**A Project Report**

**on**

**Random password generation using Python**

***Submitted by:***

**Sanjay Kumar Sharma (20715)**

**Tarun Rathi (20717)**

***In partial fulfillment for the award of the degree***

***Of***

**M.Sc**

**In**

**CYBER SECURITY**

**at**

**DEPARTMENT OF COMPUTER SCIENCE**

**MAHARAJA GANGA SINGH UNIVERSITY, BIKANER**

**NH-15, JAISALMER ROAD, RAJASTHAN(INDIA) - 334001**

**MONTH & YEAR**

**February, 20****22**

**DECLARATION**

We hereby declare that the project entitled “**Random password generation using Python**” submitted for the M.Sc Computer Science degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the SANJAY KUMAR SHARMA

Signature of the TARUN RATHI

Place:

Date:

**CERTIFICATE**

This is to certify that the project titled “**Random password generation using Python**” is the bona fide work carried out by **Sanjay Kumar Sharma** & **Tarun Rathi**, Students of M.Sc Computer Science of Department of Computer Science, Maharaja Ganga Singh University, Bikaner, Rajasthan (India) during the academic year 2019-22, in partial fulfillment of the requirements for the award of the degree of M.Sc Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Contents

[System Requirement 5](#_Toc95383380)

[Hardware Requirement: 5](#_Toc95383381)

[Software Requirement: 5](#_Toc95383382)

[Programming Languages used: 5](#_Toc95383383)

[Introduction 6](#_Toc95383384)

[Analysis 6](#_Toc95383385)

[Remember These Strong Password Best Practices 7](#_Toc95383386)

[1. Do not use sequential numbers or letters 7](#_Toc95383387)

[2. Do not include your birth year or birth month/day in your password 7](#_Toc95383388)

[3. Use a combination of at least eight letters, numbers, and symbols 7](#_Toc95383389)

[4. Combine different unrelated words in your password or passphrase 7](#_Toc95383390)

[5. Do not use names or words found in the dictionary 7](#_Toc95383391)

[6. Use a password manager to store your passwords 7](#_Toc95383392)

[7. Do not reuse your passwords 7](#_Toc95383393)

[Information You Should Never Include in Your Passwords 8](#_Toc95383394)

[Key points of Password Security 8](#_Toc95383395)

[Source Code 9](#_Toc95383396)

[Explanation 10](#_Toc95383397)

ss[Screenshots 13](#_Toc95383415)

## System Requirement

Languages and tools used to build project:

### Hardware Requirement:

* Internet connection
* Minimum 512 MB RAM
* Minimum 20 GB HDS

### Software Requirement:

* 64-bit Linux (recommended), Windows 64-bit, OSX 64-bit
* Editor : Visual studio code, Sublime
* Python 3.6 (or any upper version)

### Programming Languages used:

Python

## Introduction

Almost every person owns or operates a password-protected machine. So many online services utilize password security. Some of these services include an email account and a bank account. Digital systems store confidential information. Some of these systems are computers, phones, and safety deposit boxes (Password Protection, 2010). A password should be chosen such that it is very hard to speculate or hack. It should be complex but easy to remember.

## Analysis

A password can be a string of lettering or a secret word depending on the interests of the user. A password is what confirms to the computer or any other digital device that is protected by a secret code that the person accessing the information is who he/she claims to be (Password Protection, 2010). Safeguarding a password is very important because if an individual can access another parson’s password, then he/she is in a position to access that person’s confidential information such as bank account detail. To avoid such incidences, the password should be very strong.

A password can be made stronger and hard to hack or guess by observing a recommended procedure of password generation. One of the factors to be considered when creating a password is the length of the password. The longer it is, the more security it will offer (How to Make Strong Passwords Stronger, 2012). The minimum number of recommended characters in a password is eight. The complexity of the password is also enhanced by using the entire keyboard while creating it.

It should contain letters in lower and upper cases, punctuations, numbers, and symbols (Tips for creating a strong password, 2012). The more the characters are varied, the stronger the password. This is because password hacking software has the capability of automatically scanning for the possibilities of one’s password. The only way the cracking of a password can be avoided is through making it complex.

Acronyms have important applications in password creation. The acronyms chosen should be easy to remember. They should also relate to an event that is easy to remember (How to Make Strong Passwords Stronger, 2012). Other common mistakes that people commit when generating password include the use of words from the dictionary, spelling words backward, use of common abbreviations, the use of a sequence of repeated characters, the use of letters that are adjacent on the qwerty keyboard, and the use of personal information in the password (Tips for creating a strong password, 2012). If such mistakes are avoided while creating a password and the recommended length is observed, then the generated password will be strong enough to offer the required security.

## Remember These Strong Password Best Practices

Please take a few minutes to review these seven strong password best practices and to create new passwords for any accounts that do not follow these password guidelines:

### **1. Do not use sequential numbers or letters**

For example, do not use 1234, qwerty, jklm, 6789, etc.

### **2. Do not include your birth year or birth month/day in your password**

Remember that cyber criminals can easily find this information by snooping into your social media accounts.

### **3. Use a combination of at least eight letters, numbers, and symbols**

The longer your password and the more character variety it uses, the harder it is to guess. For example, ***M0l#eb9Qv?*** uses a unique combination of upper- and lowercase letters, numbers, and symbols.

