SEC: Lab#2 - Authentication

Author: Do Vale Lopes Miguel

Date: 28.05.2022

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

I can see two main advantages using this protocol:

- no need to reveal the password
- no replay attacks (when the challenge is a nonce)

Challenge-response protocols do not send the secret (aka the password) during authentication unlike weak protocols. If an attacker intercepts an authentication that is not encrypted, the recovery of the challenge and/or the response does not directly compromise the user password. Indeed, the attacker still needs to bruteforce using the pair challenge-response to recover the password.

The other advantage is the protection against replay attacks. In an encrypted communication, by intercepting the encrypted password in a weak authentication protocol, an attacker can replay the cipher to access the system. While in a challenge-response protocol, the attacker cannot replay a response because a new challenge is generated at each session. However, it is still possible to perform a MITM attack and usurp a session if the authentication is one-way.

Assuming the challenge is a nonce and the algorithms are adequate, the only critical point in the challenge-result protocol is the exchange of the shared secret.

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

The Yubikey must be input when registering the user because it is necessary to generate a private key in the yubikey (in the PIV) and get the corresponding public key to store it server-side.

The second time where I require the Yubikey is during the authentication of the user if the 2FA is activated. It is necessary to have access to the Yubikey in order to sign the challenge with the private key. The public key is then used server-side to verify the signature (i.e. the possession of the Yubikey with the corresponding private key).

What can you do to handle Yubilkey losses?

If the user still knows his password, we can imagine a mecanism to deactivate the 2FA by using a reset token like for the password reset. The user then inputs the reset token and is prompted to enter his password. The reset token is used as the challenge in the challenge-response protocol. If the result is valid, the server deactivates the 2FA.

Finally, all that remains is to add a private key changing mechanism where the user could input a new Yubikey to generate a new private key in the PIV and update the corresponding public key server-side when he re-activates the 2FA.

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?

An attacker can bruteforce the secret that is used to compute the response. The bruteforce is made until the computed response of the challenge corresponds to the recovered response. That is when he knows the secret.

To mitigate the attack, we derive the secret by using the hash of the password. In fact, Argon2id is a KDF which allows to inflict an important cost to the attacker. With the params used on this lab and with my computer, the attacker needs to bruteforce the hash wich takes approximately five seconds to compute. If the password is strong enough, the bruteforce is impracticable with current setup.

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

We don't need to disclose the password of our SMTP account. If the application password is leaked, we can quickly remove it from our account and generate a new if necessary.

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

The management of the PIV in a Yubikey requires a management key. By entering it, we can for example generate a new private key. There is a feature where we can use the PIN instead, but this is not recommended.

The PIN is used to authorize actions with the stored private keys in the PIV (if enabled when generating a private key). For example, we need to enter the PIN to sign data in order to process the 2FA in my application.

The PUK can be used to reset the PIN if it is blocked after too much failed attempts or is lost.

Source: PIN and Management Key