec01:$1$sedqaK7X$HJ7Itd9V0fR8tsbRUzCTm1:1016:1016::0:0:spec01:/home/spec01:/bin/sh
spec03:$1$.7EOtGYd$TqwdNDgI1LVtw4T84Y0JK0:1017:1017::0:0:spec03:/home/spec03:/bin/sh
```

Pole v řádcích mají následující význam (vysvětlení pro první řádek):

1) uživatelské jméno účtu
2) haš uživatelova hesla kde:
    a. 1 značí použitou hašovací funkci (MD5)
    b. sedqaK7X je sůl použitá při hašování (8 byte)
    c. HJ7Itd9V0fR8tsbRUzCTm1 je haš hesla
3) identifikační číslo uživatele (UID)
4) uživatelské číslo skupiny (GID)
5) klasifikace uživatele (class)
6) datum změny hesla
7) datum vypršení hesla
8) obecné informace o uživateli
9) domovský adresář uživatele
10) příkazový interpret uživatele

7.1.5 Získávání hašů hesel ze systému OS X 10.5 Leopard

Ve verzi OS X Leopard byl obsah databáze NetInfo přesunut do složky /var/db/dslocal – každý záznam databáze NetInfo je nyní soubor s příponou „plist“ (XML struktura, používaná k reprezentaci strukturovaných dat na Mac OS X platformách).

Obdobně jako u Linuxových a BSD systémů je nejjednodušší cestou k získání hašů spuštění počítače pomocí nějakého LiveCD, nebo v tzv. Single User Mode (není potřeba znát heslo a uživatel má práva root) a následné zkopírovaní potřebných souborů obsahujících haše. Postupovat lze následovně:

1) Spustíme systém v Single User Mode.
2) Připojíme disk se systémem.

    ```
    mount –uaw
    ```

3) Zkopírujeme haše.

    ```
    cp -R /var/db/shadow/hash  /hash
    ```

4) Změníme přístupová práva tak, aby vlastník, skupina i uživatel mohli provádět veškeré operace se soubory.

    ```
    chmod -Rf 777 /hash
    ```

5) Zjistíme, které User ID (UID) patří k účtu správce (soubory s haši nejsou pojmenovány podle uživatelského účtu, ale pomocí systémem vygenerovaného UID).

```
cat /var/db/dslocal/nodes/Default/users/admin.plist | grep -A 4 adminuid
```

V posledním příkazu je nutné upravit položky „admin.plist“ a „adminuid“, podle toho co je skutečně v systému.

Další možností, jak získat haše, pokud jsou uživateli k dispozici práva root je vyčtení hašů hesel z databáze pomocí Directory Service Command Line utility (dscl). Nástroj dscl je náhradou za zastaralý nástroj nidump (viz kapitola 5), který již ve verzi OS X 10.5 není obsažen. Pro extrakci hašů ze systému byl použit následující skript, který využívá dscl.

```
#!/bin/bash
    dscl . -list /Users 'authentication_authority' | \
    grep -i hash | \
    sed "s/ .*$//" | \
    while read the_name
    do
       echo
       echo "$the_name"
       the_hashfile=$( dscl . -read /users/"${the_name}" generateduid | sed 's/^.* //' )
       the_hash=`cat /var/db/shadow/hash/"$the_hashfile"`
       echo
       echo "$the_hash"
       echo "___${the_name}_NT:${the_hash:0:32}"
       echo "___${the_name}_LM:${the_hash:32:32}"
       echo "_${the_name}_NTLM:${the_hash:0:32}:${the_hash:32:32}"
       S0SHA1="${the_hash:104:48}"
       SSHA1="${the_hash:168:48}"
       [ -n "${S0SHA1//0}" ] && echo "${the_name}S0SHA1:${S0SHA1}"
       [ -n "${SSHA1//0}" ] && echo "${the_name}_SSHA1:${SSHA1}"
    done
```

Výstupem skriptu je výpis všech uživatelských účtů v systému a veškerých hašů, které s nimi souvisí. Následuje ukázka části výstupu výše uvedeného skriptu pro účet smalla01:

```
___smalla01_NT:00000000000000000000000000000000
___smalla01_LM:00000000000000000000000000000000
_smalla01_NTLM:00000000000000000000000000000000:0000000000000000000000000000000000
smalla01_SSHA1:CE70F710FA44406648761237C953D209AF39A88F04693680
```

V posledních čtyřech řádcích jsou z kompletního řetězce vybrány zajímavé haše.
NT haš (je k dispozici pouze pokud je v systému aktivní sdílení Windows).
LM haš (je k dispozici pouze pokud je v systému aktivní sdílení Windows).
NTLM haš (je k dispozici pouze pokud je v systému aktivní sdílení Windows)
zasolený haš SHA1 (SSHA1).

7.2 Lámání hašů

Testovací sestava
K měřením byl použit notebook ASUS A6KM – Q072 s následujícími parametry:
Procesor: AMD Turion 64
Chipset: Severní můstek SIS 756, Jižní můstek: SIS 964L
Velikost pevného disku: 100 GB
Operační paměť: 1024MB DDR
Operační systémy: Windows XP SP3 CZ, Ubuntu 9.04

Uživatelské účty a hesla pro testování
Pro účely testování byla vytvořena sada uživatelských účtů a hesel viz příloha A. Všechna hesla s výjimkou posledních sedmi byla náhodně vygenerována pomocí stránky http://martin.urx.cz/generator-hesel/. Z tabulky je patrné, jakým způsobem byla hesla vytvářena. Postup tvorby byl vždy od menších délek hesla k větším. Nejprve byla tvořena samostatnými sadami znaků: čísla, malá písmena abecedy, speciální znaky. Následně byly tyto samostatné sady kombinovány. Posledních sedm hesel bylo vytvořeno tak, že by měly být náchylné na slovníkové útoky.

Metodika měření času
Protože měřítkem efektivity útoků je v této práci čas jejich trvání, bylo nezbytné stanovit způsoby, jakými bude měřen.

Program Ophcrack dobu hledání hesla zobrazuje sám o sobě. Taktéž program John The Ripper zobrazuje čas hledání hesla. Doba prolamování hašů hesel byla u programu Cain & Abel byla zjišťována pomocí programu Process Explorer v11.33, který umožňuje zobrazit dobu běhu jednotlivých procesů. Pokud byla doba hledání hesla kratší než 1 vteřina, nebo rovna jedné vteřině je v tabulkách uvedená jako 0:00:01 [h:mm:ss]. Doba hledání jednotlivého hesla byla omezena na 6 hodin.

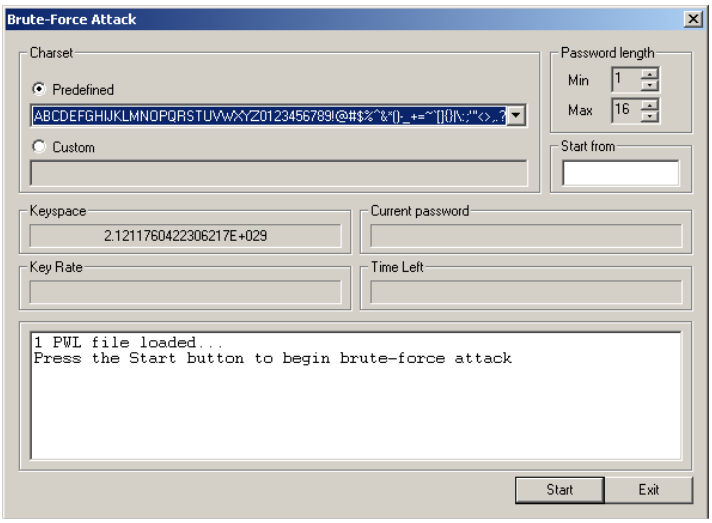
Reprezentace výsledků
Z důvodu přílišné délky tabulek naměřených výsledků byly tyto tabulky umístněny do přílohy B. Z tabulek výsledků byly vytvořeny stručné přehledy výsledků, které jsou součástí vlastní práce. Následuje vysvětlení způsobu výpočtu položek ve stručných přehledech.

Relativní úspěšnost hledání byla spočtena jako podíl celkového počtu hesel dané délky a nalezených hesel v této délce vynásobený 100. Průměrná doba prolomení haše byla spočtena jako aritmetický průměr časů hledání hesel v dané délce, přičemž za každé nenalezené heslo byla započtena doba 6 hodin.

7.2.1 Lámání PWL souborů Windows 98

Protože Windows 98 je dnes už poněkud zastaralým systémem, není k dispozici mnoho nástrojů použitelných pro útok na něj. Jediným použitelným nástrojem, který se mi podařilo nalézt je Cain & Abel [13]. Tento byl použit ve verzi 4.9.30 na Windows XP.

