Prelamovanie Hesiel

Miroslav Hájek

1. Prečo sa snažíme ochranovať počítače?

2. Ako zamedzíme prístupu nevyžiadaným osobám?

Prečo sa snažíme ochranovať počítače?

- Úložisko dôležitých a dôverných (tajných) informácií
- Významný komunikačný prostriedok
- Nástroj pre elektronický obchod
- Sprostredkovanie finančných prevodov

Ako zamedzíme prístupu nevyžiadaným osobám?

- **Autentifikácia** - potvrdenie identity
- **Autorizácia** - tabuľka prístupov
- **Utajenie** – symetrické a asymetrické šifry

„Bezpečnosť je hra na mačku a myš“

Hacking

Hacking

- pochopiť, ako daná vec funguje, hľadať v tom skratky a odhaľovať chyby

White Hat (Etický) hacker

- odhaľuje problémy zabezpečenia s cieľom zabrániť vniknutiu a zneužitiu dát

Black Hat (Škodlivý) hacking

- Trestný zákon §247 - Neoprávnený prístup do počítačového systému

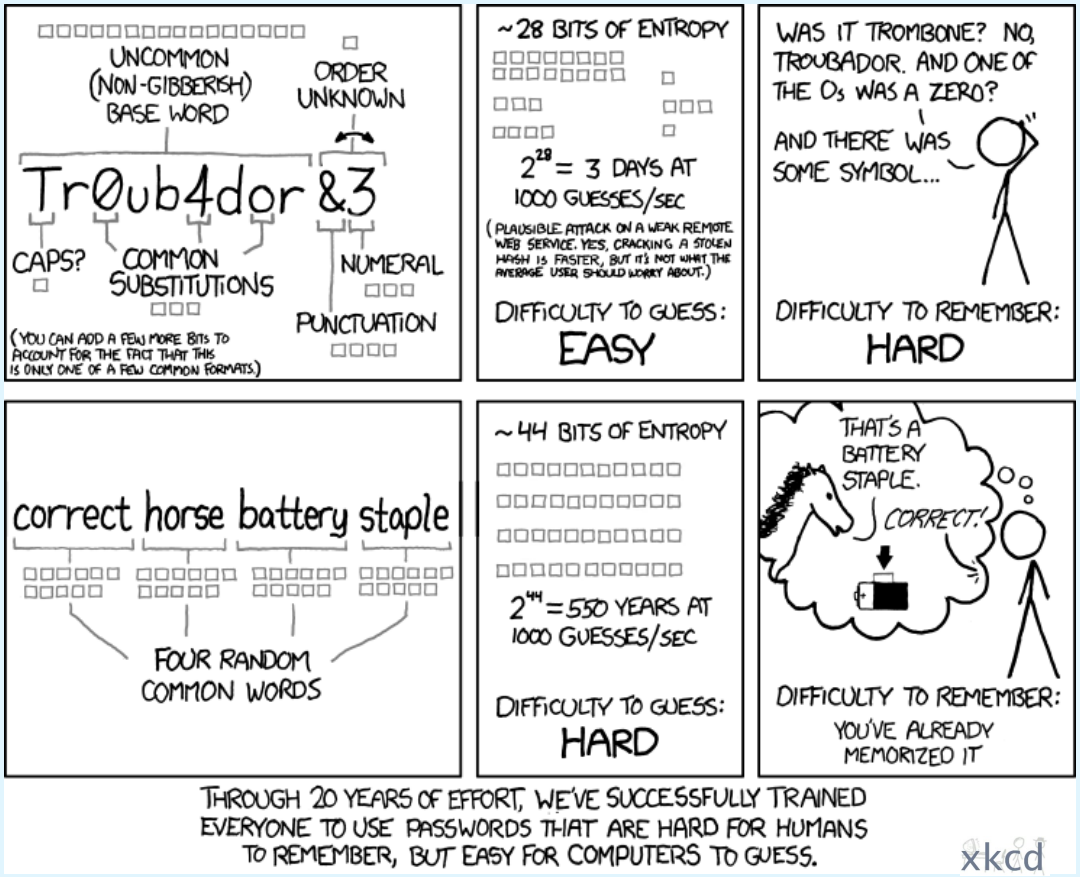
Ako si vybrať vhodné (neprelomiteľné) heslo?

