Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Ukladanie hesiel v open-source programoch

Bakalárska práca

2016
Frederik Koľbík

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Ukladanie hesiel v open-source programoch

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	2508 Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	doc. RNDr. Martin Stanek, PhD.

Bratislava, 2016
Frederik Koľbík

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Frederik Koľbík
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Ukladanie hesiel v open-source programoch
Storing passwords in open-source programs

Cieľ: Preskúmať a zhodnotiť spôsob ukladania používateľských hesiel v konkrétnych open-source programoch - aké algoritmy používajú, s akými hodnotami parametrov pracujú, prípadne aké možnosti konfigurácie pri ukladaní hesiel ponúkajú.

Vedúci: doc. RNDr. Martin Stanek, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: doc. RNDr. Daniel Olejár, PhD.

Dátum zadania: 22.10.2015

Dátum schválenia: 27.10.2015 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Abstrakt

Táto bakalárska práca sa zaoberá problematikou ukladania používateľských hesiel open-source programami. Prvá časť práce obsahuje prehľad rôznych metód a funkcií, ktoré sa používajú pri ukladaní hesiel a ich výhody a nevýhody. V druhej časti práce je prehľad konkrétnych programov, skúmame aké funkcie používajú, aké sú hodnoty parametrov a aké sú ďalšie možnosti konfigurácie pri ukladaní hesiel. Na záver práca obsahuje stručné zhrnutie a zhodnotenie súčasného stavu.

Kľúčové slová: heslo, hašovacia funkcia, bcrypt, PBKDF2, soľ, bezpečnosť

Abstract

This bachelor thesis deals with issues of storing user passwords by open-source programs. The first part of the thesis contains an overview of various techniques and functions that are used for storing passwords and their advantages and disadvantages. In the second part of the thesis there is an overview of specific programs, we look at what functions they use, what are the values of parameters and what are other configuration options for storing passwords. At the end the thesis contains a brief summary and assessment of the current state.

Keywords: password, hashing function, bcrypt, PBKDF2, salt, security

Obsah

Úvod

Ľudia v súčasnosti používajú mnoho programov a aplikácií, či už kvôli práci alebo zábave. Veľakrát je potrebné overiť identitu týchto používateľov, autentifikovať ich. Na to najčastejšie slúži prihlasovacie meno a heslo, ktoré si väčšinou volí používateľ sám. Aby bolo možné používateľa autentifikovať opätovne, je potrebné jeho prihlasovacie údaje uložiť.

Tým vyvstávajú otázky o bezpečnosti programov z hľadiska ukladania hesiel, treba zvoliť takú metódu, aby sa k heslám používateľov ľahko nedostal prípadný útočník. Riešenie poskytujú niektoré kryptografické konštrukcie, aj keď nie sú primárne určené na tento účel. Táto bakalárska práca skúma, ako ukladajú heslá vybrané open-source programy.

Práca má nasledovné členenie: prvá kapitola popisuje problematiku používateľských hesiel, poskytuje prehľad najpoužívanejších funkcií a metód ukladania hesiel, ich výhody a nevýhody a rôzne útoky na ne. Druhá kapitola obsahuje prehľad vybraných open-source programov, skúmame, aké funkcie používajú na ukladanie hesiel, aké sú ich parametre a aké možnosti konfigurácie programy poskytujú pri ukladaní hesiel. Pri neštandardných metódach poskytujeme stručnú analýzu ich bezpečnosti. V tretej kapitole je zhrnutie výsledkov a stručné štatistiky.

Kapitola 1

Ukladanie používateľských hesiel

V tejto kapitole popíšeme problematiku používania hesiel a najpoužívanejšie funkcie na ich ukladanie. Ďalej uvádzame rôzne útoky na získavanie hesiel.

1.1 Heslá

Americký Národný inštitút štandardov a technológií (NIST) [16] definuje heslo ako „reťazec znakov (písmen, čísel alebo iných symbolov) používaný na overenie identity alebo na kontrolu oprávnenia prístupu“. Heslá sú najčastejšie používaným prostriedkom na overenie identity používateľov. Jedným z hlavných faktorov bezpečnosti hesla je jeho dĺžka a „náhodnosť“. Používatelia si často volia ako heslá krátke, jednoduché slová (najčastejšie 6 až 10-znakové), ktoré sú pre nich ľahko zapamätateľné. Takto zvolené heslá sú len malou podmnožinou všetkých možných kombinácií znakov a teda sú málo „náhodné“. Navyše sú takéto heslá obsiahnuté v rôznych slovníkoch najpoužívanejších hesiel.

V tabuľke 1.1 uvádzame 25 najpoužívanejších hesiel v roku 2015 [20]. Tento zoznam je zostavený podľa viac ako dvoch miliónov uniknutých hesiel v roku 2015 a predstavuje asi 3% všetkých uniknutých hesiel v roku 2015. V prvej stovke najpoužívanejších hesiel sa vyskytujú aj rôzne značky áut, názvy filmov, roky, krstné mená či nadávky.

Jedným zo spôsobov ako zabrániť používaniu takýchto slabých hesiel, je nedovoliť používateľom zvoliť si vlastné heslo, ale náhodne im vygenerovať dostatočne dlhé, alfanumerické heslo. V praxi je však tento spôsob ťažko uskutočniteľný, pretože väčšina používateľov nie je schopná si takéto náhodne vygenerované heslá pamätať. Preto sa od používateľov pri voľbe hesla často požaduje, aby v ňom použili okrem malých písmen aj veľké písmená, číslice, prípadne iné špeciálne znaky.

Heslo	Zmena oproti 2014	Heslo	Zmena oproti 2014
1. 123456	—	14. 111111	+1
2. password	—	15. 1qaz2wsx	nové
3. 12345678	+1	16. dragon	−7
4. qwerty	+1	17. master	+2
5. 12345	−2	18. monkey	−6
6. 123456789	—	19. letmein	−6
7. football	+3	20. login	nové
8. 1234	−1	21. princess	nové
9. 1234567	+2	22. qwertyuiop	nové
10. baseball	−2	23. solo	nové
11. welcome	nové	24. passw0rd	nové
12. 1234567890	nové	25. starwars	nové
13. abc123	+1		

Tabuľka 1.1: Najpoužívanejšie heslá v roku 2015 [20]

1.2 Ukladanie hesiel

Aby mohla prebehnúť autentizácia používateľov pri prihlasovaní, tak musia byť heslá uložené v nejakej databáze. Pri autentizácii používateľa sa teda porovnáva zadané heslo pri prihlasovaní s jeho uloženým heslom. Očividne prvou možnosťou ako ukladať heslá, je ukladať ich ako otvorený text, čo však nie je bezpečné. Ak by sa totiž k databáze s heslami dostal prípadný útočník, mal by k dispozícii heslá všetkých používateľov v čitateľnej forme. Napriek tomu stále niektoré spoločnosti ukladajú heslá svojich používateľov v otvorenej forme. Príkladom môže byť webhostingová služba 000webhost, ktorej v roku 2015 unikli informácie o vyše 13 miliónoch používateľoch, medzi inými aj ich heslá, ktoré boli uložené ako otvorený text [3].

Lepšou možnosťou ako ukladať heslá je použiť nejakú transformačnú funkciu T a namiesto hesla ukladať výstup funkcie, $T(heslo)$, nazývaný odtlačok hesla. Pri prihlasovaní používateľa sa najskôr vypočíta odtlačok zadaného hesla a ten sa následne porovná s uloženým odtlačkom.

1.2.1 Útoky na heslá

Útoky na heslá rozdeľujeme na online a offline útoky. Pri online útoku háda útočník heslá priamo v prihlasovacom dialógu do systému. Online útoku sa dá jednoducho zabrániť – zablokovaním účtu po niekoľkých neúspešných pokusoch o prihlásenie alebo postupným predlžovaním času na opätovné prihlásenie s rastúcim počtom neplatných prihlásení. Pri offline útoku predpokladáme, že sa útočník dostal nejakým spôsobom

k databáze odtlačkov hesiel. V takomto prípade je obmedzený iba výpočtovou silou, ktorú má k dispozícii. V tejto časti uvažujeme len offline útoky na heslá.

Útočník zjavne môže útočiť hrubou silou, môže skúšať všetky možnosti reťazcov dĺžky $1, 2, \ldots$ a pre každý reťazec vypočítať jeho odtlačok a ten porovnávať s odtlačkami hesiel v databáze. Je zjavné, že teoreticky by takto útočník získal každé heslo. Takýto útok však v praxi nie je efektívny, s rastúcou dĺžkou reťazca narastá počet hesiel exponenciálne a vo všeobecnosti môže byť výpočet funkcie T relatívne časovo náročný.

Preto je pre útočníka rozumnejšie skúšať len niektoré heslá a vykonať tzv. slovníkový útok. Útočník si zostaví alebo inak získa zoznam najpoužívanejších hesiel, slovník, a ďalej postupuje podobne ako pri útoku hrubou silou, ale skúša len slová zo slovníka. Programy ako John The Ripper[1] alebo Hashcat[2] poskytujú aj rôzne variácie slovníkového útoku, kedy sa na heslá v slovníku používajú rôzne pravidlá, napr. zmena prvého písmena na veľké, pridanie číslice na koniec hesla, rôzne rotácie hesla, výmena znakov za číslice (a za 4, o za 0 a podobne) alebo vynechávanie a pridávanie písmen a iné. Takto môže útočník zistiť časť hesiel v rozumnom čase.

Pri útoku hrubou silou aj pri slovníkovom útoku si môže útočník dopredu predpočítať a ukladať odtlačky potenciálnych hesiel, tým pádom každý odtlačok vypočíta iba raz a potom len porovnáva odtlačky používateľských hesiel s uloženými odtlačkami. Nevýhodou však môže byť množstvo potrebnej pamäte na uloženie všetkých odtlačkov.

Kompromisom medzi časom potrebným na vypočítanie odtlačkov potenciálnych hesiel a pamäťou potrebnou na ich uloženie je útok dúhovými tabuľkami [11]. Pri zostrojovaní dúhových tabuliek začneme s vytváraním tzv. reťazí. Najprv si zoberieme nejaké heslo fixnej dĺžky a vypočítame jeho odtlačok transformačnou funkciou T. Na ten následne aplikujeme redukčnú funkciu R_1, takto dostaneme ďalšie heslo. Vypočítame jeho odtlačok a z neho redukčnou funkciou R_2 dostaneme ďalšie heslo. Takto pokračujeme, až kým nepoužijeme $t-1$ redukčných funkcií, z redukčnej funkcie R_{t-1} dostaneme posledné, t-te heslo a vypočítame si jeho odtlačok. Na konci si uložíme len počiatočné heslo a odtlačok posledného hesla. Tým sme vytvorili reťaz dĺžky t. Ďalej pokračujeme vytváraním ďalších $m-1$ reťazí, na konci teda máme zostrojenú dúhovú tabuľku, ktorá obsahuje najviac mt hesiel. V tabuľke však máme uložené len dvojice $(h_{i,1}, T(h_{i,t}))_{i=1}^{m}$, kde $h_{i,1}$ je prvé heslo v i-tej reťazi a $T(h_{i,t})$ je odtlačok posledného, t-teho hesla v i-tej reťazi.

