**Lab 5: Password Cracking**

**Install VirtualBox**

1. If you haven’t already, install one of the VirtualBox platform packages from [here](https://www.virtualbox.org/wiki/Downloads).

**Set Up the “infoset-net” Network for VirtualBox**

## **infosec-net Network Map**

The network map is as follows:

| **IP Address** | **Machine** |
| --- | --- |
| 192.168.55.100  192.168.55.101 | Windows 10 VM  Kali Linux VM |

I have configured the virtual machines in the class to be on the same virtual network so that they have internet access and so that they can talk to one another. However, you must initialize the VirtualBox network after you install VirtualBox.

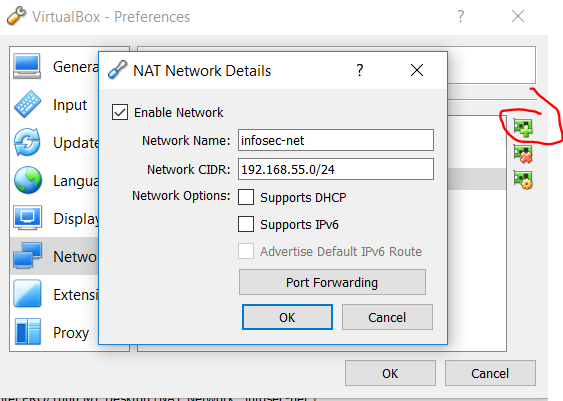
If you haven’t already done with in connection with Lab 2, do the following:

1. Create the virtual network.
   * If on Windows, download and run [this script](https://www.dropbox.com/s/923mt76knrherrm/create-infosec-natnetwork.bat?dl=0)
   * If on Mac:
     + Open Terminal application
     + Copy-paste the following lines into the terminal (each bullet is one line), and press enter
     + vboxmanage natnetwork add --netname infosec-net --network 192.168.55.0/24 --enable --dhcp off
     + vboxmanage natnetwork start --netname infosec-net

You can test whether this step was successful by navigating to Virtuabox > File > Preferences > Network, where you should see “infosec-net” in the list of networks. If you don’t see that network, then you can manually create the network from this dialog prompt by clicking the plus and using the following options.

* Enable Network: “checked”
* Network Name: “infosec-net” (case-sensitive!)
* Network CIDR: “192.168.55.0/24”
* Network Options:
  + Supports DHCP: “unchecked”
  + Supports IPv6: “unchecked”

Like this:



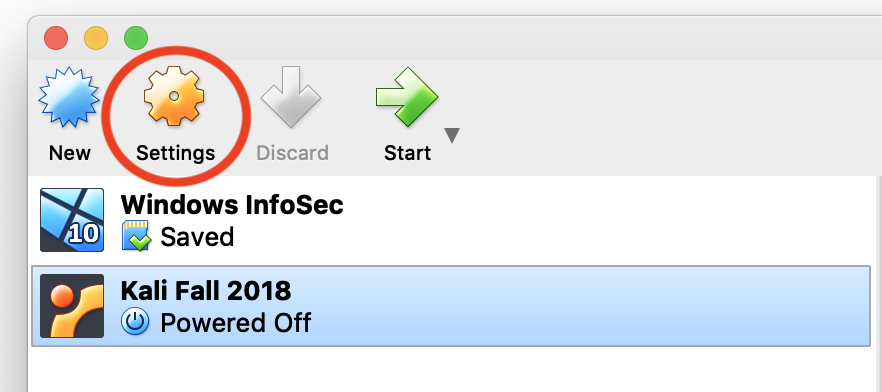
Then click ‘ok’ and ‘ok’.

