IBM 24Hr Hackathon 2022

Yenepoya Institute of Arts Science Commerce Management, Mangalore.

Team

Team Leader – Yash Oswal
Contact: +91 9834588979

Prof. Noshir Z Tarapore
Contact: +91 9326850144

Yash Oswal
Atharva Gokhale
Shivani Shete

Hasnain Merchant

Problem Statement

13. Constructing a password guesser and evaluator in Python.

Abstract

Fast-paced digitalization of the majority of our daily activities is producing massive amounts of data. This data is just like 'sugar' to the ants, 'the hackers'. We require the proper Authorization, Authentication, and Security procedures to safeguard this data. One such effective, simple to use, and affordable technique of data protection is the use of passwords. But despite all of its advantages, PASSWORDS CAN BE CRACKED!! Black hat hackers and penetration testers both employ a variety of methods to crack passwords which are mentioned below in the introduction section.

Our project "**PASS HAX**" focuses on the assessment of a password and the creation of a user/target specific dictionary of passwords sorted according to strength. This project will assist penetration testers in developing more specific dictionaries to execute attacks using dictionaries. Additionally, it will assess a user-provided password based on its length, character variety, and use of both uppercase and lowercase letters.

PASS HAX will assist the user to create a strong password.

Introduction :

Password guessing: -

- Password guessing is an online technique that involves attempting to authenticate a particular user to the system. Password cracking refers to an offline technique in which the attacker has gained access to the password hashes or database.
- Most web-based attacks on passwords are of the password guessing variety, so web applications should be designed with this in mind from a detective and preventive standpoint.
- Monitoring the system logs for failed login attempts can help identify password guessing. Clipping levels are useful to distinguish between the average user mistakenly mistyping their password and the attacker. A minimum reporting threshold level is defined by clipping levels. By using the password guessing example, a clipping threshold might be implemented so that the audit system only issues an alert if a particular user experience failed authentication more frequently than five times in one hour.

Prevention: -

Account lockouts are a common prevention measure for successful password guessing attacks. Account lockouts are used to stop attackers from easily guessing the right password by trying a huge number of different possibilities.