Pri útoku na konkrétne heslo h porovnáme jeho odtlačok o s uloženými odtlačkami $T(h_{i,t})$ pre $i = 1, ..., m$. Ak sme našli zhodu, povedzme v j-tom riadku, vieme, že hľadané heslo sa s vysokou pravdepodobnosťou nachádza na konci j-tej reťaze, teda je to heslo $h_{j,t}$. To vieme vypočítať ako pri zostrojovaní dúhovej tabuľky striedavým apli-

[1] http://www.openwall.com/john/

[2] https://hashcat.net/hashcat/

kovaním transformačnej funkcie T a redukčných funkcií $R_1, R_2, ..., R_{t-1}$ na počiatočné heslo $h_{j,1}$: $R_{t-1}(T(R_{t-2}(...(R_2(T(R_1(h_{j,1}))))...)))$. Ak sme zhodu nenašli, vypočítame $T(R_{t-1}(o))$ a opäť porovnáme s uloženými koncovými odtlačkami. Ak sme našli zhodu, vieme heslo rekonštruovať podobne ako v predošlom prípade, ak nie pokračujeme výpočtom $T(R_{t-1}(T(R_{t-2}(o))))$. Takto pokračujeme, až kým nenájdeme hľadané heslo h alebo kým sa nedostaneme na začiatok reťazí, v takom prípade žiadna reťaz heslo h neobsahuje.

Analýzu zložitosti útokov prostredníctvom dúhových tabuliek možno nájsť v prácach [6, 8].

1.2.2 Hašovacie funkcie

Ako transformačné funkcie spomenuté vyššie sa často používajú hašovacie funkcie, ako sú MD5 [19], SHA-1, SHA-256 [18] a iné. Hašovacia funkcia $H : X \to Y$ vypočíta z väčšinou ľubovoľne dlhého vstupu jeho odtlačok pevnej dĺžky. Najčastejšie je $X = \{0,1\}^*$ a $Y = \{0,1\}^n$, kde n je zvyčajne 256, prípadne 128 alebo 512. Výhodou hašovacích funkcií je ich odolnosť voči nájdeniu vzoru a odolnosť voči nájdeniu druhého vzoru. Odolnosť voči nájdeniu vzoru znamená, že z odtlačku $y \in Y$ sa nedá efektívne vypočítať $x \in X$ také, že $H(x) = y$. Odolnosť voči nájdeniu druhého vzoru znamená, že pre dané $x \in X$ sa nedá efektívne vypočítať $x' \in X$ také, že $x \neq x'$ a $H(x) = H(x')$.

Hašovacie funkcie sú konštruované tak, aby ich výpočet bol čo najrýchlejší, preto sú samostatne nevhodné na ukladanie hesiel, keďže útočník môže vypočítať milióny odtlačkov za sekundu. Navyše, ak si viac používateľov zvolí to isté heslo, bude aj ich odtlačok rovnaký, čo uľahčuje vykonanie slovníkového útoku (alebo útoku hrubou silou), pretože v takom prípade útočník získa heslá viacerých používateľov počas jedného výpočtu. Takisto sú hašovacie funkcie zraniteľné voči útoku dúhovými tabuľkami.

Pri útoku hrubou silou sa lepšie výsledky dajú dosiahnuť použitím grafického procesora GPU namiesto CPU. Podľa testov v *GPU-based password cracking* [1] sa na procesori Intel i7 920, 2,66 GHz, dá za jednu sekundu vypočítať necelých 300 miliónov MD5 odtlačkov. Naproti tomu, použitím dvoch grafických kariet Nvidia GeForce GTX 295 je možné za jednu sekundu vypočítať približne 3,2 miliardy MD5 odtlačkov. V takomto prípade sa dá prelomiť heslo zložené iba z malých písmen so 6 znakmi za približne 1 sekundu, s 8 znakmi za necelých 90 sekúnd a 6-znakové heslo zložené z malých a veľkých písmen a číslic sa dá prelomiť za približne 20 sekúnd.

Použitie soli a počítadla iterácií

Na zvýšenie bezpečnosti hašovacej funkcie pri ukladaní hesiel sa často používa tzv. soľ. Ide o reťazec znakov, ktorý sa zreťazí s heslom a až potom sa vypočíta výsledný odtlačok: $H(heslo \mid\mid sol')$ alebo $H(sol' \mid\mid heslo)$. Soľ by mala byť ideálne generovaná

náhodne pre každého používateľa a dostatočne dlhá, napr. 64 alebo 128 bitov. Týmto sa zabezpečí to, že každý používateľ bude mať s vysokou pravdepodobnosťou rôzny odtlačok svojho hesla, aj keď má rovnaké heslo ako niektorý iný používateľ. Soľ nemusí byť tajná, môže byť uložená v databáze ako otvorený text.

Pri použití soli pri ukladaní hesiel stráca útočník možnosť útočiť slovníkovým útokom alebo hrubou silou paralelne na všetky heslá. Keďže má každý používateľ s vysokou pravdepodobnosťou rôznu soľ, musel by útočník vytvoriť slovník pre každého používateľa zvlášť a útočiť na heslá jedno po druhom. Takisto sa použitím soli stáva útok dúhovými tabuľkami neuskutočniteľným, pretože útočník si nemôže dopredu vytvoriť dúhové tabuľky, musí počkať, kým získa soľ. Aj potom však môže vytvoriť dúhovú tabuľku použiteľnú len na jedného používateľa, okrem toho je konštrukcia dúhovej tabuľky rovnako zložitá ako úplné preberanie priestoru hesiel.

Keďže soľ nemusí byť utajená, bolo by možné použiť ako soľ nejaký známy údaj, napríklad používateľské meno. V takom prípade však môže útočník zostrojiť dúhové tabuľky kde ako soľ použije konkrétne používateľské mená ako sú *root* alebo *admin*. Preto by soľ mala byť skutočne vygenerovaná náhodne.

Pri ukladaní hesiel je vysoká rýchlosť výpočtu hašovacích funkcií nežiaduca, preto je vhodné tento výpočet spomaliť, napríklad iteratívnym počítaním hašovacej funkcie: $H^c(heslo) = \underbrace{H(H(...H}_{\text{c-krát}}(heslo)...))$. Číslo c sa nazýva počítadlo iterácií a označuje koľkokrát sa iteruje výpočet hašovacej funkcie. Volením rôznych hodnôt počítadla c sa dá znížiť rýchlosť výpočtu výsledného odtlačku na požadovanú úroveň. Samozrejme, tým sa zníži aj rýchlosť pri overovaní hesla, ale kým jedno overenie hesla môže trvať rádovo stovky milisekúnd, vykonanie slovníkového útoku sa môže predĺžiť až na rádovo desiatky rokov, čím sa tento útok stáva nepoužiteľným. V praxi je vhodné používať počítadlo iterácii a soľ naraz: $H^c(heslo \mid\mid soľ')$ alebo $H^c(soľ' \mid\mid heslo)$.

1.2.3 Špeciálne funkcie

Na ukladanie hesiel je odporúčané používať zložitejšie, špeciálne navrhnuté funkcie. V tejto časti popíšeme tri najpoužívanejšie takéto funkcie.

crypt

Funkcia crypt [9] je založená na symetrickej šifre DES (Data Encryption Standard) [15]. Jej parametre sú heslo a soľ (dva náhodné znaky), pri oboch sú povolené len malé a veľké písmená anglickej abecedy, číslice, bodka a lomítko, spolu 64 znakov. Z každého z prvých ôsmich znakov hesla sa vyberie spodných sedem bitov, takto sa vytvorí 56-bitový kľúč, ktorý sa použije pri niekoľkonásobnom šifrovaní konštantého reťazca (väčšinou samých núl) šifrou DES. Keďže sa pri výpočte používa len prvých

osem znakov hesla, teda veľkosť priestoru hesiel je menej ako 2^{56}, tak funkcia crypt v súčasnosti nie je považovaná za bezpečnú.

Existujú aj verzie funkcie crypt založené na hašovacích funkciách MD5, SHA-256 alebo SHA-512.

PBKDF2

Funkcia PBKDF2 [7] (skrátene z Password Based Key Derivation Function 2) je funkcia primárne určená na odvodenie symetrických kryptografických kľúčov z používateľského hesla, v praxi sa však často používa aj na ukladanie hesiel, pričom odvodený kľúč vhodnej dĺžky považujeme za odtlačok hesla.

Funkcia PBKDF2 pri odvodzovaní kľúča používa pseudonáhodnú funkciu, ktorú označíme PRF a dĺžku jej výstupu v bajtoch označíme ako l_{PRF}. Ako vstup berie funkcia PBKDF2 heslo h, soľ s, počítadlo iterácií c a číslo l_K, čo je požadovaná dĺžka odvodeného kľúča v bajtoch, a jej výstupom je odvodený kľúč K.

NIST odporúča [21], aby bola dĺžka odvodeného kľúča minimálne 14 bajtov a dĺžka náhodne generovanej časti soli (schváleným generátorom náhodných bitov) aspoň 16 bajtov. Ďalej je odporúčané, aby bolo počítadlo čo najväčšie, minimum je 1000. Ako pseudonáhodnú funkciu je odporúčané používať HMAC [17] s ľubovoľnou schválenou hašovacou funkciou.

Výpočet kľúča prebieha takto:

1. Ak $l_K > (2^{32} - 1) \cdot l_{PRF}$, tak výpočet končí, pretože požadovaná dĺžka kľúča je príliš dlhá.

2. Nech l označuje v odvodenom kľúči počet blokov dlhých l_{PRF} bajtov, zaokrúhlený nahor, a nech r označuje počet bajtov v poslednom bloku:

$$l = \lceil l_K \ / \ l_{PRF} \rceil,$$
$$r = l_K - (l - 1) \cdot l_{PRF}.$$

3. Každý blok T_i odvodeného kľúča je výstupom funkcie F, definovanej nižšie, na heslo h, soľ s, počítadlo iterácií c a index bloku:

$$T_1 = F(h, s, c, 1),$$
$$T_2 = F(h, s, c, 2),$$
$$\ldots$$
$$T_l = F(h, s, c, l).$$

   Funkcia F je definovaná ako súčet modulo 2 (XOR) prvých c iterácií použitej pseudonáhodnej funkcie PRF aplikovanej na heslo h a soľ s zreťazenú s indexom bloku i:

$$F(h, s, c, i) = U_1 \oplus U_2 \oplus \cdots \oplus U_c,$$

kde

$$\begin{aligned} U_1 &= PRF(h, s \mid\mid INT(i)), \\ U_2 &= PRF(h, U_1), \\ &\cdots \\ U_c &= PRF(h, U_{c-1}). \end{aligned}$$

$INT(i)$ vracia 4-bajtovú hodnotu celého čísla i.

4. Odvodený kľúč je prvých l_K bajtov zreťazených blokov T_1 až T_l:

$$K = T_1 \mid\mid T_2 \mid\mid \ \ldots \ \mid\mid T_l[0{:}r-1].^3$$

bcrypt

Funkcia bcrypt (algoritmus 1) [14] vychádza zo symetrickej blokovej šifry Blowfish, jej parametre sú cena c, soľ s (náhodná 128-bitová hodnota) a kľúč k, v našom prípade heslo. Najskôr sa funkciou EksBlowfishSetup, ktorá je popísaná nižšie, inicializuje stav, teda podkľúče a iné premenné používané pri šifrovaní šifrou Blowfish. Tento stav sa použije pri 64-násobnom zašifrovaní reťazca „OrpheanBeholderScryDoubt“ funkciou EncryptECB, ktorá šifruje daný vstup šifrou Blowfish v ECB móde. Výstupom funkcie je zreťazenie ceny, soli a výsledného šifrového textu.

