Password Cracking Exercise

1. Overview

The learning objective of this lab is for students to understand how password cracking tools work in general and what happens after an attacker has compromised server and has obtained a list of username and password hashes.

Lab environment: Use the pre-built Ubuntu VM that has been provided for this class. For this exercise you can optionally use your own computers. There is no risk involved in using your system, however, the guide only describes the installation process for the Ubuntu machine. If you want to install the password cracker on your own computers, you need to read the documentation included with it and install it based on your system configurations.

Submission: You need to submit a detailed lab report to describe what you have done and what you have observed. Follow the tasks and for each task answer the **Q#** specifically in your report. You may provide explanation to the observations that are interesting or surprising. You can always add code snippets or screenshots of what you have observed. You are encouraged to pursue further investigation, beyond what is required by the lab description. You can earn bonus points for extra efforts (at the discretion of your instructor). Only submit typed reports electronically. No handwritten reports accepted.

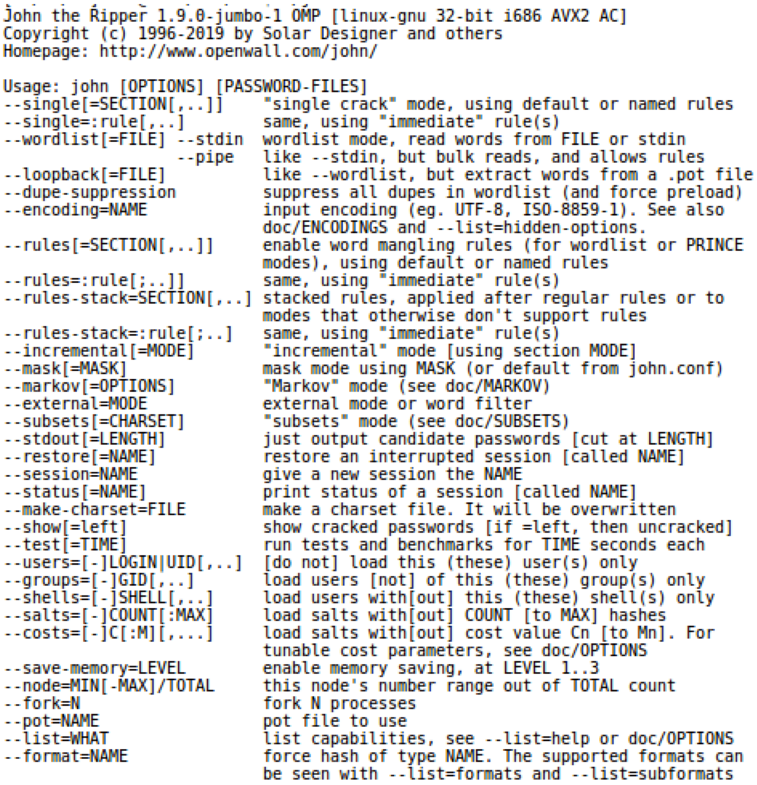
2. Tasks

Task 1: Installing John the Ripper

John the Ripper is one of the most commonly used password crackers that is freely available. Do the following steps inside your VM.

1. Click on this link: Click Here
2. Go to the second section titled: Download the latest John the Ripper jumbo release
3. Choose the first one: 1.9.0-jumbo-1 sources in tar.xz, 33 MB (file name is: john-1.9.0-jumbo-1.tar.xz)
4. Click on Save file and the file will be downloaded in the Download folder.
5. Go to the Download folder and right click on the file, select Extract Here to unzip it. You will see a folder named john-1.9.0-jumbo-1. You can move this folder to any other location if you want in order to access it easier.
6. Open a terminal, change directory to where this folder is. If you have not moved it this will be the command: `cd Downloads/john-1.9.0-jumbo-1/`
7. Change directory to the `'src'` folder: `cd src`
8. Type: `./configure && make`
9. Change directory to `'run'` folder by typing: `cd ../run/`
10. Run the program by typing: `./john`