Cain & Abel je téměř univerzálním nástrojem pro testování bezpečnosti hesel systémů Windows a také některých dalších. Program umožňuje útoky na PWL soubory pomocí slovníkového útoku a pomocí hrubé síly. Byla otestována metoda hrubé síly s nastavením, které je uvedeno na obrázku 7.4. Toto nastavení představuje „nejhorší“ možný případ, kdy heslo obsahuje: velká písmena, čísla a speciální znaky.



Obrázek 7.4: Nastavení Cain & Abel pro útok hrubou silou

Stručný přehled výsledků pro lámání PWL souborů

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:00:01
5	5	6	83	1:06:34
6	5	5	100	4:42:31

Tabulka 1: Stručný přehled výsleků pro PWL soubory

Tabulka stručného přehledu výsledků vychází z tabulky uvedené v příloze B.1. Jak je vidět z přehledu výsledků, metoda hrubé síly byla na testovacím stroji velmi úspěšná pro hesla do délky 6 znaků včetně. Na delší hesla byly útoky hrubou silou ve stanoveném časovém rámci neúčinné. Protože se jedná čistě o útok hrubou silou, rostla časová náročnost nalezení hesla exponenciálně.

7.2.2 Lámání hašů OS X 10.5

John the Ripper (JtR)
je rychlý lamač hesel, v současné době je dostupný v 11 oficiálně podporovaných modifikacích. John the Ripper běží na řadě různých platforem (Windows, DOS, BeOS, and OpenVMS). Primárním účelem bylo hledání slabých Unixových hesel. Vedle několika typů hašů z funkce crypt(3), které lze typicky nalézt na Unixových platformách jsou podporovány také šifry z Kerberos AFS a LM haše z Windows NT/2000/XP/2003. K dispozici jsou i zásuvné moduly pro další haše. John the Ripper je možné použít prakticky na všechny známější operační systémy. Program zvládá útok hrubou silou, slovníkové útoky a také hybridní útoky. John the Ripper je také možno využívat na tzv. cluster systémech. [13]

Pro útok na haše systému OS X 10.5 bylo nutné použít nástroj John The Ripper 1.7.3.1 pro Windows vylepšený o zásuvné moduly pro lámání SHA-1 hašů se solí. Program s moduly je dostupný na adrese: http://www.box.net/shared/ssp5l2zqyr.

Program byl použit v režimu „single crack“ bez parametrů. V tomto režimu se zkouší nejprve jako heslo uživatelův login s různými obměnami. Dále se zkouší hesla ze slovníku (s obměnami), který je v programu zabudován. Pokud se nepodaří heslo najít předchozími dvěma způsoby, začne se heslo hledat metodou hrubé síly.

Stručný přehled výsledků pro lámání SHA-1 hašů se solí

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:01:00
5	3	6	50	2:42:01
6	3	5	60	2:34:52
7	1	1	100	0:00:01
8	2	6	33	4:53:47
12	1	7	14	5:08:34

Tabulka 2: Stručný přehled výsledků pro lámání SHA-1 hašů se solí

Tabulka stručného přehledu výsledků vychází z tabulky uvedené v příloze B.2. Na SHA1 haše se solí byl aplikován hybridní útok, a to se projevilo zejména nalezením hesla s délkou 12 znaků za 1 vteřinu. Nahlédnutím do tabulky v příloze B.2 zjistíme, že šlo o heslo „martinmartin“, což je poměrně jednoduchá obměna slova „martin“, které pochází ze slovníku. Ostatní hesla délky náhodná hesla délky 12 znaků se nenašla. Obdobně se hybridní útok projevil při útoku na heslo s délkou 7 znaků „1martin“. U hesel s délkou 6 znaků byl hybridní útok úspěšný asi v 60 procentech případů. Stačilo v hesle „martin“ zaměnit znak „a“ za znak „@” a útok se v daném časovém rámci nezdařil. V případech, kde už John the Ripper musel využít metodu hrubé síly se zdařily útoky na hesla do osmi znaků délky, složené z malých písmen abecedy nebo číslic. V případě hesel obsahujících speciální znaky, uspěly

útoky na hesla s délkou 3 znaky. Rychlost testování hesel byla pro SHA1 haše se solí cca 1,8*10^6 kombinací za sekundu.

MacKrack
je volně šiřitelným nástrojem pro MAC OS X, který podporuje slovníkový útok i útok hrubou silou na haše: Crypt, MD5, SHA-1, zasolený SHA-1 (SSHA1). Program také umožňuje extrakci souborů obsahujících haše z verzí MAC OS X 10.2, 10.3 a 10.4. MacKrack je k dispozici na adrese: http://fsbsoftware.com/macKrack.dmg. Na verzi MAC OS X se ukázal ve verzi 1.5.2 jako **nefunkční** .

7.2.3 Lámání hašů BSD systémů

FreeBSD
FreeBSD používá pro vytvoření hašů hesel algoritmus MD5 s 8 byte solí. Z toho důvodu není možné použití rainbow tables (tabulky pro haše se by zabírali příliš mnoho místa). Další možné útoky jsou slovníkové a útoky hrubou silou. Pro otestování hašů byl použit nástroj John the Ripper v „single crack“ režimu bez parametrů (viz kapitola 7.2.2), který jako jediný nalezený v průběhu práce zvládá útok na MD5 haše se solí.

Stručný přehled výsledků pro lámání MD5 hašů se solí pro FreeBSD

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:22
3	3	6	50	3:37:53
5	1	6	17	5:51:44
6	2	5	40	3:40:06
7	1	1	100	0:00:03
12	1	7	14	5:15:00

Tabulka 3: Stručný přehled výsledků pro lámání MD5 hašů se solí pro FreeBSD

Stručný přehled výsledků vychází z tabulky v B.3. Z tabulky stručného přehledu výsledků je vidět, že JtR byl hrubou silou stoprocentně účinný pouze proti heslům o délce jeden znak. Už při délce hesla 3 znaky byla úspěšně prolomena pouze polovina hesel. Při délce hesla 5 znaků, už bylo nalezeno pouze jedno heslo a to bylo složeno pouze ze znaků malé abecedy (relativně málo odolné). Program úspěšně nalezl i některá hesla s délkou větší než pět znaků (viz příloha B.3) a to díky tomu, že při „single crack“ režimu se využívá i slovníkový útok. Rychlost testování hesel byla pro MD5 haše se solí cca 5600 kombinací za sekundu.

OpenBSD

využívá pro vytvoření hašů hesel standardně algoritmus Blowfish s 6 byte soli. Tento algoritmus je považován za velmi bezpečný. K jeho testování byl opět použit John the Ripper v „single crack“ režimu viz kapitola 7.2.2.

Stručný přehled výsledků pro lámání Blowfish hašů se solí pro OpenBSD

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:10:55
3	1	6	17	5:00:00
6	1	5	20	4:00:01
7	1	1	100	0:01:52
12	1	7	14	0:01:26

Tabulka 4: Stručný přehled výsledků pro lámání Blowfish hašů se solí pro OpenBSD

Stručný přehled výsledků vychází z tabulky uvedené v příloze B.4. Haše OpenBSD se ukázaly být velmi odolné proti útoku hrubou silou. Toto je dáno především výpočetní náročností algoritmu Blowfish. Rychlost testování hesel byla pouhých 180 kombinací za sekundu. Při této rychlosti testování hesel trvalo hledání hesla o délce jeden znak ve dvou případech ze tří více než 15 minut (viz příloha B.4). Z hesel o délce 3 znaky bylo nalezeno pouhé jedno. Výpočetní náročnost algoritmu dokonce komplikuje i slovníkovou část útoku, což se projevilo tím, že ze šestimístných hesel (skupina hesel náchylná na slovníkový útok) bylo nalezeno také pouze jedno.

7.2.4 Lámání hašů Linuxu

Ubuntu 9.04

Linuxová distribuce Ubuntu 9.04 používá pro vytvoření a uložení hašů hesel algoritmus SHA512 se solí. V průběhu práce se nepodařilo nalézt nástroj, který by byl proti těmto hašům použitelný.

Debian 5

Podobně jako FreeBSD využívá standardně Debian 5 pro vytvoření hašů hesel algoritmus MD5 s 8 byte solí. Lámání hašů hesel probíhalo stejně jako u FreeBSD viz kapitola 7.2.3.

Stručný přehled výsledků pro lámání MD5 hašů se solí pro Debian 5

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:21
3	3	6	50	3:35:14
5	1	6	17	5:52:04
6	2	5	40	3:40:53
7	1	1	100	0:00:04
12	1	7	14	5:08:35

Tabulka 5: Stručný přehled výsledků pro lámání MD5 hašů se solí pro Debian 5

Stručný přehled výsledků vychází z tabulky uvedené v příloze B.5.

Protože Debian používá pro vytváření hašů téměř stejnou metodu jako FreeBSD, jsou výsledky útoků velice podobné.

