IF3051 Strategi Algoritma

Penerapan Algoritma Bruteforce pada

Cracking Password Windows (Bruteforce Attack)

Setia Negara B. Tjaru[1]
13508054
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]if18054@students.if.itb.ac.id

Bruteforce adalah algoritma yang menggunakan pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah. Karena sangat lempang maka algoritma ini sering dipakai, apalagi pada cracking password.

Bruteforce dipakai dengan mencoba seluruh kemungkinan password. Biasanya dengan sdikit keberuntungan proses crack password ini tidak berlangsung lama karena password yang lemah dan terdiri dari hanya beberapa karakter saja.

Kata kunci :Bruteforce,bruteforce attack, password, crack, windows.

I. PENDAHULUAN

Bruteforce

Bruteforce adalah algoritma yang menggunakan pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah. Karena sangat lempang maka algoritma ini sangat gampang dimengerti. Karena gampangnya dimengertilah yang menyebabkan algoritma ini cukup populer.

Algoritma Bruteforce memecahkan masalah dengan sabgat sederhana, langsung dan jelas (obvious way).

Namun dibalik kelebihannya tersebut terdapat beberapa kekurangan karena menggunakan algoritma yang lempang makanya seringkali tidak "cerdas" dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesainnya. Kata "force" mengindikasikan "tenaga" ketimbang "otak". Kadang-kadang algoritma ini disebut juga algoritma naïf (naïve algorithm).

Algoritma *brute force* lebih cocok untuk masalah yang berukuran kecil. Dengan pertimbangan kesederhanaan dan implementasinya yang mudah. Algoritma *brute force* sering digunakan sebagai basis pembanding dengan algoritma yang lebih mangkus.

Meskipun bukan metode yang mangkus, hampir semua masalah dapat diselesaikan dengan algoritma *brute force.* Sukar menunjukkan masalah yang tidak dapat diselesaikan dengan metode *brute force*. Bahkan, ada masalah yang hanya dapat diselesaikan dengan metode *brute force*.

Contoh: mencari elemen terbesar di dalam senarai.

Kelebihan dan Kelemahan Metode Bruteforce

Kelebihan:

1. Metode *brute force* dapat digunakan untuk memecahkan hampir sebagian besar masalah (*wide applicability*).
2. Metode *brute force* sederhana dan mudah dimengerti.
3. Metode *brute force* menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan *string*, perkalian matriks.
4. Metode *brute force* menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian *n* buah bilangan, menentukan elemen minimum atau maksimum di dalam tabel (*list*).

Kekurangan :

1. Metode *brute force* jarang menghasilkan algoritma yang mangkus.
2. Beberapa algoritma *brute force* lambat sehingga tidak dapat diterima.
3. Tidak sekontruktif/sekreatif teknik pemecahan masalah lainnya.

Ken Thompson (salah seorang penemu Unix) mengatakan: "*When in doubt, use brute force*", faktanya kernel Unix yang asli lebih menyukai algoritma yang sederhana dan kuat (*robust*) daripada algoritma yang cerdas tapi rapuh.

Cracking Password

Cracking Password adalah proses mendapatkan password dari data yang disimpan ataupun dikirimkan oleh suatu system computer. Pendekatan yang umum adalah dengan cara secara berulang-ulang mencoba menebak apa sebenarya password tersebut. Penggunaan dari cracking password salah satunya adalah membantu user yang lupa atau kehilangan passwordnya untuk recovery.

Namun cracking password sering disalahgunakan oleh orang-orang tertentu untuk mendapatkan akses ke sebuah system.

Mencoba berulang kali menebak password dari sebuah system dapat dikategorikan sebagai sebuah bruteforce attack yang tentu saja menggunakan algoritma brute force dalam implementasinya.

Bruteforce attack pada cracking password ini juga sering dipakai oleh analis system untuk menguji tingkat keamanan dari sebuah system, seberapa rentan system tersebut dari serangan bruteforce attack (serangan yang paling populer).

