CS-C3130 Information Security (2021)
Exercise 1

OS Security and Access Control

The exercises are at https://infosec1.vikaa.fi/. Register with your Aalto University student number and email address and log in. Open the first exercise and launch the virtual machines.

Always store a copy of your best solutions on your own computer. The virtual machines may be reset at any time. Also, you may need them in case the exercise platform suffers data loss.

In this assignment, you will learn about access-control mechanisms and user authentication in operating systems. Everyone is encouraged to complete part A and as much of B and C as you can.

Part A: Setting up SSH authentication (10 p)

The goal of this part is to learn to use `ssh` for remote login to Linux and Unix computers. You will configure `ssh` to work with public-key authentication, without a password.

You are encouraged to find a practical tutorial on SSH online and to follow it. We will learn more about public-key cryptography and security protocols in the lectures later in the course.

1. Log into the virtual machine *SSH Server* with `ssh`. The command and password are shown in the exercise launcher.
2. Create an *ed25519* key pair for yourself in SSH. Configure your account on the server so that you can use `ssh` with public-key authentication to log in. Test that you can log in without entering the password. Also check that the related file permissions in the folder `.ssh` and its contents on the server are correct.

After completing the exercise, click on *Submit* in the exercise launcher. Your solution will then be marked.

Note that the private key must never leave your own computer. Only copy or upload the public key to the cloud.

Advice for Windows users: Windows 10 has a built-in command-line SSH client, which needs to be enabled. However, Ubuntu on Windows works far more smoothly. Do not use *putty*; even though it is very easy to download, it is unnecessarily difficult to use.

Part B: File access control in Linux (10 p)

You can find more information about Linux permissions in the lecture slides and by following the links there. There are also good explanations on the web. Before doing this exercise online, you can try it with Linux or MacOS on your own computer — assuming you are the administrator. If you are a Mac or Windows user, it is a good idea to install Linux on VirtualBox in your computer and keep it for the future. It is not required for completing this exercise, but it will be useful throughout your studies.

In the following, you are the system administrator and Alice is a project manager.

1. Connect to the *SSH Server* virtual machine. Note that `cloud-user` has root access with the `sudo` command.
2. Create three users in the system: `alice`, `bob` and `carol`. Add `alice` and `bob` into a group called `project2021`. Do not give any of these users root access.
3. Alice has asked you to do the following:
   (a) Someone has already uploaded project files to the `cloud-user` home directory. Move `project` to `alice`'s home directory.
   (b) Make `alice` the owner of `project` and everything under it.
4. Set the access rights as follows:
   (a) Any logged-in user can view the `project` directory and its files but only members of `project2021` can modify them.
   (b) Only the members of `project2021` can view and modify the `code` subdirectory and its files.

(c) Any logged-in user can run the project demo. The executable file is called `poc`.

(d) Only `alice` has access to the `confidential` directory and its contents.

After completing the exercise, click on *Submit* in the exercise launcher.

Hint: Learn how to use the following Unix commands: `sudo`, `man`, `ls`, `cd`, `mkdir`, `cp`, `mv`, `echo`, `cat`, `less`, `chown`, `chmod`, `adduser`, `addgroup`. Following the automated feedback is not a replacement for reading the instructions above. Also, find a solution that would work in practice. Impractical solutions will not earn many points in the exercise — any more than in the real world — even if you follow the instructions to the letter.

Additional information: Note that a corrupt project team member could add malicious functionality to the executable file and compromise the accounts of all users who run it. For example, can you think of a way for the project team member `bob` to get access to the documents in the `confidential` directory? In the same way, we have to trust all the executable applications installed on our personal computers and workstations. Mobile devices do not have the same problem because they implement better isolation between applications.

Part C (bonus problem). File access control in Windows (10 p)

For this exercise, you will need a Professional or Education version of Windows because `local groups` are not fully supported in the Home editions. Aalto students may want to upgrade to Windows 10 Education at no cost here. If you are a Linux or Mac user, you can install Windows into a VirtualBox VM and keep it around. You can also download ready Windows virtual machines (e.g. *MSEdge on Win10 (x64) Stable*) that are available to developers for testing with 90-day validity. A disposable VM or snapshot is the way to go if you do not trust scripts provided by the course staff, especially if you are using your employer's computer for the exercise.

Download the checker script from MyCourses to the Windows machine. Open the command prompt, go to a suitable *working directory*, and create the exercise files by running `wincheckaccess.exe /create`. Note that these files do no have any real content.

Create a new local user group `PROJECT`, users `Alice`, `Bob` and `Carol`. Set the permissions on the exercise files similarly to part B above, but in a way that is natural for Windows.

Some advice: Try to keep the settings as simple and understandable as possible. In Windows, the access rights are mainly managed through folders and inheritance, and it is usually unnecessary to set access rights on individual files. Do not use negative ACEs; instead, block inheritance on the folder. Most applications expect so-called generic permissions and may be confused by very fine-grained settings (e.g. *write* is expected to go together with *append* and the right to modify file attributes). Unlike in Linux, it is ok for a file to inherit execute rights even if the file itself is not an executable (*.exe*). An easy way to set the access rights is the Advanced Security Settings dialog, which you can open from the object's Properties dialog. The Effective Access tab and the Microsoft SysInternals tool `accesschk.exe` may be helpful for debugging your solution. Again, when reading the instructions, remember that the main goal is to find a practical solution that would be ok in a real workplace. Give yourself full control over everything so that you can administer the files (and so that you can run the check script).

For feedback and to submit your solution, run the script `wincheckaccess.exe /check <user_number>` while in the same working directory as earlier. When your solution merits some points, the checker will generate a token. Copy the token to the exercise launcher to claim your points.