Name: _Madina Martazanova____ Date: _02/18/2019___

Computer and Network Security

Lab 2: Linux Access Control and Encryption

P Drive. All students, faculty, and staff have access to 1 GB of data storage (**P** drive) while using the lab computing facilities at The College at Brockport. This service is intended to provide a convenient, temporary, data storage area. On lab computers, an icon appears on the desktop. If you are not on lab computers, the files in the **P** drive may be accessed through the web at the URL: http://filecity.brockport.edu

For this assignment, you need to use a Microsoft Windows based personal computer. Some parts of the assignment require downloading and installing new software but does not necessarily need administrative privileges. Such software may be installed on your P: drive. Thus, any computer in a computer classroom such as Drake 044 may be used. However, some software will require installation on C: drive under Program Files and hence you need access to a computer with **administrative privileges**.

1. Linux Access Control

Use **ssh** to log into the **holly.brockport.edu** server. Under your login directory, there is a folder named public_html. NOTE: **stay under your login directory and not move to other folders.**

a. Under your login directory, use ls to display the lengthy information of the directory public_html only (no any other entries are shown).

Command: ___**ls –ld** ___

Results: _**drwx--x--x 16 mmart7 students 4096 Oct 13 10:**51 .

b. Create a folder named ACL. Show its lengthy information as the only entry. **ls –ld ACL**

_ Results: __ **drwx------ 2 mmart7 students 4096 Feb 18 12:41 ACL**

c. Use umask to change the default access permission as the following requirements -- user: read and write; group: read; others: read

Command: ___**umask 133** ___

Delete the folder ACL and recreate it. Show its lengthy information as the only entry.
Command: **rm -r ACL**

Results: drw-r--r-- 2 mmart7 students 4096 Feb 18 12:53 ACL

Look at the access permission. Does it meet the requirements? ___**yes**_____

d. Keep the login directory as your current folder. Create an empty file named 1.html under ACL and assign the read and execute permissions to user, group and o/thers.

Name: _Madina Martazanova____ Date: _02/18/2019___

Command: __**touch ~/ACL/1.html**___
chmod 555 ~/ACL/1.html

e. Go to web browser and access this below URL and observe what has happened. www.itss.brockport.edu/~mmart7/ACL/1.html

Results: ___ **You don't have permission to access /~mmart7/ACL/1.html on this server.**
__

If it does not display, how to solve the problem? Your solution:

___We have to change access permission _to 755, the page was then able to display____

f. Keep the login directory as your current folder. Create a directory named test2 under public_html and assign the read and write permissions to user only (you) and no any permissions to group and others.

Command: __**mkdir test2**___
chmod 600 test2

g. Keep the login directory as your current folder. Copy 1.html from the folder test1 to the folder test2.

Command: _____ **cp ~/ACL/1.html ~/test2/1.html** _____

If it does not success, can you list the content in test2?

Results: ___**total 0**__

How to solve the problem with the minimum risks? Your solution:

We need to give the owner of the test2 execute permission as well, chmod 700 test2. This will allow only the owner to create directories and files, but restricts Group and others.

h. Copy a file from ~nyu/public_html/index.html to your ACL folder. Keep the file name. Use an appropriate editor tool to change all information to your own information, including page title, name, email, and major. Use the appropriate URL provided above to display your correct information.

The command for copying: **cp ~nyu/public_html/index.html ~/ACL/index.html**

The URL to check if the page is changed correctly: **http://www.itss.brockport.edu/~mmart7/ACL/index.html**

Name: _Madina Martazanova____ Date: _02/18/2019___

i. Use **stat** to display the Access time, the Modification time and the Change time.

Command: **stat index.html**
Results:

```
File: 'index.html'
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 811h/2065d          Inode: 4329616     Links: 2
Access: (0700/drwx------)  Uid: (91066/  mmart7)   Gid: (  405/students)
Access: 2019-02-26 18:53:34.018872596 -0500
Modify: 2019-02-26 18:53:34.018872596 -0500
Change: 2019-02-26 18:53:34.018872596 -0500
 Birth: -
```

j. Use **stat** for index.html to display the Access time, the Modification time and the Change time.