II. BRUTEFORCE ATTACK PADA CRACKING PASSWORD

Bruteforce attack sebenarnya tidak langsung dilakukan ketika mencoba meretas sebuah password terdapat beberapa cara yang umum untuk meretas password yaitu:

1. Menebak,

   sejalan dengan bruteforce, namun kandidat password yang dicobakan biasanya hanya ditebak dengan merujuk pada sesuatu yang dekat dengan yang 'mpunya' password. biasanya berupa kombinasi tanggal lahir, nama anak, istri, pacar, tempat lahir, nama ayah, merek mobil idaman, dll.

   Pada jaringan biasa dipakai password seperti 'password', 'admin', 'test', 'superuser', 'imgod' , dll.

2. Dictionary Attack

   User seringkali memilih password yang lemah , apalagi ketika internet baru pertama kali muncul sehingga pengetahuan tentang keamanan system sangatlah rendah. Contoh yang sering dipakai adalah kata-kata pada contoh Menebak diatas dan juga kata-kata yang literal terdapat pada kamus.

   Riset yang berulang dan mendalam tentang password bahwa 40% password sebenarnya sudah dapat di-crack dengan program yang sophisticated, kamus, dan informasi personal dari user.

   Dalam satu survey dari MySpace password yang didapatkan dari phising, 3,8% dari password tersebut adalah satu kata yang dapat ditemukan di dalam kamus, dan 12 persen adalah satu kata juga yang terdapat dalam kamus namu ditambahkan angka pada akhir katanya, kemudian 2 per tiga dari angka akhir tersebut adalah 1.

3. Bruteforce Attack (serangan brute-force),

   Bruteforce attack lah yang menjadi pilihan jika kedua hal diatas gagal, meskipun secara harfiah dua cara diatas sebenarnya adalah brute force juga namun yang menerapkan algortima bruteforce adalah metode yang terakhir ini.

Serangan *brute-force* adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin. Pendekatan ini pada awalnya merujuk pada sebuah program komputer yang mengandalkan kekuatan pemrosesan komputer dibandingkan kecerdasan manusia. Sebagai contoh, untuk menyelesaikan sebuah persamaan kuadrat seperti $x^2+7x-44=0$, di mana *x* adalah sebuah integer, dengan menggunakan teknik *serangan brute-force*, penggunanya hanya dituntut untuk membuat program yang mencoba semua nilai integer yang mungkin untuk persamaan tersebut hingga nilai *x* sebagai jawabannya muncul.

Teknik yang paling banyak digunakan untuk memecahkan *password*, kunci, kode atau kombinasi. Cara kerja metode ini sangat sederhana yaitu mencoba semua kombinasi yang mungkin.

Sebuah *password* dapat dibongkar dengan menggunakan program yang disebut sebagai *password cracker*. Program *password cracker* adalah program yang mencoba membuka sebuah password yang telah terenkripsi dengan menggunakan sebuah algoritma tertentu dengan cara mencoba semua kemungkinan. Teknik ini sangatlah sederhana, tapi efektivitasnya luar biasa, dan tidak ada satu pun sistem yang aman dari serangan ini, meski teknik ini memakan waktu yang sangat lama, khususnya untuk password yang rumit.

Namun ini tidak berarti bahwa *password cracker* membutuhkan decrypt. Pada prakteknya, mereka kebayakan tidak melakukan itu. Umumnya, kita tidak dapat melakukan decrypt password-password yang sudah terenkripsi dengan algoritma yang kuat. Proses-proses enkripsi modern kebanyakan hanya memberikan satu jalan, di mana tidak ada proses pengembalian enkripsi. Namun, anda menggunakan tool-tool simulasi yang

mempekerjakan algoritma yang sama yang digunakan untuk mengenkripsi password orisinal. Tool-tool tersebut membentuk analisa komparatif. Program password cracker tidak lain adalah mesin-mesin ulet. Ia akan mencoba kata demi kata dalam kecepatan tinggi. Mereka menganut "Azaz Keberuntungan", dengan harapan bahwa pada kesempatan tertentu mereka akan menemukan kata atau kalimat yang cocok. Teori ini mungkin tepat mengena pada anda yang terbiasa membuat password asal-asalan. Dan memang pada kenyataannya, password-password yang baik sulit untuk ditembus oleh program *password cracker*.

Jika sebuah system memiliki desain yang buruk untuk melindungi password yang telah tersimpan, cracking password dapat dilakukan bahkan untuk password yang kuat dan bagus ('well- chosen').