Algorithm 1 bcrypt(c, s, k)

```
1: stav ← EksBlowfishSetup(c, s, k)
2: stext ← „OrpheanBeholderScryDoubt“
3: repeat (64)
4:     stext ← EncryptECB(stav, stext)
5: return c || s || stext
```

Funkcia EksBlowfishSetup (algoritmus 2) najprv inicializuje stav funkciou InitState, ten sa následne aktualizuje volaním funkcie ExpandKey. Ak je jej druhý parameter 0, tak modifikuje stav rovnako pri šifre Blowfish, ak je nenulový, tak je stav modifikovaný mierne odlišným spôsobom. Potom sa 2^c-krát opäť aktualizuje stav funkciou ExpandKey, použitím samostatne soli a kľúča.

Výpočet funkcie bcrypt sa dá značne urýchliť použitím špeciálneho hardvéru, ako je FPGA (z angl. field programmable gate array – programovateľné hradlové pole) alebo ASIC (z angl. application specific integrated circuit – zákaznícky integrovaný obvod).

[3] Označenie $T_l[0{:}r-1]$ znamená prvých r bajtov bloku T_l.

Algorithm 2 EksBlowfishSetup(c, s, k)

1: $stav \leftarrow$ InitState
2: $stav \leftarrow$ ExpandKey$(stav, s, k)$
3: **repeat** (2^c)
4: $\quad stav \leftarrow$ ExpandKey$(stav, 0, s)$
5: $\quad stav \leftarrow$ ExpandKey$(stav, 0, k)$
6: **return** $stav$

Na CPU sa dá priemerne vypočítať za jednu sekundu 392,16 bcrypt odtlačkov s cenou 5 a 3,29 odtlačkov s cenou 12 (hodnoty sú priemerom z testov na piatich CPU: Intel Core i5-2400 3,1 GHz; AMD Opteron 6276 2,3 GHz; Intel Core i5-2520M 2,5 GHz; Intel Xeon E5540 2,53 GHz; Intel Xeon E3-1220 V2 3.1 GHz) [2]. Použitím FPGA Zynq-7000 XC7Z020 sa dá vypočítať za jednu sekundu 6 511 odtlačkov s cenou 5 a 51,95 odtlačkov s cenou 12. Na výkonnejšom FPGA Virtex-7 XC7VX785T je možné za jednu sekundu vypočítať 51 437 odtlačkov s cenou 5 a 410,4 odtlačkov s cenou 12 [22].

scrypt

V roku 2009 vytvoril Colin Percival algoritmus scrypt [13] určený na odvodzovanie kryptografických kľúčov. Tento algoritmus je možné použiť aj na vytváranie odtlačkov hesiel a je navrhnutý tak, že pri výpočte sa využíva veľké množstvo pamäte, čím sa stáva implementácia tohto algoritmu na špecializovanom hardvéri drahšou a zložitejšou.

Kapitola 2

Ukladanie hesiel v programoch

V tejto kapitole skúmame, akým spôsobom ukladajú heslá vybrané open-source programy. Z každej kategórie programov sme vybrali niekoľko najpoužívanejších alebo najzaujímavejších. Sumárne štatistiky a hodnotenia možno nájsť v kapitole 3.

Ak nie je uvedené inak, hašovacie funkcie sú použité bez počítadla iterácií. Pripomíname, že v štandardných implementáciach bcryptu je dĺžka soli 128 bitov a dĺžka odtlačku 184 bitov.

2.1 Frameworky

Framework je softvérová štruktúra, ktorá poskytuje podporu pri programovaní a vývoji iných softvérových aplikácií.

CakePHP

CakePHP je webový PHP framework pre vývoj webových aplikácií založených na PHP 4/5. Vo verzii 3.0 vytvára odtlačky hesiel pomocou vstavanej PHP funkcie password_hash s východzími nastaveniami. Funkcia password_hash [10] používa na vytvorenie odtlačku hesla ako východziu funkciu bcrypt s cenou výpočtu 10.

Django

Django je webový framework napísaný v jazyku Python. Používajú ho napríklad stránky Pinterest, Instagram alebo The Washington Post. Na ukladanie hesiel používa Django vo verzii 1.9 ako predvolenú funkciu PBKDF2 so soľou a počtom iterácií 30 000, ako pseudonáhodnú funkciu používa HMAC založený na hašovacej funkcii SHA-256. Dĺžka odtlačku je 64 bajtov. Nakonfigurovať sa dajú aj iné funkcie: PBKDF2 s funkciou HMAC-SHA-1, hašovacie funkcie MD5 a SHA-1, obe s použitím alebo bez použitia soli a funkcia crypt. Vo všetkých prípadoch je soľ 12 náhodných znakov zo sady malých a

veľkých písmen anglickej abecedy a číslic. Django ponúka na ukladanie hesiel aj funkciu bcrypt, na jej použitie je však potrebná externá knižnica. Keďže bcrypt podporuje maximálne 72-znakové heslá, Django ponúka aj mierne modifikovanú verziu bcryptu, kde sa najprv vypočíta odtlačok hesla hašovacou funkciou SHA-256 a naň sa následne použije bcrypt. V oboch prípadoch je cena výpočtu nastavená na 12. Výsledný odtlačok hesla je uložený v tvare `<algoritmus>$<počet_iterácií>$<soľ>$<odtlačok>`.

Ruby on Rails

Ruby on Rails je webový aplikačný framework napísaný v jazyku Ruby, využívajú ho viaceré známe webové aplikácie, napríklad GitHub, Hulu, SoundCloud alebo Twitch. Ruby on Rails verzie 5.0.0.beta3 ukladá heslá pomocou funkcie bcrypt, na čo používa knižnicu bcrypt-ruby.[1] Cena výpočtu je štandardne nastavená na 10.

Spring Security

Spring Security je framework, ktorý poskytuje autentifikáciu, autorizáciu a iné bezpečnostné funkcie pre aplikácie v jazyku Java. Vo verzii 4.0.4. Spring Security odporúča pri vývoji nových aplikácií používať na ukladanie hesiel bcrypt. Hodnota parametra ceny je predvolene nastavaná na 10. Spring podporuje aj použitie jednoduchých hašovacích funkcií, východzia je SHA-256 s 1 024 iteráciami a soľou, čo je náhodná 8-bajtová hodnota. Kvôli spätnej kompatibilite aplikácií je možné nakonfigurovať aj iné hašovacie funkcie s rôznym počtom iterácií, pričom sa soľ môže, ale nemusí použiť.

Symfony

Symfony je webový aplikačný PHP framework pre vytváranie webstránok a vývoj webových aplikácií. Na ukladanie hesiel odporúča Symfony verzie 3.0 používať bcrypt, prednastavená cena výpočtu je 13. Je možné použiť aj funkciu PBKDF2, jej východzie parametre sú 1 000 iterácií, dĺžka výsledného odtlačku 40 bajtov, soľ je náhodných 8 bajtov a ako pseudonáhodnú funkciu používa HMAC založený na hašovacej funkcii SHA-512. Dá sa taktiež použiť jednoduchá hašovacia funkcia, so soľou alebo bez, štandardne je použitá SHA-512 s 5 000 iteráciami.

Zend

Zend je webový aplikačný framework implementovaný v PHP 5. Verzia 2.4 používa na ukladanie hesiel funkciu bcrypt s cenou výpočtu 10. V prípade hesiel dlhších ako

[1] https://rubygems.org/gems/bcrypt-ruby

72 bajtov je odporúčané najskôr vypočítať odtlačok hesla funkciou SHA-256 a na ten použiť bcrypt.

2.2 Systémy pre správu obsahu (CMS) a wiki

Systém pre správu obsahu (CMS – z angl. Content Management System) je softvér na správu dokumentov, najčastejšieho webového obsahu. Wiki je označenie webových stránok, ktoré umožňujú používateľom pridávať a upravovať svoj obsah priamo cez webový prehliadač.

Alfresco

Alfresco je CMS napísaný v jazyku Java, ktorý spravuje obsah v rámci organizácie a poskytuje služby na riadenie tohto obsahu. Alfresco Community Edition 5.1 poskytuje tri funkcie na vytváranie odtlačkov hesiel, ktoré sa dajú nakonfigurovať. Prvou je hašovacia funkcia MD4, ktorá je z dôvodu spätnej kompatibility nastavená ako východzia. Druhou je hašovacia funkcia SHA-256. Obe hašovacie funkcie sú použité so soľou, ktorou je 8 náhodných bajtov. Poslednou je funkcia bcrypt s cenou výpočtu 10, pričom Alfresco používa na jej volanie Spring Security framework.

BIGACE

BIGACE je dynamický webový CMS napísaný v PHP. Projekt bol ukončený 17. januára 2016, poslednou verziou je BIGACE v3. BIGACE vytvára odtlačky hesiel kombináciou hašovacích funkcií MD5 a SHA-1, konkrétny odtlačok je v tvare `md5(soľ || sha1(soľ || heslo || soľ) || soľ)`. Soľ je štandardne konštantý reťazec `bv#ht*fr$EW%%&)78puoihGĎ6RSW$E%§"§QASETr3546j34fu9p8uoöijkl`.

DNN

DNN je CMS platforma určená pre veľké spoločnosti, používaná spoločnosťami ako Bank of America, Hilton alebo Samsung, jeho jadro je napísané v jazyku C#. Vo verzii 8.0 sa odtlačky hesiel vytvárajú pomocou C# triedy HashAlgorithm. Používajú sa východzie nastavenia, teda hašovacia funkcia SHA-1. Používa sa aj soľ ako 16-bajtová náhodná hodnota.

DokuWiki

DokuWiki je PHP wiki aplikácia bez potreby databázy, posledná verzia je označená ako „2015-08-10a 'Detritus'“. Na ukladanie hesiel používa ako východziu hašovaciu funkciu

MD5 so soľou (8 znakov). Použiť sa dajú aj ďalšie, nielen hašovacie funkcie: MD5 bez soli, SHA-1 bez soli, SHA-1 so 4-znakovou soľou, SHA-512 s 8-znakovou soľou, bcrypt s preddefinovanou cenou výpočtu 8, crypt s 2-znakovou soľou, MD5 s 8-znakovou soľou a počítadlom iterácií – skutočný počet iterácií je 2^c, kde c je parameter funkcie počítajúcej odtlačok hesla, štandardne je nastavený na 8, maximum je 30. Na výber sú ešte funkcie, ktoré vracajú odtlačok vo formáte, aký používa Django framework, konkrétne sú to MD5 a SHA-1, obe s 5-znakovou soľou. Soľ je vo všetkých prípadoch generovaná náhodne z malých a veľkých písmen anglickej abecedy a číslic.

Drupal

Drupal je CMS/webový framework používaný na tvorbu osobných blogov, obchodných či vládnych stránok. Drupal verzie 6 ukladá heslá pomocou hašovacej funkcie MD5, verzie 7 a 8 používajú SHA-512. Na vytvorenie samotného odtlačku používa PHP funkciu hash, ktorej parametre sú názov použitého algoritmu a dáta, ktorých odtlačok sa vypočíta. Pri oboch sa používa 8 náhodných znakov ako soľ a počítadlo iterácií, ktorých je 2^c, hodnota c je minimálne 7 a maximálne 30, východzia hodnota c je 19.

