Creating Dictionary for Attacks

Vyom Bhatia

Information Technology, SRM University

([bhatia.vyom25@gmail.com](mailto:bhatia.vyom25@gmail.com))

***Abstract-***In the today’s world, safety and security are the major issues faced by every individual. In cryptanalysis and computer security, a dictionary attack is a technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or passwords by trying hundreds or sometimes millions of likely possibilities, such as words in a dictionary. In this project, we use this approach to crack passwords to access emails, Wireless access points, encrypted zip files etc. We have used a simple C++ program to apply strings over and over again.

***Keywords – passwords; strings; dictionary; crack***

***1. Introduction***

It is often said that hacking (ethical or cracking) requires both time and patience. In cryptanalysis and computer security, password cracking is the process of recovering passwords from data that have been stored in or transmitted by a computer system. A common approach (brute-force attack) is to try guesses repeatedly for the password and check them against an available words in the dictionary.

***2. Algorithm Used***

Brute force algorithm is used to crack the passwords. Brute Force algorithm is also known as naive string matching algorithm.

[1]Exact string matching is used in search of exact occurrence of a string A compared to passwords (say string B). These algorithms are applied in biology, and especially in the segment concerning DNA chains. Much of data processing in bioinformatics involves in one way or another recognizing certain patterns within DNA, RNA or protein sequences.

Naive string matching algorithm is a type of single pattern string matching algorithms. It has no pre-processing phase, needs constant extra space. It always shifts the window by exactly one position to the right. It requires 2n expected text characters comparisons. It finds all valid shifts using a loop that checks the condition P[1....m]=T[s+1........s+m] for each of the n-m+1 possible values of s.

Consider the following example. T=ANPANMAN

P=MAN

ANPANMAN

A brute force method for string matching algorithm is shown in Figure 1:

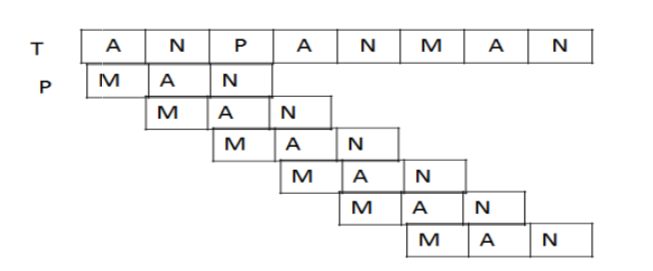


Figure 1: Naive String Matching Example

***3. Programming***

A simple C++ program is used to apply strings over and over again.

*3.1 Libraries Used*

*3.1.1.iostream*  
C++ input/output streams are primarily defined by iostream , a header file that is part of the C++ standard library (the name stands for Input/Output Stream).

*3.1.2conio*conio.h is a header file used mostly by MS-DOS compilers to provide console input/output. This header declares several useful library functions for performing console input and output from a program.Though the program was compiled on gcc compiler in Kali Linux OS, conio.h is included for compatibility reasons.

*3.1.3 fstream*

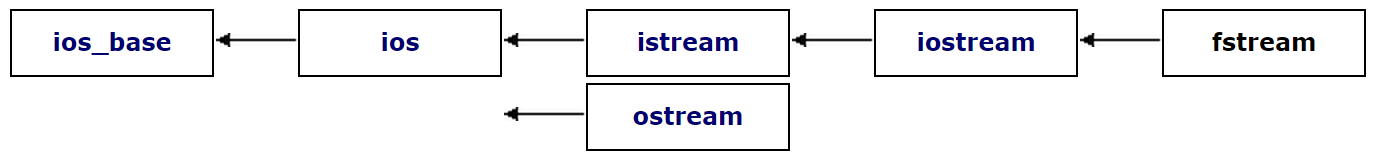


Figure 2: Hierarchical level of fstream

Input/output stream class to operate on files.  
Objects of this class maintain a filebuf object as their internal stream buffer, which performs input/output operations on the file they are associated with (if any).  
[2]File streams are associated with files either on construction, or by calling member open.A variable named file is created for handling file to store the output.

file.open("dictionary.txt", ios::out);  
Is used to create and open a file named dictionary in .txt format.