Stratégie:

- Basic16
- Comprehensive8
- Prvé písmena slov

Dvojfaktorová autentifikácia - využívať dva prvky na prehlásenie o identite

- čo viem, mám, som, robím



Prvoplánová kombinatorika hesiel

Predpis hesla	Počet možností	Čas na polovicu hesiel
8 písmen anglickej abecedy	$26^8 = 2{,}1 \times 10^{11}$	14 hodín
8 písmen slovenskej abecedy	$46^8 = 2 \times 10^{13}$	58 dní
8 písmen a 2 čísla	$9C2 \times 26^8 \times 10^2 = 7 \times 10^{14}$	1,58 roka
8 písmen (veľké, malé), číslo, špeciálny znak, neslovníkové slovo	$26^6 \times 10 \times 60 \times 7C2 = 3{,}9 \times 10^{12}$	5,7 dní
16 písmen (basic16)	$26^{16} = 4 \times 10^{22}$	158 mil. rokov

Stolný počítač dokáže uskutočniť **milión až miliardu operácií (cyklov) za sekundu**

Overenie správnych prihlasovacích údajov

```
def JeSpravneHeslo(meno, heslo):
    platny = False
    ucty = open(„ucty.txt“, „r“)

    for pouzivatel in ucty:
        m, h = pouzivatel.split(„$“)

        if meno == m and heslo == h:
            platny = True
            break

    ucty.close()
    return platny
```

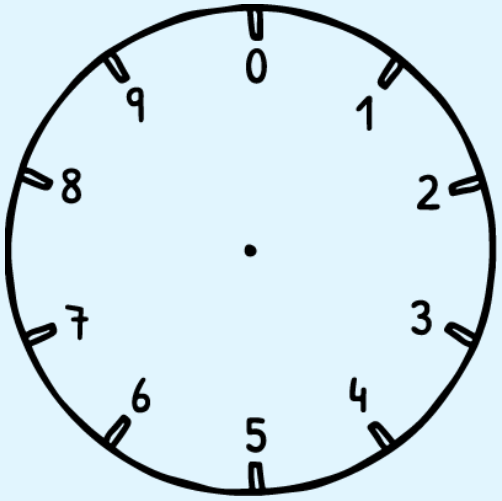
Ucty.txt

Miro**$**VelmiSilneHeslo
Jano**$**HrachNaCibuli
Ivan**$**Z4P5NaSkle

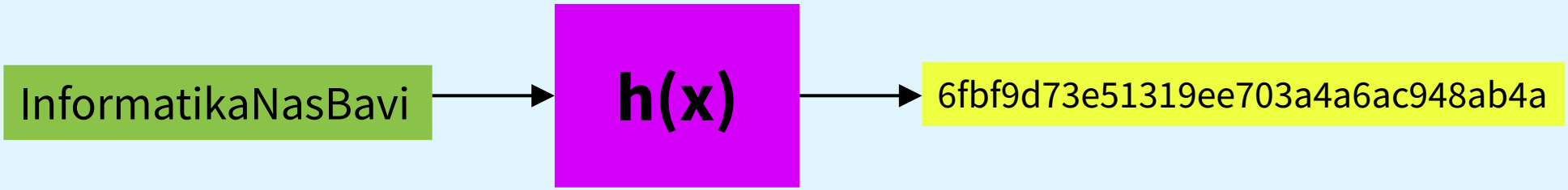
Aké bezpečnostné riziko tu vidíte?

Kryptografické hašovacie funkcie

- Otlačok obsahu správy
- **Fixná dĺžka výsledku**
  - počet bitov výsledku (hašu) je nemenný a nezáleží od dĺžky hesla
- **Jednosmerná** $\text{haš} = h(\text{heslo})$
  - z hašu nevieme zistiť spätne heslo
- **Bez kolízií** $h(\text{heslo}_1) \neq h(\text{heslo}_2)$
  - dve heslá nedajú rovnaký výsledok



Fixná dĺžka – zvyšky:
25 % 10 = 5
1581 % 10 = 1



napr. MD5, SHA

Ukladanie hesiel v zahašovanom tvare

```
from hashlib import *

def registracia(meno, heslo):
    ucty = open(„ucty.txt“, „a“)
    heslo = bytes(heslo, „utf-8“)

    fraza = sha256(heslo)
    fraza = fraza.hexdigest()
    print(f„{meno}${fraza}“, file=ucty)

    ucty.close()

registracia(„Miroslav“, „12345“)
```

Pri prístupe k súboru s heslami sú typy útokov

Hrubá sila (*Brute-Force attack*)

- vyskúšanie všetkých možností pre heslá danej dĺžky

Maska

- očakávaný tvar hesla znižujúci počet kombinácií na prehľadanie (napr. `?l?l?l?l?l?l?d?d`)

Slovníkový útok (*Dictionary attack*)

