Kryptologie pro praxi – lze WinZipem kvalitně šifrovat?

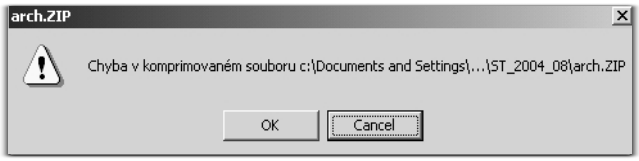
Nedávno jsme se pokoušeli s jedním známým ustavit šifrovaný e-mail. Protože z nějakého důvodu nemá rád PGP, volba padla na MS Outlook. Po určité době zkoušek, výměně řady zkušebních mailů a přečtení mnoha chybových hlášek, jsme tohoto úsilí zanechali. Proč, to je na delší povídání, ale protože jsme potřebovali problém vyřešit rychle, vrátili jsme se ke starému dobrému „WinZipu s heslem“. Je to výborný, triviální nástroj, který může danou úlohu vyřešit. Z hlediska bezpečnosti má sice své mouchy, nicméně pokud o nich budete vědět a vyvarujete se možných chyb, dostanete kvalitní šifrovací nástroj s přijatelnými nároky na administraci (v malém rozsahu, pochopitelně). WinZip navíc umožňuje posílat šifrované archivy různými mailovými klienty a pracuje s různými verzemi Windows i jiných operačních systémů, což se o šifrování integrovaném do MS Outlooku říci nedá. Aplikaci kryptografických metod ve WinZipu se věnuje článek [1], který vyšel v květnu t.r. My se seznámíme s jeho hlavními postřehy a ukážeme si, jak nejnovější verzi programu WinZip 9.0 využít pro získání vyhovující úrovně bezpečnosti i ve stavu, v jakém je. Pokud budeme v dalším hovořit o útočníkovi, budeme mít na mysli situace, kdy se šifrovaný archiv přenáší e-mailem a útočník má možnost číst zprávy a modifikovat jejich přílohy.

Co vše obsahuje archiv.zip

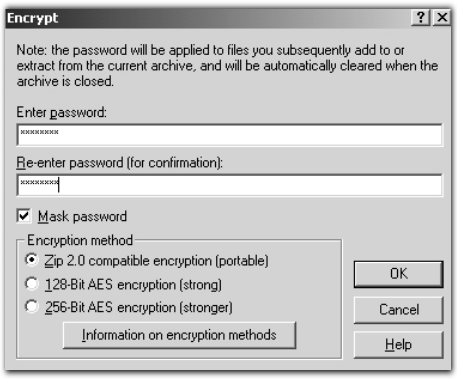
Obsah archivu *.zip je podrobně popsán ve specifikaci Info-ZIP z 3.12.2001, která je volně dostupná na ftp://ftp.info-zip.org. Podrobnosti o šifrování jsou také k dispozici – na webu http://www.winzip.com. WinZip je především určen ke komprimaci, šifrování je jen doplňkovou službou. Archiv proto může obsahovat soubory jak otevřené, tak šifrované (uvidíme, že je to první bezpečnostní chyba). Každý soubor má v archivu svůj hlavní záznam, který mu předchází (je to vlastně jeho hlavička), a potom téměř tentýž záznam, umístěný na konci celého archivu. To je centrální adresářový záznam, který slouží při dekomprimaci v případě rozložení archivu na více disket nebo CD. Při dekomprimaci konkrétního souboru si program vyžádá nejprve poslední disketu/CD a poté už jen konkrétní disketu/CD, kde je daný soubor uložen. Tyto záznamy nejsou šifrovány, což je druhá bezpečnostní chyba, a ani nejsou chráněny proti modifikaci, což je třetí chyba.

Šifrování ve WinZipu

K šifrování používá WinZip 9.0 šifrovací standard AES [2] v modu řetězení šifrového textu CBC [3], což je v pořádku. Jakmile vložíte heslo pro šifrování, je z něho vyroben klíč k AES (pomocí soli a hašovací funkce HMAC-SHA-1, [4]) a zkomprimovaný a zašifrovaný soubor je uložen s příslušnou hlavičkou do archivu. K šifrování prvního bloku otevřeného souboru se v módu CBC používá tzv. inicializační vektor (*IV*, [3]), který se ukládá do hlavičky a má být náhodný, ale předchozí verze WinZipu místo něj používaly hodnotu *CRC-32* od otevřeného souboru (čtvrtá chyba).