Z tabulky stručného přehledu výsledků je vidět, že JtR byl hrubou silou stoprocentně účinný pouze proti heslům o délce jeden znak. Už při délce hesla 3 znaky byla úspěšně prolomena pouze polovina hesel. Při délce hesla 5 znaků už bylo nalezeno pouze jedno heslo a to bylo složeno pouze ze znaků malé abecedy (relativně málo odolné). Program úspěšně nalezl i některá hesla s délkou větší než pět znaků a to díky tomu, že při „single crack“ režimu se využívá i slovníkový útok. Rychlost testování hesel byla pro MD5 haše se solí cca 5600 kombinací za sekundu.

7.2.5 Prolamování hašů Windows: XP,Vista a Seven

Prolamování hašů pomocí Webových služeb

Jako nejrychlejší a nejpohodlnější nástroj pro prolamování hašů hesel se na první pohled jeví různé webové stránky, které nabízí jakési „vyhledávače hašů“. Na stránku je uživatelem vložen haš hesla a jako odpověď stránka vrátí otevřený text hesla, ze kterého haš vznikl. Tento útok využívá metodu rainbow tables. Výhoda těchto nástrojů je zřejmá. Útočníkovi stačí pouze webový prohlížeč, internet a téměř jakýkoliv operační systém. Výsledek, ať již pozitivní nebo negativní, se vrátí téměř okamžitě.

Stránka http://hashcrack.com/index.php nabízí vyhledávání zdarma pro Windows LM haše a NT haše.

Jak je vidět z tabulky uvedené v příloze B.6 tento nástroj je schopen u LM hašů najít všechna testovaná hesla do tří znaků včetně. Dále se podařilo najít hesla náchylné na slovníkový útok „martin“ a „marTin“ o délce 6 znaků.

Z tabulky úspěšně prolomených hesel v příloze B.7 je vidět, že hashcrack.com nalezl věechna hesla do tří znaků včetně i pro NT haše. Pro NT haše bylo oproti LM hašům nalezeno

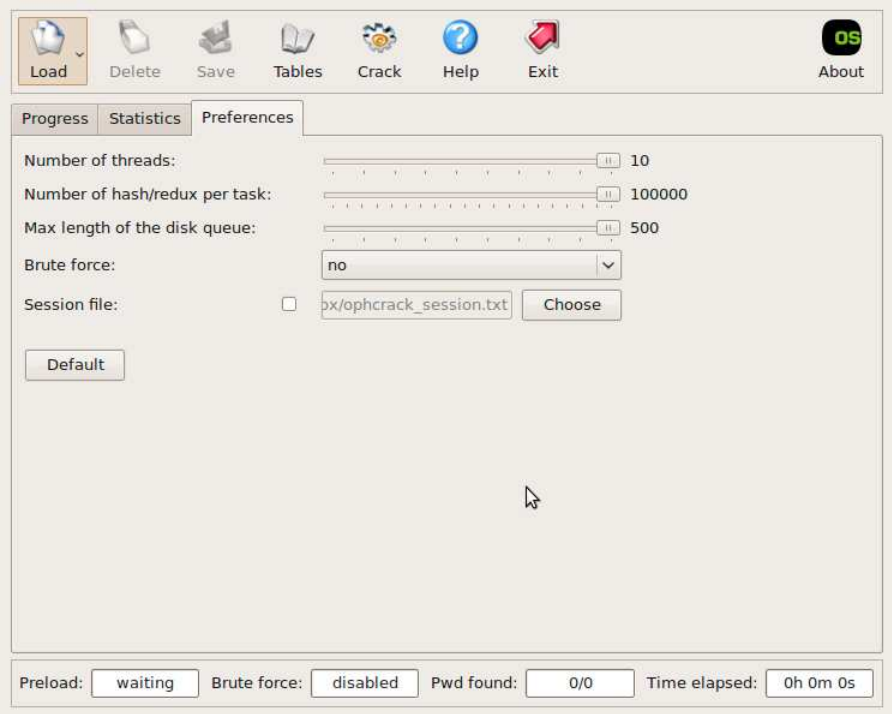
i heslo složené z pěti číslic „47991“. U NT hašů se našlo pouze jedno heslo náchylné na slovníkový útok o délce 6 znaků „martin“. Je to dáno tím, že NT haš rozlišuje malá a velká písmena v hesle.

Doba mezi vložením haše a návratem hesla nebo oznámením o nenalezení hesla byla u LM i NT hašů ve všech případech cca 5 sekund.

Projekt http://www.freerainbowtables.com/
Tento projekt má za cíl distribuovat na klienty výpočty (pomocí klientské aplikace), kdy klienti generuji rainbow tables a ty pak odesílají na server freerainbowtables.com, kde si je uživatelé mohou stáhnout. Pro jednotlivce je generování větších tabulek téměř nemožné (je potřeba velmi vysoký výpočetní výkon), ale právě při použití distribuovaných výpočtů to jde relativně rychle. Za to, že uživatel přispěje svým výpočetním výkonem na výpočty rainbow tables obdrží tzv. „crack credit“. Za tento kredit si potom může zadat do fronty výpočetního střediska projektu haše, které chce vyhledat. Ze skupiny testovaných hašů nebyl zlomen ani jeden. Aplikace pro vyhledávání je pravděpodobně mimo provoz.

Ophcrack
je lamač hašů Windows založený na technice prolamování hesel pomocí rainbow tables. Jedná se o jednu z nejlepších implementací útoku na hesla pomocí rainbow tables. Ophcrack tuto metodu dotáhl k „dokonalosti“. Program obsahuje grafické uživatelské rozhraní GTK+ a běží na systémech Windows, Mac OS X (architektura Intel CPU) stejně jako na Linuxu. Aplikace podporuje funkce a implementace v rámci: Windows, Linux, Mac OS X, lámání LM a NTLM hašů, volné tabulky pro alfanumerické LM haše, načtení hašů z lokálního SAM souboru, vzdálený (remote) SAM, umí přečíst haš ze šifrovaného SAM ,obnoveného Windows oddílu. Ophcrack podporuje i test síly hesla na poslední verzi Windows Vista. Ophcrack LiveCD běží jako malý linux systém (založen na jádře SLAX 6 distribuce). LiveCD prolamuje hesla automaticky bez potřeby instalace nebo administrátorského přístupu nebo oprávnění. Není potřeba heslo administrátora. Stejně jako Windows XP, Windows Vista a Windows Seven také používají SAM a NT haš, proto je možné prolomit i jejich haše. [14] Program je k dispozici na adrese: http://Ophcrack.sourceforge.net/

V této práci byl využit program Ophcrack verze 3.0.1 nainstalovaný na operačním systému Ubuntu 9.04 v kombinaci s rainbow tables pro Ophcrack xp_special (LM haše) a vista_special (NT haše). Pro testování bylo použito nastavení programu, které je uvedeno na obrázku 7.5.



Obrázek 7.5: Nastavení programu Ophcrack

U LM hašů nebyla testována hesla delší než 14 znaků, protože pro tato hesla již Windows ukládá pouze ve formě NT hašů. Pro nástroj Ophcrack byla do průměrné délky prolomení haše v případě neúspěšného hledání hesla započítávána pouze doba 30 minut místo obvyklých 6 hodin, protože doba hledání hesla v případě neúspěchu na testovacím stroji je pro oboje použité tabulky asi 30 minut.

Stručný přehled výsledků pro lámání LM hašů nástrojem Ophrcack

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:17
3	6	6	100	0:07:20
5	6	6	100	0:06:20
6	5	5	100	0:02:14
7	1	1	100	0:05:13
8	5	6	83	0:13:15
12	5	7	71	0:13:15

Tabulka 6: Stručný přehled výsledků pro lámání LM hašů nástrojem Ophcrack

Ze stručného přehledu (vychází z tabulky v příloze B.8) je patrné, že nástroj Ophcrack je v případě LM hašů velice rychlý a efektivní. Až do délky hesla 7 znaků měl při hledání hesel úspěšnost 100 procent a doba hledání hesel byla do 10 minut. Na delší hesla nástroj sice nebyl už tak účinný, ale i pro hesla délky 12 znaků jich bylo nalezeno cca 70 procent. Z 34 testovaných LM hašů nalezl Ophcrack 31 hesel a nejdelší úspěšné hledání zabralo na testovacím stroji asi 22 minut.

Stručný přehled výsledků pro lámání NT hašů nástrojem Ophrcack

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:33
3	6	6	100	0:07:09
5	6	6	100	0:04:52
6	5	5	100	0:03:14
7	1	1	100	0:00:21
8	1	6	17	0:25:55

Tabulka 7: Stručný přehled výsledků pro lámání NT hašů nástrojem Ophcrack

Ze stručného přehledu (vychází z tabulky v příloze B.9) vidíme, že u NT hašů Ophcrack vykázal stoprocentní úspěšnost u hesel do délky 7 znaků včetně. Z hesel o délce 8 znaků bylo nalezeno pouze „nejjednodušší“ heslo složené z číslic. U metody rainbow tables je neúspěch při hledání hesel delších sedmi znaků způsoben požadavkem na přijatelnou velikost tabulek, která je v případě tabulek vista_special cca 8 GB.