- Iba slová zo slovníka
- Pravidlá na manipuláciu s heslami

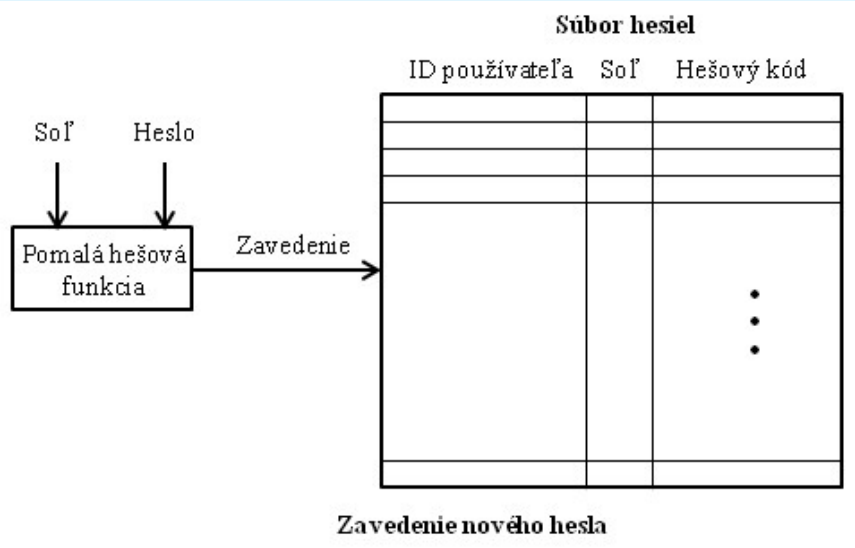
Hybridný útok – masky a slovník

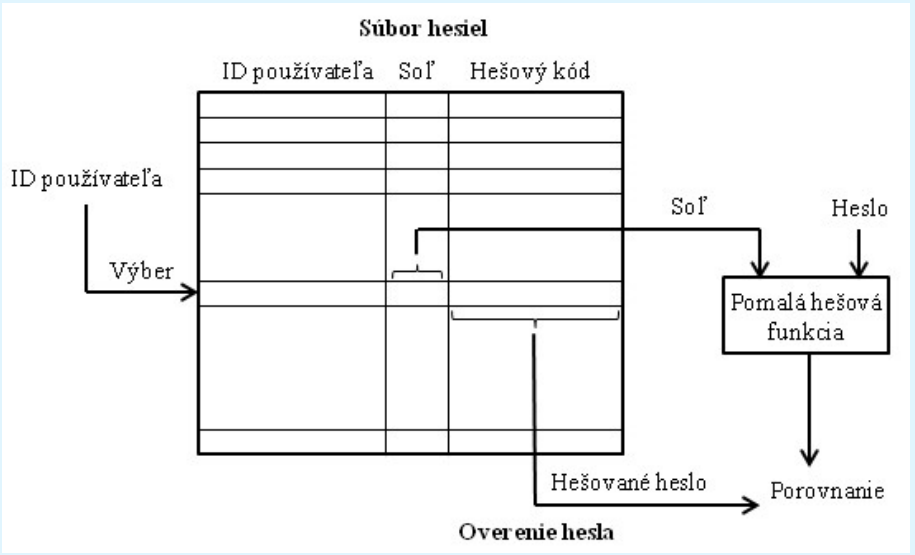
Asociačný útok – hádanie hesla podľa používateľa

Treba sa k heslám dostať! (cez napr. SQLi)

Solením hesla proti útokom hrubou silou

- Zavedenie náhodnej hodnoty do hešovania, ktorá sa ukladá spolu s heslom
- Zťažuje prelamovanie - rovnaké heslá majú rôzne haše





Funkcie na odvodenie kľúča: napr.: bcrypt, scrypt, PBKDF2, sha512crypt

```
>>> import hashlib
>>> dk = hashlib.pbkdf2_hmac('sha256', b'password', b'salt', 100000)
>>> dk.hex()
'0394a2ede332c9a13eb82e9b24631604c31df978b4e2f0fbd2c549944f9d79a5'
```

John The Ripper

```
john --test

john --single passwords.txt

john --incremental passwords.txt

john passwords.txt wordlist=“rockyou.txt”
```

HashCat

Vytvorenie zoznamu hesiel (alebo sha256sum):

```
echo -n "Password" | md5sum | tr -d " -" >> passwords.txt
```

```
for i in $(seq 1 100); do
    echo -n "$(openssl rand -base64 8)" | md5sum | tr -d " -" >> passwords.txt
done
```

hashcat

- --hash-type 0 / 1400
- --attack-mode 0
- --outfile cracked.txt
- passwords.txt
- rockyou.txt

```
 #  | Mode
====+======
 0  | Straight
 1  | Combination
 3  | Brute-force
 6  | Hybrid Wordlist + Mask
 7  | Hybrid Mask + Wordlist
 9  | Association
```

Attack-Mode	Hash-Type	Example command
Wordlist	P	hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules	MD5	hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force	MD5	hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
Combinator	MD5	hashcat -a 1 -m 0 example0.hash example.dict example.dict
Association	1	hashcat -a 9 -m 500 example500.hash 1word.dict -r rules/best64.rule

Čo vás zaujalo?

Ako si vybrať silné heslo?

V akom tvare ukladáme heslá a prečo?

Prečo keď zabudneme heslo nám ho nevedia povedať, ale musíme si vybrať nové?

Aké existujú spôsoby prelamovania hesiel?