Satu contoh nyata adalah LM hash yang dipakai oleh Microsoft Windows XP pakai, begitu juga pada versi-versi sebelumnya. Kelemahannya adalah password yang dipakai panjangnya tidak dapat melebihi 15 karakter. Kemudian LM hash mengkonversi semua karakter password tersebut menjadi uppercase kemudian memecah password tersebut menjadi dua bagian yang masing-masing berisi tujuh hingga delapan karakter. Hal ini menyebabkan tiap bagian password dapat diserang dengan brute force secara terpisah.

Berikut adalah ilustrasi betapa rentannya system ini dari serangan bruteforce.

Asumsi:

1. Password adalah 14 karakter.
2. Hanya menggunakan karakter A-Z dan a-z.
3. Tiap serangan satu kemungkinan bruteforce membutuhkan waktu 1 nanosekon.
4. Serangan dimulai dari alphabet A hingga Z baru kemudian a sampai z.
5. Diketahui bahwa password 14 karakter.

Misalkan saja password yang dipakai ialah "TTTTTTTTTTTTTTT"

Jika tanpa LM hash maka waktu yang dibutuhkan untuk cracking password tersebut adalah:

$(26+20)^{14}$ x 0.000001 sekon = 2 x 10^{17} sekon.

Waktu tersebut sama dengan 6 milyar tahun.

Sementara dengan diterapkannya LM hash pada Windows XP maka waktu yang dibutuhkan adalah:

20^7 x 2 x 0.000001 sekon = 2560 sekon.

Atau sekitar 43 menit.

Bagaimana hal tersebut dapat berbeda jauh? Hal tersebut disebabkan karena waktu yang dibutuhkan algoritma pada cracking password ini ditentukan oleh beberapa factor, yaitu:

1. Jumlah karakter yang dicobakan.
2. Jumlah karakter pada password yang ingin di-crack (retas).

II. SIMULASI CRACKING PASSWORD

Saya mencoba membuat program yang mengerjakan ilustrasi tersebut dengan Asumsi yang mirip, namun karena sepertinya memakan waktu lama maka saya membuat passwordnya hanya 8 karakter. Kemudian membandingkannya dengan LM hash yang berarti semua karakter menjadi uppercase dan pemeriksaanya dua kali yaitu 4 karakter awal dan 4 karakter akhir terpisah.

PseuduCode

Deklarasi global:

```
Karakter : array of{'A','B','C','D','E','F','G','H','I','J','
K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y',
'Z',' a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
'q','r','s','t','u','v','w','x','y','z'}

length: integer = 52
string t
```

Prosedur utama:

```
Algoritma
output ("Masukkan string password : ")
input(t)

MaxChar : integer = 8
for i: integer = 0 to MaxChar+1 do
        output("Bruteforce untuk ", i, "  karakter")
        Bruteforce(i,0,"")
```

Prosedur Rekursif Bruteforce:

```
procedure Bruteforce (input width : integer, position :
integer; input/output StringDasar string)
{Prosedur rekursif yang menerima StringDasar, width
sebagai panjang maksimal karakter, dan position sebgai
karakter keberapa yang sedang dicobakan}

Algoritma
for i:integer = 0 to length+1 do
        if position < width -1 then
                Bruteforce(width, position+1,
StringDasar+Karakter[i])
        output(StringDasar+Karakter[i])
        CekPass(StringDasar+Karakter[i])
```

Prosedur Cek Password:

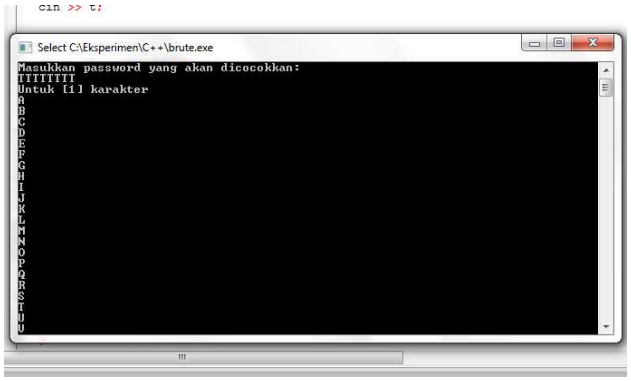
```
procedure CekPass(input password : string)
{mengecek apakah string yang ada sekarang sama dengan password}

Algoritma
if password = t then
        output("password ketemu : ",password)
        exit;
```