Ze 42 testovaných NT hašů jich Ophcrack nalezl 22. Nejdelší úspěšné hledání zabralo na testovacím stroji asi 20 minut.

John The Ripper

je univerzálním nástrojem pro testování hašů vytvářených operačními systémy. Jeho popis lze nalézt v kapitole 7.2.2. Mezi podporované haše patří i LM a NT haše Windows. Pro testování LM a NT hašů byl opět použit režim „single crack“.

Stručný přehled výsledků pro lámání LM hašů nástrojem John the Ripper

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:00:10
5	6	6	100	0:18:47
6	5	5	100	1:01:15
7	1	1	100	0:00:05
8	2	6	33	4:46:18
12	1	7	14	5:08:35

Tabulka 8: Stručný přehled výsledků pro lámání LM hašů nástrojem JtR

Stručný přehled vychází z tabulky uvedené v příloze B.10. Ze stručného přehledu výsledků vyplývá to, že hybridní útok byl úspěšný u hesel do délky 5 znaků. U skupiny hesel délky 6 znaků (skupina náchylná na slovníkový útok) se projevilo vložení speciálního znaku „@“. Prolomení těchto hesel trvalo cca dvě a půl hodiny, což je značný rozdíl oproti zbytku skupiny, kde prolomení trvalo řádově vteřiny. Ve skupině hesel o délce 6 znaků byla ještě nalezena všechna hesla. U skupiny hesel s délkou 8 znaků JtR našel už jen 2 hesla a to heslo složené z číslic a heslo složené ze znaků malé abecedy a číslic. Bylo nalezeno i jedno z hesel o délce 12 znaků, což je zásluha slovníkové části „single crack“ režimu. Rychlost testování hesel byla cca 6,7*10^6 hesel za sekundu.

Stručný přehled výsledků pro lámání NT hašů nástrojem John the Ripper

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:00:19
5	4	6	67	2:18:23
6	3	5	60	2:29:55
7	1	1	100	0:00:01
8	1	6	17	5:06:07
12	1	7	14	5:08:34

Tabulka 9: Stručný přehled výsledků pro lámání NT hašů nástrojem JtR

Stručný přehled vychází z tabulky v příloze B.11. Hybridní útok uspěl ve všech případech pouze u skupiny hesel do tří znaků včetně. U skupiny hesel délky 5 znaků, už byla pouze hesla, ve kterých nebyl žádný speciální znak. Podobně tomu bylo i ve skupině hesel délky 6 znaků. U hesla délky 7 znaků je údaj v tabulce mírně zavádějící, protože takové heslo je pouze jedno a to bylo uhodnuto díky slovníkové části útoku. U hesel délky 8 znaků bylo nalezeno pouze „nejjednodušší“ heslo složené z číslic. Díky slovníkové části útoku se našlo také jedno heslo délky 12 znaků. Rychlost testování hesel byla přibližně 9*10^6 hesel za sekundu.

Cain & Abel

Tento nástroj byl použit pro lámání PWL souborů systému Windows 98. V kapitole o lámání PWL souborů také nalezneme podrobnější popis programu. Ze zkoumaných hašů zvládá Cain & Abel slovníkové útoky a útoky hrubou silou na haše systému Windows (LM i NT). Pro útoky hrubou silou na LM haše bylo použito stejné nastavení jako pro lámání PWL souborů (kapitola 7.2.1). Pro útoky hrubou silou na NT haše byla přidána do abecedy pro útoky i malá písmena anglické abecedy, protože u NT hašů je velikost jednotlivých písmen hesla zohledněna. Na rozdíl od JtR Cain & Abel používá čistě útok hrubou silou.

Stručný přehled výsledků pro lámání LM hašů nástrojem Cain & Abel

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:00:01
5	6	6	100	0:01:54
6	5	5	100	0:49:13
7	1	1	100	0:54:40
8	1	6	17	5:53:42

Tabulka 10: Stručný přehled výsledků pro lámání LM hašů nástrojem Cain & Abel

Stručný přehled vychází z tabulky v příloze B.12. Cain & Abel byl u LM hašů účinný pro hesla až do délky 7 znaků včetně. Hledání hesel o délce do 7 znaků trvalo většinou méně než jednu hodinu. Bylo nalezeno pouze jedno heslo ze skupiny hesel o délce 8 znaků viz příloha B.12.

Stručný přehled výsledků pro lámání NT hašů nástrojem Cain & Abel

délka hesla [znaky]	počet nalezených hesel	počet hesel dané délky	relativní úspěšnost hledání [%]	průměrná doba prolomení haše [h:mm:ss]
1	3	3	100	0:00:01
3	6	6	100	0:00:01
5	6	6	100	0:07:46
6	5	5	100	4:07:33

Tabulka 11:Stručný přehled výsledků pro lámání NT hašů nástrojem Cain & Abel

Stručný přehled vychází z tabulky v příloze B.13. U NT hašů byl Cain & Abel schopen nalézt hesla do 6 znaků včetně. V případě delších hesel bylo hledání v daném časovém rámci neúspěšné. U NT hašů trvalo hledání hesel délky 6 znaků v průměru asi 4 hodiny.

Srovnání nástrojů použitých u Windows

LM haše

Jako nejúčinnější nástroj se projevil program Ophcrack, který využívá metodu rainbow tables. Tento program nalezl heslo pro 31 z 34 testovaných LM hašů. Ophcrack dokázal prolomit většinu hašů hesel o délce 12 znaků. Hledání hesel bylo časově nejméně náročné, protože nejhorším možným případem z hlediska času je prohledání celé tabulky bez úspěchu, a to zabere cca 30 minut.

John the Ripper se projevil jako druhý nejúčinnější nástroj pro hledání hesel, protože našel o 3 hesla více než Cain & Abel. Z časového hlediska v některých případech vykazoval lepší výsledky Cain & Abel v jiných naopak John the Ripper (viz příloha B.10 a příloha B.12). Ve

výsledku byly oba nástroje velmi vyrovnané. Jak John the Ripper, tak i Cain & Abel spolehlivě prolamovali LM haše, které obsahovaly hesla o délce 6 znaků včetně. Časy hledání hesel byly oproti rainbow tables u obou nástrojů lepší pokud se jednalo o krátká hesla (3 znaky), v případě delších hesel byl rychlejší Ophcrack.

Jako nejméně účinné se ukázaly útoky s využitím webových služeb, které uspěly v útoku na hesla do tří znaků včetně. Podařilo se také nalézt 2 hesla náchylná na slovníkový útok.

NT haše

Stejně jako u LM hašů i u NT hašů nejlepší výsledky vykazoval Ophcrack. Potvrdil se předpoklad, že NT haš je odolnější vůči útoku. Použitím metody rainbow tables byla spolehlivě nalezena hesla do délky 7 znaků včetně. V případě, že heslo nebylo nalezeno prohledání tabulek trvalo asi 30 minut.

Na NT haších se ukázaly přednosti nástrojů Cain & Abel a John the Ripper. Cain & Abel vykazoval větší úspěšnost při útoku hrubou silou na hesla o délce 6 znaků. Tato hesla byla prolomena všechna v čase, který se pohyboval mezi čtyřmi a pěti hodinami viz příloha B.13. Naproti tomu John the Ripper, dokázal najít jen některá hesla délky 6 znaků, ale většinou mu to díky slovníku trvalo vteřiny viz příloha B.11. Další výhodou slovníkového útoku u JtR bylo odhalení jednoho hesla o délce 7 znaků a jednoho hesla o délce 12 znaků.

Další postup prolomení bezpečnosti Windows XP, Windows Vista, Windows 7

Jednou z možných cest, jak účinně získat heslo z LM hašů a NT hašů je využití programu ElcomSoft Distributed Password Recovery od firmy ElcomSoft. Tento komerční nástroj umožňuje distribuovaný útok na hesla metodou hrubé síly. Útok může být distribuován přes internet až mezi 10000 počítačů. Navíc tento program dokáže pro lámání hašů využít výpočetní výkon některých modelů grafických karet s grafickými jádry řad 8 a 9 od firmy Nvidia. Při využití akcelerace pomocí grafické karty výrobce udává až padesátinásobné zrychlení prolamování hašů. [15]

9 ZÁVĚR

Cílem této diplomové práce bylo zjištění a otestování mechanizmů pro ukládání uživatelských jmen a hesel v moderních operačních systémech. V první části práce byly teoreticky popsány mechanizmy systémů MS Windows, Linux, FreeBSD, OpenBSD a Mac OS X.

V praktické části práce byly popsány a vyzkoušeny způsoby získání hašů hesel z operačních systémů. Následně byly na získané haše pomocí nalezených nástrojů vedeny praktické útoky.