Family Connections

Family Connections je PHP CMS určený na tvorbu privátnych sociálnych sietí. Pri ukladaní hesiel využíva verzia 3.6.2 hašovací framework PHPass [12]. Konkrétne sa používa bcrypt s cenou výpočtu 8, dá sa použiť aj MD5 so 6-bajtovou soľou a 256 iteráciami.

Foswiki

Foswiki je wiki napísaná v jazyku Perl a JavaScript. Verzia 2.1.0 ponúka v konfigurácii viacero možností ako ukladať heslá: hašovacie funkcie MD5 a SHA-1 bez soli, MD5 s 8-znakovou soľou, funkciu crypt s 2-znakovou soľou, funkciu bcrypt s cenou výpočtu 8; taktiež je možné ukladať heslá ako otvorený text. Ako predvolená funkcia je nastavená MD5 so soľou. Soľ je vo všetkých prípadoch generovaná nasledovným kódom:

```
foreach my $i ( 0 .. 7 ) {
   $salt .= $saltchars[
      (
         int( rand( $#saltchars + 1 ) ) +
         $i +
         ord( substr( $login, $i % length($login), 1 ) ) )
      % ( $#saltchars + 1 )
   ];
```

```
}
```

Saltchars je pole znakov, ktoré obsahuje veľké a malé písmená anglickej abecedy, číslice a znaky bodku a lomítko. Každý znak soli je vybraný z tohto poľa, pričom index vybraného znaku je vypočítaný nasledovne: najprv sa vyberie náhodný index poľa saltchars a k nemu sa pripočíta číslo i a ASCII hodnota i-teho znaku prihlasovacieho mena používateľa, z výsledku sa vypočíta modulo dĺžka poľa saltchars.

Takýto spôsob generovania soli je pravdepodobne ekvivalentný náhodnému výberu znakov, pretože na začiatku sa vyberá náhodný znak, poradové číslo i tento výber nijako neovplyvňuje. Znak na i-tom mieste prihlasovacieho mena môže byť vo všeobecnosti ľubovoľný, jeho ASCII hodnota, ktorá sa nakoniec pripočítava k indexu neovplyvní to, že na konci môžeme dostať ľubovoľný znak poľa saltchars s rovnakou pravdepodobnosťou.

Joomla

Joomla je CMS napísaný v jazyku PHP. Od verzie 3.2 používa na ukladanie hesiel PHP funkciu password_hash s východzími parametrami, ktorá používa funkciu bcrypt s cenou výpočtu 10.

Kliqqi

Kliqqi je PHP CMS určený na tvorbu sociálnych sietí, je pokračovaním projektu Pliggi CMS. Vo verzii 3.0.0 sú heslá ukladané pomocou hašovacej funkcie SHA-1 so soľou. Soľ je vygenerovaná nasledovným kódom: `md5(uniqid(rand(), true))`, kde funkcia uniqid vracia 23-znakový reťazec na základe aktuálneho času. Z tohto reťazca sa vypočíta MD5 odtlačok, pričom samotná soľ je prvých 36 bitov tohto odtlačku.

Liferay

Liferay je podnikový portál založený na jazyku Java, jeho hlavnou časťou je CMS, používajú ho napríklad spoločnosti Cisco, HP, Bosch alebo ČSOB. Verzia 7 ukladá heslá štandardne pomocou funkcie PBKDF2 s 8-bajtovou soľou, 128 000 iteráciami a 160-bajtovým výsledným odtlačkom, ako pseudonáhodná funkcia je použitý HMAC s hašovacou funkciou SHA-1. Použiť sa dajú aj funkcia bcrypt s cenou výpočtu 10 a hašovacie funkcie MD5, SHA-1, SHA-256 alebo SHA-384 s 8-bajtovou soľou, v takom prípade je však nutná zmena v zdrojovom kóde a jeho prekompilovanie.

Magnolia

Magnolia je digitálna obchodná platforma, ktorej jadro tvorí CMS, napísaná je v Jave. Odtlačky hesiel sú vo verzii 5.4 počítané funkciou bcrypt s cenou výpočtu 12, Magnolia na to využíva knižnicu jBcrypt.[2]

MediaWiki

MediaWiki je wiki systém napísaný v jazyku PHP, pôvodne navrhnutý pre Wikipediu. V súčasnosti ju používajú aj iné projekty nadácie Wikimedia, napríklad Wiktionary a Wikimedia Commons. Verzia 1.26.2 používa na ukladanie hesiel ako východziu funkciu PBKDF2 s HMAC-SHA-256 ako pseudonáhodnou funkciou, dĺžka soli je 16 bajtov, počet iterácií je 10 000 a dĺžka výsledného odtlačku je nastavená na 128 bajtov. V konfiguračnom súbore je možné zmeniť východziu funkciu na bcrypt s cenou výpočtu 9.

Pimcore

Pimcore je CMS založený na frameworku Zend a napísaný v PHP. Vo verzii 4 sa ukladajú MD5 odtlačky hesiel, pričom soľ nie je povinná a je možné nastaviť, či sa má s heslom zreťaziť spredu alebo zozadu. V nastaveniach je možné zmeniť funkciu na počítanie odtlačkov hesiel na password_hash, ktorá používa bcrypt s cenou výpočtu 10. Dĺžka hesiel je v Pimcore obmedzená na 30 znakov.

TYPO3

TYPO3 je CMS napísaný v jazyku PHP. Na ukladanie hesiel vo verzii 7.6.5 používa TYPO3 upravený framework PHPass [12], kde sa používa MD5 so 6-bajtovou soľou a počítadlom iterácií, ktorých je 2^c, c je prednastavené na 14, minimum je 7 a maximum je 24. Dá sa nastaviť aj bcrypt s východzou cenou výpočtu 7 a funkcia PBKDF2, ktorá sa používa s pseudonáhodnou funkciou HMAC-SHA-256, soľ je 16 náhodných bajtov, dĺžka výsledného odtlačku je 32 bajtov a počet iterácií je 25 000.

WordPress

WordPress je CMS pre blogy a webstránky napísaný v PHP. WordPress je v súčasnosti najpopulárnejší CMS, používa ho viac ako 60 miliónov webstránok. Štandardne pri ukladaní hesiel WordPress používa framework PHPass [12], konkrétne MD5 so 6-bajtovou soľou a 64 iteráciami. Vďaka popularite WordPressu existuje množstvo pluginov, ktoré poskytujú bezpečnejšie spôsoby ukladania hesiel.

[2] http://www.mindrot.org/projects/jBCrypt/

XWiki

XWiki je wiki softvér dizajnovo prispôsobený použitiu rôznych rozšírení a je napísaný v Jave. Verzia 8.0 ukladá heslá pomocou hašovacej funkcie SHA-512, pričom sa používa náhodných 32 bajtov ako soľ, na to využíva triedu MessageDigest v Jave.

Yellow

Yellow je jednoduchý CMS na tvorbu malých blogov, wiki a webových stránok, napísaný v jazyku PHP. Vo verzii 0.6.4 sú na výber dve funkcie na ukladanie hesiel: SHA-256, ktorá je nastavená ako východzia a pri ktorej sa používa 32-bajtová soľ a bcrypt s východzou cenou výpočtu 10.

2.3 Systémy riadenia výučby

Systém riadenia výučby (LMS – z angl. learning management system) je softvér na organizáciu a administráciu výuky.

Chamilo

Chamilo je LMS napísaný v PHP. Verzia 1.10.4 využíva pri ukladaní hesiel framework Symfony. Konkrétne sa používa bcrypt s cenou výpočtu 4 alebo hašovacie funkcie MD5 a SHA-1, obe bez soli. Tiež sa dajú heslá uložiť ako otvorený text.

ILIAS

ILIAS je LMS napísaný v jazyku PHP. Od verzie 5 používa na ukladanie hesiel funkciu bcrypt s cenou výpočtu 8, predošlé verzie používali funkciu MD5 bez soli. Na samotné heslo sa najprv použije PHP funkcia str_pad, ktorá vytvorí reťazec dĺžky 4-krát dĺžka hesla, pričom heslo je v tomto reťazci v strede a zvyšok reťazca je vyplnený SHA-1 odtlačkom soli použitej pri výpočte bcryptu. Z tohto reťazca sa vypočíta ďalší odtlačok pomocou funkcie hash_hmac:[3]

```
hash_hmac(
        'whirlpool',
        str_pad($raw, strlen($raw) * 4, sha1(salt), STR_PAD_BOTH),
        this->getClientSalt(),
        true
)
```

[3] http://php.net/manual/en/function.hash-hmac.php

a až na ten sa použije bcrypt. Premenná $raw je zadané používateľské heslo.

Dôvod prečo sa používa najprv funkcia hash_hmac nie je v kóde ani v dokumentácii uvedený, domnievame sa, že sa počíta odtlačok hesla touto funkciou kvôli tomu, že funkcia bcrypt akceptuje heslá maximálnej dĺžky 72 bajtov. Preto sa môžu bez ujmy používať aj dlhšie heslá, keďže funkcia hash_hmac vracia s horeuvedenými parametrami 64-bajtový odtlačok.

Moodle

Moodle je LMS napísaný v PHP. Verzia 3.0.3 ukladá heslá pomocou funkcie bcrypt s cenou výpočtu 10, prostredníctvom PHP funkcie password_hash. Je možné použiť aj tzv. rýchly výpočet, vtedy sa cena nastaví na hodnotu 4.

Sakai

Sakai je LMS napísaný v Jave. Vo verzii 10.7 sú odtlačky hesiel počítané hašovacou funkciou SHA-256 s použitím 4-bajtovej soli.

2.4 Cloud Management

CloudStack

CloudStack je softvér poskytujúci služby na tvorbu a manažovanie platformy cloud computingu, napísaný je v jazyku Java. Verzia 4.7 ukladá heslá štandardne pomocou hašovacej funkcie SHA-256 s 32-bajtovou soľou. Ďalšie možnosti sú hašovacia funkcia MD5 bez soli a funkcia PBKDF2, ktorá používa 64-bajtovú soľ, ako pseudonáhodná funkcia je použitý HMAC s hašovacou funkciou SHA-1, počet iterácií je 100 000 a dĺžka výsledného odtlačku je 512 bajtov, zmena je však možná až s prekompilovaním programu.

Eucalyptus

Eucalyptus je softvér na tvorbu privátnych cloudových prostredí, je napísaný v Jave. Vo verzii 4.2.1 sa na ukladanie hesiel používa funkcia crypt založená na hašovacej funkcii SHA-512 s východzím počtom iterácií 5 000 a 16-znakovou soľou, znaky sú vyberané zo sady malých a veľkých písmen anglickej abecedy, číslic, bodky a lomítka.

OpenNebula

OpenNebula je platforma určená na spravovanie infraštuktúry distribuovaných datacentier napísaná v jazyku C++. Heslá sú vo verzii 4.14 ukladané hašovacou funkciou

SHA-1 bez použitia soli.

OpenStack

OpenStack je softvérová platforma pre cloud computing napísaná v jazyku Python. Verzia nazvaná „Mitaka“ ukladá heslá pomocou funkcie crypt založenej na hašovacej funkcii SHA-512 s počtom iterácií 10 000, minimum iterácií je 1 000 a maximum je 100 000. Soľ je 16 náhodných znakov zo sady malých a veľkých písmen anglickej abecedy, číslic, bodky a lomítka. OpenStack na to využíva knižnicu PassLib.[4]

ownCloud

OwnCloud je softvér na tvorbu a používanie webových úložísk, je napísaný v PHP. Odtlačky hesiel vo verzii 9.0 sú počítané funkciou bcrypt s cenou výpočtu 10, prostredníctvom PHP funkcie password_hash.

