Report

Configuration

On the server side, change the configuration for sending email in mailer.rs

```
pub fn send_mail(email: &String, subject: &String, token: &String) {
    let smtp_address = "smtp.gmail.com";
    let username = "PUT YOUR EMAIL HERE";
    let password = "PUT_YOUR_APPLICATION_KEY_HERE";

    let email = EmailBuilder::new()
        .to(email.to_string())
        .from(username)
        .subject(subject.to_string())
        .text(token.to_string())
        .build()
        .unwrap()
        .into();
```

Question

> What are the advantages of a challenge-response authentication compared to a weak authentication protocol?

With a challenge-response authentication, the password is sent only once at registration. This decreases the risk that a man in the middle intercepts the password.

> In your application, when do you require the user to input its Yubikey? Justify.

When registering to store the user's public key on the server side

During the authentification, if the 2FA is activate.

If 2FA authentication is enabled, we can require the yubikey to reset the password [not implementend for this homework]. Thus, if the user's email is hacked, an attacker can't reset the password.

> What can you do to handle Yubilkey losses?

As for the password reset, we can sent a token by email. The user can then enter this token to configure a new yubi key. In addition to the token, you can also require the password.

Another possibility is to ask the user to authenticate several keys, but the user only needs one key to authenticate. Thus, if the user loses his key, he can authenticate with another key he owns.

Reference : https://support.yubico.com/hc/en-us/articles/360013647620-Losing-Your-YubiKey

> An attacker recovered the challenge and the associated response (in the password authentication). How does this allow an attacker to perform a bruteforce attack? What did you implement to make this attack hard?

It can compute the tag with hmac on the challenge and compares it with the associated response.

It must, however, derive a key with argon2 from its password list or generate a key list.

This attack can be done locally on his computer.

The best way is to implement a password policy to make this attack more difficult.

No measures have been implemented in the lab to make the brute-force attack more difficult (Apart from using argon2 to derive the key).

> For sending the email, what are the advantages of using an application password?

The app password is easily revocable. In addition, in the context of a google account, a user password allows access to many more resource than the application password gives.

It can also be used to reduce access. For example, the one I was using was limited to a Windows device.

> In the Yubikey, what is the purpose of the management key, the pin and the puk?

- PIN

A PIN permits to perform actions such as creating a digital signature. It is a code that the user must enter.

The PIN is blocked if the user enters three incorrect PINS consecutively.

- Management key

The management key is required for PIV operations. This is a 24 bytes 2DES key. Nevertheless, it is possible to set a default key or "or you can choose to not set a Management Key, instead using the PIN for these operations."

- PUK

For the puk, it is clearly documented in the documentation : "The PUK can be used to reset the PIN if it is ever lost or becomes blocked after the maximum number of incorrect attempts. "

Unlike the PIN, if the PUK is blocked, then a complete reset is required

Reference : https://developers.yubico.com/yubikey-piv-manager/PIN_and_Management_Key.html