Obr. 1 Chybové hlášení může být také výsledkem útoku



Obr. 2 Pozor, jako defaultní se nabízí slabší šifrování

CRC-32 vyzařuje informaci

Tato chyba se může projevit u krátkých souborů s nízkou redundancí, například u souborů obsahujících PIN, telefonní čísla, soubory typu ANO/NE, obecně soubory mající do 32 bitů informace. Může to být nevýhodné i v případě, kdy útočník potřebuje pouze zjistit, zda v šifrovaném archivu je uložen jemu známý soubor (například jeho obchodní katalog nebo jeho nabídka do výběrového řízení). To vše podle *CRC-32* snadno zjistí a nemusí ani útočit na použitou šifru. Proto byl ve verzi 9.0 tento způsob šifrování (označovaný AE-1) nahrazen šifrováním AE-2, které spočívá pouze v tom, že *IV* je nulový a CRC-32 se neukládá. Nicméně z důvodu zpětné kompatibility musí verze 9.0 umět dešifrovat i AE-1, přičemž způsob šifrování je nechráněně uložen v hlavičce souboru (pátá chyba). Pokud chce útočník testovat přítomnost známého souboru, postupuje následovně: Změní způsob šifrování z AE-2 na AE-1 a vypočte *CRC-32* inkriminovaného souboru, které vloží do archivu. Z reakce příjemce pak zjistí, jestli se správně trefil do *CRC-32*. WinZip totiž *CRC-32* u AE-1

musí kontrolovat po dešifrování, jinak vyhlásí chybu. Pokud *CRC-32* souhlasí, je zde slušná pravděpodobnost, že testovaný soubor v archivu skutečně je. Obdobně může útočník postupovat, je-li na výběr několik možností krátkých otevřených souborů. Může tak například zkoušet hádat PIN přenášený v šifrovaném souboru, atp.

Otevřené názvy souborů, délka před a po kompresi

Z hlaviček v archivu vyzařují cenné informace o šifrovaných souborech (další chyba). Je to především jejich název, délka před kompresí a po ní. Název souboru přitom mnohdy řekne vše. Příkladem mohou být názvy souborů vlada_demise.doc, Lubos_a_ruzove_slipy.jpg, slibna_nabidka_Gripeny.ppt, vyvoz_Very_do_Ciny.doc, PIN.txt apod. Délka před kompresí může také okamžitě napovědět, zda je v archivu šifrován soubor, který útočník zná. Také srovnání délek před kompresí a po ní může napovědět o jaký typ souboru jde (txt, jpg, pdf apod.). V případě, že jsou takto přenášeny seznamy přístupových hesel, získává útočník cenné informace o jejich lexikální struktuře.

Záměrná změna typu komprese nebo změna koncovky

Protože služební údaje o typu komprese a jméno otevřeného souboru jsou uvedeny otevřeně v hlavičce, může je útočník libovolně měnit (další chyba). Změní-li útočník například v hlavičce typ komprese na nulovou kompresi, WinZip na straně příjemce soubor odšifruje a dekomprimuje (nulovou dekompresí). Při pokusu otevřít daný dokument (třeba soubor.xls) však příslušný program (zde Excel) zobrazí nesmysly, protože mu soubor typu xls ve skutečnosti není předkládán (je to původní komprimovaný soubor). Útočník pak zachytí e-mail v němž si adresát stěžuje, že mu došly nesmysly a jako jeho komunikující protějšek si od něj vyžádá „narušený soubor" (třeba z důvodu, aby zjistil, jestli nebyl zavirovaný na jeho počítači). Pak ho jednoduše dekomprimuje skutečnou metodou a má čistá data. Stejně to probíhá, pokud útočník změní koncovku otevřeného souboru (například z xls na doc). Program, který ho interpretuje, ukáže nesmysly stejně jako u změny komprese.

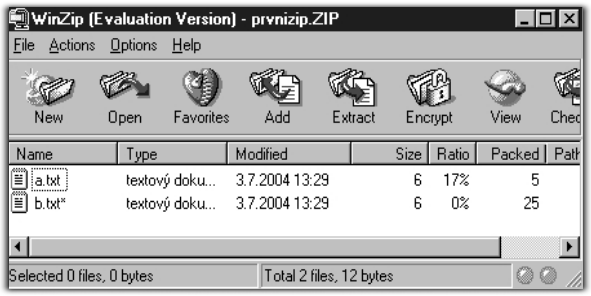
Míchání šifrovaných a nešifrovaných souborů

Zdánlivá výhoda WinZipu se mění v nástroj útoku. Útočník může do archivu jednoduše přidat svůj nešifrovaný soubor (třeba nakup_akcii_na_zitra.xls, seznam_uprav_platu.xls) nebo původní šifrovaný soubor s neznámým obsahem může zaměnit za svůj podvržený nešifrovaný soubor. Protože v archivu jsou ještě další šifrované soubory, WinZip příjemce si jako obvykle vyžádá tajné heslo a poté rozbalí všechny soubory. Místo domnělého souboru však rozbalí soubor útočníka. Příjemce se domnívá, že soubor byl zašifrován tajným heslem, takže jeho obsahu věří. Podobně může útočník nevhodný soubor z archivu zcela vymazat (třeba trestni_oznameni.doc, snizeni_platu.xls) a libovolně upravovat originální nešifrované soubory. Chyba v tomto případě pramení z absence kryptografické kontroly integrity celého archivu.

