COM-402 exercises 2023, session 3: Access Control and Authentication

Exercise 3.1

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

Exercise 3.2

- What is the difference between ACLs and capabilities?

Exercise 3.3

- Describe an advantage and a disadvantage of RBAC.

Exercise 3.4

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

```
$ls -l /etc/shadow
-rw-r--- 1 root shadow 1359 sep 24 22:13 /etc/shadow
```

- There is a program called `unix_chkpwd` that is owned by the user `root` and the group `shadow`. It has the setgid bit set:

  ```
  -rwxr-sr-x 1 root shadow 38912 fév 14 2019 /usr/sbin/unix_chkpwd
  ```

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

Exercise 3.5

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 3.6

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 3.7

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

Exercise 3.8

Imagine that an attacker is able to intercept the service ticket and the encrypted session key delivered by the TGS to a client.

- What are the mechanisms that prevent the attacker to use this information to obtain access to the service mentioned in the ticket?

Exercise 3.9

- Describe an attack that is possible in Kerberos if no pre-authentication is used.
- Is this type of attack completely avoided with pre-authentication?

Exercise 3.10

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

Solutions to the Exercises

Solution 3.1

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 3.2

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 3.3

- 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 3.4

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 3.5

- 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 3.6

- This makes sure that a fake website (e.g. playpal 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 note impersonate the user.

Solution 3.7

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 3.8

- The main protection is given by the encrypted session key. In order to obtain the service, the client needs to produce an authenticator when presenting the ticket. To create this authenticator the session key is required. The attacker only has the encrypted session key. To decrypt it, they would need the session key of the previous step (which is the user's password hash).
- Additionally, the ticket contains the IP address of the client. The attacker could only use the ticket if they were also capable of using the client's IP address.

Solution 3.9

Without pre-authentication, anybody can ask the AS for a ticket granting ticket and an encrypted session key. The session key of the TGT is encrypted with the password hash of the corresponding user. An attacker can then try to bruteforce the user's password by trying to decrypt the session key with the hashes of possible passwords.

With pre-authentication, the user has to prove that they know the password hash by generating an authenticator in order to receive a TGT.

The attacker can no more ask for some data that will allow them to crack the password. However, if they are patient, they can wait until a user logs in and requests a ticket. If they can observe the traffic, they can run the same attack or try to bruteforce the password from the authenticator. The attack is still possible, but the window of opportunity is much smaller.

Solution 3.10

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 cliking on "Log out the device shown" for each session.