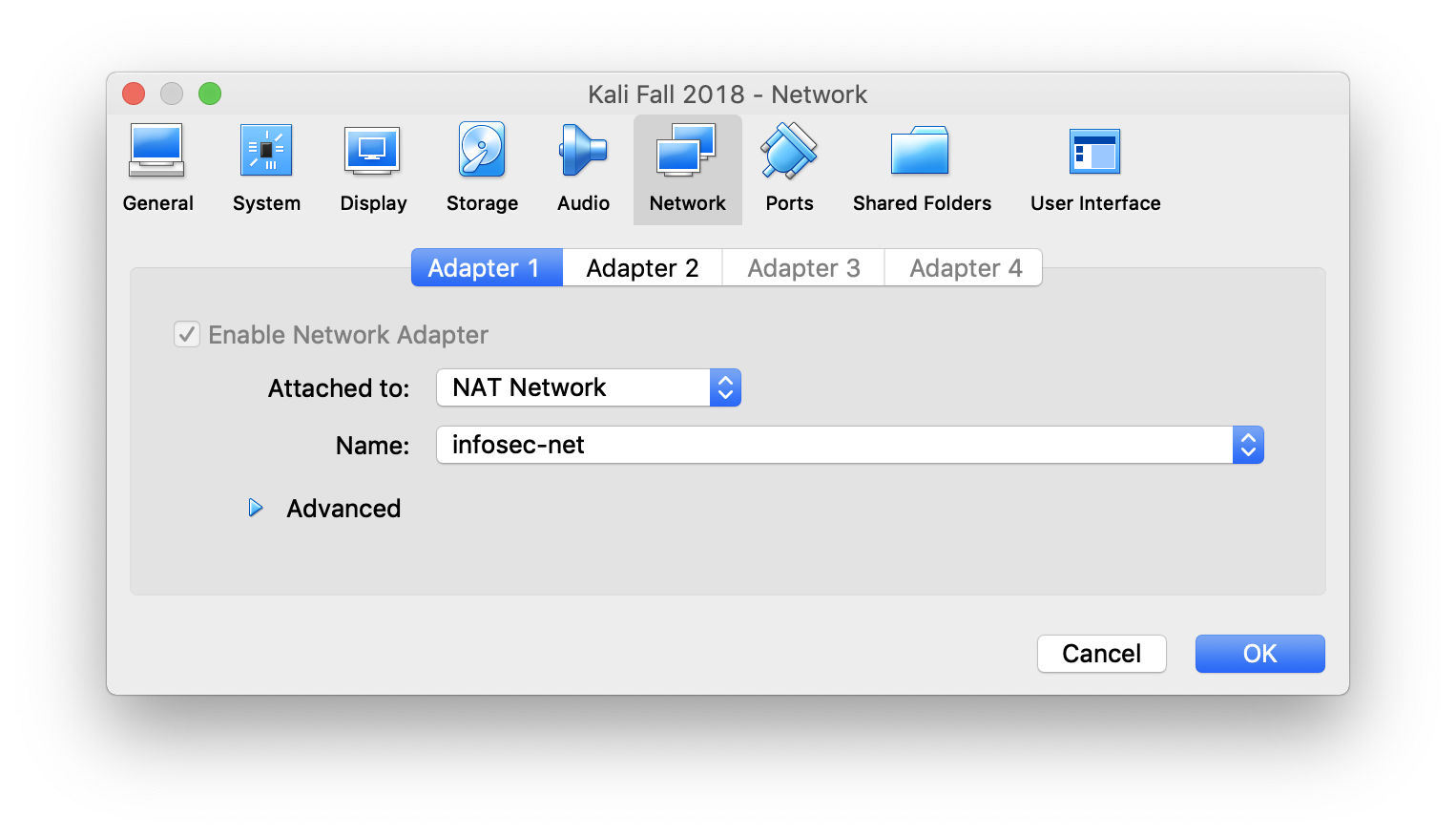
**Download and Setup the Kali VM**

1. Download a copy of the Kali Linux .ova file to your computer. If you didn’t