Rainbow Table Attack: Cracking UNIX Passwords

Lab Environment

1. Following the tutorial to install Kali Linux. https://www.youtube.com/watch?v=GpTIM9OroIY
2. You can set the root as:
   - Username: root
   - Password: dees
3. Prerequisite

   You need the OpenCL Intel runtime. http://registrationcenter-download.intel.com/akdlm/irc_nas/9019/opencl_runtime_16.1.1_x64_ubuntu_6.4.0.25.tgz. The main page is https://software.intel.com/en-us/articles/opencl-drivers, Just use the Ubuntu version for Kali, It will say its unsupported but will install anyways. And it works!!

```
root@kali:~/Downloads# tar -zxvf opencl_runtime_16.1.1_x64_ubuntu_6.4.0.25.tgz
root@kali:~/Downloads# cd opencl_runtime_16.1.1_x64_ubuntu_6.4.0.25/
root@kali:~/Downloads/opencl_runtime_16.1.1_x64_ubuntu_6.4.0.25# ls
EULA.txt  install_GUI.sh  install.sh  pset  PUBLIC_KEY.PUB  rpm  silent.cfg
root@kali:~/Downloads/opencl_runtime_16.1.1_x64_ubuntu_6.4.0.25# . install_GUI.sh
```

4. Type to check the configuration: hashcat –b

```
root@kali:/# hashcat -b
hashcat (v3.10) starting in benchmark-mode...

OpenCL Platform #1: Intel(R) Corporation
========================================
- Device #1: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz, 290/1162 MB allocatable, 2MCU
```

Task 1: Crack Password Using Rainbow Table

1. Overview
   1.1. **rtgen**: generate rainbow tables
   1.2. **rsort**: sort the rainbow table
   1.3. **rcrack**: find the password
2. Display command:

```
root@kali:~# ls
build    Documents  hashcat  Music     Public     Videos
Desktop  Downloads  intel    Pictures  Templates
root@kali:~# mkdir passwordcrack
root@kali:~# ls
build    Documents  hashcat  Music          Pictures  Templates
Desktop  Downloads  intel    passwordcrack  Public    Videos
root@kali:~# cd passwordcrack/
root@kali:~/passwordcrack# mkdir rainbow
root@kali:~/passwordcrack# cd rainbow/
root@kali:~/passwordcrack/rainbow# rtgen --help
RainbowCrack 1.6.1
Copyright 2003-2015 RainbowCrack Project. All rights reserved.
http://project-rainbowcrack.com/

usage: rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_ind
ex chain_len chain_num part_index
       rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_ind
ex -bench

hash algorithms implemented in alglib0.so:
    lm, plaintext_len limit: 0 - 7
    ntlm, plaintext_len limit: 0 - 15
    md5, plaintext_len limit: 0 - 15
    sha1, plaintext_len limit: 0 - 20
    sha256, plaintext_len limit: 0 - 20

example: rtgen md5 loweralpha 1 7 0 1000 1000 0
         rtgen md5 loweralpha 1 7 0 -bench
root@kali:~/passwordcrack/rainbow#
```

3. Generate a rainbow table
   3.1. Type : rtgen md5 loweralpha 5 5 0 1000 10000 0

```
root@kali:~/passwordcrack/rainbow# rtgen md5 loweralpha 5 5  0 10000 100000 0
rainbow table md5_loweralpha#5-5_0_10000x100000_0.rt parameters
hash algorithm:         md5
hash length:            16
charset:                abcdefghijklmnopqrstuvwxyz
charset in hex:         61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 7
8 79 7a
charset length:         26
plaintext length range: 5 - 5
reduce offset:          0x00000000
plaintext total:        11881376

precomputation of this rainbow table is finished
```

4. Questions
   4.1 Explain each parameter
5. Find the generated file
   5.1 The generated file is stored under the folder /usr/share/rainbowcrack

```
root@kali:~/passwordcrack/rainbow# ls -la /usr/share/rainbowcrack/
total 5032
drwxr-xr-x   2 root root    4096 Sep 22 22:29 .
drwxr-xr-x 453 root root   20480 Sep 19 15:43 ..
-rw-r--r--   1 root root   34976 Jul 16  2015 alglib0.so
-rwxr-xr-x   1 root root     771 Jul 16  2015 charset.txt
-rw-r--r--   1 root root 1600000 Sep 22 22:11 md5_loweralpha#1-7_0_10000x100000_0.rt
-rw-r--r--   1 root root 1600000 Sep 22 22:31 md5_loweralpha#3-3_0_10000x100000_0.rt
-rw-r--r--   1 root root 1600000 Sep 22 22:21 md5_loweralpha#5-5_0_10000x100000_0.rt
```

5.2 Sort the rainbow table

```
root@kali:~/passwordcrack/rainbow# rtsort /usr/share/rainbowcrack/md5_loweralpha#5-5_0_10000x1
00000_0.rt
/usr/share/rainbowcrack/md5_loweralpha#5-5_0_10000x100000_0.rt:
546029568 bytes memory available
loading rainbow table...
sorting rainbow table by end point...
writing sorted rainbow table...
```

5.3 Find the password

5.3.1 Generate the md4 of the password “abced”

5.3.2 Type rcrack to find the password

```
root@kali:~/passwordcrack/rainbow# echo -n abcedef |  md5sum
2ed01372d1149baa5a79d25e5eda3372  -
root@kali:~/passwordcrack/rainbow# rcrack /usr/share/rainbowcrack/md5_loweralpha#5-5_0_10000x1
00000_0.rt -h  ebb080afaac3a990ad3f1d0f21742fac
489778790 bytes memory available
1 x 1600000 bytes memory allocated for table buffer
160000 bytes memory allocated for chain traverse
disk: /usr/share/rainbowcrack/md5_loweralpha#5-5_0_10000x100000_0.rt: 1600000 bytes read
searching for 1 hash...
plaintext of ebb080afaac3a990ad3f1d0f21742fac is abced
disk: thread exited

statistics
---------------------------------------------------------
plaintext found:                                1 of 1
total time:                                     6.43 s
  time of chain traverse:                       6.41 s
  time of alarm check:                          0.00 s
  time of wait:                                 0.01 s
  time of other operation:                      0.02 s
time of disk read:                              0.00 s
hash & reduce calculation of chain traverse:  49990000
hash & reduce calculation of alarm check:     74
number of alarm:                                64
speed of chain traverse:                        7.80 million/s
speed of alarm check:                           0.07 million/s

result
---------------------------------------------------------
ebb080afaac3a990ad3f1d0f21742fac  abced  hex:6162636564
```

Reference

- https://www.exploit-db.com/docs/104.pdf