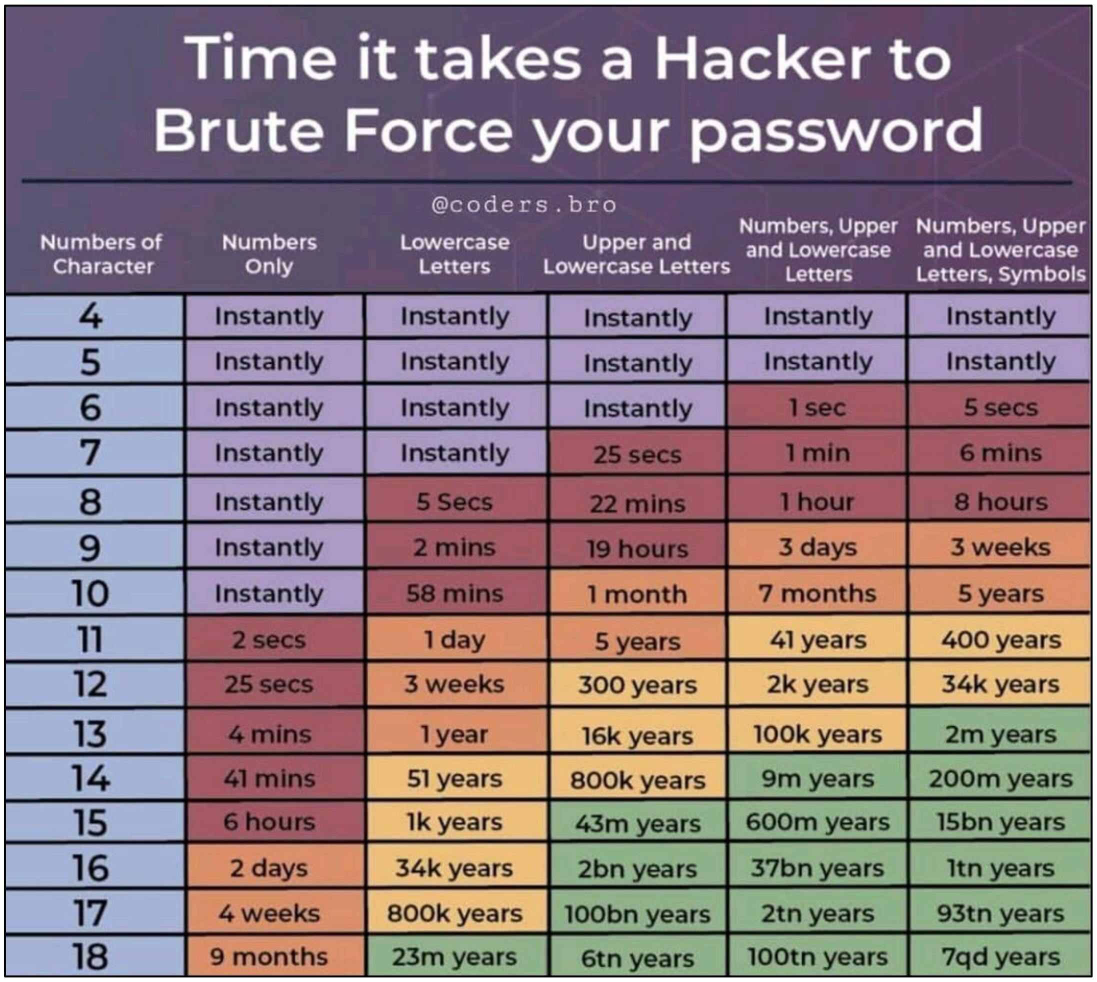
Password Cracking: -

- Password cracking is considered an offline attack because the attacker has gained access to a password hash for a particular account or the entire password database.
- Most password databases store the passwords as hashes rather than clear text. These one-way cryptographic hashes are created by running the plaintext password through a hashing algorithm such as MD5, LM, NT Hash (MD4), etc.

- The attacker will attempt to crack the password with a dictionary, hybrid, and then finally a brute force method if suitably motivated to achieve the plaintext password.

Prevention:

- Strong password policies that specify the right length, complexity, expiration, and rotation of passwords help prevent successful password cracking attempts.
- Another preventive precaution is having a robust system security that prevents the attacker from ever getting access to the password database.



Time it takes a Hacker to Brute Force your password

@coders.bro

Numbers of Character	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 Secs	22 mins	1 hour	8 hours
9	Instantly	2 mins	19 hours	3 days	3 weeks
10	Instantly	58 mins	1 month	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	6tn years	100tn years	7qd years

Understanding Password Cracking:

Password cracking involves acquiring valid passwords. You can do this in several ways, including:

- Via various types of attack
- Recovery and exploitation of passwords stored on the system
- Use of password decryption software
- Social engineering