Mac OS X
Verze OS X 10.5 poskytuje poměrně kvalitní zabezpečení s využitím šifrovacího algoritmu SHA1 se solí. Nicméně algoritmus SHA1 se dnes již nedoporučuje používat, protože se považuje za prolomený. Na SHA1 haše byly vedeny hybridní útoky pomocí nástroje John the Ripper. Vysoce úspěšné útoky byly pouze útoky na kratší hesla (3 znaky), a nebo některá hesla ze slovníku.

FreeBSD
Většina Linuxových systémů pro uložení hesel stále používá MD5 haš, který je sice prolomený, ale dá se zesílit tím, že se použije vícenásobně. Z toho plyne, že je teoreticky možné hesla prolomit hrubou silou, ale při použití dostatečně kvalitního hesla a soli to není jednoduché. Prakticky vedené útoky potvrdily předpoklad vysoké odolnosti hašů proti útokům. Už při délce hesla 3 znaky byla úspěšnost prolomení hašů jen padesátiprocentní. Pro hesla delší než 3 znaky se útok podařil pouze u hesel, která byla nalezena díky slovníkové části hybridních útoků vedených pomocí programu John the Ripper.

OpenBSD
Implementace hašovacího algoritmu Blowfish představuje pro útočníka velmi obtížný problém. Zatím nebyly oznámeny žádné úspěchy v jeho kryptoanalytickém prolamování.

Díky velké výpočetní náročnosti algoritmu se podařilo prolomit hybridním útokem pomocí nástroje John the Ripper pouze několik málo hašů. Jednalo se buď o velmi krátká hesla (1 znak) a nebo velmi jednoduchá slova ze slovníku. Z důvodu velké výpočetní náročnosti se stává problémem i slovníkový útok zahrnující velký počet slov.

Ubuntu
Tato distribuce začala ve verzi 9.04 používat pro hašování hesel algoritmus SHA512, na který se nepodařilo nalézt žádný nástroj k útoku.

Debian 5
používá k hašování hesel téměř identickou implementaci jako FreeBSD. Výsledky testování byly velice podobné výsledkům dosaženým u FreeBSD.

Windows 98
Windows 98 využívá k uložení hesel PWL soubory. Na tyto se podařilo vést útoky hrubou silou pomocí nástroje Cain & Abel. Pro hesla o délce 6 znaků a menší byly útoky velmi úspěšné. Bylo nalezeno 20 z 21 testovaných hesel do délky 6 znaků. V případě delších hesel nebylo prolomeno žádné.

Windows XP, Windows Vista, Windows XP
Tyto systémy využívají poměrně slabých hašů LM haš (Windows XP) a NT haš (Windows Vista a Windows 7), které je možné velmi rychle prolomit pomocí tzv. rainbow tables. V praxi se ukázalo že nástrojem Ophcrack lze během třiceti minut získat až dvanáctimístná hesla z LM hašů a šestimístná hesla z NT hašů na běžném domácím počítači.

Proti heslům uloženým pomocí LM haš s délkou 6 znaků a méně, byly se stoprocentní úspěšností vedeny útoky pomocí nástrojů Cain & Abel (hrubou silou) a John the Ripper (hybridní). U těchto útoků se ale časová náročnost při delších heslech oproti rainbow tables pohybuje v řádu hodin.

Ukládání hesel pomocí NT Haš se v praxi ukázalo jako odolnější vůči útokům. Při použití nástroje Ophcrack (rainbow tables) byla prolomena hesla o délce 7 znaků do 30 minut. Při útocích hrubou silou (Cain & Abel) se dařilo spolehlivě získat hesla o délce 6 znaků v čase mezi čtyřmi a pěti hodinami. V případě hybridního útoku nástrojem John the Ripper se dařilo získat hesla stejné délky v závislosti na tom, zda-li byla obsažena ve vestavěném slovníku programu.

Doporučení pro volbu hesel
Heslo by nemělo mít žádný smysl, aby nebylo možné jej nalézt pomocí slovníkového útoku. Délka hesla by měla být alespoň osm znaků, což se u většiny systémů projevilo jako dostatečná délka. Osm znaků je dostatečných za předpokladu, se nejedná o heslo složené jen z písmen malé abecedy nebo číslic. Je vhodné do hesla zahrnout čísla, malá písmena, velká písmena a speciální znaky. Zahrnutí speciálních znaků do hesel se projevilo nejpozitivněji.

Výjimkou mezi operačními systémy byl systém Windows XP, který využívá k uložení hesel LM hašů. Při použití rainbow tables proti tomuto systému nejsou bezpečná ani hesla s délkou 12 znaků. Jednoduchým řešením tohoto problému je použití hesla s délkou větší než 14 znaků. Při použití hesla delšího než 14 znaků dojde k uložení hesla pouze pomocí silnějšího NT haše.

LITERATURA

[1] MENŠÍK, Miroslav. *INFORMATIKA MODUL 4 BEZPEČNOST OPERAČNÍHO SYSTÉMU A SÍŤOVÉ KOMUNIKACE*. Brno: 2004.

[2] JOHANSON, Jesper. *Security Watch The Most Misunderstood Windows Security Setting of All Time.* [Online] [cit. 16. 12. 2008] <http://technet.microsoft.com/cs-cz/magazine/cc160954(en-us).aspx>

[3] TODOROV, Dobromir. *Mechanics of User Identification and Athentication*. Taylor & Francis Group, 2007. ISBN 978-1-4200-5219-0.

[4] RainbowCrack, Project. *Project RainbowCrack.* [Online] 2008. [cit. 11. 12. 2008] <http://www.antsight.com/zsl/rainbowcrack/>

[5] BOER, Bosselaers. *An Attack on the Last Two Rounds of MD4*. [Online] 15. Oct. 1991. [cit. 11.12. 2008] <http://citeseer.ist.psu.edu/denboer91attack.html.>

[6] Public Interest, Inc. debian.org. *Co je Debian?* [Online] Public Interest, Inc. , 12. 5 2009. [cit. 14. 5. 2009] <http://www.debian.org/>

[7] Ltd., Canonical. *About Ubuntu.* [Online] Canonical Ltd., 2009. [cit 3. 3. 2009] <http://www.ubuntu.com/>

[8] FreeBSD, Projekt. *O systému FreeBSD*. [Online] 2008. [Citace: 11. 12. 2008.] <http://www.freebsd.cz/cs/about.html>

[9] MAZIERES, David a PROVOS, Niels. *A Future-Adaptable Password Scheme*. [Online] 11. 6. 1999. [cit: 3. 12. 2008] <http://www.usenix.org/events/usenix99/provos/provos.pdf>

[10] DRIBIN, Dave. *How Mac OS X Implements Password Authentication*. [Online] 7. 4. 2006. [cit. 3. 12. 2008.] <http://www.dribin.org/dave/blog/archives/2006/04/07/os_x_passwords/>

[11] Software, Lastbit. *Pwl Files*. Lastbit Software. [Online] [citace: 10. 5. 2009] <http://www.lastbit.com/vitas/pwl.asp>

[12] Microsoft. *Nástroj Syskey*. Microsoft TechNet. [Online] 21. 1. 2005. [cit 23. 4. 2009] <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/cs/library/ServerHelp/d66896e2-534d-4956-a7d3-0b1df1678e7b.mspx?mfr=true>

[13] Project, Openwall. *John the Ripper password cracker*. Openwall Project. [Online] 16. 8. 2008. [cit. 8. 12. 2009] <http://www.openwall.com/john/>

[14] Ophcrack. *Ophcrack*. [Online] [cit. 11. 12. 2008.] <http://ophcrack.sourceforge.net/>

[15] Ltd., Elcomsoft Co. *ElcomSoft Distributed Password Recovery*. [Online] Elcomsoft Co. Ltd., 2009. [cit. 15. 5. 2009] <http://www.elcomsoft.com/edpr.html>

[16] MONTORO, Massimiliano. *Info*. oxid.it. [Online] [cit 8. 12. 2008.] <http://www.oxid.it/cain.html>

[17] MACUR, Jiří. *INFORMATIKA MODUL 1 POČÍTAČ A OPERAČNÍ SYSTÉMY*. Brno 2004.