get a copy in class, download it here:

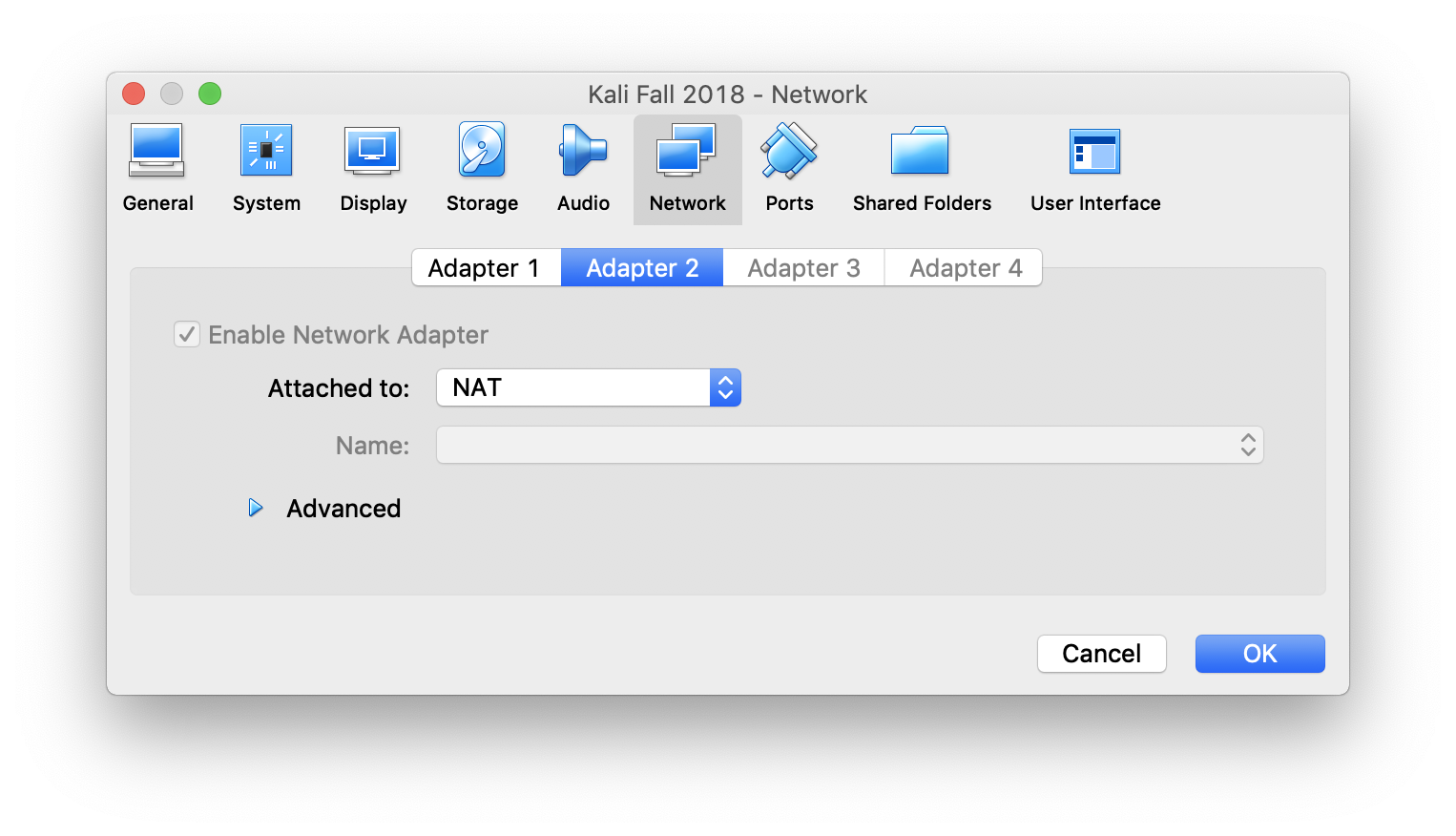
<https://www.dropbox.com/s/r93i8o7ibs43yeq/Kali%20Lite%20x64%202018.ova?dl=0>