Methods of attacking passwords:-

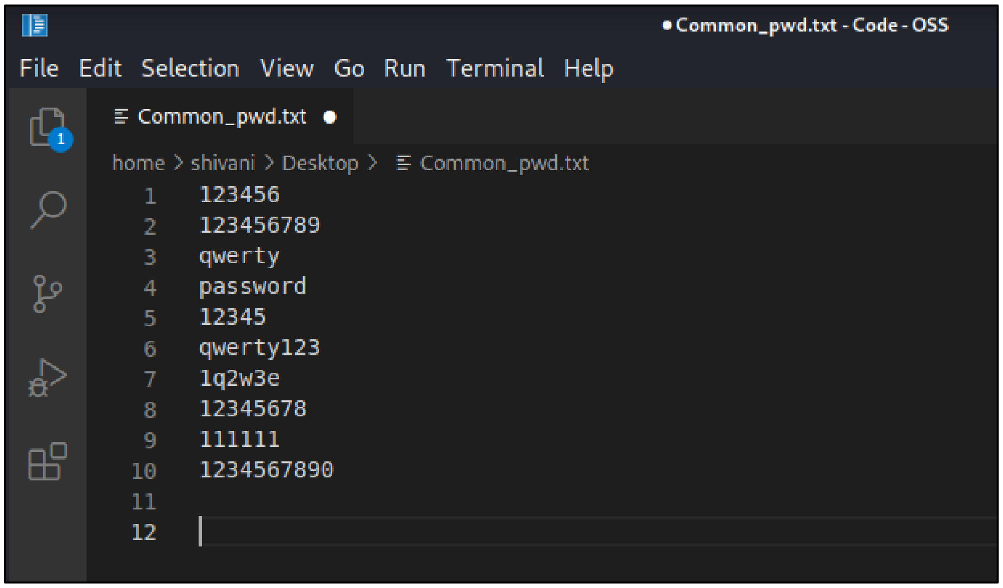
1. *Guessing Attack:*

In the guessing attack, perpetrators are successful when they are able to guess a person's password. This can occur if a user has selected a blank password. It can also occur if the user has chosen a simple password such as "password."

2. *Dictionary Attack:*

With a dictionary attack you load a file of dictionary words into the password cracking tool, and if the password is one of the words within the dictionary file it is cracked.

- The dictionary method simply directs the password cracking tool to use a supplied list of words as potential passwords.
- The tool will encrypt the supplied word using the matching password algorithm, and compare the resulting hash with the hash in the database.
- If the two hashes match, then the plaintext password is now known.
- If the dictionary method is unsuccessful then the hybrid approach will likely be attempted.


●Common_pwd.txt - Code - OSS
File Edit Selection View Go Run Terminal Help
Common_pwd.txt ●
home > shivani > Desktop > Common_pwd.txt

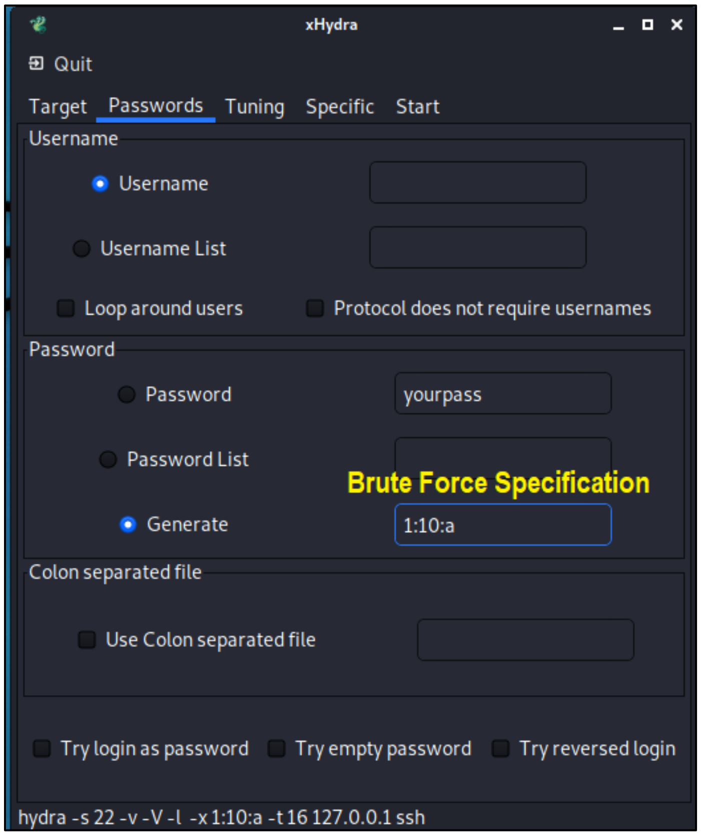
```
123456
123456789
qwerty
password
12345
qwerty123
1q2w3e
12345678
111111
1234567890


```

3. *Brute Force Attack:*

- Password brute forcing involves simply attempting every possible password until the correct match is found.