Scalr

Scalr je PHP platforma určená na spravovanie multi-cloudovej infraštruktúry. Scalr Community Edition verzie 5.10 ukladá heslá pomocou hašovacej funkcie SHA-256 bez použitia soli.

2.5 Bugtrackery

Bugtracker je softvérová aplikácia, ktorá sleduje hlásené chyby a nedostatky softvéru.

Bugzilla

Bugzilla je bugtracker napísaný v jazyku Perl, pôvodne vyvíjaný pre projekt Mozilla. Verzia 5.0.2 na ukladanie hesiel používa hašovaciu funkciu SHA-256, soľ je 8 náhodných znakov zo sady malé a veľké písmená anglickej abecedy a číslice.

Flyspray

Flyspray je bugtracker napísaný v PHP. Heslá sú vo verzii 1.0 štandardne ukladané hašovacou funkciou MD5 bez soli. V konfiguračnom súbore sa dá použitá funkcia zmeniť na hašovaciu funkciu SHA-1 bez použitia soli alebo na funkciu crypt.

[4] http://pythonhosted.org/passlib/index.html

MantisBT

MantisBT je PHP bugtracker. Odtlačky hesiel sú vo verzii 1.2.19 počítané hašovacou funkciou MD5 bez soli. Kvôli kompatibilite s predošlými verziami je podporovaná aj funkcia crypt a aj ukladanie hesiel v otvorenej forme.

2.6 E-Commerce

E-commerce softvér sa používa na tvorbu a správu internetových obchodov.

Magento

Magento je e-commerce platforma napísaná v jazyku PHP. Magento Community Edition verzie 2.0.5 ukladá heslá pomocou hašovacej funkcie SHA-256 s 32-bajtovou soľou, je možné použiť aj verziu bez soli. Taktiež je možné aj funkciu MD5, so soľou alebo bez.

nopCommerce

NopCommerce je e-commerce softvér založený na frameworku ASP.NET MVC a je určený na tvorbu internetových obchodov, napísaný je v jazyku C#. Verzia 3.70 ukladá heslá hašovacou funkciou SHA-1 s 5-bajtovou soľou.

OpenCart

OpenCart je systém na manažovanie internetových obchodov napísaný v PHP. Vo verzii 2.2.0.0 sú heslá ukladané pomocou hašovacej funkcie SHA-1 s 9 náhodnými znakmi, ktoré sa vyberajú z malých a veľkých písmen anglickej abecedy a číslic, ako soľou, výsledný odtlačok má tvar `sha1(soľ || sha1(soľ || sha1(heslo)))`.

osCommerce

OsCommerce je PHP systém na správu online obchodov. Verzia 2.3.4 počíta MD5 odtlačky hesiel s 6-bajtovou soľou a 1024 iteráciami, na to používa framework PHPass [12], v ňom je možné nastaviť aj funkciu bcrypt s cenou výpočtu 10.

OXID

OXID je e-commerce softvér na tvorbu e-shopov, napísaný je v PHP. OXID eShop Community Edition verzie 4.9.7 ukladá heslá pomocou hašovacej funkcie SHA-512 so soľou, tou je 32 náhodných bajtov.

PrestaShop

PrestaShop je softvér na tvorbu a správu internetových obchodov, je napísaný v jazyku PHP. Heslá sú vo verzii 1.6 štandardne ukladané pomocou funkciou bcrypt s cenou výpočtu 10. Dá sa použiť aj hašovacia funkcia MD5 so soľou – 56 náhodných znakov vyberaných z malých a veľkých písmen anglickej abecedy a číslic.

Sylius

Sylius je e-commerce platforma napísaná v PHP. Verzia 0.17.0 ukladá heslá pomocou funkcie PBKDF2 s 1 000 iteráciami, dĺžka soli je 20 bajtov, dĺžka konečného odtlačku je 40 bajtov a ako pseudonáhodná funkcia je použitá HMAC-SHA-512.

X-Cart

X-Cart je PHP e-commerce platforma. Verzia 4.7.5 využíva pri ukladaní hesiel framework PHPass [12], konkrétne sa používa bcrypt s cenou výpočtu 11.

2.7 Blogy, fóra, sociálne siete

b2evolution

B2Evolution je systém na tvorbu blogov, multi-blogov a diskusných fór, napísaný je v PHP. Verzia 6.6.8 ukladá heslá pomocou hašovacej funkcie MD5 so soľou (8 náhodných znakov vybraných z malých a veľkých písmen anglickej abecedy a číslic).

Beehive Forum

Beehive je PHP projekt na vytváranie diskusných fór. Verzia 1.4.7 používa na ukladanie hesiel funkciu bcrypt s cenou výpočtu 10.

Diaspora

Diaspora je distribuovaná sociálna sieť napísaná v jazyku Ruby. Odtlačky hesiel sú vo verzii 0.5.8.0 sú počítané funkciou bcrypt s cenou výpočtu 10.

Discourse

Discourse je platforma napísaná v jazykoch Ruby a Javascript určená na tvorbu diskusných fór a chatov. Heslá sú vo verzii 1.5 ukladané pomocou funkcie PBKDF2,

ako pseudonáhodná funkcia je použitá funkcia HMAC-SHA-256, počet iterácií je štandardne nastavený na 64 000, dĺžka soli je 8 bajtov a dĺžka výsledného odtlačku je 32 bajtov.

Ghost

Ghost je blogovacia platforma napísaná v Javascripte. Vo verzii 0.7.9 sú odtlačky hesiel počítané funkciou bcrypt s cenou výpočtu 10.

phpBB

PhpBB je systém na vytváranie diskusných fór, je napísaný v PHP. Od verzie 3.1 používa na ukladanie hesiel funkciu bcrypt s cenou výpočtu 10. Je možné použiť aj hašovaciu funkciu MD5 so soľou (6 náhodných znakov zo sady malých a veľkých písmen anglickej abecedy, číslic, bodky a lomítka) alebo bez nej.

punBB

punBB je PHP systém na tvorbu diskusných fór. Verzia 1.4.4 počíta odtlačky hesiel pomocou hašovacej funkcie SHA-1 s 12-bajtovou soľou, odtlačok vyzerá nasledovne: `sha1(soľ || sha1(heslo))`.

2.8 Účtovníctvo

FrontAccounting

FrontAccounting je účtovnícky softvér pre malé a stredné podniky. Napísaný je v PHP a verzia 2.4 ukladá heslá pomocou hašovacej funkcie MD5 bez soli.

jBilling

JBilling je podnikový fakturačný systém napísaný v Jave. JBilling Community Edition verzie 4.1.1 používa na počítanie odtlačkov hesiel framework Spring Security. Ako východzia je nastavená funkcia bcrypt s cenou výpočtu 10, je možné použiť aj hašovacie funkcie: MD5 bez soli, MD5 so 16-bajtovou soľou, SHA-1 s 20-bajtovou soľou alebo SHA-256 s 32-bajtovou soľou, na to je však potrebné kód prekompilovať. Tiež je možné ukladať heslá ako otvorený text.

Siwapp

Siwapp je aplikácia na správu a vytváranie faktúr, je napísaná v jazyku Ruby. Vo verzii 0.4.2-beta sú heslá ukladané pomocou funkcie bcrypt s cenou výpočtu 10, pričom sa používa knižnica bcrypt.[5]

2.9 Iné

Cacti

Cacti je PHP nástroj na tvorbu grafov sieťovej prevádzky. Verzia 0.8.8h počíta odtlačky hesiel pomocou hašovacej funkcie MD5 bez soli.

eHour

EHour je time-tracking softvér napísaný v Jave. Pri ukladaní hesiel využíva verzia 1.4.3 Spring Security Framework, konkrétne používa hašovaciu funkciu SHA-1 so soľou – tou je náhodné 4-ciferné číslo: `(int) (Math.random() * 10000)`.

Check_MK

Check_MK je nástroj na automatizované sledovanie stavov počítačových sietí, napísaný je v jazyku Python. Vo verzii 1.2.8 používa pri ukladaní hesiel hašovaciu funkciu MD5. Na začiatku sa takto postupne buduje reťazec m: na začiatku obsahuje heslo a soľ (tou je 6-ciferné číslo – miliónnásobok aktuálneho unixového času v sekundách skonvertovaného na číslo z intervalu $< 0, 1)$), potom sa vypočíta osobitne odtlačok o_1. Do reťazca m sa sa l-krát (l je dĺžka zadaného hesla) pridá niektorý znak odtlačku o_1:

```
mixin = md5(password + salt + password).digest()
for i in range(0, len(password)):
        m.update(mixin[i % 16])
```

Následne sa vykoná cyklus, počas ktorého sa do reťazca m pridá nula alebo prvý znak hesla:

```
i = len(password)
while i:
        if i & 1:
                m.update('\x00')
        else:
                m.update(password[0])
```

[5] https://rubygems.org/gems/bcrypt

```
        i >>= 1
```

Po skončení cyklu sa z reťazca m vypočíta odtlačok o_2. Potom sa buduje v cykle reťazec m_2, tisíckrát sa doňho pridá heslo, odtlačok o_2 (v kóde je to final) alebo soľ:

```
for i in range(1000):
        m2 = md5()
        if i & 1:
                m2.update(password)
        else:
                m2.update(final)

        if i % 3:
                m2.update(salt)

        if i % 7:
                m2.update(password)

        if i & 1:
                m2.update(final)
        else:
                m2.update(password)
```

Z reťazca m_2 sa nakoniec vypočíta výsledný odtlačok.

V takomto počítaní odtlačku hesla nevidíme výrazné slabiny, ale ani pozitíva. Lepšie by bolo použiť jednoduché počítadlo iterácií, prípadne aj s viac ako 1 000 iteráciami.

iDempiere

IDempiere je podnikový informačný systém napísaný v jazyku Java. Verzia 3.1 počíta odtlačky hašovacou funkciou SHA-512 s 8-bajtovou soľou a počítadlom iterácií, ktorých je 1 000.

Mautic

Mautic je PHP platforma na marketingovú automatizáciu. Verzia 1.4.0 používa pri ukladaní hesiel Symfony framework, konkrétne funkciu bcrypt s cenou výpočtu 12.

OpenMRS

OpenMRS je aplikácia, ktorá umožňuje navrhovať systémy lekárskych záznamov, je často používaná v rozvojových krajinách. Aplikácia je napísaná v Jave a vo verzii

1.11.6 ukladá heslá pomocou hašovacej funkcie SHA-512 so soľou, ktorá je generovaná nasledovným kódom:

```
encodeString(
        Long.toString(System.currentTimeMillis())
         + Long.toString(rng.nextLong())
);
```

Z náhodného 64-bitového čísla a aktuálneho času v milisekundách sa vypočíta SHA-512 odtlačok, ktorý sa použije ako soľ.

Soľ by stačilo generovať ako 8 náhodných bajtov, keďže sa vygeneruje náhodné 64-bitové číslo, aktuálny čas v milisekundách tento náhodný výber neovplyvňuje. Počítanie odtlačku vygenerovaného čísla funkciou SHA-512 môžeme vnímať ako snahu o zvýšenie entropie soli (pozri kapitolu 3).

PacketFence

PacketFence je systém na kontrolu sieťovej komunikácie, napísaný je v jazyku Perl. Verzia 6.0.1 používa na ukladanie hesiel funkciu bcrypt s cenou výpočtu 8.

pfSense

PfSense je firewall založený na operačnom systéme FreeBSD.[6] Napísaný je v PHP a odtlačky hesiel sú vo verzii 2.3 počítané funkciou bcrypt s cenou výpočtu 10.

