|  |
| --- |
| SET 8D |
| Crack4Beginner Python Application |
| COMP7481 – Carly Orr |

|  |
| --- |
| Danny Lieu  Mar 26, 2018 |

Table of Contents

[Background 2](#_Toc509858213)

[What it involves 2](#_Toc509858214)

[To-do lists 2](#_Toc509858215)

# Background

As a BTECH student, I am really interested of how one can crack another’s password and how long that process takes. Beside that, I have finished the course named COMP 8506 which have allowed me to use multiple cracking and reconnaissance tools to penetrate a system. Throughout the course, I realized that people were the weakest link in network security and most of the time it was the user’s fault when the system was hacked. For example:

* The user uses a simple password which contains only digits or letters
* The password is the user’s birthday, phone number, address, etc.
* The password is stored in a visible place
* The password is not being updated for a long time

Inspiring from [howsecureismypassword.net](https://howsecureismypassword.net/), I want to do some experimental exercise to see how hard it takes to crack a password. Because of time limitation, I will implement a password cracking application to crack a password length of 8 or below.

# What it involves

The program will implement two types of password cracking: brute-force attack and dictionary attack. Brute-force attacking will in most cases take longer than a dictionary attack. The goal of the dictionary attack is to reduce the amount of time to crack a password compared to the brute-force attack by using a pre-generated list of possible passwords. The application will be implemented in Python and will provide a friendly GUI for better user interactions.

# To-do lists

In order to achieve the project’s goals, here are what I will need to do:

* Do some research of what kinds of technique are used in password cracking process
* Review the existing online source code to understand what it takes to implement an application
* Design how the application will be implemented in term of algorithm uses, multithreading techniques, etc.
* Implement the application using Python
* Test and debug to see if the application can be enhanced