Hashchain a jeho využitie - Chlebovec

Zreťazené hashovacie funkcie - skrátene hashchain má využitie v informatike a autentizácii. Jedná sa o formu aplikácie hashovacích funkcií viacnásobným hashovaním výsledného hashu rovnakou hashovacou funkciou. Pri prvom použití hashovacej funkcie je použitý hash na vstupné dáta, ktoré môžu byť definované napríklad používateľom, jedná sa o plaintext akejkoľvek dĺžky.

Výsledkom hashovacej funkcie je reťazec fixnej dĺžky. Dĺžka reťazca po aplikovaní hashovacej funkcie je daný jej charakteristikou, napríklad: sha256 má 256-bitový výstup, ktorý sa v znakovej sade prejaví ako 64-znakový výstup, pričom je každý znak 4-bitový. Hashovacie funkcie sú caps-sensitívne, čo znamená, že na výsledný hash reťazec má vplyv malé i veľké písmeno. Viacnásobným hashovaním môžeme autentizovať napríklad používateľa s jeho univerzálnym heslom, pričom každé overenie používateľa bude znamenať posun o -1 hash. Z výstupu hashovacej funkcie je pôvodná informácia nedopočítateľná (jednocestný model hashovacích funkcií).

Príklad: Server si uloží 1000*hash(vstup) a používateľ pri autentizácii použije 999*hash(vstup). Server pri prevzatí hashu vykoná jeho opätovný hash a porovná, či sa zhoduje s jeho uloženým. V prípade, že áno, používateľa autentizuje.Pri ďalšom overení už bude server očakávať vstup 998*hash(vstup), pričom on si uloží 999*hah(vstup).

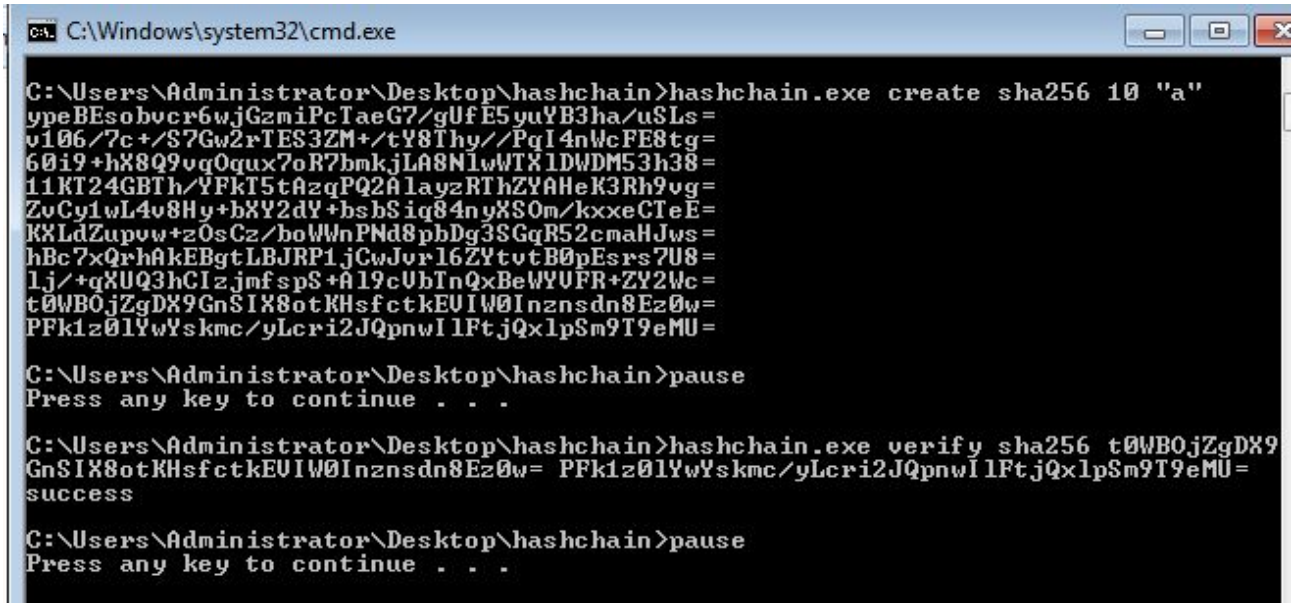
Výhody: Univerzálnosť, rýchlosť, lavínovitosť (malá zmena, znak, caps a pod. úplne zmení hash na nepoznanie), jednocestnosť
Nevýhody: Možnosť kolízie, vypočítanie rovnakého hashu z rôznych vstupov

Domáca úloha - Hashchain v jazyku C

Domácu úlohu som realizoval pre overenie funkčnosti programu z Github repozitára. Projekt som realizoval v Linuxe, ale nakoľko problémom s inštaláciou a použitím prekladača MingW som testovanie musel realizovať na Windowse 7 vo virtuálnom stroji.

Dlho som bojoval s kompiláciou, nakoľko bolo potrebné pri kompilácii nastaviť umiestnenie knižníc kryptografického nástroja openssl, nakoľko kompilátor stále hlásil, že umiestnenie nepozná.Cestu bolo nutné zmeniť na umiestnenie C:\BATCHES\include s prepínačom -I pri kompilácii. Následne kompilátor nebol schopný nalinkovať lcrypto knižnice, ktoré boli potrebné. Riešením ich bolo presunúť z umiestnenia C:\BATCHES\lib do lib knižnice prekladača MingW: C:\Program Files\CodeBlocks\MinGW\lib.

Program je po skompilovaní do .exe formátu možno spustiť z 32-bitového operačného systému Windows z príkazového riadka, nakoľko potrebuje ďalšie parametre pre svoj chod. Využíva parametre príkazového riadka. Pre jeho ľahšie a pohodlnejšie ovládanie a kompiláciu som vytvoril sériu kratkých batch (.bat) súborov, ktoré demonštrujú využitie hashchain funkcií s dĺžkou 10 so vstupom jednoznakového reťazca “a” na viacerých hashovacích funkciách.



Okrem funkcie pre vytvorenie hash štruktúry (create) obsahujú batch súboru aj spustenie programu s funkciou na overenie hashov (verify), kedy je jeden vstupný hash v base64 formáte hashovaný zadanou hashovacou funkciou a porovnáva sa s druhým zadaným hashom. Vstupy používateľa a výstupy systému pre hashe sú v base64 formáte pre user-friendly reprezentáciu binárneho hashu. Pri porovnaní hashov sú hashe dekódované a následne hashované. Ak sa hashe po hashovaní ľavého hashu zhodujú, program vypíše success, v opačnom prípade failure. Porovnáva sa vždy hash n a hash n+1 (následujúci).

Použitie programu

Príkaz pre kompiláciu: `gcc hashchain.c -o hashchain.exe -lcrypto -I C:\BATCHES\include`
Príkaz pre vytvorenie hashchainu: `hashchain.exe create hashovacia_funkcia dĺžka "vstup"`
Príkaz pre porovnanie hashov hashchainu: `hashchain.exe verify hashovacia_funkcia hash1_base64 hash2_base64`
Funkcie sú obsiahnuté v batch súboroch a vzorovo obsahujú odtestovanie týchto hashovacích funkcií s overením hashov č.9 a č.10:

- md5
- sha1
- sha256
- sha512

Referát vyhotovil: Martin Chlebovec, FEI, 3. roč. Bc., Počítačové siete