Command: _____**stat index.html**

k. After the user types and executes some common commands, such as cat, the three times of the file can be changed.

What operation/command is used to make the file's Access time changed only?__**touch -a**_

What operation/command is used to make the file's Change time changed only?__**touch -r** __

What operation/command is used to make the three times changed to a same value?_**touch -c**

l. Use the manual of touch and find out how to use touch to manually change the access time to a particular value.

Command to look at the manual: ___**man touch**___

Take an example to manually revise the file's access time only:

____ **Touch –a 1.html__**

2. Hashes and File Integrity

For this part, you will work on the Kali Linux virtual machine set up in Assignment 1.

We remarked earlier that relying on the hash MD5 to verify the integrity of a downloaded file is discouraged. Are there problems? Let us explore it.

a. Start the Kali virtual machine you had set up in Assignment 1. Login and open the Firefox browser. Access http://www.itss.brockport.edu/~nyu/security/ and download the file **Hello1**.

Copy it to the **~/Documents** folder. Also, download the file **hello2**. Copy it also to the **~/Documents** folder.

b. To determine the MD5 hash for the two files, open a terminal window. Change directory to **~/Documents**. To determine the MD5 hash, type the command:

```
md5sum hello1 hello2
```

c. Do the two files have identical MD5 hash value?

da5c61e1edc0f18337e46418e48c1290 hello1
da5c61e1edc0f18337e46418e48c1290 hello2

Yes, the two files have the same hash values.

d. Ensure that it reads as below (where 8 hex digits have been deleted)

```
-- -- 61 e1 ed c0 f1 83 37 e4 64 18 e4 8c -- --
```

Provide the missing digits: **__da5c … 1290_____**

e. You must find that the two files have the same MD5 hash. On the basis of this, what do you conclude? Do these two files have the same executable content but just different names?

The two files having identical hash values is not likely, the hash value has probably been altered in one of the files to be passed off as harmless.

f. To execute these files, first change the permissions by typing

```
chmod 700 hello1 hello2
```

g. Execute the first program by typing the command: **./hello1**
What do you observe?

Output: Hello World.

h. Now execute the second program by typing the command: **./hello2**
What do you observe?

This program is evil!!!
Erasing hard drive…1Gb…2Gb… just kidding!
Nothing was erased.

i. On the basis of what you observed, what do you conclude? Do these two files have the same executable content but just different names?

Name: _Madina Martazanova____ Date: _02/18/2019___

Based on the execution of the second file, these two files do not have the same executable content.

j. Explain the security implications of this MD5 collision.

MD5 hash values can be modified to look like those of trusted files, so the malicious file can appear to be a copy of a trusted file.

k. Determine **`SHA1`** hash of the two files (using the command **`sha1sum`**).

SHA1 for hello1: **8f42c29f6ac45423d2a7dd614d666a26e39f29ee hello1** (**sha1sum hello1)**

SHA1 for hello2: **dfce366c23c88044ad57a5eaa7d5420024a7fd14 hello2 (sha1sum hello2)**

l. On the basis of this, what do you conclude? Do these two files have the same executable content but just different names?

No,They don't have the same executable content.

m. Use **od** to view the binary content of hello1 and hello2. Commands:

od hello1
od hello2

n. Use **diff** to view the difference between hello1 and hello2. Command:

diff hello1 hello2

(they're both different)

o. Use **hexdump** to convert the binary file. (hexdump helloX > helloX.txt) Commands:

hexdump hello1> hello1.txt
hexdump hello2 > hello2.txt

p. Use **diff** to view the difference between the text files. Command:
diff hello1.txt hello2.txt

q. Observe the diff results between the two text files and write down where they are different?
Diff

118,120c118,120 *//Lines 118 – 120 differ in hello1.txt*

122,124c122,124 *//Lines 122-124 differ in hello2.txt*

r. Sum up the binary numbers in different codes and see if their sums are same?

X = 120c118 (Hex) = 1 0010 0000 1100 0001 0001 1000 (Bin) = 18923800(dec)
Y = 124c122 (Hex) = 100100100110000010010010 (Bin) = 19185954(dec)

Therefore, X != Y.
Alternatively, X xor Y != 0.