]project-open[

]project-open[je softvér na riadenie projektov napísaný v jazyku Tcl. Heslá sú vo verzii 4.0 ukladané pomocou hašovacej funkcie SHA-1 so soľou (40 náhodných znakov vybraných z malých a veľkých písmen anglickej abecedy a číslic).

Redmine

Redmine je webový softvér na riadenie projektov napísaný v jazyku Ruby. Verzia 3.2.1 využíva na ukladanie hesiel hašovaciu funkciu SHA-1 s 8-bajtovou soľou. Odtlačok hesla je uložený v tvare `SHA1(soľ || SHA1(heslo))`.

Secrets for Android

Secrets for Android je androidová aplikácia na manažovanie a ukladanie hesiel, používateľovi si stačí pamätať iba jedno hlavné heslo (master password). Vo verzii 2.4.5 sa

[6] https://www.freebsd.org/

z hlavného hesla počíta odtlačok funkciou bcrypt, pomocou ktorého sa neskôr odvodí kľúč na šifrovanie hesiel symetrickou šifrou. Cena výpočtu je maximálna hodnota, pri ktorej sa vygeneruje kľúč za menej ako 0,9 sekúnd.

X2CRM

X2CRM je informačný systém pre manažment vzťahov so zákazníkmi, napísaný je v PHP. Na počítanie odtlačkov hesiel používa verzia 6.0 funkciu PBKDF2 s HMAC-SHA-256 ako pseudonáhodnou funkciou, pričom dĺžka soli je 24 bajtov, počet iterácií je nastavený na 32 768 a dĺžka výsledného odtlačku je 24 bajtov.

Kapitola 3

Zhrnutie a zhodnotenie stavu

V tejto kapitole zhrnieme naše doterajšie výsledky a uvedieme stručné štatistiky.

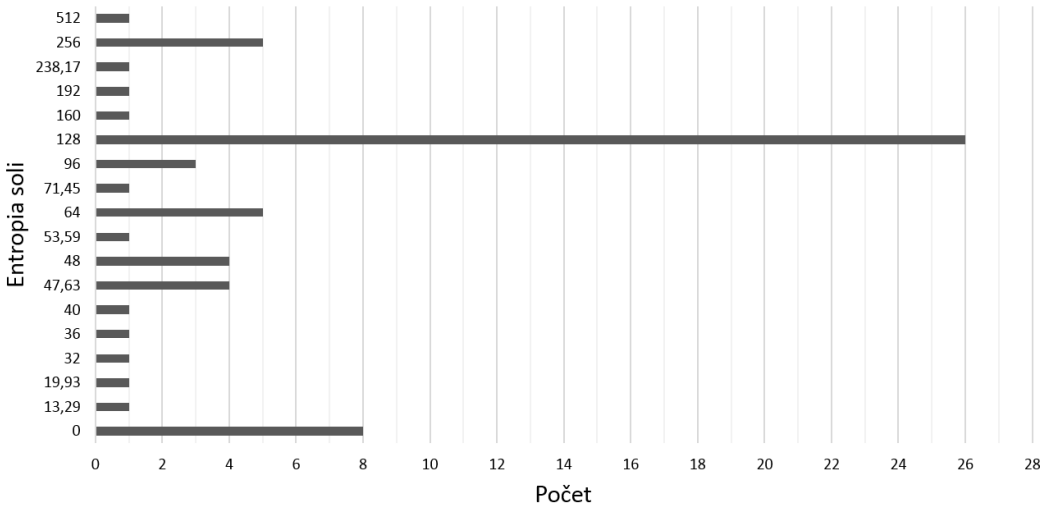
3.1 Soľ a iterácie

Ak sa ako soľ používajú náhodne generované bajty, programy na to využívajú štandardné knižnice, ktoré implementujú generátory pseudonáhodných čísel. Ak sa pri generovaní soli vyberajú znaky, väčšinou sa vyberajú náhodne zo sady malých a veľkých písmen anglickej abecedy a číslic (spolu 62 znakov), prípadne doplnenej o bodku a lomítko (spolu 64 znakov). Rozdiel medzi týmito metódami generovania soli je vo výslednej entropii soli. Všeobecne, entropia vyjadruje, koľko bitov informácie obsahuje nejaká správa. Entropiu soli H môžeme vypočítať vzťahom $H = l \cdot \log_2 n$, kde l je dĺžka soli a n je počet rôznych stavov. Napríklad, pri bajtoch sa n rovná 256, keďže jeden bajt môže nadobúdať 256 rôznych hodnôt.

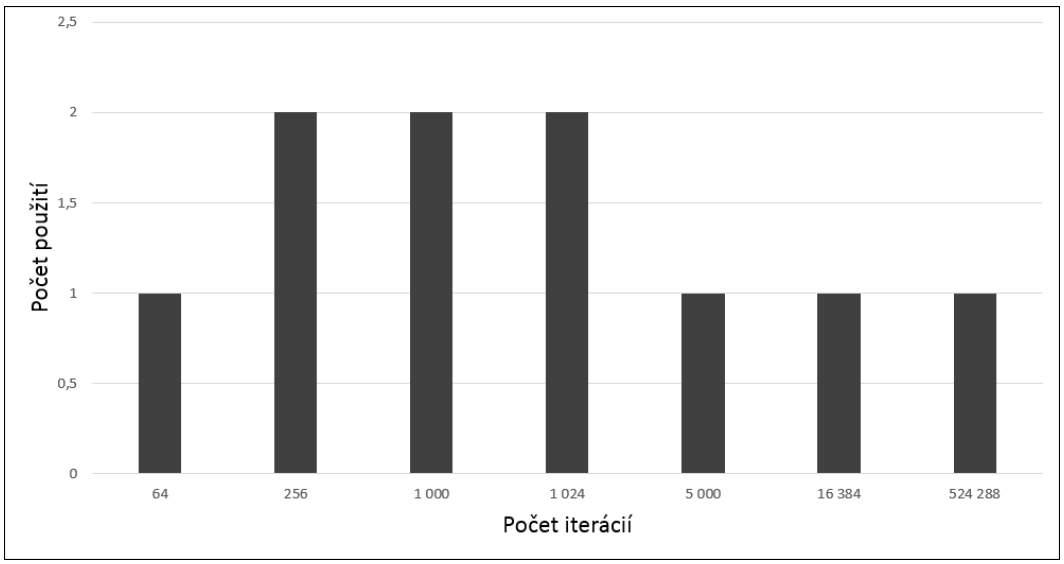
Entropia 16-znakovej soli, kde sa znaky vyberajú z množiny o veľkosti 62, je $16 \cdot \log_2 62 \approx 95{,}27$ bitov, entropia soli, ktorou je 4-ciferné číslo, je $4 \cdot \log_2 10 \approx 13{,}29$ bitov. Teda 8-bajtová soľ a 8-znaková soľ nie sú z hľadiska entropie ekvivalentné, entropia prvej je $H_1 = 8 \cdot \log_2 256 = 64$ bitov a entropia druhej je $H_2 = 8 \cdot \log_2 64 = 48$ bitov, ak sa znaky vyberajú zo 64-prvkovej množiny. Ak by sme v tomto prípade chceli dosiahnuť entropiu tiež aspoň 64 bitov, potrebovali by sme soľ dlhú minimálne $\lceil \frac{64}{\log_2 64} \rceil = 11$ znakov.

Graf na obrázku 3.1 ukazuje, akú dlhú soľ, resp. koľko bitov entropie majú soli použité pri výpočtoch odtlačkov hesiel východzími funkciami programov z kapitoly 2. Najpočetnejšia entropia a aj dĺžka soli je 128 bitov, keďže funkcia bcrypt používa soľ len tejto dĺžky. Na druhom mieste je entropia 0 bitov kvôli prípadom, kedy sa soľ vôbec nepoužíva.

Ak sa počítajú odtlačky hesiel pomocou hašovacích funkcií, používa sa počítadlo iterácií minimálne. Graf na obrázku 3.2 koľko iterácií a koľko programov ich používa



Obr. 3.1: Entropia soli pri východzích funkciách



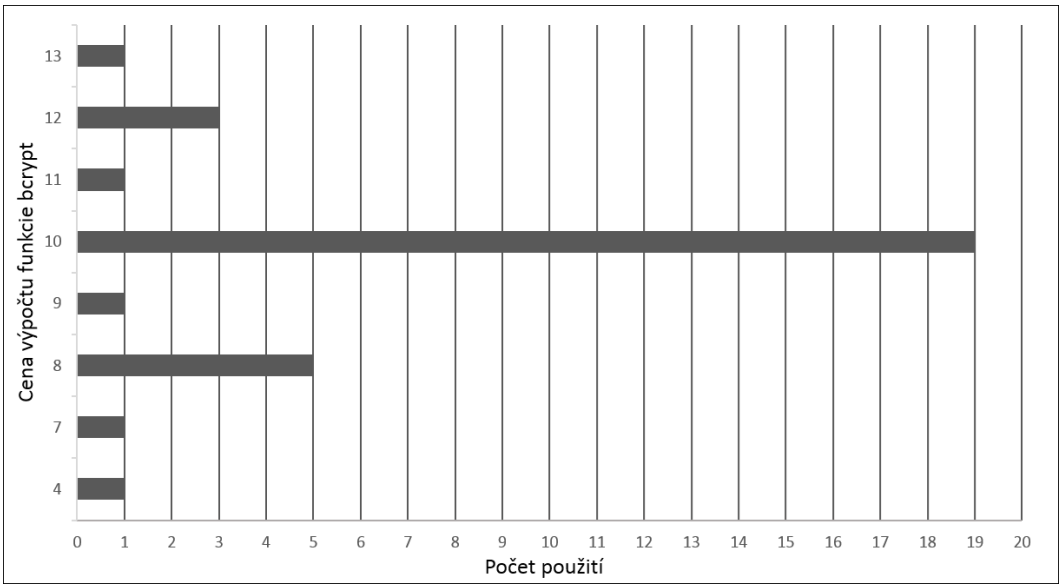
Obr. 3.2: Počty iterácií hašovacích funkcií

v spojení s hašovacími funkciami, pripomíname, že 16 384 = 2^{14} a 524 288 = 2^{19}. Najmenšie dve hodnoty a hodnoty 1 000, 1 024 a 16 384 sú použité s funkciou MD5, hodnota 1 024 je použitá s funkciou SHA-256 a hodnoty 1 000, 5 000 a najvyšší počet iterácií sú použité s funkciou SHA-512.

Kvôli rýchlosti výpočtu hašovacích funkcií je potrebné použiť čo najviac iterácií tak, aby to používatelia sotva postrehli pri prihlasovaní. Použiť len 64 alebo 256 iterácií s funkciou MD5 nepovažujeme za bezpečné.

3.2 Parametre funkcií bcrypt a PBKDF2

Na obrázku 3.3 je graf, ktorý vyjadruje, koľkokrát sú použité jednotlivé ceny výpočtov. Najviac sa používa cena výpočtu 10, keďže táto hodnota je nastavená v mnohých knižniciach a implementáciách bcryptu ako východzia.

