VAJA: Izdelava zgoščenih vrednosti (ang. hashing)

Namen

V sodobnem digitalnem svetu je varovanje integritete in avtentičnosti podatkov ključnega pomena. Namen te vaje je udeležence seznaniti z uporabo zgoščevalnih (ang. hash) funkcij, ki so temelj informacijske varnosti. S praktičnimi primeri zgoščevanja (ang. hashing) in preverjanja zgoščevalnih vrednosti bomo raziskali, kako lahko te tehnike uporabimo za zagotavljanje, da so podatki ostali nespremenjeni med shranjevanjem ali prenosom. Vaja omogoča razumevanje temeljnih konceptov kriptografije in poudarja pomen kriptografskih zgoščevalnih funkcij v procesih zagotavljanja varnosti in zasebnosti.

Uvod

Kriptografske hash funkcije so temeljni gradniki v svetu informacijske varnosti, ki omogočajo enosmerno pretvorbo vhodnih podatkov v edinstveno vrednost fiksne dolžine, znano kot zgoščevalna vrednost (ang. hash). Zaradi svojih lastnosti, kot so hitrost izračuna in odpornost na kolizije, so hash funkcije nepogrešljive pri preverjanju celovitosti podatkov, avtentikaciji in drugih varnostnih aplikacijah. S pomočjo orodja OpenSSL bomo spoznali osnovne postopke za ustvarjanje in preverjanje hash vrednosti.

Vajo bomo izvajali v operacijskem sistemu Kali Linux.

Naloge:

1. Prenos datotek iz strežnika:

Odprite terminal v sistemu Kali Linux. Iz spletnega mesta https://cybersec.ltfe.org/vaje2/ prenesite datoteke »letter_to_grandma.txt«, »sample.img_SHA256.sig« in »sample.img«

Za prenos datoteke uporabite ukaz:ca

```
wget https://cybersec.ltfe.org/vaje2/letter_to_grandma.txt
```

```
wget https://cybersec.ltfe.org/vaje2/sample.img
```

```
wget https://cybersec.ltfe.org/vaje2/sample.img_SHA256.sig
```

2. Prikaz datoteke: Vpišite spodnji ukaz za prikaz vsebine datoteke letter_to_grandma.txt:

```
┌──(kali㉿kali)-[~]
└─$ cat letter_to_grandma.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies
you sent me. I got them this morning and I have already eaten half
of the box! They are absolutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.
```

3. Izračun zgoščene vrednosti
   3.1. Iz terminalnega okna izvedite spodnji ukaz za izračun zgoščene vrednosti tekstovne datoteke. Ukaz bo uporabil algoritem SHA-2-256 za generiranje hash vrednosti. Hash vrednost bo prikazana na zaslonu, ko bo OpenSSL izračunal vrednost.

```
┌──(kali㉿kali)-[~]
└─$ openssl sha256 letter_to_grandma.txt
SHA2-256(letter_to_grandma.txt)=
28c110d47c3f0293717329324a00533b1b687413eaaccdb449982a4aa0764ce1
```

Pozorni bodite na format izpisa. OpenSSL prikaže uporabljen algoritem za izračun hash vrednosti, SHA-256, sledi ime datoteke, ki se uporablja kot vhodni podatek. Hash vrednost SHA-256 je prikazana za enačajem.

Zgoščevalne funkcije so uprorabne za preverjanje integritete podatkov, ne glede na to, ali gre za sliko, pesem ali preprosto tekstovno datoteko. Najmanjša sprememba povzroči popolnoma drugačni prstni odtis. Prstni odtisi se lahko izračunajo pred in po prenosu primerjajo. Če se vrednosti ne ujemajo, so bili podatki med prenosom spremenjeni.

   3.2. Spremenimo datoteko letter_to_grandma.txt in ponovno izračunajmo prstni odtis. Izvedite spodnji ukaz za odprtje urejevalnika besedil nano v ukazni vrstici:

```
┌──(kali㉿kali)-[~]
└─$ nano letter_to_grandma.txt
```

V nano spremenite prvi stavek iz ‘Hi Grandma’ v ‘Hi Grandpa’. Opazimo, da spreminjamo samo en znak, ‘m’ v ‘p’. Po spremembi pritisnite tipki <CONTROL+X> za shranjevanje spremenjene datoteke. Pritisnite ‘Y’ za potrditev imena in shranite datoteko. Pritisnite tipko <Enter> in izšli boste iz nano, da nadaljujete z naslednjim korakom.

