Password Cracking menggunakan Brute Force Attack

IF2211 Strategi Algoritma

Ichwan Haryo Sembodo 13512008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512008@stei.itb.ac.id

***Abstract*—Seperti yang sudah diketahui, *Brute Force* adalah salah satu cara (algoritma) untuk melakukan sesuatu dengan 'paksa', mengapa dengan 'paksa'? karena algoritma ini menggunakan semua kemungkinan yang ada dan mengandalkan faktor keberuntungan (*Lucky*) untuk menemukan pemecahan masalahnya. Dengan algoritma ini tidak terelakan bahwa *password* atau kombinasi karakter pun juga dapat dibobol / diretas oleh algoritma ini. Kasus yang bersangkutan tentang peretasan / pembobolan suatu *password* atau suatu masalah tentang kombinasi karakter dinamakan *Brute Force Attack*. *Brute Force Attack* sebenarnya hanya istilah saja, diistilahkan seperti itu karena tujuan dari penggunaan *Brute Force* yang tidak menguntungkan pihak yang diretas (*Attack*).**

***Index Terms*—Brute Force, Brute Force Attack, Algoritma Brute Force, Password Cracking.**

I. PENDAHULUAN

A. DEFINISI BRUTE FORCE

Algoritma *Brute Force* adalah algoritma yang menggunakan cara sangat sederhana untuk menemukan solusinya, yaitu dengan cara mencoba semua kemungkinan yang ada. Dengan kata lain semakin banyak kemungkinannya maka semakin lama proses pencarian solusinya.

Algoritma *Brute Force* adalah algoritma yang dapat menemukan semua solusi karena algoritma ini mencoba semua kemungkinan solusi yang ada dan mengandalkan faktor keberuntungan (*Lucky*) untuk menemukan solusinya, namun jika faktor keberuntungan (*Lucky*) tersebut tidak didapat maka algoritma ini adalah *worst-algorithm* (algoritma yang terlalu lama memakan waktu untuk memecahkan suatu masalah).

B. DEFINISI PASSWORD CRACKING

Password Cracking atau bisa disebut peretasan *password* atau pembobolan *password* adalah suatu proses yang bertujuan untuk membobol suatu *password* dengan cara mencoba semua kemungkinan atau pun dengan menggunakan referensi yang lain, misal kamus data yang menjurus pada *password* yang ingin ditemukan.

Kasus *Password Cracking* ini 'biasanya' digunakan dengan tujuan yang tidak baik (tidak menguntungkan suatu pihak), misalkan kasus pembobolan *password* Facebook, pembobolan *password* e-mail.

Pada jaman sekarang *password* yang disimpan pada suatu database biasanya sudah di-enkripsi sedemikian rupa sehingga tidak dapat dibaca dari luar, dan untuk membaca database *password-password* tersebut hanya

dapat dilakukan satu arah yaitu oleh pemegang kunci dari enkripsi database *password* tersebut. Dengan demikian melakukan decrypt pada database tersebut mungkin dilakukan namun hanya akan memperpanjang waktu yang dibutuhkan untuk menemukan password yang ada. Sehingga konsep *Password Cracking* bukanlah melakukan decrypt pada database *password* namun mencoba semua kemungkinan *password* yang ada tanpa memperdulikan database *password* tersebut.

C. DEFINISI BRUTE FORCE ATTACK

Brute Force Attack adalah peretas *password* atau kriptografi yang tidak melakukan *decrypt* informasi untuk mendapatkan *password* atau kriptografi yang dibutuhkan. *Brute Force Attack* seperti yang dikatakan diatas yaitu hanya istilah untuk *Brute Force* yang digunakan untuk meretas (tidak menguntungkan suatu pihak). Prinsip kerjanya juga sama seperti *Brute Force* yaitu dengan mencoba semua kemungkinan yang ada, cepat dan lamanya suatu proses berdasarkan seberapa panjang *password* atau kriptografi yang akan diretas. Jadi jika *password* atau kriptografi itu pendek maka proses *Brute Force* untuk menemukan solusinya tidak akan memakan waktu yang lama.

