Exercise 02 – Instructor Manual

May 25, 2023

Overview

This exercise provides hands-on experience with Cryptography to perform brute force password attacks using '*John The Riper'* tool

Learning Objectives

- Understand basics of password management in Unix/Linux based systems
- Understand brute force attacks on password.
- Ability to apply basic regular expression understanding to provide appropriate patterns to *John The Ripper* tool

Reading Material

1. https://www.openwall.com/john/doc/RULES.shtml
2. https://miloserdov.org/?p=5477
3. https://en.wikipedia.org/wiki/Passwd

Prerequisites and environment familiarity

Student should be familiar with Linux and command line terminal usage.
Student should be familiar with use of `/etc/shadow` file structure
You will be given access to Kali Linux VM where the tool JTR (john) is installed. Information about the IP address of this VM and login process will be provided by TA for this course.

Description

The assignment is to carry out a password cracking exercise using open source software "John The Ripper" (https://www.openwall.com/john/) on a given system. You will be provided 20 entries of /etc/shadow file corresponding to usernames `cruser`<NN>, where NN can have value between 11 to 99 (including both). The passwords based on dictionary words consisting ofyour first name or last name and these names will be given to you in a file. You need to use this list of names as the dictionary to launch the dictionary base attack. Both the files "shadow" and "usernames" (to be used as dictionary words) will be available in the home directory `/home/cmsc626/HA01` on this VM machine.

To facilitate cracking, the passwords are constructed in one of the following ways and have minimum length of 8 and max length of 12.

a. Name itself e.g. if the dictionary word is "`ramrustagi`", then password can be "`ramrustagi`"
b. Name suffixed with some digits (123…). For example, if name is "`Ram`", then password could be "`Ram`NNNNN", where N can be between 1 to 9 such as "`Ram12345`" or "`Ram11111`".

c. Name suffixed or prefixed intermingled with digits. For example, if the name is "`Ram`", then password can be "NN`Ra`N`m`NN", where N can be between 1 to 9.
d. Letters of name (if length is less than 8, then appended with some number of digits to make it at least 8 characters) shuffled randomly, with at least two letters capitalized.

Your assignment is as follows:

a. Run the John The Ripper (JTR) tool with appropriate options for constructing the proper rule to launch the dictionary attack.
b. After you discover the password, verify by login to that user account with the password. After login to that user account, create a file named "password" in the home directory of the user which contains the password.

Explanation and Hints

a. First run the tool *john* with just simple dictionary words as it is.
b. Run the tool *john* with a rule that adds digits at the end of the word.
c. Either create all possible shuffle of the given word(s) and run the tool with these words with capitalization
d. Run the brute force check for all possible combinations of all alphabets of max length 12.

Exercise Setup

1. The program `usercreate.py` generates 20 users with username as `cruser1`, `cruser2`, …, `cruser02` with password generated based on the input word list. The program requires two command line arguments as below:
    a. <wordlist>: This first argument corresponds to filename which contains 20 or more words. These can be names of the students of the class or any other English dictionary words Some of the names should be less than 8 characters. These words should preferably be in lowercase letters. An example of such file "`example-usernames.txt`" is provided as a sample.
    b. <passwordlist>: This second argument corresponds to filename which is generated by the program. Each line contains 3 field separated by Comma(', '). First field corresponds to the Linux login (username) e.g. `cruser1`. The second field corresponds to one of the words from the input file given as first argument. The last (third) field corresponds to derived password from the word in second field. This will be the actual password that can be used to login into the system.
2. The program outputs a list of shell commands that should be used to create the user with the specific derived password (as in field3 of passwordlist). An example of such a command is given below:

```
sudo -S useradd -m -s /bin/bash -p
'$6$q3VT92yoZ9jF8/Ge$RWPJb/bApppo/NUmwGSDg0tdfC92fMBhG
t7dAgmBFCymmQ3fRv.ZURxjUw3UdaVsSeFhd6Xhs1i4FcbEhDoMA.'
cruser1
```

3. Executing the above commands will result in actual creation of users with username as `cruser`<N> along with corresponding password.