   3.3. Sedaj, ko je bila datoteka spremenjena in shranjena, ponovno izvedite isti ukaz za generiranje zgoščene vrednosti SHA-2-256 datoteke.

```
┌──(kali㉿kali)-[~]
└─$ openssl sha256 letter_to_grandma.txt
SHA2-256(letter_to_grandma.txt)=
663a781df0ee5d88d74b7ec001f79c48b8a1744fd800668819c6a3bc1b1ac48f
```

Vprašanje: Ali se nova zgoščena vrednost razlikuje od vrednosti izračunane v 3.1 točki? Kako se razlikuje?

   3.4. Uporabiti je mogoče tudi zgoščevalni algoritem z daljšo dolžino bitov, na primer SHA-2-512. Za generiranje zgoščene vrednosti SHA-2-512 datoteke letter_to_grandma.txt uporabite spodnji ukaz:

```
┌──(kali㉿kali)-[~]
└─$ openssl sha512 letter_to_grandma.txt
SHA2-512(letter_to_grandma.txt)=
c63ec8dcfaaa11470c5aace0df81926d8ee9a05a6433a9b0f442bb7e8b0470b7a31e
8eaacf8527892abac6369a19d3df51812d6343740f32b8b157fdf1b2998d
```

3.5. Uporabite sha256sum in sha512sum za generiranje SHA-2-256 in SHA-2-512 zgoščene vrednosti datoteke letter_to_grandma.txt:

```
┌──(kali㉿kali)-[~]
└─$ sha256sum letter_to_grandma.txt
663a781df0ee5d88d74b7ec001f79c48b8a1744fd800668819c6a3bc1b1ac48f
letter_to_grandma.txt
```

```
┌──(kali㉿kali)-[~]
└─$ sha512sum letter_to_grandma.txt
c63ec8dcfaaa11470c5aace0df81926d8ee9a05a6433a9b0f442bb7e8b0470b7a31e
8eaacf8527892abac6369a19d3df51812d6343740f32b8b157fdf1b2998d
letter_to_grandma.txt
```

Vprašanje: Ali se zgoščena vrednost ustvarjena z sha256sum in sha512sum ujema z vrednostmi ustvarjenima v 3.3 in 3.4 točki? Obrazložite.

Opomba: SHA-2 je priporočeni standard za izračunavanje zgoščenih vrednosti. Čeprav SHA-2 še ni bil ogrožen, računalniki postajajo vse močnejši. Pričakuje se, da bo ta naravna evolucija kmalu omogočila napadalcem, da prelomijo SHA-2.

SHA-3 je najnovejši algoritem za izračunavanje zgoščenih vrednosti in bo sčasoma postal nadomestilo za družino zgoščevalnih algoritmov SHA-2.

4. Preverjanje prstnih odtisov

Kot je bilo že omenjeno, je pogosta uporaba prstnih odtisov preverjanje integritete datotek. Sledite spodnjim korakom za uporabo prstnih odtisov SHA-2-256 za preverjanje integritete datoteke sample.img, ki je bila prenesena z interneta.

4.1. Skupaj z datoteko sample.img je bil prenesen tudi sample.img_SHA256.sig. sample.img_SHA256.sig je datoteka, ki vsebuje prstni odtis SHA-2-256, ki ga je izračunala spletna stran. Najprej uporabite ukaz cat za prikaz vsebine datoteke sample.img_SHA256.sig:

```
┌──(kali㉿kali)-[~]
└─$ cat sample.img_SHA256.sig
fecad654dd24bbfebf9d1de25482e13d89733f35ede9678c8100c6dba477b410
sample.img
```

4.2. Uporabite SHA256sum za izračun prstnega odtisa SHA2-256 datoteke sample.img:

```
┌──(kali㉿kali)-[~]
└─$ sha256sum sample.img
fecad654dd24bbfebf9d1de25482e13d89733f35ede9678c8100c6dba477b410
sample.img
```

Vprašanje: Ali je bila datoteka sample.img prenesena brez napak? Razložite.

Opomba: Čeprav je primerjava prstnih odtisov relativno robustna metoda za zaznavanje napak pri prenosu, obstajajo boljši načini za zagotovitev, da datoteka ni bila spremenjena. Orodja, kot je gpg, zagotavljajo veliko boljšo metodo za zagotovitev, da prenesena datoteka ni bila spremenjena s strani tretjih oseb in je dejansko datoteka, ki jo je želel objaviti založnik.