1. In VirtualBox, go to “File” > “Import Appliance”, then browse to the “Kali Lite x64 2018.ova” file downloaded in the previous step.
2. After the virtual machine imports, select the Kali VM and click the settings icon:

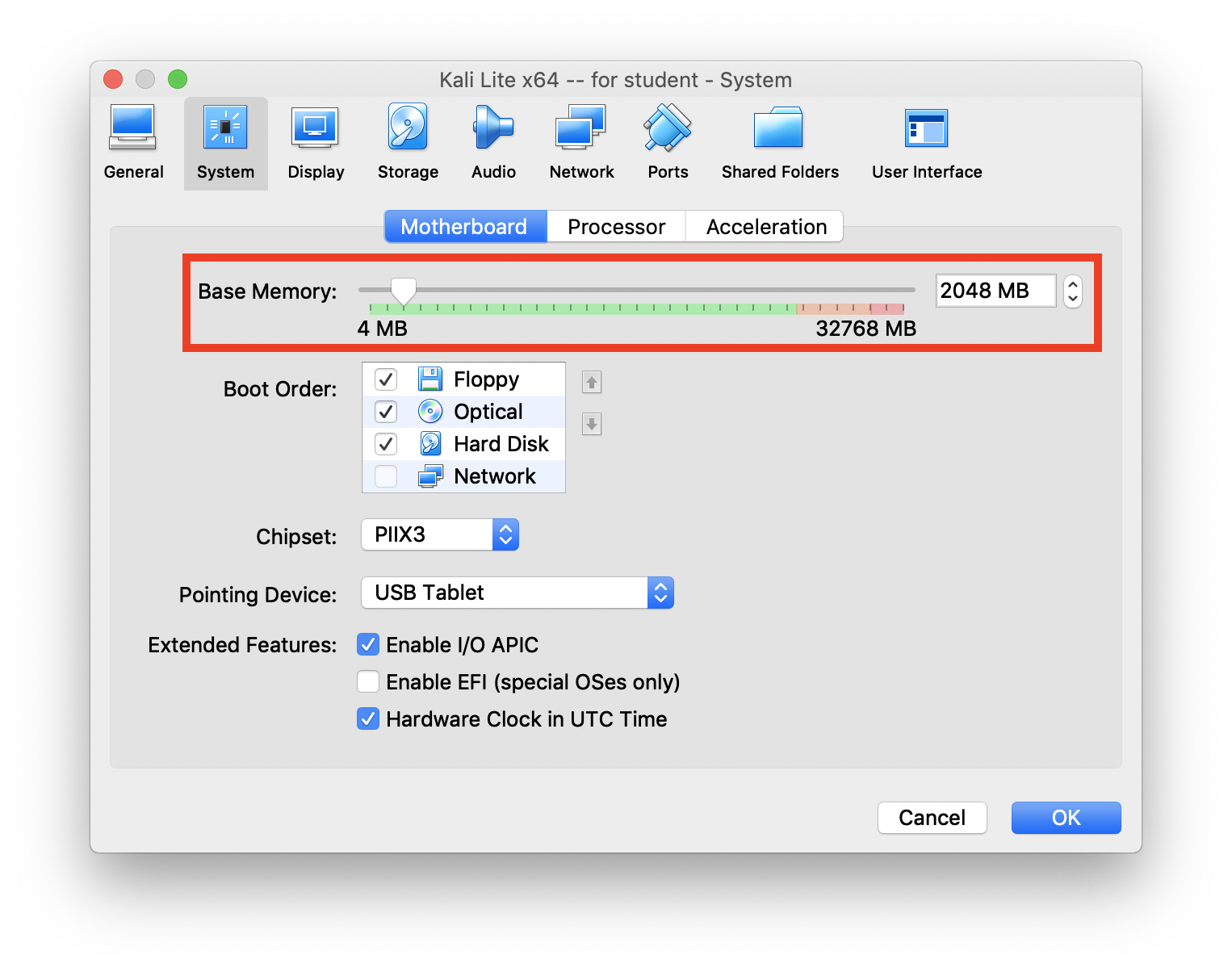


Under the “Network” pane, make sure that “Adapter 1” is set to “NAT Network” with the name “infosec-net”:

Similarly, make sure that “Adapter” is set to “NAT”:



1. Under the system pane, set the “Base Memory” to at least 2048 MB of RAM.



You can check how much RAM you have installed on your laptop on Windows searching for and running the “System Information” app, and look at how many gigabytes are listed next to “Installed physical memory.”

On a Mac, you can check how much RAM you have installed on your laptop by going to the  menu, selecting “About this Mac,” and look at how many gigabytes are listed next to “Memory.”

1. Start the imported Kali VM appliance.
2. The username is “root” and the password is “toor” (“root” backwards).
3. You can now delete the “.ova” file if you’re low on disk space.

**Part 1: Testing Passwords**

1. Visit the following URL:

<https://lowe.github.io/tryzxcvbn/>

1. Try out different passwords to see how strong they are by looking at the statistics under “guess times.”

**Optional**: If you want to learn more about password strength estimation, see this video and paper:

<https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler>

**Part 2: Checking an Account for an Existing Data Breach**

Check to see if one of your online accounts has already been breached. Visit: <https://haveibeenpwned.com>

Type in one of your email accounts or usernames to see if it has already been compromised in a data breach.

Next visit: <https://haveibeenpwned.com/Passwords>

Try out some passwords to see if they have already been compromised in a data breach.

Finally, visit: <https://haveibeenpwned.com/NotifyMe>

Sign up to be notified when one of your accounts is breached in the future.

**Question**: Was one of your accounts breached? If so, which one?

**Part 3. Sign-up for Two Factor Authentication**

Visit <https://twofactorauth.org> and browse through the categories to find an online service that you regularly use (e.g., Gmail, Snapchat, Instagram, Facebook, etc.). Click the  icon in the “Docs” column to learn how to set two factor authentication for that service.

**Question**: What is the name of the service you set up for 2FA?

**Part 4: Online Password Attack**

1. Open a terminal window in your Kali VM, and type cd /usr/share/wordlists.
2. The ls command displays which files are in this folder.
3. Unzip the rockyou.txt.gz wordlist:

gunzip rockyou.txt.gz

1. Type the ls command to see that rockyou.txt.gz is now rockyou.txt.
2. Type wc -l rockyou.txt to see how many entries are in this password dictionary file (‘-l’ contains a lower-case ‘L’).

This file comprises all unique passwords from the 32 million RockYou password breach you read about in your reading.

1. Launch an online password attack using THC-Hydra. Type the following (all on one line):

hydra -V -l istheory -P rockyou.txt https-get://is.theorizeit.org/auth/

**Note:** The trailing slash (‘/’) is needed.

Where:

* + hydra is the password cracking tool to execute
  + -V means verbose, and will show you the username and password combination being attempted
  + -l istheory sets “istheory” as the login name. Note: that’s a lowercase ‘L.’
  + -P rockyou.txt is the password dictionary file to use.
  + https-get means a GET request over HTTPS. Note that Hydra supports many protocols (e.g., ftp, ssh).
  + is.theorizeit.org/auth/ is the password-protected URL to be accessed.

**Note:** you can also use THC-Hydra with web forms: <http://insidetrust.blogspot.com/2011/08/using-hydra-to-dictionary-attack-web.html>

**Question:** What was the password (Scan the results to find the line beginning with [443][http-get])?

**Question:** Approximately how many passwords a second were you able to try? **Hint:** You may need to calculate this from the start and end time along with number of guesses made.

