**BLG 460E SECURE PROGRAMMING**

HW2

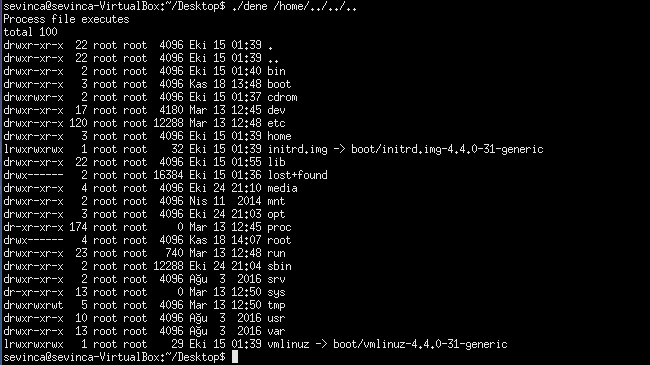
**Ahmet Göktuğ SEVİNÇ**

**150140120**

**13.03.2019**

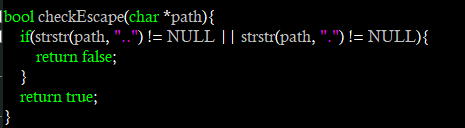
**Q1)**

In the first question we were asked to analyze the given code and determine how a attacker can skip validation performed in code. The code expects *“/home*” path from the user. To be able to compare if the valid path is sent or not, it just checks first 5 characters of the entered path and compares it with the “*/home*”. If they match, code considers path as a valid path. However, this validation process enables attacker to use paths that are not canonical. For example, attacker can use *“/home/..”* path. Since first 5 characters are *“/home”*, this path will pass the validation test but instead of accessing *home* directory, we will access to the *root* directory. Below image shows list of directories in *root* path.



Also, since we have access to the *root* directory, we can access any path in the file system. For example, we can use *“/home/../dev”* path or other paths inside *dev* directory.

To prevent directory traversal attacks, we can request user to enter canonical paths and prevent usage of other paths. To do this in given code, we can check if the path user entered is containing any escape characters such as “.., . “. If we add below code;



We obtain following result;



**Q2)**

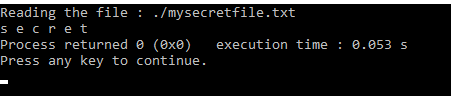
In the second question we were asked to analyze the code and explain and try to find a solution to the possible attacks. In the code, we have a file called “mysecretfile.txt” that we don’t want to be accessible. There are several ways to attack the written code and some of them are listed below:

* We can access the file by using “., ..” operators.

**Linux:**



**Windows:**

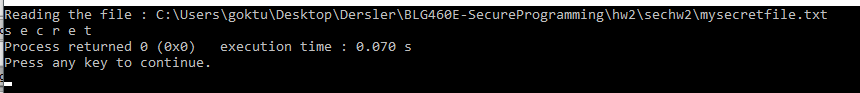


* We can access the file by giving its full path.

**Linux:**

**C:\Users\goktu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\q2-Linux1.png**

**Windows:**

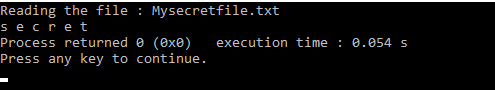


* On Windows operating system, reading file operation is not case sensitive. So, we can use that property to access the file but in Linux we cannot apply this operation.

**Linux:**



**Windows:**



To overcome those attacks, firstly I converted all letters of the given path to lowercase characters and then checked if the given path contains *mysecretfile.txt*. If it contains, I prevented to access to the path.

