November 29, 2020

Introduction to cyber security 156360

semester a 2020-2021

hw # \_5\_

machon tal english speakers

|  |  |
| --- | --- |
| Teudat zehut | Student Name |
| 209927128 | Tamar Harizy |
| 006798775 | Gila Odes |

**Question1**

**(a) “**Give two advantaged of capabilities over ACLs.**”**

Stored with the object, therefore, simple to use and see what is going on

Easy to determine & change rights to a resource making it user-friendly

**(b) “**Give two advantages of ACLs over capabilities**”**

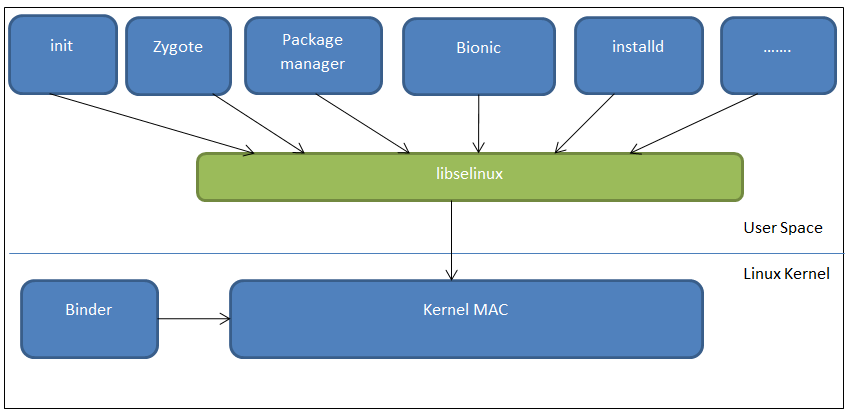
Easy to delegate

Easy to add/delete users (opposite of ACL where adding and deleting users is more combersome)

**Question 2 “Explain what kind of access policy is used by SEAndroid”**

SEAndroid (Security Enchancements for andriod) is an access policy that was initially built to allow for the use of SELinux in Andriod products. Nowadays, however, the SeAndriod has expanded their goal to be more than just an enabler of SELinux – but rather to be an overall framework for implementing SELinux’s Mandatory Access Control (MAC) and middleware access control (MMAC) on Android Devices. They do this by adding support to the Linux Kernel and allowing the user the space to do the following:

* protect their work from misuse by having a priviledege demons system
* using sandboxes to isolate apps from tehmainn system to avoid the case of malicious apps e=affecting the main system server
* prevent priviledge escalation by apps (as they are separate from the main kernel)
* It allows centralized control and analysis policies in main server
* allows applications to be controlled at both installation and at runtime using the MMAC policy

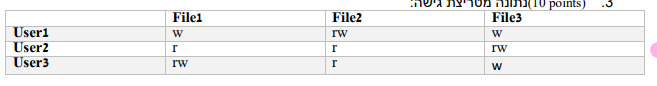


sources cited:

Image: <https://hsc.com/DesktopModules/DigArticle/Print.aspx?PortalId=0&ModuleId=1215&Article=66>

Info: <https://software.intel.com/content/www/us/en/develop/articles/android-security-customization-with-seandroid.html>

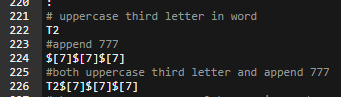
Question 3

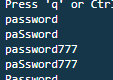


We propose that the model being used in the above table is that of the Bel LaPadula Model which has two basic principles that it follows: There is no read up and there is no write down. User2 has the highest access, therefore, any file he creates he has the ability to read and write to that file – but – others cannot read what he has written they can only write to the file – this is the case with File3. In File2 the fact that user1 can both read and write to the file shows that he has created the file. The user2 can read this file as he can read down but he cannot write to the file as it is of lower clearance and write down is not allowed. the same would go for user3 who must therefore, be of a higher clearance to user1. This is why with File1 which is evidentially created by User3 as he has both write ad read access, user2 can only read as the file is of lower clearance so no writing but user1 can write up to the file and cannot read it as his clearance is of a lower level.

**Question 4 \_ John the Ripper:**

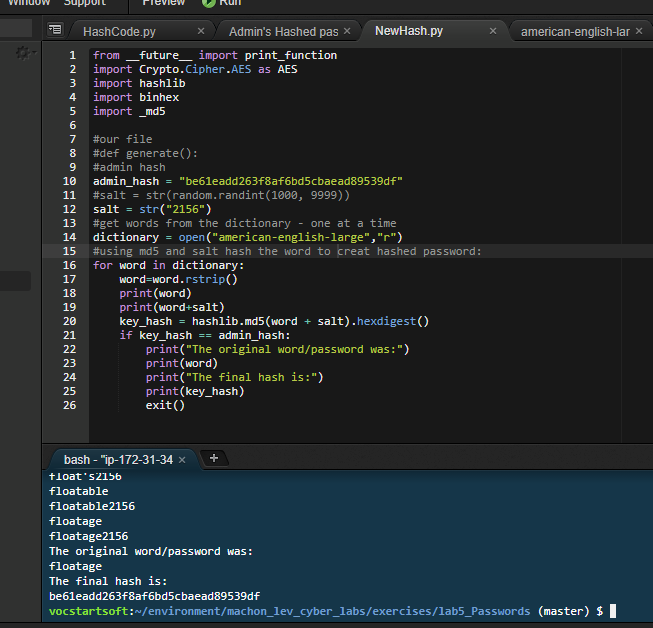
In order t answer this question we sent john a small dictionary with the word “password” to see what rules John-the-Ripper used. We then downloaded the rules of John onto our server and added the following commands:

This concurs with the suggestions given in the homework as to how to “crack” the hashed UserChallenge password (that the password has the 3rd letter capatalised and that 777 is appended to the end.) the results when tested on “password” where as follows:



We then saved the UserChallenge in a file on our server and asked john to crack the password using the american-english-large dictionary and the code command:

john ./UserChallange

**Question 5 \_ CTF Challenge:**

As one can see in the screenshot 🡪

We went through each word in the dictionary and hashed it using the md5 digest and the given salt. As soon as our hashed outcome was equivalent to the admin user’s hashed password the program stopped. We printed out each word and its hash value. Therefore, We were able to decipher that the password of the admin must be the last word printed before the program stopped which was “Floatage”. Hence, we have cracked the password using a brute-force technique.