**Part 5: Offline Attack Using Hashcat**

**Note**: If you want, you can install Hashcat on your host computer, where you'll get massive speed improvements. Hashcast needs to be able to directly interface with the CPU/GPU in order to perform at a high level—it can’t so as well in a virtualized environment.

For Windows or Linux, download Hashcat here: <https://hashcat.net/hashcat>. For Mac, I recommend that you first install Homebrew (follow instructions at <https://brew.sh>), then run the command, “brew install hashcat”.

**Cracking in Hashcat**

1. On your Kali machine, open up a terminal and navigate to your home directory (/root/).
2. Hashcat uses a command-line interface. To see available options and syntax, type:

hashcat --help

1. Download a copy of “office2john.py” and “hashcat.doc” by entering the following terminal command:

wget <https://anthonyvance.com/files/hashcat.doc>

wget <https://anthonyvance.com/files/office2john.py>

1. First, a dictionary attack will be performed against a password-protected Word document. The following command uses a Python script to obtain the hash of the Word document password:

python office2john.py hashcat.doc

1. In the output you’ll see the name of the file followed by the type. The type is shown with a $ at the beginning and end of it. You’ll need to copy the type and everything until “:::”. For example, the hash looks like the following, all on one line:

$oldoffice$1\*04477077758555626246182730342136\*b1b72ff351e41a7c68f6b45c4e938bd6\*0d95331895e99f73ef8b6fbc4a78ac1a

Save the hash into a file in your home directory.

Note: make sure the entire hash is on one line within the text file. **Don’t add extra spaces at the end.** If you get a “line-length exception” in the next step, make sure there’s not a typo in the beginning of the hash.

1. While still in your home dir, run the following command (all on one line). Reference the hash file you just created and choose an arbitrary name for an output file. Once the password is cracked, you will read your output file to see the cracked password. It will be appended to the end of the hash following a colon (:) symbol.

**Note:** In the commands below, the “<< >>” notation means to replace that text with the names of the actual files you are using.

hashcat --force -a 0 -m 9700 --potfile-disable -o <<outputFileName.txt>> <<HashInputFileName.txt>> /usr/share/wordlists/rockyou.txt

Or alternatively, if you prefer to do it without making an input file, put the hash string right in the terminal, surrounded by single quotes:

hashcat --force -a 0 -m 9700 --potfile-disable -o <<outputFileName.txt>> '[hash string]' /usr/share/wordlists/rockyou.txt

Where the switches correspond to:

* --force This is necessary to get hashcat to run in a VM environment (it doesn’t normally like to).
* -a 0 Straight dictionary attack against the hash
* -m <<Office\_Flag>> The corresponding flag for the version of Office in use (see table in “hashcat –help”)
* --status: Provides an update of the status of the process without giving a prompt
* -o <<Output\_File>>: The location where the cracked hashes will be saved. The results will also be saved on the .pot file, unless otherwise specified. In our case, we disabled it.
* <<Hash>>: The saved password hash.
* <<Dictionary>>: The list of words that will be used to try and crack the password.

**Question:** What is the password for “hashcat.doc”?

**Question:** Roughly how many passwords per second can Hashcat try on a .doc file? Cracking speeds are in the following format:

H/s Hashes per second

KH/s Kilohashes per second (Thousands of hashes per second)

MH/s Megahashes per second (Millions of hashes per second)

GH/s Gigahashes per second (Billions of hashes per second)

TH/s Terahashes per second (Trillions of hashes per second)

PH/s Petahashes per second (Quadrillions of hashes per second)

**Note**: Your results would be orders of magnitude faster if you weren’t running Hashcat in a VM.

1. Refer to the [benchmark output for a Brutalis](https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40). (Running benchmark on your own machines is breaking on scrypt for some reason.). Units are expressed as (unit)H/s, or hashes per second, [thusly](https://hashcat.net/forum/thread-4908.html).