    You should be able to see the following screen. This shows that you have successfully installed the program. The list shows various options available to use while running the program.



```
John the Ripper 1.9.0-jumbo-1 OMP [linux-gnu 32-bit i686 AVX2 AC]
Copyright (c) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION[,..]]     "single crack" mode, using default or named rules
--single=:rule[,..]         same, using "immediate" rule(s)
--wordlist[=FILE] --stdin   wordlist mode, read words from FILE or stdin
                  --pipe    like --stdin, but bulk reads, and allows rules
--loopback[=FILE]           like --wordlist, but extract words from a .pot file
--dupe-suppression          suppress all dupes in wordlist (and force preload)
--encoding=NAME             input encoding (eg. UTF-8, ISO-8859-1). See also
                            doc/ENCODINGS and --list=hidden-options.
--rules[=SECTION[,..]]      enable word mangling rules (for wordlist or PRINCE
                            modes), using default or named rules
--rules=:rule[;..]]         same, using "immediate" rule(s)
--rules-stack=SECTION[,..] stacked rules, applied after regular rules or to
                            modes that otherwise don't support rules
--rules-stack=:rule[;..]    same, using "immediate" rule(s)
--incremental[=MODE]        "incremental" mode [using section MODE]
--mask[=MASK]               mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]          "Markov" mode (see doc/MARKOV)
--external=MODE             external mode or word filter
--subsets[=CHARSET]         "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]           just output candidate passwords [cut at LENGTH]
--restore[=NAME]            restore an interrupted session [called NAME]
--session=NAME              give a new session the NAME
--status[=NAME]             print status of a session [called NAME]
--make-charset=FILE         make a charset file. It will be overwritten
--show[=left]               show cracked passwords [if =left, then uncracked]
--test[=TIME]               run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..]   [do not] load this (these) user(s) only
--groups=[-]GID[,..]        load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]      load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]      load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:M][,...]      load salts with[out] cost value Cn [to Mn]. For
                            tunable cost parameters, see doc/OPTIONS
--save-memory=LEVEL         enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL      this node's number range out of TOTAL count
--fork=N                    fork N processes
--pot=NAME                  pot file to use
--list=WHAT                 list capabilities, see --list=help or doc/OPTIONS
--format=NAME               force hash of type NAME. The supported formats can
                            be seen with --list=formats and --list=subformats
```

Take a look at the following documentation files:

- More information about the options can be found in john-1.9.0-jumbo-1 > doc > OPTIONS.
- The definition of modes such as *wordlist*, and *incremental* are described in john-1.9.0-jumbo-1 > doc > MODE.
- Examples on how to use the system are shown in john-1.9.0-jumbo-1 > doc > EXAMPLES.
- The john-1.9.0-jumbo-1 > doc > CONFIG file describes the configurations that you can modify. If you want to modify any of the configurations you need to change john-1.9.0-jumbo-1 > run > john.conf file. For example, if you want to emit a status line whenever a password is cracked find the following line in john.conf and change the ‘N’ to ‘Y’: CrackStatus = N.
  StatusShowCandidates is also useful in showing the number of password guesses tried.

The result of your password cracking will be stored in a file called john.pot. When you use multiple modes of cracking, the new results will be appended to this file.

Task 2: Cracking a set of passwords

You should have downloaded two files for this exercise. The *target.txt* file is a list of username and password hashes (md5) you are trying to crack. The *dictionary.txt* is a list of common words that you can use as a dictionary while running the password cracking attacks. Your goal in this task is to crack as many of the password hashes as possible from the target.txt file provided. You should try multiple modes with different variations. For example, you can try wordlist mode by using the command: `./john -wordlist=<PATH TO DICTIONARY>/dictionary.txt -format=raw-MD5 <PATH TO TARGET FILE>/target.txt`

The wordlist is the simplest mode and will just try words from the dictionary. You can then try adding some mangling rules to each word in the dictionary by using the command `./john -wordlist=<PATH TO DICTIONARY>/dictionary.txt -format=raw-MD5 -rules <PATH TO TARGET FILE>/target.txt`

Remember that the `incremental` mode will continue generating guesses for a very long time and may never finish, so you would want to try this option at the end, and manually stop this at some point (by pressing Ctrl-C).

Q1: You should try at least 3 different modes/techniques. Submit the list of passwords you were able to crack. Either show the plaintext passwords with their hashes, or with the username. Report on which modes you used. Show the command lines you used when cracking the passwords.

Task 3: Check your own password

Your goal in this task is to try and crack one of your own passwords. In order to try this, choose one of the passwords you personally use for one of your accounts (or do this for all of your passwords). Then Run this command in terminal to find the MD5 hash value of your password.

```
   printf '%s' "yourpassword" | md5sum
```

This will print the md5 hash value on the screen for you. Copy this md5 value and write it to a file with a similar pattern as the target file in the previous task (or just add it to to the end of the target file). Try running the password cracker in any mode you choose (single-crack, wordlist with or without rules, markov, incremental, etc...) for at least 5 minutes.

Q2: Report whether or not your password has been cracked and which mode you used. You can optionally try all modes or a combination of them, or run it for much longer than 5 minutes to see how strong your password is. Note that you should not include your password or its hash in the report, just explain whether or not your password was cracked, and what approach you used for cracking it.