

COM-402 exercises 2024, session 5:

Access control and authentication

Exercise 5.1

- Explain what is the most important principle for access control in IT systems.

Exercise 5.2

- What is the difference between ACLs and capabilities?

Exercise 5.3

- Describe an advantage and a disadvantage of RBAC.

Exercise 5.4

Linux systems store password hashes and other information about user accounts in a file called `shadow`. The file has the following access rights:

What could be the reason for the `setgid` to be set?

Exercise 5.5

When mandatory access control (MAC) is used to protect the integrity of a system:

- Can a subject write to objects on levels above or below it (write-up or write-down)? Explain.
- Give an example of how an operating system can use MAC to protect its integrity.

Exercise 5.6

Consider the Universal 2nd Factor (U2F) authentication system.

- Explain why the name of the visited website is added to the information signed by U2F.
- Explain why a U2F second factor is better than an OATH based one time password (OTP).

Exercise 5.7

- Is a biometric authentication system with a false acceptance rate of 0.01% a good system?

Exercise 5.8

What is the role of the AS and TGS in Kerberos? Why are they separate?

Exercise 5.9

- When you change your password on the website of Twitter, you can still access Twitter from your smartphone, without giving the new password. How is this possible?

Solutions to the Exercises

Solution 5.1

The most important principle is the *least privilege principle*. It mandates that each subject should only have the minimum set of privileges required for it to fulfill its function.

Solution 5.2

ACLs and capabilities both describe discretionary access rights that subjects have on objects. With ACLs, the rights are attached to the corresponding objects. Capabilities are rights that are attached to the corresponding subject (e.g. users or programs)

Solution 5.3

- RBAC makes it simpler to manage access rights.
- It is difficult to define roles that match the least privileges closely. So you either end up having many slightly different roles or roles that are too liberal.

Solution 5.4

The program can be run by any user (see the last x in `-rwxr-sr-x`). The `setgid` bit means that even when run by any user, it will run in the group `shadow`. This group has the right to read, but not modify, the file `/etc/shadow` (see the second r in `-rw-r-----`). Thus, this program could be used by any user for operations that only need to read the hashes but not modify them. For example, it could check a user's password, which is exactly what it is used for.

Here is a typical list of files with the `setgid` bit set on a Linux system.

```
find / -perm /g=s -group shadow -printf "%M %u %g %p\n" 2>/dev/null
root shadow -rwxr-sr-x /usr/bin/chage
root shadow -rwxr-sr-x /usr/bin/expiry
root shadow -rwxr-sr-x /usr/sbin/pam_extrausers_chkpwd
root shadow -rwxr-sr-x /usr/sbin/unix_chkpwd
```

Solution 5.5

- When protecting integrity, we want to prevent that subjects with lower integrity levels can write to higher levels. Subjects can not write to levels above them (no write-up). The goal is to preserve the integrity of the higher level objects.
- Processes that interact with the internet are assigned a low level of integrity. Only few directories are assigned to that level. All other directories have higher levels and can not be accessed by those processes. This is true even if the user owning the process has DAC rights to write into these directories.

Solution 5.6

- This makes sure that a fake website (e.g. paypal instead of paypal) can not play man-in-the-middle and ask you to sign a challenge it got from the original server.
- U2F uses signatures made with asymmetric keys. If the server were to be hacked, the attacker would only find public keys and could not impersonate the user.

Solution 5.7

The false acceptance rate (FAR) alone does not provide enough information to judge the quality of a biometric authentication system. For example, the same system could have a false rejection rate (FRR) of 80%, which would make it quite unusable.

(Biometric authentication systems can be judged by their equal error rate (EER) which is reached when the sensitivity is selected such that FER and FAR are equal.)

Solution 5.8

The role of the Authentication Server (AS), is mainly for *authenticating* the user, granting it the Ticket Granting Ticket. The role of the Ticket Granting Server (TGS), is mainly for *checking access control* allowed to the authenticated user. It grants the user a session ticket, could be used for specific services. This allows for reuse of the TGT if the user wants to access multiple service, enabling separation of concern.

Solution 5.9

Twitter uses OAuth2 to authenticate its users. When you installed twitter on your phone, you had to give the password that was valid at that time. The twitter client used OAuth2 with your user username and password to get an access token. That token is stored in your phone and gives access to Twitter independently of any password change.

If you go to the “settings and privacy” page of your twitter account you can find all active tokens on “Apps and sessions”. You can invalidate single tokens by clicking on “Log out the device shown” for each session.