4. User also needs a copy of `/etc/shadow` file since that file can only be read by sudo user. Thus, create a copy of shadow file in the student's home directory with list of those users whose passwords are expected to be cracked. For example,
   `sudo grep ^cruser /etc/shadow >/home/<loginid>/shadow`

   A student should use this shadow file for cracking the password.

Evaluation Support

1. The program `john` (John the Ripper) should be installed on the system. This can be done by using the following.
   `sudo apt install john`
2. By default, the configuration file used by john is /etc/john/john.conf. Modifying this file requires superuser privileges and thus, in general, user cannot modify this file. For password cracking, a regular non-sudo user should create a directory .john in the home directory and first copy the default configuration into this directory e.g.
   `mkdir ~/.john`
   `cp /etc/john/john.conf ~/.john`
3. A user should create its configuration rules in this copied file. When john is invoked, it first looks at this configuration file in subdirectory `.john` in the home directory.
4. The password cracking will be done in multi steps.
   a. Step 1: Invoke the john program in its simplest form as below. This will crack those passwords which corresponds to login name as password.
      `John –wordlist=<wordlistfile> <shadowfile>`

      It will display the cracked passwords along with corresponding usernames.
   b. To see this later, use the following command
      `John -show <shadowfile>`
   c. To crack more password, we need to create rules for wordlist in the configuration file ~/.john/john.conf. In this file, below the section [List.Rules:Wordlist], create rules as per worked out options. Some examples are given below
      i. Append 1 digit from 0 to 9 at the end of word
         `$[0-9]`
      ii. Append 2 digits from 0 to 9 at the end of word
         `$[0-9][0-9]`
      iii. Create similar rules for appending 3, 4 or 5 digits at the end
      iv. Prepend 2 digit at the beginning of the word. Similarly, add more rules for prepending 3, 4 or more digits at the beginning of the word
         ^[0-9]^[0-9]
      v. Reverse the word and capitalize the first letter
         `rc`
      vi. Rotate the word on the left by 2 positions and capitalize the first letter
         `{{c`
      vii. Make a combination of such rules and try to crack the passwords

viii. Check that password size must be greater than 7 or less than 13.
`>7`
`<D`
The letter A corresponds to 10, B corresponds to 11 and so on.

d. In general, a student should be able to crack 4 to 5 words depending upon how randomly words were chosen to crate the passwords.

Variation and Enhancement.

The program createusers.py should be modified to create password with other combinations of wordlist e.g. rotating the words by few positions, Appending special characters e.g, "$@%" etc. Of course such creations will take lot more time to crack the passwords often running days.

Note: The program john is CPU hungry and may take up all the CPU time. So, this should be monitored and when a program is running for more than 24 hours, the TA/admin should kill that program. Continuous 100% CPU usage results in warning at the VMWare console level which should be avoided.

An alternative to John The Ripper tool is `hashcat` (https://hashcat.net/hashcat/). This can be explored for password cracking. If the systems have GPU support, it is likely to lot more efficient than `john`.

Assessment and Rubric

Please do submit the following:

1. The commands that you used to invoke the password cracker (2 marks)
2. List of username and corresponding password (cracked by you) along with the time taken by the tool to discover each password. Tool can report that information when password is cracked. The marks allotment is as follows.
    a. 2 marks for first 2 password
    b. 2 marks for next 2 passwords
    c. 2 marks for 2 more passwords
3. (2 marks) For each user whose password is cracked, login to the machine with cracked password and create a file "password" in the home directory of the user (whose password is cracked" which will contain the cracked password. (2 marks). For example if password for user cruser1 is "xyz12345", the file /home/cruser1/password should contain the text "xyz12345".
4. Challenges faced and how did you address these (2 marks)

Note

Any plagiarism activity will result in penalties of being awarded 0 marks.

<end of Exercise 02 – Instructor Manaual>