Seperti yang dikatakan diatas bahwa *Brute Force Attack* tidak melakukan *decrypt* informasi . Hal inilah yang menjadikan *Brute Force* menjadi senjata yang disegani. Semua jenis enkripsi data dapat ditangani oleh *Brute Force* karena algoritma ini menggunakan semua kemungkinan yang ada untuk menemukan solusinya bukan melakukan *decrypt* pada informasi yang telah disediakan database yang akan diretas. Dengan adanya *Brute Force Attack* sebenarnya data-data yang ada tidak ada yang aman, semua dapat diretas. Namun dalam melakukan peretasan ini *Brute Force* membutuhkan waktu yang lama, 1 menit, 1 jam, 1 bulan bahkan 1 tahun cepat dan lambatnya *Brute Force* sangat dipengaruhi oleh kompleksitas suatu password itu sendiri.

Ada beberapa cara untuk mempercepat proses *Brute Force* ini salah satunya dengan menggunakan *Dictionary* (kamus). Yang dimaksud dari kamus disini adalah pencarian tidak dilakukan untuk semua kemungkinan namun pencarian dibatasi pada kamus yang ada. Misal mencari semua *password* atau kriptografi yang mirip dengan kamus ‘KBBI’ yang menjadi masalah adalah apakah benar-benar solusi dari *Brute Force* itu ada pada kamus.

Walaupun sudah terdapat cara untuk mempercepat proses *Brute Force* namun *Brute Force* masih memiliki masalah pada huruf besar dan huruf kecil. Jika huruf besar dan huruf kecil dikombinasikan maka dengan bantuan kamus pun proses *Brute Force* akan mengalami kesulitan untuk menemukan solusi yang tepat. Maka dari itu ada beberapa cara agar *password* atau kriptografi aman dari algoritma *Brute Force* yaitu dengan mengkombinasikan huruf besar dan huruf kecil, lalu angka dan simbol. Seperti yang sudah dijelaskan jika cepat lambatnya proses *Brute Force* bergantung pada panjang atau pendeknya suatu *password* atau kriptografi sehingga lebih panjang lebih baik. Pencegahan peretasan untuk akun online biasanya dapat dicegah dengan suatu sistem yang otomatis akan melakukan *Lock* pada user yang salah melakukan *password* sebanyak 3x.

II. BRUTE FORCE ATTACK

A. METODE BRUTE FORCE ATTACK

Brute Force Attack adalah metode yang digunakan untuk meretas *password* atau kriptografi dengan algoritma *Brute Force*. Kecepatan peretasan ini bergantung pada panjang pendeknya *password* atau kriptografi yang ingin dipecahkan.

Cara untuk menghitung berapa jumlah kemungkinan yang ada dari panjang suatu *password* atau kriptografi dapat dihitung dengan

$N = L^{(Min)} + L^{(Min+1)} + L^{(Min+2)} + \ldots + L^{(Max)}$

N = jumlah kemungkinan yang ada

L = jumlah karakter yang ada

Min = panjang minimum dari *password* atau kriptografi

Max = panjang maximum dari *password* atau kriptografi

Misalkan kita ingin mengetahui berapa jumlah kemungkinan yang ada jika suatu *password* terdiri dari 7 karakter dan semua karakter *password* hanya terdiri dari *alphabet* yaitu 'abcdefghijklmnopqrstuvwxyz' yang berjumlah 26. Maka perhitungannya adalah

$N = 26^1 + 26^2 + 26^3 + \ldots + 26^{27} = 8353082582$

Jadi ada 8353082582 kemungkinan yang harus di coba untuk menemukan *password* yang diinginkan. Jumlah tersebut adalah *WORST-CASE* jikalau ternyata kombinasi password yang ada ternyata berada pada paling akhir yaitu 'zzzzzzz'. Karena algoritma *Brute Force* juga bergantung pada faktor keberuntungan (*Lucky*) maka *BEST-CASE* nya adalah algoritma ini hanya perlu melakukan 1 kali percobaan jika ternyata kombinasi *password* nya adalah 'aaaaaaa'.