### **4. Combine different unrelated words in your password or passphrase**

This makes it difficult for cyber criminals to guess at your password. Do not use phrases from popular songs, movies, or television shows. Use three or four longer words to create your passphrase. For example, ***9SpidErscalKetobogGaN***.

### **5. Do not use names or words found in the dictionary**

Substitute letters with numbers or symbols to make it difficult to guess the password. Or deliberately use spelling errors in the password or passphrase. For example, P8tty0G#5dn for “patio garden.”

### **6. Use a password manager to store your passwords**

Do not store your passwords in a document on your computer. Make sure you’re using the password manager tool provided to you by the IT/support team to store all professional and personal passwords.

### **7. Do not reuse your passwords**

Every device, application, website, and piece of software requires a unique and strong password or PIN. Remember, if a cyber criminal does guess one of your passwords, they will use this to attempt hack into all of your personal and professional accounts.

## Information You Should Never Include in Your Passwords

When updating and creating new passwords, please do not include the following information in your passwords:

* Your pet’s name.
* Your birthday or that of family members.
* Any words related to your hobby, job, or interests.
* Part of your home address including, city/town, street, house/apartment number, or country.
* Your name or the name of a family member.
* Cyber criminals research their victims online looking for clues that can help them hack your password. And they will any clues about you, where you live, your interests, and your family to strategically guess at your password.

## Key points of Password Security

There are key points of password security that users must know in order to reduce the likelihood of a hacker cracking their password and thus gaining access to their device.

* Most importantly, passwords must be long and complex.
* Long and complex passwords require more effort and time for a hacker to guess.
* ***Passwords should contain at least ten characters*** and have a combination of characters such as commas, percent signs, and parentheses, as well as upper-case and lower-case letters and numbers.
* *Users should never write down their passwords*, as that makes it easier for the passwords to be stolen and used by someone else.
* Also, ***never use the same password*** for two or more devices, as hackers who break into one machine will try to use the same password to take control of others.
* If it’s hard to guess a strong password **use our program to get a strong and secure password.**

## Source Code

**import random**

**import string**

**print('hello, Welcome to Password generator!')**

**length = int(input('\nEnter the length of password: '))**

**lower = string.ascii\_lowercase**

**upper = string.ascii\_uppercase**

**num = string.digits**

**symbols = string.punctuation**

**all = lower + upper + num + symbols**

**temp = random.sample(all,length)**

**password = "".join(temp)**

**print(password)**

## Explanation

## Import – is the keyword used to import the other python library already created for ready to use in any further programs. With the *Import* keyword we can use a complete predefined library or a part of it. To use a library in our program, library name is followed by the *import* keyword so here we are going to use some library in our code as follows.

## Random - Random module or library is used to perform the random generations. We are making use of *random.sample* module here. If you will observe in the output all characters will be unique. random.sample() never repeats characters. If you don’t want to repeat characters or digits in the random string, then use random.sample() but it is less secure because it will reduce the probability of combinations because we are not allowing repetitive letters and digits. Random module or library is our first tool to generate totally unique passwords each time.

## String – String Module or library is used to predefine the characters set to be used in our code. The string module contains a number of useful constants, classes and a number of functions to process the standard python string. Here we are going to use some of *sub-modules* or classes in string library. These *sub-modules* or classes or String library are as follows:

1. ***string.ascii\_letters***: Concatenation of the ascii (upper and lowercase) letters – here it will allow all the Upper and Lower case characters supported by the *ASCII standards.*
2. ***string.ascii\_lowercase:*** All lower case letters – this will allow to use all the lower case characters supported by *ASCII standards.*
3. ***string.ascii\_uppercase:*** All Upper case letters – with this sub module we can use all the UPPER CASE characters supported by the *ASCII standards.*
4. ***string.digits:*** The string ‘0123456789’. – As it suggest with this sub-module we can use all the digits, all from 0-9 in a random way.
5. ***string.punctuation:*** Here this sub-module called ***string.punctuation*** will allows us to use some of the punctuation String of ASCII characters which are considered punctuation characters in the C locale.

## Print – print is prebuilt or inbuilt function to show a message on screen for user. Here we are greeting and welcoming users to our program to begin random password generation.

## Length – here we have declared a variable in code to ask user for length of the password. Here while taking input from user we have used the int method in input function, so user only can put an integer number for length of the password. If the user puts anything non-integer as length of password, it will show error. So user have to an integer as input to run program successfully.

## Now here we have declared some variables to store string modules character set, are as follows:

## Lower – a sub-module of string library to use all lowercase characters.

## Upper – another sub-module of sting module to use all UPPER CASE CHARACTERS.

## NUM - It is also an sub-module of string module to use all the digits in our password.

## Symbol – with this, we can use all the punctuation characters in our code.

## All – this is the combination of all string sub-modules, and this will store our password

## Temp – here we temporarily store our password after randomize the characters. For randomizing the characters, here we have used *random.sample* function from *random* module. In this function we have passed two position arguments i) all – it contains all the character set and, ii) length – for length limit as how long our password should be.

## Here length of the password is already defined by the use as we have taken password length from user as length = int(input('\nEnter the length of password: '))

## This will randomize the characters for the length given by user.

## Password – here we put our temporary password to variable called ‘Password’ by joining Temp with empty sting by *"".join(temp)*

## Now our password is ready to be show or print for user. Here we use:

## Print – to print the randomized password stored in *Password* variable.

## 

## Screenshots



