Lab 7 - Password Cracking

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The purpose of this lab is to learn more about passwords and password complexity. For this lab, you will use a web based password analyzing tool at <https://www.grc.com/haystack.htm> provided by Gibson Research Corporation.

If your password is the “needle” then the ability to hide your password depends on making the “haystack” as big as possible. You will also learn that some of the “truths” about passwords are myths. For example, which of the following two passwords is stronger, more secure, and more difficult to crack?

D0g…………………

PrXyc.N(n4k77#L!eVdAfp9

We have been told that clearly the second password is the better one because it is more secure. It is also impossible to remember. The Gibson Research tool, however, will show that the first password is not only easier to remember, but is 95 times more difficult to crack that the second password.

1. **Read the entire documentation on the web site, you will find it very interesting and informative. The information on this site is part of your course work and will appear on future tests.**

**Exercise 1: Using the 10 most common passwords used in the world.**

This list was compiled by PCMag.com and is something every “bad guy” has memorized. If your password is among this list, you may as well hand over your wallet or purse to bad guy right now.

1. Enter the password list below and record the Search Space Size, as a power of 10 and Offline Fast Attack Scenario. The first parameter measures the size of the haystack, and the second measures the speed of cracking based on current PC installed cracking tools.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **Password** | **Search Space Size (power of 10)** | **Offline Fast Attack Scenario** |
| 1 | password | 2.17 \* 10^11 | 2.17s |
| 2 | 123456 | 1.11 \* 10^6 | 0.0000111s |
| 3 | qwerty | 3.21 \* 10^8 | 0.00321s |
| 4 | abc123 | 2.24 \* 10^9 | 0.0224s |
| 5 | letmein | 8.35 \* 10^9 | 0.0835s |
| 6 | monkey | 3.21 \* 10^8 | 0.00321s |
| 7 | myspace1 | 2.90 \* 10^12 | 29.02s |
| 8 | password1 | 1.04 \* 10^14 | 17.41 mins |
| 9 | link182 | 8.06 \* 10^10 | 0.806s |
| 10 | <your first name> | 5.45 \* 10^13 | 9.08 mins |

**Exercise 2: Adding Complexity and Length to Password**

1. Now you will analyze how the search space and complexity influence the ability to crack the password.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **Password** | **Search Space Size (power of 10)** | **Offline Fast Attack Scenario** |
| 1 | 460 | 1.11 \* 10^3 | 0.0000000111s |
| 2 | 4609 | 111 \* 10^4 | 0.000000111s |
| 3 | 4d6A09 | 5.77 \* 10^10 | 0.577s |
| 4 | 4d6A09 | 5.77 \* 10^10 | 0.577s |
| 5 | 4d6A0%9 | 7.06 \* 10^13 | 11.76s |
| 6 | SeNeCa | 2.02 \* 10^10 | 0.202s |
| 7 | SeNeCa/ | 3.24 \* 10^13 | 5.41 mins |
| 8 | SeNeCa// | 2.76 \* 10^15 | 7.66 hours |
| 9 | SeNeCa//// | 1.99 \* 10^19 | 6.33 years |
| 10 | SeNeCa//?? | 1.99 \* 10^19 | 6.33 years |

1. Clearly the “SeNeCa//??” password is easier to remember than “4dA0%9”. What conclusion can your draw from the above Exercise: (write 3-4 sentences to explain your conclusion)

The conclusion I can draw is having repetition of symbols and dictionary words, you able to make a strong memorable password. Attackers are unaware of the length of a password. Thus, making it long but memorable will be beneficial for the user.

**Exercise 3: Cracking Hashes.**

All operating systems store passwords as hash values, either MD5 or SHA-1. There are various tools designed to steal the password hash value. For these tools to work, however, the hacker needs local access to the machine. (If unauthorized people have local access to a workstation, you have a larger security problem than just passwords). Once he/she has captured the hash values, the value is compared offline to a database of hash values to find a match. If the hacker finds a match to the hash value he\she assumes that must be the password. Take the following passwords in the table below and convert to hash values.

1. Navigate to the web page <http://passwordsgenerator.net/md5-hash-generator>
2. Enter the following passwords to convert to MD5 hash values. Copy the hash value to the table below.
3. Navigate to the web page <https://crackstation.net> . Read the documentation on the web site.
4. Use your phone or wrist watch to record the approximate time it takes to crack the password hash. (in seconds)
5. Enter the Captcha code and Click Crack Hashes

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **Password** | **MD5 Hash Value** | **Approximate Cracking Time** |
| 1 | password | 5F4DCC3B5AA765D61D8327DEB882CF99 | 2.17s |
| 2 | password1 | 7C6A180B36896A0A8C02787EEAFB0E4C | 2.920s |
| 3 | Passw0rd | D41E98D1EAFA6D6011D3A70F1A5B92F0 | 3.756 |
| 4 | P@ssw0rd | 161EBD7D45089B3446EE4E0D86DBCF92 | 1.016 |
| 5 | P@ssw0rd. | 4D934E4CDE0DCE1D9B3ECAF84F5672B2 | 0.819 |
| 6 | P@ssw0rd.. | 628C98267EDFD4766DB2BE05E3B2105F | did not find |

1. What conclusion can you make, from the above exercise, about the optimum, character mix? (write 3-4 sentences to support your answer)

Having repeating characters within the password as well as all other types of characters increases the password strength tremendously. Having all character types increases the total search space, increasing amount of possibilities. Repeating characters easily increase length of a password and hard to crack as the cracker does not know how long the password when they are trying to crack it.

1. What does padding (repetition of a character) do for the hacker and for us? (one sentence)

Padding increases length of a password making it harder to crack, yet easy to remember.

**Grading:**

* **LearnName\_Lab7\_Password.docx** – complete the tables and questions
* submit the lab file using the link on MySeneca