B. ALGORITMA BRUTE FORCE ATTACK

Dalam melakukan pencarian *password* yang ada *Brute Force Attack* menggunakan algoritma yang sangat sederhana, yaitu dengan mencoba semua kombinasi yang karakter yang ada dengan *pseucode* berikut

```
Array of String KombinasiKarakter
String Kombinasi
Do
{
        Kombinasi = KombinasiMaker()
        If (Kombinasi == password)
        {
                Selesai
        }
        Else
        {
                KombinasiKarakter.add(Kombinasi)
        }
}while(tidak ada kombinasi lagi)
```

Penjelasan untuk *pseucode* diatas adalah sebagai berikut. Array of String yang dinamakan KombinasiKarakter adalah array yang digunakan untuk menyimpan kombinasi yang telah dicoba agar tidak terjadi pengecekan yang bertumpuk, maksutnya adalah agar kombinasi yang sudah di cek tidak di cek lagi. *Method* KombinasiMaker adalah suatu *method* yang mengembalikan string kombinasi yang belum pernah diakses. String Kombinasi adalah string yang digunakan untuk menyimpan sementara kombinasi string yang ada dan mencobanya, jika cocok maka selesai jika tidak kombinasi tersebut dimasukan kedalam KombinasKarakter agar tidak dicek lagi.

C. APROKSIMASI WAKTU

Aproksimasi waktu algoritma *Brute Force Attack* atau bisa disebut *Brute Force* sangat bergantung pada komputer yang digunakan, yang dimaksud disini adalah spesifikasi yang dimiliki oleh komputer tersebut. Misal menggunakan komputer dengan spesifikasi rendah untuk meretas *password* maka akan memakan waktu yang lama.

Kemampuan suatu komputer untuk mencoba beberapa *password* dalam 1 detik telah diklasifikasikan sebagai berikut

1. Kelas komputer A yang dapat memcoba 10.000 *password*/detik

2. Kelas komputer A yang dapat memcoba 100.000 *password*/detik

3. Kelas komputer A yang dapat memcoba 1.000.000 *password*/detik

4. Kelas komputer A yang dapat memcoba 10.000.000 *password*/detik

5. Kelas komputer A yang dapat memcoba 100.000.000 *password*/detik

6. Kelas komputer A yang dapat memcoba 1.000.000.000 *password*/detik

Contoh perbandingan aproksimasi waktu antar kelas komputer yang telah diklasifikasikan untuk beberapa kasus:

1. Kasus 10 karakter yang terdiri dari angka

Numerals	0123456789						
Password		Class of Attack					
Length	Combinations	Class A	Class B	Class C	Class D	Class E	Class
2	100	Instant	Instant	Instant	Instant	Instant	Instant
3	1000	Instant	Instant	Instant	Instant	Instant	Instant
4	10,000	Instant	Instant	Instant	Instant	Instant	Instant
5	100,000	10 Secs	Instant	Instant	Instant	Instant	Instant
6	1 Million	1½ Mins	10 Seconds	Instant	Instant	Instant	Instant
7	10 Million	17 Mins	1½ Mins	1½ Mins	Instant	Instant	Instant
8	100 Million	2¾ Hours	17 Mins	1½ Mins	10 Seconds	Instant	Instant
9	1000 Million	28 Hours	2¾ Hours	17 Mins	1½ Mins	10 Seconds	Instant

2. Kasus 36 karakter yang terdiri dari huruf besar dan huruf kecil dan angka

Upper Case Alpha	ABCDEFGHIJKLMNOPQRSTUVWXYZ						
Lower Case Alpha	abcdefghijklmnopqrstuvwxyz						
Numerals	0123456789						
Password		Class of Attack					
Length	Combinations	Class A	Class B	Class C	Class D	Class E	Class
2	1,295	Instant	Instant	Instant	Instant	Instant	Instant
3	46,656	4 Secs	Instant	Instant	Instant	Instant	Instant
4	1.6 million	2½ Mins	16 Seconds	1½ Seconds	Instant	Instant	Instant
5	60.4 million	1½ Hours	10 Mins	1 Min	Instant	Instant	Instant

3. Kasus 52 karakter dengan huruf besar dan huruf kecil

Mixed Alpha	AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz						
Password		Class of Attack					
Length	Combinations	Class A	Class B	Class C	Class D	Class E	Class
2	2,704	Instant	Instant	Instant	Instant	Instant	Instant
3	140,608	14 Secs	< 2 Secs	Instant	Instant	Instant	Instant
4	7.3 Million	12½ Mins	1¼ Mins	8 Secs	Instant	Instant	Instant
5	380 Million	10½ Hours	1 Hour	6 Minutes	38 Secs	4 Secs	Instant
6	19 Billion	23 Days	2¼ Days	5½ Hours	33 Mins	3¼ Mins	19 Secs
7	1 Trillion	3¼ Years	119 Days	12 Days	28½ Hours	3 Hours	17 Mins
8	69 Trillion	169½ Years	17 Years	1½ Years	62 Days	6 Days	15 Hour
9	2.7 Quadrillion	8,815 Years	881 Years	88 Years	9 Years	322 Days	32 Days

