SEC: Lab#3

Author: Do Vale Lopes Miguel

Date: 13.06.2022

TLS

Client:

- Added server certificate as root certificate (certificate pinning)
- Removed the accept invalid certificates and accept invalid hostnames:

  ```
  .danger_accept_invalid_certs(true)
  .danger_accept_invalid_hostnames(true)
  ```

- Accepted built-in system certificates (the use of built-in certificates should be safe unless the user previously added a malicious certificate in the system)
- Change in the TLS accepted versions:
  - 1.2 as minimal version (for backward compatibility)
  - 1.2 as maximal version (1.3 is not available but is better)

Server:

- Change in the TLS accepted versions:
  - 1.2 as minimal version (for backward compatibility)
  - 1.2 as maximal version (1.3 is not available but is better)

Input validation

- Enforce server-side input validations whenever needed
- Centralized functions in file validator.rs

Phones

- Added function to validate swiss phone according to the format:

  ```
  0xxxxxxxxx
  ```

Usernames

- Added function to validate usernames according to the format:

  ```
  - case insensitive
  - only ascii alphanum + underscores
  - max 32 chars
  - min 1 char
  ```

Passwords

- Added function to validate passwords according to the policy:

  ```
  - At least one digit \[0-9\]
  - At least one lowercase char [a-z]
  - At least one uppercase char [A-Z]
  - At least one special char [.!@#$%^&{}\[\]:;<>,?\\/~_+\-=|'\*\(\)]
  - At least 8 chars in length, but no more than 64.
  ```

Access control

- Centralized access with casbin
  - Model:

```
    [request_definition]
    r = sub, obj

    [policy_definition]
    p = sub, obj

    [role_definition]
    g = _, _

    [policy_effect]
    e = some(where (p.eft == allow))

    [matchers]
    m = g(r.sub, p.sub) && r.obj == p.obj
    ```

  - Policy:

    ```
    # policies
    p, HR, changePhone
    p, HR, addUser
    p, StandardUser, changeOwnPhone

    # role inheritance
    g, HR, StandardUser
    ```

- Add general messages for missing authentication and forbidden access

  ```
  const FORBIDDEN_MSG: &'static str = "forbidden";
  const UNAUTHENTICATED_MSG: &'static str = "unauthenticated";
  ```

Logging

- Use of simplelog and teminal logger with custom config to add date to the logs:

```rust
// main.rs
// Set up logger
let config = ConfigBuilder::new()
    .set_time_format_custom(format_description!("[day].[month].[year] [hour]:[minute]:[second]"))
    .build();

TermLogger::init(
    LevelFilter::Info,
    config,
    TerminalMode::Mixed,
    ColorChoice::Auto
).unwrap();
```

- Add logging in main, actions and database
- Failed input validation, database updates, errors and normal behavior have been logged

Others

- Usernames are lowercased
- Remove sensitive information sent by the server in show_users endpoint
  - Created new struct UserInfo to send

    ```rust
    #[derive(Serialize, Deserialize, Clone, Debug)]
    struct UserInfo {
        username: String,
        phone_number: String,
    }
    ```

  - Changed function values in Database

    ```rust
    pub fn get_all_user_info() -> Result<Vec<UserInfo>, Box<dyn Error>> {
        Ok(DB.borrow_data()?.data
            .values()
            .cloned()
            .map(|u| {
                UserInfo::new(String::from(u.username()),
                    String::from(u.phone_number()))})
            .collect())
    }
    ```

- Hashed password in Database and changed login
  - Use of Argon2id

    ```
    Version: 19
    Memory: 64 KiB
    Number of passes: 3
    Number of lanes: 4
    Output length: 64B
    ```

  - Verify password even if the user does not exist
  - Hash default password accounts (**Test1234.** replaces default_pass)