klima106_M_J 17.1.2006 14:29 Stránka 19

Kryptologie pro praxi – chraňme passwordy

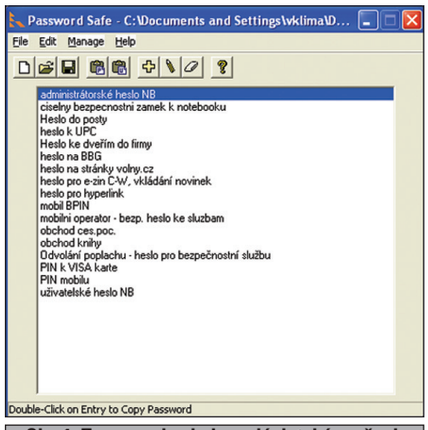
O heslech, či chcete-li slangově o passwordech, toho bylo napsáno mnoho, a mnoho ještě napsáno bude. Stále je to základní metoda autentizace ve většině prostředků, které používáme. Proč? Je nejlevnější, svede ji naprogramovat skoro každý, a skoro každý ji také dokáže používat. Hesla, kódová slova PIN a přihlašovací jména nás však převálcovala. Pro počítačově gramotného člověka se jich za chvíli nakupí tolik, že je ve své paměti prostě nemůže všechny udržet. Také nám se stalo, že jsme nejen zapomněli hesla nebo přihlašovací jména, ale zapomněli jsme dokonce i to, kde jsme si je uložili pro případ, že bychom je zapomněli. Také se stalo, že jsme takové záznamy nebo staré maily s těmito údaji smazali při nějaké čistce nebo omylem. Na pomoc proto přicházejí různé programy, které nám umožňují vytvořit a udržovat naši malou tajnou databázi passwordů a jmen bezpečně zašifrovanou.

Trezor na hesla

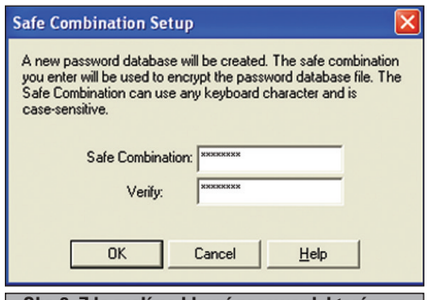
Jedním z freewarových programů obsahujících kvalitní šifru je i populární Password Safe (softwarový trezor na hesla, viz http://www.schneier.com/passsafe.html) pro Windows *(obr. 1)*. Je výjimečný z toho důvodu, že jej původně navrhl a dozoroval Bruce Schneier, známý odborník na počítačovou bezpečnost a kryptografii. Dnes si ukážeme, že i mistr tesař se umí pěkně utnout. Password Safe je opravdu jednoduchá a užitečná utilita, se kterou se naučí pracovat každý. Vlastně ji stačí spustit a na počátku zadat jméno souboru, do kterého se jako do trezoru budou zašifrovávat všechny ty tajné PIN, jména a hesla apod., a zvolit si jeden hlavní password, kterým to bude zašifrováno. Časem pak můžeme dojít do stavu, kdy si budeme pamatovat jen tenhle hlavní password. Do vytvořeného souboru pak vkládáme jednotlivé záznamy, které si pak můžeme zase vybírat a používat je v různých aplikacích. Soubor se záznamy je zašifrován algoritmem Blowfish, který je považován za bezpečný (je také z dílny B. Schneiera). Zašifrovaný soubor je pak změtí nicneříkajících bajtů. Pokud neznáme hlavní password, nemůžeme ho odšifrovat.