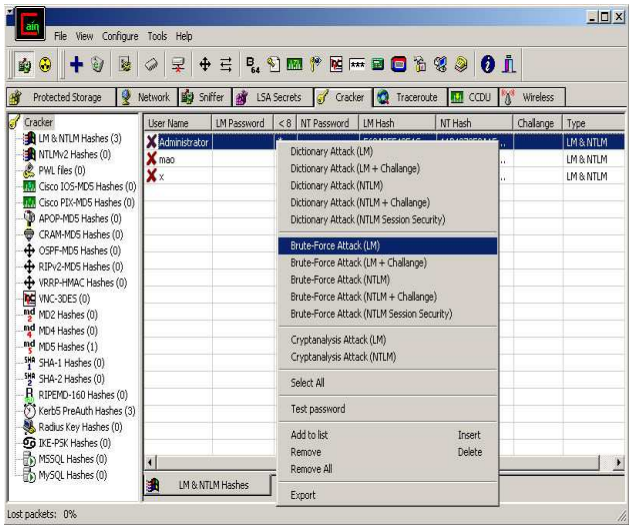
Seperti yang terdapat pada tabel contoh diatas, dapat dikatakan bahwa semakin hebat suatu komputer maka waktu yang dibutuhkan untuk meretas suatu password semakin singkat..

III. CONTOH PROGRAM BRUTE FORCE ATTACK

A. Figures and Tables

A. Cain and Abel

Cain and Abel adalah alat *recovery password* untuk sistem operasi Microsoft. Program ini memungkinkan melakukan *recovery* berbagai jenis *password* dengan mengendus jaringan, *cracking password* terenkripsi menggunakan *Dictionary-attack*, *Brute Force Attack* dan serangan *kriptoanalysis*, merekam percakapan VoIP, decoding *password* yang teracak, memulihkan kunci jaringan *wireless*, mengungkap *password cache* dan menganilisis *routing protocol*. Program ini tidak memanfaatkan kerentanan perangkat lunak atau bug yang tidak diperbaiki. Mencakup beberapa aspek keamanan/kelemahan yang ada dalam protocol standar, metode otentikasi dan mekanisme *caching*. Tujuan utama program ini adalah melakukan pemulihan (*recovery*) *password*.

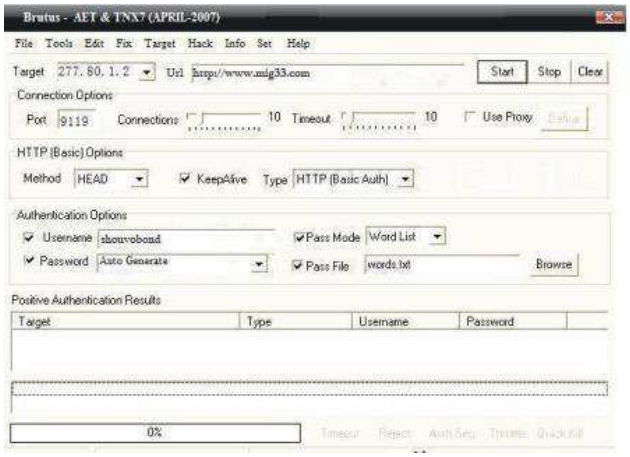


Contoh screenshot program cain and abel

Cain and Abel telah dikembangkan dengan harapan akan berguna bagi administrator jaringan, guru, konsultan keamanan/professional, staff forensic, vendor keamanan perangkat lunak, tester penetrasi professional dan semua orang yang berencana untuk menggunakannya untuk

alasan yang jelas dan etis. Pihak pembuat program ini tidak akan membantu atau mendukung setiap aktivitas illegal yang dilakukan dengan program ini. Perlu diperhatikan bahwa ada kemungkinan pemakaian software ini bisa menyebabkan kerusakan/kehilangan data dan pihak pembuat program ini tidak bertanggung jawab atas hal tersebut.