3. Cracking Linux Passwords with John and Johnny on Kali

a. Open a terminal window On Kali. Kali uses **`/etc/passwd`** and **`/etc/shadow`** files to keep track of user credentials. An inspection of **`/etc/shadow`** shows that for all but one account login is prevented with an **`*`** or **`!`** placed in the second field.

The only account that permits login is: **_root__**

b. Add an account for yourself, so you do not have to login as **`root`** all the time. Replace **`jdoe1`** with your Brockport login ID in the example below:

```
useradd –m jdoe1 –G sudo –s /bin/bash
```

The **`–m`** option creates a home directory for the user as **`/home/jdoe1`** and the **`–G`** option allows the user to join the **`sudo`** group. The **`–s`** option sets the login shell.

Set the password for the user by typing the command: **`passwd jdoe1`**

You will be prompted for the password. Type **pa$$w0rd**. Confirm again.

c. Add an account for yourself with your first name as the username. This account is not included in the **`sudo`** group.

Command used: **useradd –m madina –s /bin/bash**

Set your password as **`qwertyui`**

d. Type the following two commands to check the **`/etc/passwd`** and **`/etc/shadow`** files.

```
tail –n 2 /etc/passwd /etc/shadow
```

e. In order to crack the password information, we need to add the hashed password information to the **`/etc/passwd`** file. Type the following command:

```
unshadow /etc/passwd /etc/shadow >~/Documents/passwd.txt
```

f. Move to the **Documents** directory. Display **passwd.txt** and see what it contains. Since we are only interested in the first line (corresponding to root) and the last two lines that were added by us, you may edit the **passwd.txt** file, remove all other lines, and save the edited file as **crack.hash**. Alternatively, you may type the following two commands:

```
head –n 1 passwd.txt >  crack.hash
tail –n 2 passwd.txt >> crack.hash
```

You will be left with exactly three lines in **crack.hash**.
You may check this by typing the command: **wc –l crack.hash**

g. We are ready to crack the passwords in **crack.hash**.
Type the command: **john crack.hash**

h. Abort execution after five minutes. At this stage only one hash remains to be cracked.

Remaining hash to be cracked corresponds to the account: **mmart7**

i. Open the application **Johnny** which has a graphical user interface. Open the password file **crack.hash**. You should see the three lines of the password file. Uncheck the lines corresponding to the accounts for which the passwords have already been determined and displayed, leaving only one line checked.

j. At this stage, pretend as a hacker who is trying to guess the password, perhaps knowing some personal information about the account holder – car driven, spouse or children's names, favorite sports team, etc. Press **Guess password** and enter the guess. The application would check that and would respond if that is indeed correct. Keep trying various possibilities and eventually enter **pa$$w0rd** as a guess, at which stage Johnny will accept it.

k. Return to the command window and type the command: **john --show crack.hash**

Show the output below:

root:toor:0:0:root:/root:/bin/bash
mmart7:pa$$w0rd:1000:1000::/home/mmart7:/bin/bsh
madina:qwertyui:1002:1002::/home/madina:/bin/bash
3 password hashes cracked, 0 left

l. Access http://www.itss.brockport.edu/~nyu/security/ and download passwd1.txt. Now you are ready to crack the password file.

m. While some of the easy passwords will be cracked within seconds, others may take several hours or may be even days. Run the program sufficiently long, until you crack at least 8-9 passwords, if not all.

Name: _Madina Martazanova____ Date: _02/18/2019___

Name: _Madina Martazanova____ Date: _02/18/2019___

List of cracked passwords

	Username	Plaintext password	Order in which cracked
1.	**root**	w1zaed	2
2.	**john**	wizard	1
3.	**tom**	tomkatz	1
4.	**helena**	helena	**1**
5.	**kristi**	letmein	1
6.	**lisa**	12345678	1
7.	**steve**	abc123	1
8.	**arlette**	Phoenix9	1
9.	**kathy**	sk8board	**3**
10.	**doug**	crabcola	4

Comment Summary