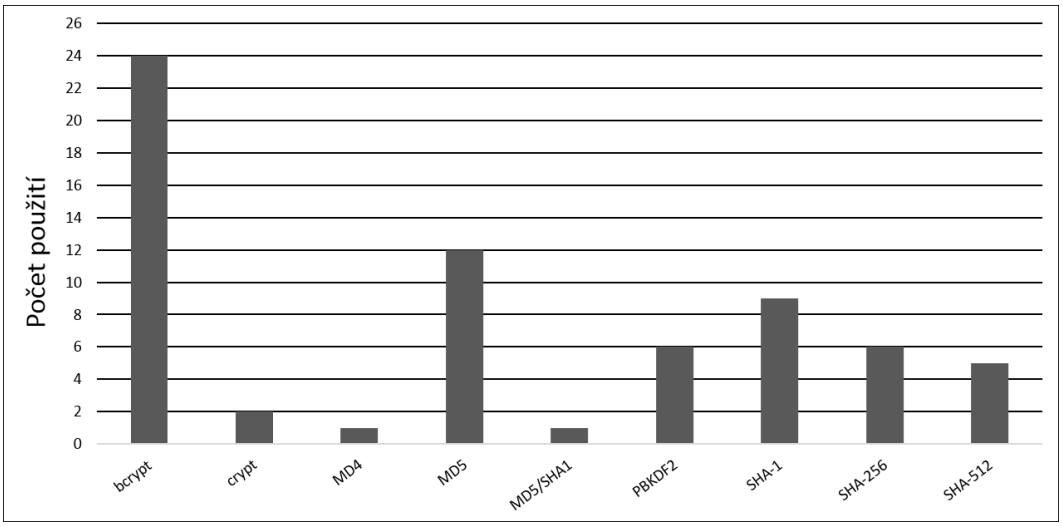


Obr. 3.3: Ceny výpočtu vo funkcii bcrypt

Tabuľka 3.1 ukazuje počet použití rôznych hodnôt parametrov funkcie PBKDF2. Je vidieť, že v rôznych programoch sú použité rôzne hodnoty, teda neexistuje nejaké štandardné nastavenie parametrov, ktoré by sa často používalo.

Entropia soli	Počet	Počet iterácií	Počet	Dĺžka odtlačku v bajtoch	Počet	Pseudonáhodná funkcia	Počet
64	3	1 000	2	24	1	HMAC-SHA-1	3
71,45	2	10 000	1	32	2	HMAC-SHA-256	5
128	2	25 000	1	40	2	HMAC-SHA-512	2
160	1	30 000	2	64	2		
192	1	32 768	1	128	1		
512	1	64 000	1	160	1		
		100 000	1	512	1		
		128 000					

Tabuľka 3.1: Štatistika parametrov funkcie PBKDF2



Obr. 3.4: Použité funkcie

3.3 Použité metódy

Graf na obrázku 3.4 zobrazuje aké funkcie používajú programy ako východzie a počet použití. Vidíme, že viac ako tretina skúmaných programov používa na ukladanie hesiel funkciu bcrypt, ktorá je považovaná za bezpečnú, to vnímame pozitívne. Na druhej strane sa ešte stále používajú hašovacie funkcie MD4 a MD5 bez soli, čo nepovažujeme za bezpečné. Jedným z možných dôvodov, prečo sa používajú, je spätná kompatibilita s predošlými verziami programov.

V niektorých prípadoch sa používajú hašovacie funkcie vnorene, napr. takto: `sha1(soľ || sha1(heslo))`, alebo sa používajú v kombinácii s druhou: `md5(soľ || sha1(soľ || heslo || soľ) || soľ)`. Takéto používanie hašovacích funkcií nepovažujeme za ničím prínosné, lepšie by bolo použiť počítadlo iterácií.

Presne tretina skúmaných programov podporuje aj použitie iných, niekedy aj bezpečnejších funkcií ako sú východzie. Do niektorých aplikácií, napr. WordPress, sa zasa dajú doinštalovať rôzne pluginy, ktoré poskytujú bezpečnejšie funkcie na ukladanie hesiel. V mnohých, najmä webových aplikáciách sa používajú alebo dajú použiť aj iné autentifikačné metódy, napr. protokol OAuth [4, 5]. To sú ďalšie z možných dôvodov, prečo sa pri ukladaní hesiel používajú menej bezpečné funkcie.

3.4 Celkové zhrnutie

V tabuľke 3.4 uvádzame kompletné zhrnutie o ukladaní hesiel programami z kapitoly 2.

Tabuľka 3.2: Zhrnutie

Programy	Jazyk	Predvolená funkcia	Soľ	Počet iterácií/cena výpočtu	Dĺžka odtlačku	Podpora inej funkcie
Frameworky						
CakePHP	PHP	bcrypt	16 B	10	184 b	nie
Django	Python	PBKDF2	12 znakov	30 000	512 b	áno
Ruby on Rails	Ruby	bcrypt	16 B	10	184 b	nie
Spring Security	Java	bcrypt	16 B	10	184 b	áno
Symfony	PHP	bcrypt	16 B	13	184 b	áno
Zend	PHP	bcrypt	16 B	10	184 b	nie
Systémy pre správu obsahu (CMS) a wiki						
Alfresco	Java	MD4	8 B	—	128 b	áno
BIGACE	PHP	MD5/SHA-1	konštanta	—	128 b	nie
DNN	C#	SHA-1	16 B	—	160 b	nie
DokuWiki	PHP	MD5	8 znakov	—	128 b	áno
Drupal	PHP	SHA-512	8 znakov	524 288	512 b	nie
Family Connections	PHP	bcrypt	16 B	8	184 b	áno
Foswiki	Perl	MD5	8 znakov	—	128 b	áno
Joomla	PHP	bcrypt	16 B	10	184 b	nie
Kliqqi	PHP	SHA-1	36 b	—	160 b	nie
Liferay	Java	PBKDF2	8 B	128 000	1 280 b	áno
Magnolia	Java	bcrypt	16 B	12	184 b	nie
MediaWiki	PHP	PBKDF2	16 B	10 000	1 024 b	áno
Pimcore	PHP	MD5	—	—	128 b	áno
TYPO3	PHP	MD5	6 B	16 384	128 b	áno
WordPress	PHP	MD5	6 B	64	128 b	áno
XWiki	Java	SHA-512	32 B	—	512 b	nie
Yellow	PHP	SHA-256	32 B	—	256 b	áno
Systémy riadenia výučby						
Chamilo	PHP	bcrypt	16 B	4	184 b	áno

ILIAS	PHP	bcrypt	16 B	8	184 b	nie
Moodle	PHP	bcrypt	16 B	10	184 b	nie
Sakai	Java	SHA-256	4 B	—	256 b	nie
Cloud Management						
CloudStack	Java	SHA-256	16 B	—	256 b	áno
Eucalyptus	Java	crypt	16 znakov	5 000	512 b	nie
OpenNebula	C++	SHA-1	—	—	160 b	nie
OpenStack	Python	crypt	16 znakov	10 000	512 b	nie
ownCloud	PHP	bcrypt	16 B	10	184 b	nie
Scalr	PHP	SHA-256	—	—	256 b	nie
Bugtrackery						
Bugzilla	Perl	SHA-256	8 znakov	—	256 b	nie
Flyspray	PHP	MD5	—	—	128 b	áno
MantisBT	PHP	MD5	—	—	128 b	áno
E-commerce						
Magento	PHP	SHA-256	32 B	—	256 b	áno
nopCommer	C#	SHA-1	5 B	—	160 b	áno
OpenCart	PHP	SHA-1	9 znakov	—	160 b	nie
osCommerce	PHP	MD5	6 B	1 024	128 b	áno
OXID	PHP	SHA-512	32 B	—	512 b	áno
PrestaShop	PHP	bcrypt	16 B	10	184 b	áno
Sylius	PHP	PBKDF2	20 B	1 000	320 b	nie
X-Cart	PHP	bcrypt	16 B	11	184 b	nie
Blogy, fóra, sociálne siete						
b2evolution	PHP	MD5	8 znakov	—	128 b	áno
Beehive Forum	PHP	bcrypt	16 B	10	184 b	nie
Diaspora	Ruby	bcrypt	16 B	10	184 b	nie
Discourse	Ruby	PBKDF2	8 B	64 000	256 b	nie
Ghost	Javascript	bcrypt	16 B	10	184 b	nie
phpBB	PHP	bcrypt	16 B	10	184 b	áno
punBB	PHP	SHA-1	12 B	—	160 b	nie
Účtovníctvo						
FrontAccour	PHP	MD5	—	—	128 b	nie
jBilling	Java	bcrypt	16 B	10	184 b	áno
Siwapp	Ruby	bcrypt	16 B	10	184 b	nie
Iné						
Cacti	PHP	MD5	—	—	128 b	nie
eHour	Java	SHA-1	4-ciferné číslo	—	160 b	nie

Check_MK	Python	MD5	6-ciferné číslo	1 000	128 b	nie
iDempiere	Java	SHA-512	8 B	1 000	512 b	nie
Mautic	PHP	bcrypt	16 B	12	184 b	nie
OpenMRS	Java	SHA-512	64 B	—	512 b	nie
PacketFence	Perl	bcrypt	126 B	8	184 b	nie
pfSense	PHP	bcrypt	16 B	10	184 b	nie
]project-open[	Tcl	SHA-1	40 znakov	—	160 b	nie
Redmine	Ruby	SHA-1	8 B	—	160 b	nie
Secrets for Android	Android	bcrypt	16 B	podľa výkonu	184 b	nie
X2CRM	PHP	PBKDF2	24 B	32 768	192 b	nie

Záver

V tejto bakalárskej práci sme sa venovali problematike používania používateľských hesiel, popísali sme v praxi najpoužívanejšie funkcie na ukladanie hesiel a útoky na ne. Poskytli sme prehľad vybraných open-source programov a popísali spôsoby, akým ukladajú používateľské heslá. Na záver sme poskytli stručné zhrnutie a zhodnotenie získaných výsledkov.

Naše výsledky považujeme za mierne pozitívne prekvapivé. Najpoužívanejšie z vybraných programov ukladajú heslá bezpečným spôsobom. Všetky programy používajú štandardné funkcie na ukladanie hesiel, väčšina z nich tiež používa soľ, niektoré aj počítadlo iterácií. Na druhej strane pri tých programoch, ktoré používajú napr. MD5 bez soli nevidíme dôvod, prečo by to tak malo byť, keďže vo všetkých najpoužívanejších programovacích jazykoch existujú funkcie PBKDF2 alebo bcrypt ako vstavané alebo existujú knižnice, kde sú implementované.

Naše odporúčanie pri ukladaní hesiel je používať funkciu bcrypt s cenou výpočtu aspoň 10 alebo funkciu PBKDF2 s dostatočným počtom iterácií (najlepšie desiatky alebo stovky tisíc) a s náhodne generovanou, dostatočne dlhou, ideálne aspoň 32-bajtovou, soľou.

Literatúra

[1] Marcus Bakker and Roel Van Der Jagt. GPU-based Password Cracking. *University of Amsterdam, System and Network Engineering, Amsterdam, Research*, 2010.

[2] Markus Dürmuth and Thorsten Kranz. On Password Guessing with GPUs and FPGAs. In *Technology and Practice of Passwords*, pages 19–38. Springer, 2015.

[3] Thomas Fox-Brewster. 13 Million Passwords Appear To Have Leaked From This Free Web Host, 2015. `http://www.forbes.com/sites/thomasbrewster/2015/10/28/000webhost-database-leak/`.