Versi Cain and Abel terbaru lebih cepat dan berisi banyak fitur baru seperti APR (*Arp Poison Routing*) yang memungkinkan *sniffing* di *switched* LAN. *Sniffer* dalam versi ini juga dapat menganalisa protokol terenkripsi seperti SSH-1 dan HTTPS, dan berisi filter untuk menangkap berbagai mekanisme otentikasi. Versi baru ini juga memonitor otentikasi *routing protocol* dan kamus dan *brute force cracker* untuk semua algoritma *hashing* umum dan untuk otentikasi spesifik, kalkulator *password/hash*, serangan *kriptoanalysis*, decoder password dan beberapa utilitas yang tidak begitu umum terkait dengan jaringan dan sistem keamanan.



Contoh screenshot program Brutus

Menggunakan Brutus akan mengajarkan anda banyak tentang sistem, karena mensimulasikan serangan nyata. Untuk membuat baik pengguna simulasi serangan Brutus, seorang administrator harus memperhatikan apakah usaha *break-in* akan dicatat, dan apakah *timeout* dikeluarkan setelah beberapa kali gagal *login* – ini dapat dengan mudah dilihat pada Brutus.

B. Brutus

Brutus adalah jenis *cracker password* yang berbeda. Brutus bekerja online, mencoba membobol telnet, POP3, FTP, HTTP, RAS atau IMAP dengan hanya mencoba untuk login sebagai pengguna ang sah. Brutus meniru serangan dari luar seperti kenyataannya (tidak seperti *cracking password* aplikasi lain yang mensimulasikan serangan *internal*) dan dengan demikian berfungsi sebagai alat keamanan audi berharga.

Brutus dapat berjalan dalam modus *single user* (mencoba masuk keakun pengguna tunggal dengan mencoba kombinasi *password* yang berbeda) atau dengan mencoba daftar kombinasi *user/password* dari file. Aplikasi akan memindai *host* untuk layanan yang dikenal dan dapat dengan mudah dimodifikasi untuk *break-in* laanan *custom* lain yang membutuhkan *login* interaktif dari sebuah *username* dan *password*.

IV. KESALAHAN UMUM

Istilah *Brute Force Attack* sama saja dengan *Brute Force*. Dinamakan *Brute Force Attack* karena tujuan dari *Brute Force* itu sendiri adalah untuk meretas sesuatu (*attack*)

V. KESIMPULAN

Teknik peretasan menggunakan *Brute Force* (*Brute Force Attack*) adalah teknik/cara yang sangat kuno dan memakan waktu yang lama disbanding dengan teknik peretasan *modern*. Walupun dikena dengan teknik kuno namun *Brute Force* sendiri adalah ancaman yang berbahaya karena hampir semua *password* atau kriptografi dapat dipecahkan dengan algoritma ini, meskipun memakan waktu yang lama.

Untuk mempersulit *Brute Force Attack* dalam

melakukan peretasan, dalam pembuatan *password* atau kriptografi paling sedikit buatlah 6 karakter password dengan kombinasi huruf besar, huruf kecil dan angka. Dengan demikian maka *Brute Force Attack* akan membutuhkan waktu yang lebih lama untuk memcahkan *password* tersebut. Hampir tidak ada *password* yang tidak dapat dipecahkan oleh *Brute Force Attack* namun kita dapat me-*reduce* atau mempersulit *Brute Force Attack* agar *Brute Force Attack* memakan waktu yang lama dalam meretas *password* kita.

REFERENCES

Algoritma Brute Force (2014).ppt. Munir, Rinaldi.

www.tech-faq.com/brute-force-attack.html. tanggal akses 18 Mei 2014 pukul 19.00

http://www.hoobie.net/brutus/ tanggal akses 18 Mei 2014 pukul 19.45

www.oxid.it/cain.html tanggal akses 18 Mei 2014 pukul 20.00

www.computerhope.com/jargon/b/brutforc.htm tanggal akses 18 Mei 2014 pukul 20.00

Munir. Rinaldi, "IF2211 STRATEGI ALGORITMA dikat kuliah Strategi Algoritma", Departemen Teknik Informatika, 2009.

http://hackingspirits.com/eth_hac/tools/brute_force.html tanggal akses 18 Mei 20.00

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, 18 Mei 2014

Ichwan Haryo Sembodo 13512008