Jak se bránit možným útokům

Na úvod poznamenejme, že jsme se zde nezmiňovali o několika dalších drobnostech, které nemají tak velký vliv na bezpečnost, jako výše uvedené chyby. Jedná se například o použitý generátor náhodných znaků, který WinZip používá v metodě AE-2 a který není tak kvalitní, jak by měl, nicméně při dodržení níže uvedených zásad nebude tato chybička využitelná.

Při plánování použití WinZip k ochraně přenášených souborů musíme v každém případě nejprve zvážit, do jaké míry pro nás výše uvedené hrozby a slabiny představují reálné riziko, a podle toho se zařídit. Řadu věcí lze řešit organizačně. Chcete-li mít jistotu, nabídneme vám trochu paradoidní, zato však poměrně jistý postup: Zásadně používejte jako vstup do šifrování jen jeden soubor a zvolte vždy



Obr. 3 soubory označené křížkem jsou v archivu uloženy zašifrovaně

metodu AES se 128 nebo 256 bitovým klíčem. Jméno šifrovaného souboru nesmí vyzařovat informaci o obsahu. Pokud by délky souboru před a po zašifrování mohly vyzařovat pro vás cenné informace, je nutné použít metodu dvojího archivu: Do prvního, otevřeného archivu vložíme inkriminovaný soubor a ještě nějaké „smetí", tj. náhodně vygenerované soubory (mohou mít jména odstran.to, smeti.xxx atp.), kterými se snažíme délku našeho souboru zamaskovat. Můžeme použít například taktiku jednotné délky celého archivu. Výsledek našeho snažení musíme pojmenovat tak, aby opět nevyzařoval žádnou informaci, a připsat mu vhodnou mnemotechnickou koncovku, například takto: archiv.zip.vnitrni. Vzniklý soubor pak umístíme do druhého, šifrovaného archivu, který pojmenujeme třeba k_odeslani.zip. Maskování délky vypadá sice na první pohled komplikovaně, avšak je dlužno podotknout, že to také komplikovaný problém je, a že běžné programy a pracovní postupy si s ním pro jistotu nelámou hlavu vůbec. Naštěstí ale v běžném provozu nejsme příliš často v situaci, kdy musíme informaci o délce za každou cenu tajit. To ovšem neznamená, že tuto hrozbu můžeme zcela pustit ze zřetele. Metodu dvojího archivu použijeme i v případě, pokud je souborů pro archivaci více.

Pokud nám WinZip zahlásí nějakou chybu, rozhodně nebudeme vzniklý soubor nikomu odesílat ani hlásit jeho *CRC* apod. V případě chyby raději dotyčného kontaktujeme jinak, přičemž postupujeme velmi obezřetně – „špatně dešifrované nesmysly" mohou ve skutečnosti obsahovat citlivá otevřená data. Při příjmu vnořených souborů typu k_odeslani.zip je vhodné použít kontextové menu a dát obsah archivu dekomprimovat (a odšifrovat) do vhodně chráněného adresáře. Poté se odstraní umělá koncovka ze vzniklého souboru archiv.zip.vnitrni a vzniklý archiv.zip dáme opět pomocí kontextového menu dešifrovat do adresáře.

Uvedený, snad trochu paranoidní postup nikomu nevnucujeme a každý si může vyhodnotit, nakolik jsou uvedená doporučení pro něj aktuální. Kdo chce, má tak možnost používat WinZip bezpečněji.

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

LITERATURA

[1] T. Kohno: Analysis of the WinZip encryption method, http://eprint.iacr.org/2004/078.pdf

[2] V. Klíma, T. Rosa: Kryptologie pro praxi (6) – nejpoužívanější šifry, ST č. 11/2003, str. 16

[4] V. Klíma, T. Rosa: Kryptologie pro praxi (8) – funkce HMAC, ST č. 2/2004, str. 17

[3] V. Klíma, T. Rosa: Kryptologie pro praxi (4) – operační mód, ST č. 9/2003, str. 16

[5] E-archivy článků na http://cryptography.hyperlink.cz a http://crypto.hyperlink.cz