Seznam použitých zkratek:

AP	Authentication Package
ASCII	American Standard Code for Information Interchange
BSOD	Blue Screen Of Death
DES	Data Encryption Standard
HMAC-MD5	Hash Message Authentication Code MD5
JtR	John the Ripper
LDAP	Lightweight Directory Access Protocol
LM	LAN Manger
LSA	Local Security Autority
MD4	Message Digest 4
MD5	Message Digest 5
MS	Microsoft
MS GINA	Microsoft Graphical Identification and Authentication
MS-CHAPv2	Microsoft Challenge-handshake authentication protocol version 2
NT	New Technology
NTLM	NT LAN Manger
NIS	Network Information Service
OS	Operační Systém
PWL	Pass Word List
SAM	Security Accounts Manager
SAS	Secure Attention Sequence
SHA	Secure Hash Algorithm
SP	Service Pack
SSP	Security Service Provider
SSHA1	Salted SHA-1
SYSKEY	System Key

Seznam příloh:

Příloha A: Tabulka uživatelských účtů a jejich hesel vytvořených pro testování

Příloha B: Tabulky úspěšně nalezených hesel

Příloha C: Optický disk s elektronickou verzí diplomové práce

PŘÍLOHA A: TABULKA UŽIVATELSKÝCH ÚČTŮ A HESEL VYTVOŘENÝCH PRO TESTY

Abeceda hesla	Uživatelský účet	Heslo
čísla 0 až 9		
	num01	9
	num03	380
	num05	47991
	num08	96736226
	num12	304927007616
	num20	17459139944125610382
	num30	859478752955871840976231593599
malá písmena abecedy bez diakritiky		
	smalla01	a
	smalla03	abc
	smalla05	oemjs
	smalla08	froaroet
	smalla12	choeswiewiag
	smalla20	chiucrlaroesiadrluth
	smalla30	moawluzoefleqoebriexoeroucrouf
speciální znaky		
	spec01	+
	spec03	&*=
	spec05	*=%.;
	spec08	*:.+:==+
	spec12	?%?=?&:*&&;:
	spec20	*%?;=:&?:.%*.%.:.%=*
	spec30	:??=*&.%=%?;&*?;;;=:&&?;.*%+&;
kombinace malých a velkých písmen abecedy bez diakritiky		
	mixeda03	pFF
	mixeda05	OTbGp
	mixeda08	TzTOFnaH
	mixeda12	gkfNOGYHvutJ
	mixeda20	zQysRnHzxZJYuqJcNcqh
	mixeda30	hOwvegFFPIFctsmthyTrSOfXpsEYah
kombinace malých a velkých písmen abecedy bez diakritiky a čísel 0 až 9		

	mixednuma03	I9T
	mixednuma05 mixednuma08	xrZx8 3gswVGMp
	mixednuma12	Hyc66m8K42N2
	mixednuma20	hxeghL4QJuwxvfI1bUae
	mixednuma30	58IqH4ug3JX80gL2mfbxwGX4yW0bPU
kombinace malých a velkých písmen abecedy bez diakritiky a čísel 0 až 9 a speciálních znaků		
	mixednumaspec03	%jp
	mixednumaspec05	G;3It
	mixednumaspec08	kyu1ST=4
	mixednumaspec12	$TmNJbHM88S;
	mixednumaspec20	Q7:j9KaH&YpkIh0XmtqW
	mixednumaspec30	gN96g#NM=4vL8L#:rwT#k*KO*;?a&.
hesla náchylná na slovníkový útok		
	word01	martin
	word02	1martin
	word03	martinmartin
	word04	123dog
	word05	marTin
	word06	m@rtin
	word07	m@rTin

PŘÍLOHA B: TABULKY ÚSPĚŠNĚ NALEZENÝCH HESEL

B.1: Tabulka úspěšně nalezených hesel pro PWL soubory

Účet	heslo	čas prolomení [h:mm:ss]
num01	9	0:00:01
num03	380	0:00:01
num05	47991	0:09:19
smalla01	a	0:00:01
smalla03	abc	0:00:01
smalla05	oemjs	0:06:09
spec01	+	0:00:01
spec03	&*=	0:00:02
mixeda03	pFF	0:00:01
mixeda05	OTbGp	0:05:13
mixednuma03	I9T	0:00:01
mixednuma05	xrZx8	0:11:39
mixednumaspec03	%jp	0:00:01
mixednumaspec05	G;3It	0:07:01
word01	martin	5:13:12
word04	123dog	2:37:16
word05	marTin	5:13:20
word06	m@rtin	5:14:43
word07	m@rTin	5:14:06

B.2: Tabulka úspěšně nalezených hesel pro SHA1 haše se solí

Účet	heslo	SHA-1 haš se solí	čas prolomení [h:mm:ss]
num01	9	10EF58306F5328546F632D1C70D1BF4B319E29642E91AAE2	0:00:01
num03	380	B41D7F53205F873B6BD85B931FB4AB60BC7F8A04CAB531A4	0:00:03
num05	47991	73EBCD7CD76982B1F7803157FEC9168682098B760FE848F1	0:01:47
num08	96736226	882D7C17DC2EE5386559C5AE93360FA6E3EBB13830A8DDD7	3:04:37
smalla01	a	CE70F710FA44406648761237C953D209AF39A88F04693680	0:00:01
smalla03	abc	C5490830D747EFE29EE44179E3D4973F6357BAA0E3D66655	0:00:01
smalla05	oemjs	52F25F6146202FDFAB18BF937A194AEDE728213AB21AB54C	0:01:05
smalla08	froaroet	EC65956C5C671129FFFB4FB2E28535A9A3A16D23953E21CB	2:18:03
spec01	+	F094A95EDB85303C1330D2535EA92BB3AEC96A15DC668B7B	0:00:01
spec03	&*=	A3536C74F499EE74948519DEB5FDE556BF8A6024F0F973A5	0:03:15
mixeda03	pFF	7463D66BA90FF57FB047BB080BF81711C923B6805A79AD05	0:01:56
mixeda05	OTbGp	BF7BE55C84EC0C8D6FA3627062DD8A2730A6F573BAB70D06	0:51:18
mixednuma03	I9T	4BE8297C401457F4F03DD6A729C47E89147AC6F4B26D0471	0:00:44
mixednumaspec03	%jp	F837BB7F8304A20A4F0803B28FC391CB73C9AD1EF79335B0	0:02:18
word01	martin	E96D1171065194434CBE70EBD6289974F53F0A94F96C192C	0:00:01
word02	1martin	8DF0140B3FF65801882F2C9D6B40B304C9F6277D53323FC9	0:00:01
word03	martinmartin	584CD748A073ECF9669EDC7792F5E3B7C04CE774601B3C4D	0:00:01
word04	123dog	473965179995B368E665973122094820AA43B9918CF2EE10	0:00:52
word05	marTin	95D37F1AE04EA274197A8DABBE83FE3CD18A8C95931EDFA9	3:28:20

B.3: Tabulka úspěšně nalezených hesel pro MD5 haše se solí pro FreeBSD

účet	heslo	MD5 haš se solí	čas prolomení [h:mm:ss]
num01	9	1gzzJVCQs$kDiHlEpqxjOuq3.zFccbE1	0:00:31
num03	380	1exZ5zGGt$z8qrMfX5ENISUvUyJPIPO.	0:04:42
smalla01	a	1VHyjNxue$DBElogLW19NRNBtVQBpgd1	0:00:01
smalla03	abc	1RXTY7owK$TDKg2HO70OatrC0uwV.iv1	0:00:01
smalla05	oemjs	1aWM7Sd.h$QLzYCghlX6SkFA.hxCTah.	5:10:25
spec01	+	1sedqaK7X$HJ7Itd9V0fR8tsbRUzCTm1	0:00:34
mixednuma03	I9T	1jCPAfkuM$q5h293OUbVb7z9iDdpXFk1	3:42:32
word01	martin	$1$3p4bVqxw$qJsY/23XO8hfk1dwFTpMG1	0:00:01
word02	1martin	1/ufn/o.B$pjVfLD3u2pZlKeBvcFlOi.	0:00:03
word03	martinmartin	1Edz.bxr2$CbmMg8KOa54AZ/7o3YQy10	0:00:03
word04	123dog	1Uy67wdA5$jezXaiVEVPyCtmYEOW8rK1	4:00:34

B.4: Tabulka úspěšně nalezených hesel pro Blowfish haše se solí pro OpenBSD

účet	heslo	Blowfish haš se solí	čas prolomení [h:mm:ss]
num01	9	$2a$06$EPYJGnY.2d8T9n.8FmkGvuAF1AOM25leNsbUd0zpn7objgTkXf2Iu	0:16:22