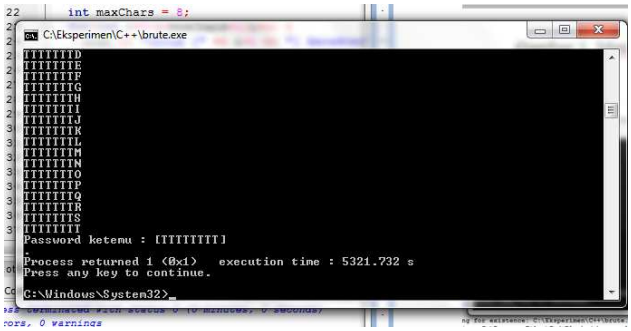
Prosedur Timer:

```
Procedure timer()
{menghitung waktu dari awal main sampai password ditemukan}
```

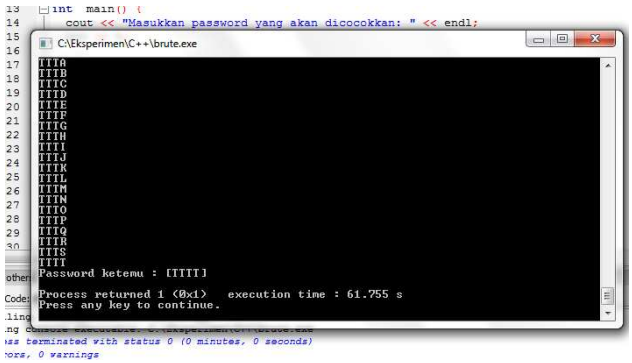
Screenshot running program:



Gambar 1. Meminta masukan password



Gambar 2. Password biasa ditemukan



Gambar 3. Password LM hash ditemukan

Dapat dilihat pada screenshot butuh waktu 5321 detik untuk memecahkan password yang ditulis biasa sementara untuk yang menyerupai LM hash hanyabuth waktu 124 detik (62 x 2).
Berarti Pada LM hash password menjadi lemah 42 kali lipat pada password yang mempunyai lebar 8 karakter.

Saya sebenarnya mencoba melakukan cracking password pada windows saya sendiri. Pada awalnya dengan software yang sudah terkenal untuk membobol password UNIX dan DOS NT yaitu JohntheRipper baru kemudian mencoba untuk membuat program sendiri. Namun OS yang saya pakai adalah Windows Seven, karena ketatnya pengamanan user acoount pada OS ini sehingga saya tidak berhasil melakukannya. Sehingga saya hanya membuat ilustrasi bagaimana program ini bekerja dan berapa perkiraan waktu yang dibutuhkan untuk meretas password pada OS terdahulu jika metode hashing digunakan.

II. KESIMPULAN

1. Algoritma Bruteforce dapat dipakai untuk Bruteforce Attack untuk memecahkan password windows.
2. Algoritma Bruteforce termasuk algoritma yang solid dan robust, algoritma ini hampir dapat memecahkan semua masalah di dunia informatika, bahkan ada beberapa masalah yang hanya dapat dipecahkan dengan algortima bruteforce
3. Bruteforce Attack sangat bergantung pada dua hal berkaitan dengan performansinya, yaitu jumlah panjang password yang akan diretas dan jumlah himpunan karakter yang akan dicobakan pada password tersebut.
4. Windows NT versi XP dan versi-versi sebelumnya dapat dengan mudah diretas dengan algoritma bruteforce karena adanya hash yang membagi dua lebar password, yang berarti memperpendek jumlah karakter yang harus di bruteforce. Dan fasilitas meng-uppercase-kan password, yang berarti jumlah karaktr yang

perlu di coba berkurang menjadi setengah jika password hanya berisi karakter alfabetis.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

Setia Negara
13508054

Referensi

http://en.wikipedia.org/wiki/Brute_force_attack
http://en.wikipedia.org/wiki/Password_cracking
http://virologi.info/virologist/modules/news/index.php?storytopic=4
http://www.openwall.com/john/
http://osix.net/modules/article/?id=455