**Question:** How much slower is Hashcat in cracking .doc MS Office documents (option 9700, “<= 2003 $0|$1, MD5 + RC4”) compared to Office 2013 documents (option 9600)?

**Question:** How does an offline password attack compare with the online hydra attack you attempted earlier?

**Part 6. Cracking Linkedin Hashes Using Hashcat**

In this section, you’ll see how many hashes you can recover from the 2016 LinkedIn password breach. This breach of 177,500,189 unsalted SHA1 password hashes represents the data of all LinkedIn users as of 2012. Among these passwords, only 61,829,207 are unique.

However, in interest of your time, this section will require you to crack only 500,000 of these passwords. After you complete this lab, you’re welcome to crack all of the LinkedIn hashes. Ask me for a copy.

1. Download a copy of the file LinkedIn\_HalfMillionHashes.txt (also available [here](https://raw.githubusercontent.com/deargle/deargle.github.io/master/security-assignments/LinkedIn_HalfMillionHashes.txt)), all on one line:

wget https://anthonyvance.com/files/LinkedIn\_HalfMillionHashes.txt

1. Open a terminal. To get your feet wet, perform a straight dictionary attack using the rockyou.txt wordlist again, as follows (one line):

hashcat --force -m 100 --potfile-disable --remove --outfile=LinkedIn\_cracked.txt LinkedIn\_HalfMillionHashes.txt /usr/share/wordlists/rockyou.txt

Note: This command may take 5–10 minutes to run. To see the status of a running job in Hashcat, press the “s” key (it might take up to 15 seconds for Hashcat to report its status).

**Note:** These commands use the --remove flag. This will remove cracked hashes from the input file. So, if you run these commands more than once without changing anything, it won't crack anything after the first time.

Hashcat will report how many passwords it “recovered” when it finishes. You can always count the number of lines in your outfile (LinkedIn\_cracked.txt) to see how many you’ve cracked so far, total:

wc -l LinkedIn\_cracked.txt

Or count the number of passwords left (it started with half a million):

wc -l LinkedIn\_HalfMillionHashes.txt

To see hashes cracked in real time, in another terminal shell, type the command: tail -f LinkedIn\_cracked.txt. Type control+c to exit the tail command.

**Question:** How many passwords were you able to recover using this command?

1. Run another attack that uses a rules-based method (one line):

hashcat --force -m 100 --potfile-disable --remove --outfile=LinkedIn\_cracked.txt LinkedIn\_HalfMillionHashes.txt -r /usr/share/hashcat/rules/best64.rule /usr/share/wordlists/rockyou.txt

Rules apply common patterns to password dictionaries to crack even more hashes. You can read about rules in Hashcat here: <https://hashcat.net/wiki/doku.php?id=rule_based_attack>.

The “best64.rule” is one of the most effective sets of Hashcat rules. It is continually refined using input and testing from the password cracking community. You can view the contents of the best64.rule here:

<https://github.com/hashcat/hashcat/blob/master/rules/best64.rule>

You can read an explanation of these set of rules here:

<http://kaoticcreations.blogspot.com/2011/09/explanation-of-hashcat-rules.html>

**Question:** How many additional passwords were you able to recover using this rules-based attack?

**Optional:** Experiment with other rules found in /usr/share/hashcast/rules.

1. Run another attack that uses a hybrid method that uses a dictionary attack combined with a “mask,” which is a pattern that is appended to each password in the password dictionary:

hashcat --force -m 100 --potfile-disable --remove --outfile=LinkedIn\_cracked.txt LinkedIn\_HalfMillionHashes.txt -i -a 6 /usr/share/wordlists/rockyou.txt ?d?d

The “?d?d” at the end means to append two digits between 0–9 each at the end of each password in the rockyou.txt password dictionary.