num03	380	$2a$06$geMeKTGbHfuOEo3Dk80iq.nZk1DGRpTvgNq7/Db076xFPjjdGP6GG	0:00:01
smalla01	a	$2a$06$Y9kK9eExBkrPJqQzfuCxS.9K5oZYk2zaqYZ2O5.HAQcS3VHSNAkJ2	0:00:06
spec01	+	$2a$06$a4EGcWTMq/QOsnmXadI.NeHzXVBuqY0uWaNF.c4UOMTzlmfILkM4.	0:16:18
word01	martin	$2a$06$0HT7Uh/8Zlr7w01cSJBhGeVKDI5fm8eGGe.fkhOguu8lbUdD8CIdG	0:00:08
word02	1martin	$2a$06$VDxOKxCTXFYqCq52PgbYsefKmGwbkhwMhG1WEvGY.9sjb6nc1flIS	0:01:52
word03	martinmartin	$2a$06$b/CETLeJBqI7pttMrdfHaeccz8jbkvVkws2nOxCAQeE3wdLv9TRha	0:01:26

B.5: Tabulka úspěšně nalezených hesel pro MD5 haše se solí pro Debian

Účet	heslo	MD5 haš se solí	čas prolomení [h:mm:ss]
num01	9	1V1.A70Us$BOeUQY3FiVxhMZT6jhqfU0	0:00:31
num03	380	1eg.3G08u$OU/N4wvAt13zfiqSrc92.1	0:04:48
smalla01	a	$1$0uUWPsgn$ierdJIFwkJuR8wU0ljPFu/	0:00:01
smalla03	abc	1mdD1kWiT$sLekEBLkvS2xp4jyytfMU.	0:00:01
smalla05	oemjs	1gtXNysOg$zx57O888VYya.rG.HSOFq0	5:12:21
spec01	+	1pvmtvqIG$cH.lpBW22xc3Mi1Ii9CAA1	0:00:31
mixednuma03	I9T	$1$9MggBHBU$6yl3wUe4gxkjGKrsKMzyn0	3:26:33
word01	martin	1ZglW3VjR$QM8zX3fp/ghDO9gh/XjBd1	0:00:01
word02	1martin	$1$5NJtt/9X$f774Bw968/T7E5DWHnWNY.	0:00:04
word03	martinmartin	1VQ8579wQ$ZkGl/NBdeIid3igsW3PZV.	0:00:06
word04	123dog	1LbvYJtV/$22VxcIILtJxJdr5Bn7qkB/	4:05:20

B.6: Tabulka úspěšně nalezených hesel pro LM haše pomocí webových služeb

Účet	heslo	LM haš
num01	9	09752A3293831D17AAD3B435B51404EE
num03	380	5CD2A18668F8A1CCAAD3B435B51404EE
smalla01	a	7584248B8D2C9F9EAAD3B435B51404EE
smalla03	abc	8C6F5D02DEB21501AAD3B435B51404EE
spec01	+	C9E04F959D38CCB0AAD3B435B51404EE
spec03	&*=	A456A0FB7154F33EAAD3B435B51404EE
mixeda03	pFF	EA6527FD755060BEAAD3B435B51404EE
mixednuma03	I9T	2C2724515ABFDD23AAD3B435B51404EE
mixednumaspec03	%jp	537B02D8AA067B76AAD3B435B51404EE
word01	martin	CB829955C5787ACAAAD3B435B51404EE
word05	marTin	CB829955C5787ACAAAD3B435B51404EE

B.7: Tabulka úspěšně nalezených hesel pro NT haše pomocí webových služeb

Účet	heslo	NT haš
num01	9	90AD6AB281C4AE016E5A7564C307A7E8
num03	380	2B7E20443D479152413D558C180E063C
num05	47991	0606D9A3097812CF2B2B551B8BEC72BD
smalla01	a	186CB09181E2C2ECAAC768C47C729904
smalla03	abc	E0FBA38268D0EC66EF1CB452D5885E53
spec01	+	9B0E9CD1456256B6C6A3664BE284E517
spec03	&*=	E727804223F614E0DA55F6F9BB42B28C
mixeda03	pFF	695A2A3942B1EE426F694ED8A18684CD
mixednuma03	l9T	6B51E08CEB34ED70CEF89F3845531C4B
mixednumaspec03	%jp	FA507AC217AF189395F1CDAC4D792E10
word01	martin	D643DA71A76DD7EB97226D3C5AFEB01B

B.8: Tabulka úspěšně nalezených hesel pro LM haše pomocí Ophcrack

Účet	heslo	LM haš	čas prolomení [h:mm:ss]
num01	9	09752A3293831D17AAD3B435B51404EE	0:00:16
num03	380	5CD2A18668F8A1CCAAD3B435B51404EE	0:01:41
num05	47991	7239A6B0656FAD8DAAD3B435B51404EE	0:06:00
num08	96736226	D8C2696D88417AC5C81667E9D738C5D9	0:21:27
num12	304927007616	04AF06C562CA71008978313ED85774F2	0:04:49
smalla01	a	7584248B8D2C9F9EAAD3B435B51404EE	0:00:30
smalla03	abc	8C6F5D02DEB21501AAD3B435B51404EE	0:05:11
smalla05	oemjs	36F5F68EFF305FFFAAD3B435B51404EE	0:06:10
spec01	+	C9E04F959D38CCB0AAD3B435B51404EE	0:00:05
spec03	&*=	A456A0FB7154F33EAAD3B435B51404EE	0:05:07
spec05	*=%.;	E0634096C16B162FAAD3B435B51404EE	0:12:35
spec08	*:.+:==+	EFDAA9BBECDE7312C9E04F959D38CCB0	0:00:43
spec12	?%?=?&:*&&;:	12DDD58FBBA1BBDB57444CE406A11B0E	0:06:39
mixeda03	pFF	EA6527FD755060BEAAD3B435B51404EE	0:14:40
mixeda05	OTbGp	FCBE402ABE856D70AAD3B435B51404EE	0:06:45
mixeda08	TzTOFnaH	FCA1A947FB573D7E5ACDCD7C247FA83A	0:01:48
mixeda12	gkfNOGYHvutJ	FBD236C08D73C643627FB175B9F57646	0:13:19
mixednuma03	I9T	2C2724515ABFDD23AAD3B435B51404EE	0:02:15
mixednuma05	xrZx8	8D5A629565BAD6DDAAD3B435B51404EE	0:05:01
mixednuma08	3gswVGMp	3411D087F6EED6388B0EA5A7DF135B03	0:03:02
mixednuma12	Hyc66m8K42N2	EDCC80D3D44FBC7EF7C69CDFB1E5E78A	0:07:08
mixednumaspec03	%jp	537B02D8AA067B76AAD3B435B51404EE	0:15:06
mixednumaspec05	G;3lt	CDD80A145A211FD2AAD3B435B51404EE	0:01:28
mixednumaspec08	kyu1ST=4	25D48A8CCC30606FFF17365FAF1FFE89	0:22:28
mixednumaspec12	$TmNJbHM88S;	B1FA02D38B48D4C7D27EEECBD91D706D	0:07:24
word01	martin	CB829955C5787ACAAAD3B435B51404EE	0:04:32
word02	1martin	35D7A2A64A05F923AAD3B435B51404EE	0:05:13
word03	martinmartin	F0A90DFA3ACDCD39E238065355FD3F58	0:09:43
word04	123dog	B65B9941F2388936AAD3B435B51404EE	0:03:04
word05	marTin	CB829955C5787ACAAAD3B435B51404EE	0:00:01
word06	m@rtin	F3300BA07671142DAAD3B435B51404EE	0:01:46
word07	m@rTin	F3300BA07671142DAAD3B435B51404EE	0:01:46

B.9: Tabulka úspěšně nalezených hesel pro NT haše pomocí Ophcrack

Účet	heslo	NT haš	čas prolomení [h:mm:ss]
num01	9	90AD6AB281C4AE016E5A7564C307A7E8	0:00:14
num03	380	2B7E20443D479152413D558C180E063C	0:03:25
num05	47991	0606D9A3097812CF2B2B551B8BEC72BD	0:00:17
num08	96736226	8FA0EBC6510D739D000CD08A6DB15439	0:05:29
smalla01	A	186CB09181E2C2ECAAC768C47C729904	0:00:36
smalla03	abc	E0FBA38268D0EC66EF1CB452D5885E53	0:06:04
smalla05	oemjs	53607D0EAF6D2953E23AD1A596188D3B	0:05:57
spec01	+	9B0E9CD1456256B6C6A3664BE284E517	0:00:50
spec03	&*=	E727804223F614E0DA55F6F9BB42B28C	0:19:32
spec05	*=%.;	4949A77C9E5727BC6FC4ECC6D2ACC4A3	0:02:59
mixeda03	pFF	695A2A3942B1EE426F694ED8A18684CD	0:07:06
mixeda05	OTbGp	2E02687944682C9D464BB7B1A2892E83	0:06:26
mixednuma03	I9T	6B51E08CEB34ED70CEF89F3845531C4B	0:00:21
mixednuma05	xrZx8	3FC679456646BFF64BC3D41D77BA5B80	0:05:08
mixednumaspec03	%jp	FA507AC217AF189395F1CDAC4D792E10	0:06:28
mixednumaspec05	G;3It	E8CBB3FF49B58B0A54E9D32C1F850FDF	0:08:27
word01	martin	D643DA71A76DD7EB97226D3C5AFEB01B	0:06:01
word02	1martin	EF835E1C8C1A5BE951C62FA4D8CF4311	0:00:21