- Brute forcing will eventually yield the password, but the question is whether it will return the plaintext password quickly enough (days, months, or years) for it to still be of value.

- A variation on typical password brute forcing that can greatly increase the speed with which the correct password can be retrieved is a precomputation brute force attack.



Username

[✓ Username]

Username List

☐ Loop around users ☐ Protocol does not require usernames

Password

Password: yourpass

Password List

Brute Force Specification

[✓ Generate]: 1:10:a

Colon separated file

☐ Use Colon separated file

☐ Try login as password ☐ Try empty password ☐ Try reversed login

```
hydra -s 22 -v -V -l  -x 1:10:a -t 16 127.0.0.1 ssh
```

The above screenshot means that it will run passwords from 1 to 10 characters with lowercase letters only.
User is free to generate their own brute force pattern.

4. *Syllable Attack:*

The syllable attack is a combination of brute force attack and dictionary attack. The technique usually is used when the password is known to be a nonexistent word.

5. *Rule-Based Attack:*

The rule-based attack is used when the perpetrator is able to get some information about the password, usually following some form of enumeration that has identified the password policy in place for an organization. For example, if the policy indicates that the length of the password is not fewer than eight characters and must contain at least numbers and a special character, the perpetrator will adjust and customize the cracking tool for this.

6. *Hybrid Attack:*

A hybrid attack is used to find a password that is a dictionary word with combinations of characters prepended or postponed to it.

- The hybrid approach to password cracking still leverages a word list (dictionary), but makes alterations to the word before putting the guess through the hashing algorithm.

- Common alterations made by hybrid crackers include prepending or appending numbers or symbols to the password, changing the case of the letters in the word, making common symbol or number substitutions for letters (e.g., replacing an “o” with a “0”).

7. *Rainbow Attack*

- The rainbow attack technique makes use of rainbow tables, which are collections of all password hashes that are appropriate for a particular algorithm. Rainbow tables are tables of precomputed password-hash combinations, sometimes within specific confines such as an upper limit on password length or only including the more common symbols.
- While rainbow tables can simplify password cracking to a simple table lookup to locate the relevant password hash, building these rainbow tables takes a long time.

How to strengthen your password?

1. Password should be atleast 12 characters in length and should be the combination of uppercase letters, lowercase letters, special characters, and numbers.
2. Passwords should not contain any personal information such as name, birthdate, pet’s name, etc.
3. Consecutive keyboard characters such as “qwerty”, “asdfgh”, “vbnm” should be strictly avoided.
4. Make sure that you don’t save your passwords in plain text or keep them written in notebooks and never trust anyone with your password.
5. Change your passwords regularly and do not use same passwords for multiple applications. Do not login through untrusted platforms or sites.

Summary of ‘Pass Hax’

Platform : All computers with Python installed in it.

Tools Used : Python 3.10.4, Visual Studio Code, PyCharm were used in the development of the ‘Pass Hax’ project.

Usage of SDKs & APIs : No SDKs and APIs were used in the project.

Libraries used in Project :

Following libraries were used in the development of the project.

6. *colorama* (external) : This library makes ANSI escape character sequences (for producing colored terminal text and cursor positioning) work under MS Windows. This is used in our project to beautify the terminal experience while interacting with the program.
7. *time* (in-built) : Python's time module provides a function for getting local time from the number of seconds elapsed since the epoch called localtime()
8. *sys* (internal) : This library provides various functions and variables that are used to manipulate different parts of the Python runtime environment.
9. *random* (in-built) : Python has a built-in module that you can use to make random numbers.

All Components of the project were created during the 24HR duration of the Hackathon.