Zvláštní ochrana hlavního passwordu

Protože hlavní password chrání mnoho dalších passwordů a PIN i jiných důvěrných informací, bude muset být chráněn obzvláště dobře. Na každý password se však dá útočit tzv. slovníkovým útokem. Vezme se například slovník českých slov (nebo pravděpodobných passwordů) a zkoušejí se všechny výrazy, které obsahuje. Rozhodně se to nezkouší ručně, ale speciálně napsaným programem, který vlastně provádí tutéž kontrolu hlavního passwordu *(obr. 2)* jako program Password Safe. Často je u takového souboru uložena



Obr. 1 Trezor na hesla je malá databáze všech možných hesel a PIN



Obr. 2 Zde zvolíme hlavní password, kterým se bude vše šifrovat

hodnota *H*(*psw*), kde *H* je hašovací funkce. Jak víme, z této hodnoty nelze password určit (viz vlastnosti hašovacích funkcí [1]), ale můžeme passwordy zkoušet jeden po druhém a kontrolovat vypočtenou hodnotu *H*(*psw*) proti hodnotě uložené. Jelikož se jedná o nejčastější techniku, existují dokonce slovníky, které ukládají už dvojice (*psw*, *H*(*psw*)). Příslušný lušticí program pak přečte hodnotu *H*(*psw*) z hlavičky zašifrovaného souboru a jen kontroluje, jestli je uložena také ve slovníku na druhém místě. Takové slovníky existují například pro hašovací funkci MD5 a všechny možné alfanumerické řetězce do určité délky. Netřeba říkat, že tento slovník je seřazen podle druhé hodnoty, takže zjištění, zda *H*(*psw*) v něm je, je okamžité. Aby hlavní password byl ochráněn proti tomuto slovníkovému útoku, generuje se při jeho ukládání náhodný řetězec, takzvaná sůl, a k zašifrovanému souboru se uloží hodnota *sůl, H*(*psw*, *sůl*). Hodnotu soli nemohou slovníky předem znát, proto jsou na takovou ochranu krátké. Hlavní myšlenkou je, aby útočník, pokud bude chtít password zjistit procházením slovníku, musel provést příliš mnoho výpočtů. Proto se v hlavičce souboru dokonce neukládá *H*(*psw*, *sůl*), ale například *H1000*(*psw*, *sůl*), kde *H1000* je tisíckrát za sebou volaná hašovací funkce. Na vyzkoušení jednoho passwordu tak útočník musí vynaložit 1000krát více strojového času než v předchozím případě. Password Safe používá podobnou techniku, kde *H1000* vyžaduje dvakrát výpočet SHA-1 a tisíckrát volání algoritmu Blowfish. Výsledek se pak porovná s uloženou hodnotou *H1000* na počátku zašifrovaného souboru. Datová struktura zašifrovaného souboru je tato: *sůl*, *H1000*, *sůl2*, *IV*, *CBC_BF*(délka dat 4 B, 4 B nulové, *jméno1*, *password1*, *poznámky1*, ... , *jménoN*, *passwordN*, *poznámkyN*). Přitom *CBC_BF*(...) jsou zašifrovaná námi uložená data. Data se šifrují opět algoritmem Blowfish v modu Cipher Block Chaining (CBC, [2]) s náhodně voleným inicializačním vektorem *IV* a klíč k algoritmu se vypočte jako *SHA-1*(*psw*, *sůl2*).

Kde je chyba?

Útočník, který chce zjišťovat password, nemusí počítat hodnotu *H1000*, ale pouze *SHA-1*(*psw*, *sůl2*), aby vypočetl klíč a jím odšifroval data prostřednictvím *CBC_BF*(délka dat 4 B, 4 B nulové, *jméno1*, *password1*, *poznámky1*, ...). Když volí nesprávný password, po odšifrování prvního bloku dostane náhodná data, a pouze s pravděpodobností 2^{-32}= asi 10^{-9} dostane poslední čtyři bajty nulové. Na zjištění, zda password je správný nebo špatný, postačí tedy jedno volání *SHA-1* a jedno volání algoritmu Blowfish. Předchozí popsaná ochrana passwordu tisícinásobným šifrováním byla proto naprosto zbytečná a navíc jako bezpečnostní opatření zavádějící. Inu, i mistr tesař se utne. Bylo to velké překvapení, když profesionální lamači passwordů z firmy ElcomSoft toto zjistili (viz http://permalink.gmane.org/gmane.comp.security.bugtraq/20432). Podobné techniky ochrany passwordů proti slovníkovým útokům používá více programů, například program PGP, ale takováto „zkratka“ od profesionála se tam neobjevuje.

Vlastimil Klíma, Tomáš Rosa
v.klima@volny.cz, trosa@ebanka.cz