Password Hash - zoeken in groot text bestand ***

Snel zoeken in password hash lijst. Je leert wat een zoek algoritme is en gaat een zoekalgoritme implementeren.

Wat ga je leren?

- met hash functies werken
- PHP en files lezen en doorzoeken
- een programmeer-opdracht opdelen in stapjes; een plan maken
- wat een zoeklagoritme is
- implementeren van een zoekalgortime

Inleiding

Er is een zeer lange lijst met password hashes. Je kan daarin zoeken of jouw password is gehacked. Dit kan je online doen, maar je wilt je super-goede-geheime password waarschijnlijk niet zomaar naar een of andere site sturen. We gaan dus onze eigen lokale password validation maken. En de uitdaging is wie de snelste code kan maken. Hoe snel kan jij een hash vinden in een half miljard password hashes. Ik kan je vertellen dat dat op een i5 laptop met SSD onder de paar seconden moet kunnen.

Have I been powned?

Kijk op https://haveibeenpwned.com/Passwords je kunt daar je password invullen en controleren of het een 'bekend' (dus gehacked) wachtwoord is. Deze wachtwoorden zijn onder hackers bekend en zijn dus eigenlijk niet (meer) veilig. Maar als je de controle wilt uitvoeren zonder jouw wachtwoord over het internet te sturen dan zal je zelf iets moeten maken. En ja, de site heeft een secure connectie via SSL, maar wat als de programmeur stiekem een lijstje bijhoud van alle wachtwoorden die worden geprobeerd?

Case

We gaan dus onze eigen versie maken van 'Have I been powned'. Daarvoor moet je het grote bestand met hashes downloaden. Deze staat op de genoemde site. Kies de versie in het format SHA-1 ordered by hash.

De file ziet er als volgt uit:

```
000000005AD76BD555C1D6D771DE417A4B87E4B4:4
00000000A8DAE4228F821FB418F59826079BF368:3
00000000DD7F2A1C68A35673713783CA390C9E93:630
00000001E225B908BAC31C56DB04D892E47536E0:5
00000006BAB7FC3113AA73DE3589630FC08218E7:2
00000008CD1806EB7B9B46A8F87690B2AC16F617:4
0000000A0E3B9F25FF41DE4B5AC238C2D545C7A8:15
0000000A1D4B746FAA3FD526FF6D5BC8052FDB38:16
0000000CAEF405439D57847A8657218C618160B2:15
0000000FC1C08E6454BED24F463EA2129E254D43:40
```

Elke regel bestaat uit een password hash, een dubbele punt en een getal. Dit laatste getal geeft aan bij hoeveel hacks het password is gevonden. Het enige dat je moet is dus een password hash zoeken.

Hints

Onder Linux kun je met het wc (word count) commando vrij snel bepalen hoeveel regels het bestand heeft. Je zult er snel achter komen dat gewoon de file doorlopen en regel voor regel kijken of je de juiste hash hebt gevonden niet werkt. Dit ligt natuurlijk wel aan de snelheid van jouw computer/laptop. Probeer maar eens in te schatten hoelang het duurt voordat je bijvoorbeeld 10% van de file heb doorzocht. Ik heb een algoritme genaakt dat op elke eenvoudige laptop de hash binnen 1 seconden vind.

Als je gaat nadenken over een strategie bedenk dan hoe jij het handmatig zou aanpakken?

Deel nu het probleem op in stapjes en test elke stapje. Wat zijn je stapjes en hoe wil je het aanpakken.

Maak het probleem eenvoudiger.

Als je er nog niet uitkomt, maak het probleem dan eenvoudiger. Bijvoorbeeld je hebt 20 hashes:

```
1200
1201
2011
2045
```

```
2234
3400
4000
4001
4010
4098
4099
5332
8020
8100
8201
8245
8376
8898
8999
9010
```

Stel je zoekt de hash van het een wachtwoord en de hash is 9500. Ga jij nu 20x maal kijken of de hash gelijk is of zie je dit sneller? En stel je zoekt naar de hash 8020, ga je dan alle hashes vergelijken of die je dit op een snellere manier. Probeer een plan te maken hoe je zelf in zo min mogelijk stappen bepaald of een hash in de lijst staat. Deze strategie kun je dan mogelijk ook toepassen op het grote bestand.

Maar een plan en bespreek dat eventueel met je docent voordat je begint. Een goed plan is het halve werk!

--

Revision #4
Created 2020-04-17 12:34:54 UTC by Max
Updated 2020-06-20 09:32:32 UTC by Max