Dr. Palvi Aggarwal

CS 5352/6352: Computer Security
Assignment 2
Total Points: 100 Points
Submission Date: 02/15/2024 at 11:59 pm

Name _____
Student ID _____

Instructions: Please carefully read the following instructions.

- Please make sure that you add your name and student ID as part of your assignment.
- Please include detailed description of your observations and finding in the report.
- Please submit your report as PDF and your code as a zipped folder.
- Please use the following naming convention for submission: Name_AssignmentNumber

Lab environment: This lab has been tested on our pre-built Ubuntu 20.04 VM, which can be downloaded from the SEED website.

Question 1: Please answer the following questions.

a. **(5 points)** List all the attributes that are stored for an individual user in a shadow file.
b. **(5 points)** Where do the passwords in Windows system get stored?

Question 2: Please perform the following tasks.

a. **(15 points)** Write a well-documented Python program that creates 30 users and uses passwords from the commonPasswdFile.txt (provided in the Assignment 2 folder). Display the contents of Password file and Shadow file. Which algorithm has been used to store the passwords?
b. **(15 points)** Write a Python program to compute the necessary hash of the passwords (commonPasswdFile.txt) to compare them with the passwords stored in shadow file. The program should output all the username: password combinations after successful crack. Please list all the cracked combinations in the report and provide how long it took to crack the password with and without the salt.
    - **Challenge:** Change the user passwords to a more complex one. Provide how long it took to crack the password with and without the salt.

Question 3: Please answer the following questions.

a. **(5 points)** Please provide a step-by-step demonstration of how to Set-UID programs can be used to display contents of shadow file for seed user.
b. **(5 points)** What is real User ID (UID) and Effective User ID (EUID)? How does Set-UID programs affect EUID? Please explain with an example.

Question 4: Please download the Labsetup-Assignment1.zip to perform the following tasks.
Note: You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or

surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits.

(20 Points) Task 1: Manipulating Environment Variables
In this task, we study the commands that can be used to set and unset environment variables. We are using Bash in the seed account. The default shell that a user use is set in the /etc/passwd file (the last field of each entry). You can change this to another shell program using the command chsh (please do not do it for this lab). Please do the following tasks:

- Use printenv or env command to print out the environment variables. If you are interested in some environment variables, such as PWD, you can use "printenv PWD" or "env |grep PWD".

- Use export and unset to set or unset environment variables. It should be noted that these two commands are not separate programs; they are two of the Bash's internal commands (you will not be able to find them outside of Bash).

(20 Points) Task 2: Passing Environment Variables from Parent Process to Child Process
In this task, we study how a child process gets its environment variables from its parent. In Unix, fork () creates a new process by duplicating the calling process. The new process, referred to as the child, is an exact duplicate of the calling process, referred to as the parent; however, several things are not inherited by the child (please see the manual of fork () by typing the following command: man fork). In this task, we would like to know whether the parent's environment variables are inherited by the child process or not.

a. Please compile and run "myprintenv.c" and describe your observation. Because the output contains many strings, you should save the output into a file, such as using a.out > child (if a.out is your executable file name).
b. Now comment out the printenv() statement in the child process case and uncomment the printenv() statement in the parent process case. Compile and run the code again and describe your observation. Save the output in another file.
c. Compare the difference of these two files using the diff command. Please draw your conclusion.

(20 Points) Task 3: Environment Variables, execve() and system()
In this task, we study how environment variables are affected when a new program is executed via execve(). The function execve() calls a system call to load a new command and execute it; this function never returns. No new process is created; instead, the calling process's text, data, bss, and stack are overwritten by that of the program loaded. Essentially, execve() runs the new program inside the calling process. We are interested in what happens to the environment variables; are they automatically inherited by the new program?

a) Please compile and run "myenv.c" and describe your observation. This program simply executes a program called /usr/bin/env, which prints out the environment variables of the current process.
b) Change the invocation of execve() in the code to the following line and describe your observation. execve("/usr/bin/env", argv, environ)
c) Please draw your conclusion regarding how the new program gets its environment variables.

d) How would you use system() function to do the above task? Please explain the difference between system() and execve() system calls.