**Question:** How many passwords were you able to recover using this hybrid attack?

If you would like to try using a different character set for your mask, you can use the following masks below. Note that each mask below is for one character. If you wanted to test four digits at the end of each password, the mask would be: ?d?d?d?d.

?l = abcdefghijklmnopqrstuvwxyz

?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ

?d = 0123456789

?s = !"#$%&amp;'()\*+,-./:;&lt;=&gt;?@[\]^\_`{|}~

?a = ?l?u?d?s

?b = 0x00 - 0xff

**Optional:** Another common password pattern is to prepend digits at the beginning of passwords. If you would like to try this mask, run the following command:

hashcat --force -m 100 --potfile-disable --remove --outfile=LinkedIn\_cracked.txt LinkedIn\_HalfMillionHashes.txt -i -a 7 ?d?d /usr/share/wordlists/rockyou.txt

**Optional**: Want even more practice? You can download the massive Troy Hunt "Have I Benn Pwened" SHA1 password hash list on the bottom of [this page](https://haveibeenpwned.com/Passwords).

Also, see [Daniel Miessler's wordlist collection](https://github.com/danielmiessler/SecLists/tree/master/Passwords) for more wordlists besides rockyou to try.

**Part 7. Secure Password Hashing**

Refer to the [benchmark output for a Brutalis](https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40).

**Question:** How much slower is Hashcat in cracking bcrypt hashes compared to SHA1 hashes?

**Question:** Would it be feasible to crack the passwords of the LinkedIn breach if LinkedIn had used bcrypt instead of SHA1? Why or why not?

**Part 8. Create a Targeted Wordlist Using CeWL**

CeWL (Custom Word List Generator) is a command-line tool that creates custom wordlists from a target website. This can be useful for cracking a password of an organization or individual that also has a website or social media profile. Because people often use information about themselves or their organization when creating passwords, custom wordlists can be very effective.

Imagine that you exfiltrated the following MD5 hash from a database on neurosecurity.byu.edu:

**cf4aff530715824c055892438a1ab6b2**

**You want to create a custom dictionary using the words on neurosecurity.byu.edu to see if you can crack the hash.**

1. Create a custom dictionary using CeWL for the website neurosecurity.byu.edu:

cewl -v -d 2 -m 5 -w custom\_dict.txt https://neurosecurity.byu.edu

Where:

* “v” runs CeWL in verbose mode.
* “d” is the depth to “spider” or crawl the website
* “m” is the minimum word length
* “w” “custom\_dict.txt” is the name of your new custom wordlist or dictionary.

Give the command a minute or two to complete.

1. Check how many entries are in the custom\_dict.txt file:

wc -l custom\_dict.txt

1. Look at the words in custom\_dict.txt:

less custom\_dict.txt

1. Permute the words in the custom\_dict.txt wordlist using the “best64” rule, and append the output to custom\_dict.txt (all one line):

﻿hashcat -a 0 custom\_dict.txt -r /usr/share/hashcat/rules/best64.rule

--stdout >> custom\_dict.txt

1. Check how many entries are in the custom\_dict.txt file now:

wc -l custom\_dict.txt

1. Run Hashcat using custom\_dict against the MD5 hash (all one line):

hashcat --force -a 0 -m 0 cf4aff530715824c055892438a1ab6b2 custom\_dict.txt

**Question:** What is the plaintext of the hash?

**Learn more:**

|  |  |
| --- | --- |
| /var/folders/6z/h6v1p8vj7835y4fxzqq3jdch0000gn/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/31nyO6DhPAL._SX322_BO1,204,203,200_.jpg | “Hash Crack: Password Cracking Manual (v2.0)” by Joshua Picolet.  A password cracking reference guide written by one of the world’s foremost professional password crackers.  Amazon: <http://a.co/d/fTtI7DF> |
| * <http://arstechnica.com/security/2013/03/how-i-became-a-password-cracker/> * Official Hashcat documentation: <http://hashcat.net/wiki/> | |