Relatório - Projeto SIO

(Entrega 3)

Trabalho realizado por:
- Abel Teixeira, Nº 113655;
- Filipe Sousa, Nº 114196;
- Tiago Albuquerque, Nº 112901.

Introdução:

Este projeto consiste no desenvolvimento de um sistema seguro para a gestão e manipulação de documentos em organizações. Inclui funcionalidades de autenticação de utilizadores (Subjects), gestão de permissões e controlos de acesso, bem como operações de upload, download e exclusão de documentos.

Overview:

- Python
- Flask
- SQLite

Funcionalidades implementadas:

1. **Autenticação e gestão de sessões**:
    a. **Criação de Sessões Seguras**: Geração de sessões autenticadas para os sujeitos com tempo de expiração de 60 min.
    b. **Validação de Sessões**: Verificação da validade da sessão baseada no id da mesma, evitando acesso não autorizado como, por exemplo, tentativa de acesso com a sessão expirada.
    c. **Assumir e libertar Roles**: Os sujeitos podem assumir ou liberar roles durante uma sessão, com verificações de permissões associadas.

2. **Gestão de Sujeitos e Organizações**:
    a. **Criação de Sujeitos e Organizações**: Registo seguro de sujeitos e organizações, que inclui a verificação de argumentos (e.g. usernames e emails únicos, verificação da segurança da password etc.).
    b. **Alteração de Status de Sujeitos**: Alteração de status (`active`, `suspended`) de sujeitos com validação de permissões.
    c. **Listar Sujeitos**: A listagem de sujeitos tem o suporte de filtros (e.g. mesma organização ou role).

3. **Controle de Roles e Permissões**:
    a. **Criação e gestão de Roles**: Definição de roles com permissões associadas para controlar o acesso a certas funcionalidades na organização;
    b. **Adição e remoção de Permissões**: Alterações de permissões associadas a roles.
    c. **Associação de sujeitos a Roles**: Manipulação das associações entre sujeitos e roles dentro de uma organização.

4. **Gestão de Documentos**:
    a. **Upload de Documentos Seguros**: Upload de documentos que vão ser encriptados com AES e controle de integridade via HMAC.
    b. **Download de Documentos Encriptados**: Retorno de arquivos com metadados de integridade e chaves criptográficas associadas.
    c. **Exclusão de Documentos**: Remoção de documentos da organização.
    d. **Controle de Acesso a Documentos (ACL)**: Configuração de permissões específicas para documentos, vinculadas a roles. (*)

5. **Criptografia e Segurança**:
    a. **Proteção de Arquivos**: Uso do *AES* (256 bits) para encriptação do seu conteúdo e *RSA* (2048 bits) para troca segura de keys.
    b. **Hashing Seguro de passwords**: Utilização de *PBKDF2* para armazenar passwords.
    c. **Timestamps**: Implementação de timestamps para evitar ataques de replay e garantir integridade aos pedidos.
    d. **Integridade dos Dados**: Verificação com HMAC para garantir que arquivos não foram alterados.(**)

6. **Operações de encriptação e keys**:
    a. **Geração e gestão de Keys AES**: Criação de chaves únicas para cada documento.
    b. **Desencriptação segura no Cliente**: Rotinas para descodificar e verificar integridade no lado do cliente.
    c. **Gerenciamento de Chaves RSA**: Uso de chaves públicas e privadas para criptografia assimétrica e comunicação segura.

Decisões efetuadas pelo grupo:

- Não achámos oportuno a utilização de uma role ACL;
- O Repositório ao ser iniciado pede uma password (da **PRIVATE_KEY** do mesmo) para poder ser desbloqueado e inicializado.

- Pedidos (Cliente - Servidor):
    - Todos os sessions ID foram passados pelos headers dos pedidos;
    - O resto das informações dos mesmos foram passados ou pelo pelos parâmetros (*'params'*) ou pelo corpo (*'body'*):
        - De notar que todas estas informações foram encriptadas com recurso à **PUBLIC_KEY** do Repositório.
        - São desencriptados à chegada na api com a **PRIVATE_KEY** do Repositório.

- Respostas (Servidor - Cliente):
    - Quando contém dados sensíveis, são encriptadas com a **PUBLIC_KEY** do sujeito que fez o pedido.
    - Nesse caso, a resposta é composta pelo conteúdo encriptado e pelo nome da key do sujeito.
    - Quando o sujeito(cliente) recebe a resposta é pedida a password da **PRIVATE_KEY** do sujeito para poder ver o conteúdo desencriptado da resposta.

- Outros pontos:
    - **Validação de Timestamps:** Todos os pedidos incluem um timestamp encriptado, que é validado no servidor para evitar ataques de replay.
    - **Tratamento de Erros:** Em caso de falhas, como chaves inválidas ou falta de permissões, o sistema retorna mensagens de erro claras.
    - **Proteção de Credenciais:** Nenhuma password foi armazenada como texto simples no código-fonte ou em repositórios. A gestão de passwords segue práticas como hashing com PBKDF2 e salt dinâmico.

Roles e Permissões:

- **Roles admissíveis** no nosso projeto**:**
    - ‘manager’
    - ‘supervisor’
    - ‘member’
    - ‘guest’

- **Permissões admissíveis** no nosso projeto:
    - *Permissões nos documentos**:***
        - DOC_READ
        - DOC_DELETE
        - DOC_ACL
    - *Permissões nas organizações:*
        - SUBJECT_NEW
        - SUBJECT_DOWN
        - SUBJECT_UP
        - DOC_NEW

- **Permissões relativas a Roles:**
    - ROLE_NEW
    - ROLE_DOWN
    - ROLE_UP
    - ROLE_MOD

Permissões para cada Role:

- **‘manager’:**
    - DOC_READ
    - DOC_DELETE
    - DOC_NEW
    - DOC_ACL
    - ROLE_NEW
    - ROLE_DOWN
    - ROLE_UP
    - ROLE_MOD
    - SUBJECT_NEW
    - SUBJECT_DOWN
    - SUBJECT_UP

- **‘supervisor’:**
    - DOC_READ
    - DOC_DELETE
    - DOC_NEW
    - DOC_ACL
    - ROLE_NEW
    - SUBJECT_DOWN
    - SUBJECT_UP

- **‘member’:**
    - DOC_READ
    - DOC_DELETE
    - DOC_NEW

- **‘guest’:**
    - DOC_READ

Observações:

*Não foi implementado na sua totalidade

**Após implementar a verificação de integridade dos documentos começou a aparecer um bug na desencriptação onde aparecem 4 caracteres desconhecidos em todos os documentos desencriptados antes do conteúdo dos mesmos.

Notas:

- De notar que foi escrito um **README_COMMANDS.md** com o objetivo de apoiar no teste do software.
- O **README.md** contém todos os comandos e os seus argumentos devidamente documentados para ajudar na utilização. O mesmo também contém as Roles criadas por nós com as respectivas permissões.