word04	123dog	03F0A973BE0CFDF6BB252A48EB24C05B	0:05:00
word05	marTin	5292303D744CCBEFA057187876BA15EE	0:00:18
word06	m@rtin	586CD98F7CCA16A62432F79D7CC7D2AF	0:00:10
word07	m@rTin	F06E24CA7F9060D1713F31904CB44BE9	0:04:43

B.10: Tabulka úspěšně nalezených hesel pro LM haše pomocí John The Ripper

Účet	heslo	LM haš	čas prolomení [h:mm:ss]
num01	9	09752A3293831D17AAD3B435B51404EE	0:00:01
num03	380	5CD2A18668F8A1CCAAD3B435B51404EE	0:00:01
num05	47991	7239A6B0656FAD8DAAD3B435B51404EE	0:00:17
num08	96736226	D8C2696D88417AC5C81667E9D738C5D9	0:19:13
smalla01	a	7584248B8D2C9F9EAAD3B435B51404EE	0:00:01
smalla03	abc	8C6F5D02DEB21501AAD3B435B51404EE	0:00:01
smalla05	oemjs	36F5F68EFF305FFFAAD3B435B51404EE	0:00:12
spec01	+	C9E04F959D38CCB0AAD3B435B51404EE	0:00:01
spec03	&*=	A456A0FB7154F33EAAD3B435B51404EE	0:00:31
spec05	*=%.;	E0634096C16B162FAAD3B435B51404EE	1:37:27
mixeda03	pFF	EA6527FD755060BEAAD3B435B51404EE	0:00:02
mixeda05	OTbGp	FCBE402ABE856D70AAD3B435B51404EE	0:00:22
mixeda08	TzTOFnaH	FCA1A947FB573D7E5ACDCD7C247FA83A	0:30:16
mixednuma03	I9T	2C2724515ABFDD23AAD3B435B51404EE	0:00:01
mixednuma05	xrZx8	8D5A629565BAD6DDAAD3B435B51404EE	0:14:23
mixednuma08	3gswVGMp	3411D087F6EED6388B0EA5A7DF135B03	4:18:32
mixednumaspec03	%jp	537B02D8AA067B76AAD3B435B51404EE	0:00:22
mixednumaspec05	G;3It	CDD80A145A211FD2AAD3B435B51404EE	0:00:01
word01	martin	CB829955C5787ACAAAD3B435B51404EE	0:00:01
word02	1martin	35D7A2A64A05F923AAD3B435B51404EE	0:00:05
word03	martinmartin	F0A90DFA3ACDCD39E238065355FD3F58	0:00:05
word04	123dog	B65B9941F2388936AAD3B435B51404EE	0:00:13
word05	marTin	CB829955C5787ACAAAD3B435B51404EE	0:00:01
word06	m@rtin	F3300BA07671142DAAD3B435B51404EE	2:33:01
word07	m@rTin	F3300BA07671142DAAD3B435B51404EE	2:33:01

B.11: Tabulka úspěšně nalezených hesel pro NT haše pomocí John the Ripper

Účet	heslo	NT haš	čas prolomení [h:mm:ss]
num01	9	90AD6AB281C4AE016E5A7564C307A7E8	0:00:01
num03	380	2B7E20443D479152413D558C180E063C	0:00:03
num05	47991	0606D9A3097812CF2B2B551B8BEC72BD	0:00:26
num08	96736226	8FA0EBC6510D739D000CD08A6DB15439	0:36:44
smalla01	a	186CB09181E2C2ECAAC768C47C729904	0:00:01
smalla03	abc	E0FBA38268D0EC66EF1CB452D5885E53	0:00:01
smalla05	oemjs	53607D0EAF6D2953E23AD1A596188D3B	0:16:07
spec01	+	9B0E9CD1456256B6C6A3664BE284E517	0:00:01
spec03	&*=	E727804223F614E0DA55F6F9BB42B28C	0:00:44
mixeda03	pFF	695A2A3942B1EE426F694ED8A18684CD	0:00:20
mixeda05	OTbGp	2E02687944682C9D464BB7B1A2892E83	0:10:23
mixednuma03	I9T	6B51E08CEB34ED70CEF89F3845531C4B	0:00:14
mixednuma05	xrZx8	3FC679456646BFF64BC3D41D77BA5B80	1:23:22
mixednumaspec03	%jp	FA507AC217AF189395F1CDAC4D792E10	0:00:33
word01	martin	D643DA71A76DD7EB97226D3C5AFEB01B	0:00:01
word02	1martin	EF835E1C8C1A5BE951C62FA4D8CF4311	0:00:01
word03	martinmartin	9E94BDB778E46E2E97569660A61D42C9	0:00:01
word04	123dog	03F0A973BE0CFDF6BB252A48EB24C05B	0:00:14
word05	marTin	5292303D744CCBEFA057187876BA15EE	0:29:20

B.12: Tabulka úspěšně nalezených hesel pro LM haše pomocí Cain & Abel

účet	heslo	LM haš	čas prolomení [h:mm:ss]
num01	9	09752A3293831D17AAD3B435B51404EE	0:00:01
num03	380	5CD2A18668F8A1CCAAD3B435B51404EE	0:00:01
num05	47991	7239A6B0656FAD8DAAD3B435B51404EE	0:01:25
smalla01	a	7584248B8D2C9F9EAAD3B435B51404EE	0:00:01
smalla03	abc	8C6F5D02DEB21501AAD3B435B51404EE	0:00:01
smalla05	oemjs	36F5F68EFF305FFFAAD3B435B51404EE	0:02:28
spec01	+	C9E04F959D38CCB0AAD3B435B51404EE	0:00:01
spec03	&*=	A456A0FB7154F33EAAD3B435B51404EE	0:00:01
spec05	*=%.;	E0634096C16B162FAAD3B435B51404EE	0:03:30
mixeda03	pFF	EA6527FD755060BEAAD3B435B51404EE	0:00:01
mixeda05	OTbGp	FCBE402ABE856D70AAD3B435B51404EE	0:00:57
mixeda08	TzTOFnaH	FCA1A947FB573D7E5ACDCD7C247FA83A	5:22:14
mixednuma03	I9T	2C2724515ABFDD23AAD3B435B51404EE	0:00:01
mixednuma05	xrZx8	8D5A629565BAD6DDAAD3B435B51404EE	0:02:01
mixednumaspec03	%jp	537B02D8AA067B76AAD3B435B51404EE	0:00:01
mixednumaspec05	G;3It	CDD80A145A211FD2AAD3B435B51404EE	0:01:05
word01	martin	CB829955C5787ACAAAD3B435B51404EE	0:53:06
word02	1martin	35D7A2A64A05F923AAD3B435B51404EE	0:54:40
word04	123dog	B65B9941F2388936AAD3B435B51404EE	0:28:01
word05	marTin	CB829955C5787ACAAAD3B435B51404EE	0:53:08
word06	m@rtin	F3300BA07671142DAAD3B435B51404EE	0:55:55
word07	m@rTin	F3300BA07671142DAAD3B435B51404EE	0:55:57

B.13: Tabulka úspěšně nalezených hesel pro NT haše pomocí Cain & Abel

účet	heslo	NT haš	čas prolomení [h:mm:ss]
num01	9	90AD6AB281C4AE016E5A7564C307A7E8	0:00:01
num03	380	2B7E20443D479152413D558C180E063C	0:00:01
num05	47991	0606D9A3097812CF2B2B551B8BEC72BD	0:11:06
smalla01	a	186CB09181E2C2ECAAC768C47C729904	0:00:01
smalla03	abc	E0FBA38268D0EC66EF1CB452D5885E53	0:00:01
smalla05	oemjs	53607D0EAF6D2953E23AD1A596188D3B	0:03:57
spec01	+	9B0E9CD1456256B6C6A3664BE284E517	0:00:01
spec03	&*=	E727804223F614E0DA55F6F9BB42B28C	0:00:01
spec05	*=%.;	4949A77C9E5727BC6FC4ECC6D2ACC4A3	0:11:25
mixeda03	pFF	695A2A3942B1EE426F694ED8A18684CD	0:00:01
mixeda05	OTbGp	2E02687944682C9D464BB7B1A2892E83	0:03:21
mixednuma03	I9T	6B51E08CEB34ED70CEF89F3845531C4B	0:00:01
mixednuma05	xrZx8	3FC679456646BFF64BC3D41D77BA5B80	0:12:35
mixednumaspec03	%jp	FA507AC217AF189395F1CDAC4D792E10	0:00:01
mixednumaspec05	G;3It	E8CBB3FF49B58B0A54E9D32C1F850FDF	0:04:10
word01	martin	D643DA71A76DD7EB97226D3C5AFEB01B	4:51:20
word04	123dog	03F0A973BE0CFDF6BB252A48EB24C05B	2:14:40
word05	marTin	5292303D744CCBEFA057187876BA15EE	4:46:00
word06	m@rtin	586CD98F7CCA16A62432F79D7CC7D2AF	4:34:19
word07	m@rTin	F06E24CA7F9060D1713F31904CB44BE9	4:11:24