“How secure is your password?”

New Mexico

Supercomputer Challenge

Final Report

April 5, 2017

Capital-1

Capital High School

Team Member(s):

Ariel Arrellin

Alec Nordby

Teacher(s):

Irina Cislaru

Barbara Teterycz

Summary:

Modern day technology has come a long way from simple circuits and morse code. Passwords are keyphrases that are used to protect your information, and for that matter, the information of super-sensitive government agencies such as the NSA, CIA, FBI, and even the secret service. People use increasingly easy passwords that usually correlate with personal data, such as birthdays, addresses, and pet names. These passwords are easy to guess by hackers, and can result in the loss of personal data and/or millions of dollars of revenue.The purpose of the project was to discover what variables come into play when it comes to selecting a password for a secure account. For this purpose, we used a Python MD5 hash decoder and used it to bruteforce a set of passwords. The time it took changed based upon variations of passwords and length. With a long enough password, it will be borderline impossible to crack without using a multi-processor supercomputer.

Problem Statement:

What defines a good password and how can we implement those characteristics into our own personal passwords and more importantly what goes into decoding it?

Method:

The first step in our experiment was to find a sufficient base code and modify it so that it was suitable for our experiment.This came in the form of a “crack2.py” python file that we found online and modified so it ran with a directory. We edited “crack2.py” hash password by first typing a set of numbers no more than 6 into a md5 encoder. We then connected the directory to the actual crack2.py code and entered passwords into the spaces that the code would be guessing from. We ran the program and let the clock tick by as the cracker ran through passwords using dials. We gathered what results we could, and extrapolated the data that would take longer than a human lifespan to calculate.

Validity of Study:

The validity of our project can be proven through the data we collected. To the point where some passwords would take longer than a day we extrapolated that data by using the number of possible combinations or the amount of characters and variations of those characters. An example would be the password ‘DeUtch’, which contains 6 characters and 2 possible variations of each character, so the data to extrapolate the number of characters would be 6^2, which is 36 possible variations if we mix in the case sensitive properties of most passwords. This changes exponentially if you add symbols and a number set from 0-9.

Conclusion:

Passwords today are only as easy as we make them. By using completely random, or even encrypted, forms of our passwords we can become more safe from those who would wish to invade our systems. Of course eventually new scripts will be created to break down the walls of security that you have put up, your information will be displayed for everyone to see and your bank account emptied by a cyber thief. Keep up with your passwords, make them more intricate, and be careful what you do/say online.

Results (String of letters combinations to solve):

Letters/string	Possible combinations using lowercase only then lowercase and capitals	Solve time
cat	26^3	4 microseconds
udidntput	26^8	1 day
abcdjk	26^6	8 milliseconds

TurtletUrt	52^9	1 month
PossiblyAPassword	52^17	118 billion years
BleuMoon	52^8	22 minutes

More Charts (Words and Symbols):

String	Possible combination	Solvetime
Dragon+Hunter	82^{13}	3 seconds
cat=dog	82^{7}	3 min
Catapult#turtle	82^{15}	52 quadrillion years
Sony%Namco	82^{10}	130 thousand years

And More Charts (Word, Symbol, and Words):

String/Number/Symbol string	Possible combinations	Solvetime
A#23	92	2 hundred microseconds
@ctor1	92^6	400 milliseconds
P@tM8Mat	92^8	9 hours
K11l3rB3@aSt092	92^15	16 billion years

Achievement:

- We successfully managed to decrypt MD5 hash passwords up to 6 characters using numbers, 5 characters letters with numbers, and 4 characters using symbols, numbers, and letters.
- We got a dictionary attack working to a certain extent.
- We extrapolated from our findings, how long it would take to decode passwords based on length and possible variations.

Acknowledgments:

We wanted to thank our teachers Ms. Teterycz and Ms. Cislaru for supporting us through the project. We also wanted to thank the Supercomputing challenge for having us in Socorro and providing resources that helped us through the coding process.

Citation

http://stackoverflow.com/questions/11367553/brute-force-script-in-python-3-2

http://stackoverflow.com/questions/17064472/python-brute-forcing-very-basic

http://www.miraclesalad.com/webtools/md5.php

http://www.howtogeek.com/195430/how-to-create-a-strong-password-and-remember-it/

http://stackoverflow.com/questions/1240852/is-it-possible-to-decrypt-md5-hashes http://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-passwords-part-1-principles-technologies-0156136/