[4] Eran Hammer-Lahav. RFC 5849, The OAuth 1.0 Protocol. 2010. `https://tools.ietf.org/pdf/rfc5849.pdf`.

[5] Dick Hardt. RFC 6749, The OAuth 2.0 Authorization Framework. 2012. `https://tools.ietf.org/pdf/rfc6749.pdf`.

[6] Jin Hong and Sunghwan Moon. A Comparison of Cryptanalytic Tradeoff Algorithms. *Journal of Cryptology*, 26(4):559–637, 2013.

[7] Burt Kaliski. RFC 2898; PKCS# 5: Password-Based Cryptography Specification Version 2.0. 2000. `https://tools.ietf.org/pdf/rfc2898.pdf`.

[8] Byoung-Il Kim and Jin Hong. Analysis of the Non-Perfect Table Fuzzy Rainbow Tradeoff. In *Information Security and Privacy*, pages 347–362. Springer, 2013.

[9] Linux Programmer's Manual. CRYPT(3). `http://man7.org/linux/man-pages/man3/crypt.3.html`.

[10] PHP Manual. password_hash function. `http://php.net/manual/en/function.password-hash.php`.

[11] Philippe Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *Advances in Cryptology-CRYPTO 2003*, pages 617–630. Springer, 2003.

[12] Openwall. Portable PHP password hashing framework. `http://www.openwall.com/phpass/`.

[13] Colin Percival. Stronger Key Derivation via Sequential Memory-hard Functions. *Self-published*, pages 1–16, 2009.

[14] Niels Provos and David Mazieres. A Future-Adaptable Password Scheme. In *USENIX Annual Technical Conference, FREENIX Track*, pages 81–91, 1999.

[15] FIPS PUB. 46-3: Data Encryption Standard (DES). *National Institute of Standards and Technology*, 1999. `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`.

[16] FIPS PUB. 140-2: Security Requirements for Cryptographic Modules. *National Institute of Standards and Technology*, 2001. `http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf`.

[17] FIPS PUB. 198-1 The Keyed-Hash Message Authentication Code (HMAC). *National Institute fo Standards and Technology*, 2008. `http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf`.

[18] FIPS PUB. 180-4 Secure Hash Standard (SHS). *National Institue for Standards and technology*, 2015. `http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf`.

[19] Ronald Rivest. RFC 1321; The MD5 Message-Digest Algorithm. 1992. `http://tools.ietf.org/pdf/rfc1321.pdf`.

[20] SplashData. Announcing Our Worst Passwords of 2015, 2016. `https://www.teamsid.com/worst-passwords-2015/`.

[21] Meltem Sönmez Turan, Elaine Barker, William Burr, and Lily Chen. Recommendation for Password-Based Key Derivation. *NIST special publication*, 800:132, 2010.

[22] Friedrich Wiemer and Ralf Zimmermann. High-Speed Implementation of bcrypt Password Search Using Special-Purpose Hardware. In *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, pages 1–6. IEEE, 2014.

Zoznam programov

Každý záznam obsahuje tri odkazy – odkaz na stránku projektu, odkaz na dokumentáciu a odkaz na repozitár zdrojového kódu. Ak sú v zázname len dva odkazy, tak prvý je odkaz na projekt a druhý je odkaz na repozitár a dokumentácia je ľahko dostupná z hlavnej stránky projektu.

1. CakePHP
   `http://cakephp.org/`
   `http://book.cakephp.org/3.0/en/index.html`
   `https://github.com/cakephp/cakephp`

2. Django
   `https://www.djangoproject.com/`
   `https://docs.djangoproject.com/en/1.9/`
   `https://github.com/django/django`

3. Ruby on Rails
   `http://rubyonrails.org/`
   `http://api.rubyonrails.org/`

4. Spring Security
   `http://projects.spring.io/spring-security/`
   `http://docs.spring.io/spring-security/site/docs/4.0.4.RELEASE/apidocs/`
   `/overview-summary.html`
   `https://github.com/spring-projects/spring-security`

5. Symfony
   `http://symfony.com/`
   `http://symfony.com/doc/current/book/security.html`
   `https://github.com/symfony/symfony`

6. Zend
   `http://framework.zend.com/`
   `http://framework.zend.com/manual/current/en/index.html`
   `https://github.com/zendframework/zend-crypt`

7. Alfresco
   https://www.alfresco.com/
   http://docs.alfresco.com/community/concepts/welcome-infocenter_community.html
   https://github.com/Alfresco/community-edition

8. BIGACE
   http://www.bigace.de/
   http://wiki.bigace.de/
   https://github.com/bigace/bigace3

9. DNN
   http://www.dnnsoftware.com/
   https://github.com/dnnsoftware/Dnn.Platform

10. DokuWiki
    https://www.dokuwiki.org/
    https://github.com/splitbrain/dokuwiki

11. Drupal
    https://www.drupal.org/
    https://www.drupal.org/documentation/administer
    https://github.com/drupal/drupal

12. Family Connections
    https://www.familycms.com/index.php
    https://github.com/ryanhowdy/fcms

13. Foswiki
    http://foswiki.org/Home/WebHome
    http://foswiki.org/System/WebHome
    https://github.com/foswiki/distro

14. Joomla
    https://www.joomla.org/
    https://docs.joomla.org/Main_Page
    https://github.com/joomla/joomla-cms

15. Kliqqi
    http://www.kliqqi.com/
    https://github.com/Kliqqi-CMS/Kliqqi-CMS

16. Liferay
    https://www.liferay.com/web/guest/home
    https://dev.liferay.com/discover/portal/-/knowledge_base/7-0/what-is-liferay
    https://github.com/liferay/liferay-portal

17. Magnolia
    https://www.magnolia-cms.com/
    https://documentation.magnolia-cms.com/display/DOCS

18. MediaWiki
    https://www.mediawiki.org/wiki/MediaWiki
    https://doc.wikimedia.org/mediawiki-core/master/php/index.html

19. Pimcore
    https://www.pimcore.org/
    https://www.pimcore.org/wiki/
    https://github.com/pimcore/pimcore

20. TYPO3
    https://typo3.org/
    https://docs.typo3.org/
    https://github.com/TYPO3/TYPO3.CMS

21. WordPress
    https://wordpress.org/
    https://codex.wordpress.org/Main_Page
    https://core.trac.wordpress.org/browser

22. XWiki
    http://www.xwiki.org/xwiki/bin/view/Main/WebHome
    https://github.com/xwiki/xwiki-platform

23. Yellow
    http://datenstrom.se/yellow/
    https://github.com/datenstrom/yellow

24. Chamilo
    https://chamilo.org/
    https://github.com/chamilo/chamilo-lms

25. ILIAS
    http://www.ilias.de/docu/goto_docu_root_1.html
    https://github.com/ILIAS-eLearning/ILIAS

26. Moodle
    https://moodle.org/
    https://docs.moodle.org/30/en/Main_page
    https://github.com/moodle/moodle

27. Sakai
    https://www.sakaiproject.org/
    https://github.com/sakaiproject/sakai

28. CloudStack
    http://cloudstack.apache.org/index.html
    https://git-wip-us.apache.org/repos/asf?p=cloudstack.git;a=summary

29. Eucalyptus
    http://www8.hp.com/us/en/cloud/helion-eucalyptus-overview.html
    https://github.com/eucalyptus/eucalyptus

30. OpenNebula
    http://opennebula.org/
    https://github.com/OpenNebula/one

31. OpenStack
    http://www.openstack.org/
    http://docs.openstack.org/
    https://github.com/openstack/keystone

32. ownCloud
    https://owncloud.org/
    https://doc.owncloud.org/
    https://github.com/owncloud/core

33. Scalr
    http://www.scalr.com/
    https://scalr-wiki.atlassian.net/wiki/display/docs/Home
    https://github.com/Scalr/scalr

34. Bugzilla
    https://www.bugzilla.org/
    https://www.bugzilla.org/docs/
    https://git.mozilla.org/?a=project_list;pf=bugzilla

35. Flyspray
    http://www.flyspray.org/
    https://github.com/Flyspray/flyspray

36. MantisBT
https://www.mantisbt.org/index.php
https://www.mantisbt.org/docs/master-1.2.x/en/administration_guide.html
https://github.com/mantisbt/mantisbt

37. Magento
https://magento.com/
http://devdocs.magento.com/
https://github.com/magento/magento2

38. nopCommerce
http://www.nopcommerce.com/default.aspx
http://docs.nopcommerce.com/display/nc/nopCommerce+Documentation
https://github.com/nopSolutions/nopCommerce

39. OpenCart
http://www.opencart.com/
http://docs.opencart.com/
https://github.com/opencart/opencart

40. osCommerce
https://www.oscommerce.com/
https://github.com/osCommerce/oscommerce2

41. OXID
https://www.oxid-esales.com/en/home.html
https://github.com/OXID-eSales/oxideshop_ce

42. PrestaShop
https://www.prestashop.com/
https://github.com/PrestaShop/PrestaShop

43. Sylius
http://sylius.org/
https://github.com/Sylius/Sylius

44. X-Cart
http://www.x-cart.com/
http://help.x-cart.com/

45. b2evolution
http://b2evolution.net/
https://github.com/b2evolution/b2evolution

46. Beehive Forum
    http://www.beehiveforum.co.uk/
    https://github.com/BeehiveForum/BeehiveForum

47. Diaspora
    https://joindiaspora.com/
    https://wiki.diasporafoundation.org/Main_Page
    https://github.com/diaspora/diaspora

48. Discourse
    http://www.discourse.org/
    https://github.com/discourse/discourse

49. Ghost
    https://ghost.org/
    https://github.com/TryGhost/Ghost

50. phpBB
    https://www.phpbb.com/
    https://area51.phpbb.com/
    https://github.com/phpbb/phpbb

51. punBB
    http://punbb.informer.com/
    http://punbb.informer.com/wiki/start
    https://github.com/punbb/punbb

52. FrontAccounting
    http://frontaccounting.com/
    https://github.com/FrontAccountingERP/FA

53. jBilling
    http://www.jbilling.com/
    http://www.jbilling.com/documentation/developers

54. Siwapp
    http://www.siwapp.com/
    https://github.com/siwapp/siwapp

55. Cacti
    http://www.cacti.net/
    http://svn.cacti.net/viewvc/

56. eHour
https://ehour.nl/
https://github.com/te-con/ehour

57. Check_MK
http://mathias-kettner.com/check_mk.html
https://github.com/opinkerfi/check_mk

58. iDempiere http://www.idempiere.org/
http://wiki.idempiere.org/wiki/Main_Page
https://bitbucket.org/idempiere/idempiere

59. Mautic
https://www.mautic.org/
https://github.com/mautic/mautic

60. OpenMRS
http://openmrs.org/
https://wiki.openmrs.org/display/docs/Home
https://github.com/openmrs/openmrs-core

61. PacketFence
http://packetfence.org/
https://github.com/inverse-inc/packetfence

62. pfSense
https://www.pfsense.org/
https://doc.pfsense.org/
https://github.com/pfsense/pfsense

63. |project-open[
http://www.project-open.com/
http://www.project-open.com/en/index.html

64. Redmine
http://www.redmine.org/projects/redmine
http://www.redmine.org/projects/redmine/repository/show/branches

65. Secrets for Android
https://play.google.com/store/apps/details?id=net.tawacentral.roger.secrets
https://github.com/rogerta/secrets-for-android

66. X2CRM
http://www.x2crm.com/
http://wiki.x2crm.com/wiki/Main_Page
https://github.com/X2Engine/X2CRM