Assignment #2 : Password Cracking Tools (10th March 2025)

Objectives-

To help you gauge the level of security offered by the typical password systems. The goal is to crack as many passwords as possible in the inputs described below. This can take hours on fast laptops/ desktops, if you do not take advantage of multiple cores or GPU.

Students should be able to:

1. Recognize the limitations of the typical password schemes
2. Sketch the cracking algorithms, and use cracking tools

Background: Required Reading-

1. Password Cracking Ideas
2. Some Well Known Cracking Tools
3. ./Word-Lists and Hash-Dumps
4. The following are readily available "word lists" in the context of choosing good passwords. This is obviously not an exhaustive list. Note also that some files are populated at multiple sites.
5. https://github.com/danielmiessler/SecLists/ is the security tester's companion. It is a collection of multiple types of lists used during security assessments. List types include usernames, passwords, URLs, sensitive data grep strings, fuzzing payloads, and many more. It includes a collection of password lists. It includes the *RockYou* lists.
6. `/usr/share/wordlists/*` of the Kali Linux distribution.
7. http://contest-2010.korelogic.com/wordlists.html wordlists used in "Crack Me If You Can" contest of DEFCON 2010.
8. https://wiki.skullsecurity.org/Passwords Passwords that were leaked or stolen from sites.
9. http://gdataonline.com/downloads/
10. http://www.justpain.com/ut_maps/wordlists/
11. http://weakpass.com/lists
12. http://www.adeptus-mechanicus.com/codex/hashpass/hashpass.php
13. http://www.openwall.com/wordlists/
14. Mark Burnett, Today I Am Releasing Ten Million Passwords, Feb 9, 2015.

1. SubLab-1: Cracking MD5 Password Hashes using Hashcat/John the Ripper

MD5 examples shown for three users.

student:	29e08fb7103c327d68327f23d8d9256c
jsmith:	f6a0cb102c62879d397b12b62c092c06
jtripper:	c8645ebb3300e01459f7554dcbee024f

Crack the passwords of *student*, *jsmith*, and *jtripper* and include the screenshot in the report for each user.

2. SubLab-2: Cracking Windows NTLM Password Hashes using Hashcat

Extract the plain NTLM hash from your Windows OS and crack the password. Include the screenshots with username and cracked password. You can use existing **dictionaries with rules**, if password is not recovered then prepare a new dictionary with partial password and use **best64.rule** to crack it.

3. SubLab-3: Cracking SHA512 Password Hashes using Hashcat/John the Ripper

Since MD5 is considered "broken", Linux distributions have moved to using salted *SHA512* password hashes (crypt id 6, i.e., $6$), which are several orders of magnitude more difficult to brute-force. Example lines from machine M2:

```
root:$6$JL.TO.lFJwABA7sa$fy8wh8dIxHg59.vpWDPo1Xotmz3snOVTo0dGa
y0m.nNhya13GQZmXu2eTNmu5bGMfYjHWss70u0dq6n4JOs9f1
```

(**Hint**:

i. First letter of the password is in uppercase.
ii. Use ***example.dict*** wordlist given in Google Classroom.)

```
iiit:$6$R6WMQ54VhiUKK8wV$YgHU9GJAhGUffgm0ixWPwsv6A4J5Y3v6eeVje
jn85D9pU6mScZm338C.YOt0/C2M3sfMVdO5BkLAKVganONBg/
```

(**Hint**:

i. The password consists of ***iiitv*** *or* ***iiitp*** *or* ***iiitbh***, special character i.e @ and ***2014*** or ***2015*** or ***2016*** in any order. One letter is in uppercase. Create your own directory.
ii. The password ends with &. Use existing or create your rule.)

```
jazz:$6$NlyiS0mI6ud2FVX5$Bqkm1CpE6ZRzvFzhjT.8auBby9uIK9aizkp6Q
clpJJx6sQj8J3R95EtdiAF2h//arcg/8N6AMX4a3p5syfobC.
```

Given that password is having exactly 8 characters, first character as #, second character is in uppercase, rest characters are in lowercase and ends with digit 1.

`(`**`Hint`**`: Use mask instead of dictionary/wordlist).`

Crack the passwords of *root*, *iiit*, *jazz* and include the screenshot in the report for each user.