*3.2 Variables*

*3.2.1 Integer*Seventeen integer variables are defined where each variable represents one character in the password string.

*3.2.2Character*  
One array of size seventeen is declared to hold the string. The size of array increases after each increment of the for loop.

Following adds a null character at the end of each string.

key[a]='\0';

*3.3 Control Loop*

Like most programming languages, C++ provides flow control structures to implement repetition. These control structures, generally known as loops, allow us to write blocks of code that are executed a given number of times, or simply executed until certain condition is met. The advantage is that we need not write that block of code many times in the source code.

*3.3.1 For loop*The following loop is used to transfer control to the succeeding character in the string.

for(a=0;a<17;a++)

ASCII values for each character is incremented in the following loop.

for(a=32;a<=126;a++)

Where the variable ‘a’ is substituted for all the variables used in the nested loops. This particular loop is nested seventeen times for each character in the string.

*3.4 Typecasting*

Typecasting is making a variable of one type, such as an int, act like another type, a char, for one single operation.[3] To typecast something, simply put the type of variable you want the actual variable to act as inside parentheses in front of the actual variable. (char)a will make 'a' function as a char.

key[0]=(char)p;

This converts all the integer based ASCII’s into characters.

***4. Complexity***

[1]Brute Force algorithm takes time O((nm+1)m), and this bound is tight in the worst case. T5he worst case running time is thus O((n-m+1)m). The running time of Naive String Matching algorithm is equal to its matching time, since there is no preprocessing.

***5. Tools Used***

Following services were successfully cracked using this dictionary

*5.1 Wireless Access Points*

*5.1.1Operating System*

* Linux, Kali/ Backtrack OS

*5.1.2 Tool*

* Fern (Kali built in)

*5.2 Email Accounts*

The hotmail and gmail accounts are cracked using

*5.2.1Operating System*

* Windows 7/8

*5.2.2 Tool*

* em@il cracker

*5.3 Encrypted Zip Files*

The encrypted zip files are decrypted using

*5.3.1Operating System*

* Windows 7/8

*5.3.2 Tool*

* Ultimate Zip Cracker

*5.4 General Hit and trial*

Apart from the above mentioned scenarios, any login page can be attacked

*5.4.1Operating System*

* Any operating system

*5.4.2 Browser*

* Mozilla Firefox v20.0

*5.4.3 Plug-in*

* FireForce

***6. Issues Encountered***

*6.1 Issues Faced in Generating the Dictionary*

Hardware Deficiency: The high end machines with 8GB ram, quad core Ci7, though are considered as expensive and high end machines does not have the required hardware to execute the entire dictionary in a decent amount of time.

# 

*6.2 Issues faced in applying the dictionary*

*6.2.1. Lack of Appropriate Tools*Though the dictionary is created with the required passwords, specific tools are required to attack various security gateways.

*6.2.2 Time Required*Cracking passwords is a lengthy process which requires a lot of time and patience.

***6. Future Scope and Conclusion***

Though 25-GPU cluster cracks every standard Windows password in <6 hours. [4]A password-cracking expert has unveiled a computer cluster that can cycle through as many as 350 billion guesses per second. It's an almost unprecedented speed that can try every possible Windows passcode in the typical enterprise in less than six hours.  
Enhancement of tools that allow the user to upload .cpp files to attack instead of using .txt files. This saves a great amount of space. The .txt files may take up to 250GB of space whereas .cpp files may take up negligible space.  
Modern wireless access points have set a minimum of password length of 8 characters. So we will have to change the reference string length to 8-17. This will reduce the execution time by at most a day and reduce the required storage by 150GB.

Separate modules of each string length have to be created as a computer cannot be expected to run continuously for that length of time. So, the modules for each string length can be executed separately.

**References**

[1] String Matching Methodologies:A Comparative Analysis (ISSN 0975-9646) - Akhtar Rasool,Amrita Tiwari, Gunjan Singla,Nilay Khare Department of computer Science & Engg. Maulana Azad National Institute of Technology

[2] https://www.daniweb.com/software-development/cpp/threads/6542/fstream-tutorial

[3]https://www.cprogramming.com/tutorial/ lesson11.html

[4] http://arstechnica.com/security/ 2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours/