Segurança Informática e nas Organizações

Universidade de Aveiro

Tiago Costa, Tiago Almeida, João Pinto

Segurança Informática e nas Organizações

Dept. de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

Tiago Costa, Tiago Almeida, João Pinto
tiagocosta@ua.pt (114629), tiagoalmeida@ua.pt (113106),
joaop1@ua.pt (113093)

29 de janeiro de 2025

Conteúdo

Capítulo 1

Introdução

Com o presente relatório, o grupo que realizou este projeto pretende rever o mesmo, verificando se cumpre os requisitos referidos no standard ASVS, publicado pela OWASP. Para o efeito, foi escolhido um capítulo que o grupo considerou relevante: o Capítulo 4, relativo ao Access Control. Ao longo do relatório serão discutidos os vários pontos mencionados neste capítulo, justificando se são pertinentes no contexto da aplicação desenvolvida e, no caso positivo, é feita uma análise relativa ao cumprimento (ou não) dos requisitos.

Este relatório reflete o empenho do grupo em desenvolver uma solução que atenda às necessidades de segurança e funcionalidade requeridas pelo projeto, bem como a conformidade com boas práticas de segurança definidas no ASVS. A estrutura apresentada inclui a descrição geral do sistema, as principais decisões de implementação e as conclusões sobre os resultados obtidos e melhorias futuras.

Capítulo 2

Metodologia

Ao longo deste capítulo são descritos os métodos que o grupo seguiu para fazer uma análise do sistema à luz dos requisitos do capítulo escolhido do ASVS. A metodologia aplicada compreende as etapas de compreensão dos requisitos propostos no ASVS, bem como uma revisão do sistema, testes executados, identificação de melhorias e o processo de elaboração do presente relatório.

2.1 Revisão da Arquitetura do Sistema

Numa fase inicial foi feita uma revisão geral do sistema, tendo já em mente o capítulo escolhido. Consequentemente, esta revisão incidiu mais na implementação dos mecanismos de controlo de acesso, uma vez que já tinha sido estabelecido o capítulo do ASVS a abordar.

2.2 Compreensão dos Requisitos do ASVS e Testes

No que diz respeito ao capítulo escolhido do ASVS, o grupo fez uma análise mais detalhada das normas descritas no Capítulo 4. Este capítulo especifica requisitos para um controlo de acesso robusto. Foram identificados requisitos relevantes para o sistema desenvolvido e, posteriormente, feita uma análise do sistema sob o ponto de vista dessas mesmas secções. Também a relevância de cada requisito foi devidamente justificada. Assim sendo, cada tópico será categorizado numa das seguintes categorias:

- **Não aplicável**: O requisito não é relevante no contexto do sistema.
- **Implementado**: O requisito é cumprido justificadamente.
- **Não implementado**: O requisito não foi implementado.

Na grande maioria dos requisitos, a análise do sistema é suficiente para justificar o seu cumprimento (ou não cumprimento). Eventualmente, nos requisitos em que seja possível e útil realizar testes, serão apresentados esses mesmos testes como justificação.

Capítulo 3

Análise do Capítulo 4 do ASVS: Access Control

3.1 General Access Control Design (V4.1)

3.1.1 Secção V4.1.1

Descrição

"Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed."

A partir da descrição, conseguimos definir que ponto 4.1.1 exige que os controlos de acesso sejam implementados e verificados em uma camada confiável no lado do servidor, independentemente de haver mecanismos de controle no cliente. Isso é importante porque qualquer lógica de acesso implementada no cliente pode ser facilmente manipulada ou ignorada por um atacante.

Aplicabilidade

No sistema desenvolvido, os controlos de acesso são aplicadosa apartir do uso de ACLs (Access Control Lists) definidas e geridas pelos utilizadores que têm permissões para o fazer dentro de cada organização. Posto isto, este requisito é **aplicável**.

Avaliação

As ACLs residem no servidor, pelo que é válido afirmar que as regras de controlo de acesso são aplicadas numa camada segura e confiável. Para cada comando que requer autorização, a aplicação verifica se o utilizador possui uma role com a permissão necessária e se esta role está ativa dentro da organização.

No exemplo abaixo podemos ver como é realizada a verificação de permissões do lado do servidor. É usado o exemplo da função "role_add()", mas o formato é o mesmo para as restantes funções:

```
@app.route("/role/add", methods=["POST"])
def role_add():
    # Operacoes de obtencao de dados e verificacao da sua validade
    # ...

    # Filtrar apenas as roles que est o ativas
    active_subject_roles = [role for role in sessions[
        received_session_id]["ROLES"] if organizations[
        organization_name]["ACL"][role]["STATUS"] == "ACTIVE"]

    # Verificar se essa role tem a permissao
    for role in active_subject_roles:
        if "ROLE_NEW" in organizations[organization_name]["ACL"][
            role]["PERMISSIONS"]:

            # Verificar se a role que esta a ser adicionada ja
                existe
            if new_role in organizations[organization_name]["ACL"]:
                return json.dumps({"Error": "Role already exists."
                    }), 400

            organizations[organization_name]["ACL"][new_role] = {
                "PERMISSIONS": [],
                "STATUS": "ACTIVE"
            }

            return json.dumps({"status": "success", "message": "
                Role created"}), 200

    return json.dumps({"Error": "Subject missing permissions to add
         role"}), 400
```

3.1.2 Secção V4.1.2

Descrição

"Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized."

O que é pedido neste ponto é que utilizadores não possam alterar as permissões, sejam suas, de outros ou de documentos, se não estiver autorizado para o fazer.

Aplicabilidade

No sistema desenvolvido é possível alterar a ACL, tanto de organizações como de documentos. Por este motivo, este tópico é relevante e **aplicável**, uma vez que é importante que estas modificações sejam acessíveis apenas a utilizadores autorizados.

Avaliação

Este ponto está implementado na aplicação. Para alterar as permissões de uma role, sejam permissões dentro da organização ou de um documento específico, é necessário ter uma role com permissões como `ROLE_MOD` ou `DOC_ACL`. Além disso, um *subject* não pode atribuir-se a si mesmo uma role com as permissões necessárias, pois estas precisam lhe ser atribuídas por alguém com permissões específicas.

3.1.3 Secção V4.1.3

Descrição

"Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege."

O ponto V4.1.3 do ASVS exige que seja seguido o **Princípio do Mínimo Privilégio (PoLP)**, garantindo que os utilizadores só possam acessar funções, dados, URLs e recursos para os quais possuem autorização explícita. Além disso, o sistema deve ser protegido contra ataques de **spoofing** e outros com fins de **elevação de privilégios**.

Aplicabilidade

Este requisito é **aplicável** ao sistema, dado que comandos específicos exigem permissões para serem executados.

Avaliação

Este ponto está **implementado** no nosso sistema.

Implementação do Princípio do Mínimo Privilégio (PoLP):
Comandos que exigem permissões específicas requerem obrigatoriamente o envio de uma `session file` como argumento. Ao receber este ficheiro, a API verifica server-side se o utilizador possui uma role ativa com as permissões necessárias. Caso contrário, o comando é rejeitado.

Proteção contra Spoofing:
Ataques de spoofing são mitigados porque:

- Todas as `session files` são armazenadas e verificadas no servidor.
- As roles enviadas pelo cliente não são utilizadas para validação, servindo apenas como referência informativa.

Proteção contra Replay:
Payloads interceptados não podem ser reutilizados, pois são encriptados com um *shared secret* e incluem um `nonce` exclusivo.

Proteção contra Roubo de Sessão:
As `session files` são encriptadas usando AES-CBC com uma chave compartilhada, protegendo a integridade das permissões.

3.1.4 Secção V4.1.5

Descrição

"Verify that access controls fail securely including when an exception occurs."

O ponto V4.1.5 requer que os controlos de acesso sejam implementados de forma a garantir que, em caso de falha ou exceção, o sistema não conceda acesso indevido ou realize ações não autorizadas.

Aplicabilidade

Os parâmetros dos comandos são escolhidos pelos utilizadores e, caso estejam mal, podem levar a API a executar um erro. Por esse motivo este ponto é **aplicável** no nosso sistema.

Avaliação

No nosso sistema, tentamos identificar os erros de forma a que sempre que são enviados parâmetros incorretos, uma mensagem de erro esclarecedora acompanhada de um código de erro HTTP 400 é devolvida.

Exemplo de erro de tempo de validade de um session file expirado:

```
@app.route("/role/add", methods=["POST"])
def role_add():
    # ...

    # Verificar se a sessao esta dentro do tempo valido
    current_time = datetime.now()
    expiry_time = datetime.strptime(sessions[received_session_id]["
        EXPIRY_TIME"], "%d-%m-%Y %H:%M:%S")
    if current_time > expiry_time:
        sessions.pop(received_session_id)
        return json.dumps({"Error": "Session expiry time reached.
            Create a new session."}), 400

    # ...
```

Melhorias Propostas

- **Códigos de erro mais específicos:** Substituir o uso genérico do código 400 por códigos mais apropriados, como 403 para falta de permissões e 401 para falhas de autenticação.
- **Erro default 500:** Fazer uso de um try-except para cada rota da API de forma a capturar exceções não previstas e devolver um código genérico 500 para evitar a exposição de mensagens sensíveis.

3.2 Operation Level Access Control (V4.2)

3.2.1 Secção V4.2.1

Descrição

"Verify that sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records."

O ponto 4.2.1 exige que dados sensíveis e a API estejam protegidos contra ataques de **Insecure Direct Object Reference (IDOR)**, que podem permitir ações como criar, visualizar, atualizar ou excluir registros de outros utilizadores sem autorização.

Aplicabilidade

Este requisito é **aplicável**, dado que o sistema lida com dados sensíveis enviados por métodos HTTP para a API, que podem ser intercetados e alterados com fim maliciosos.

Avaliação

Este ponto foi **implementado** no sistema a partir das seguintes medidas:

- **Ausência de Identificadores Diretos na URL:** As rotas são estáticas e os dados são transmitidos no corpo do *payload*, eliminando a possibilidade manipulação direta de identificadores.
- **Validação Server-Side:** Todas as interações são validadas no servidor antes de qualquer ação ser executada.
- **Identificadores Não-Previsíveis:** IDs são gerados de forma aleatória (UUIDs), reduzindo o risco de adivinhação.

3.2.2 Secção V4.2.2

Descrição

"Verify that the application or framework enforces a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF protects unauthenticated functionality."

O ponto V4.2.2 requer que a aplicação implemente mecanismos robustos para evitar ataques de **Cross-Site Request Forgery (CSRF)** em funcionalidades autenticadas e proteções eficazes contra automação ou CSRF em funcionalidades não autenticadas.

Aplicabilidade

Este requisito **não se aplica**, pois a arquitetura do sistema elimina os vetores clássicos de ataques CSRF.

Justificativa

- Todas as interações com a API exigem autenticação explícita e validação de sessão, eliminando a possibilidade de um atacante realizar requisições não autorizadas em nome do usuário.
- O sistema não utiliza navegadores para gerenciar cookies ou tokens de autenticação, removendo o principal vetor de ataque associado ao CSRF, que depende de cookies enviados automaticamente pelo navegador.
- Os *payloads* enviados entre cliente e servidor são encriptados utilizando um *shared secret* e protegidos com HMAC para garantir integridade e autenticidade. Mesmo que um atacante consiga interceptar ou tentar forjar requisições, elas serão invalidadas pela falta de um HMAC válido.

Em resumo, a arquitetura do sistema não apresenta os pressupostos necessários para a exploração de ataques CSRF, tornando este requisito irrelevante para o contexto atual.

3.3 Other Access Control Considerations (V4.3)

3.3.1 Secção V4.3.1

Descrição

"Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use."

O ponto V4.3.1 exige que interfaces administrativas implementem mecanismos de autenticação multi-fator (MFA) para prevenir acessos não autorizados. Isso garante que mesmo que as credenciais de um administrador sejam comprometidas, um atacante ainda precisará de um segundo fator (como um código temporário, autenticação por aplicativo ou token físico) para acessar a interface administrativa.

Aplicabilidade

Devido à arquitetura do sistema, este ponto **não é aplicavel**, uma vez que não existe uma interface administrativa tradicional. Invés disso, o sistema usa ACLs com roles e permissões associadas.

Justificativa

No nosso sistema, não existe uma interface administrativa tradicional acessível via navegador ou outro meio. Toda a interação administrativa é feita através de comandos específicos enviados à API, e esses comandos seguem os mesmos padrões de segurança aplicados a outras operações, como:

- Uso de sessões autenticadas.
- Encriptação de payloads com um shared secret.
- Validação de permissões baseada em roles.

Conclusão: Como não há uma interface administrativa no sentido convencional, o requisito de MFA para essas interfaces não se aplica diretamente.

Melhorias Propostas

- Integrar MFA utilizando uma segunda camada de autenticação, como um código temporário gerado por uma aplicação (e.g., Google Authenticator) ou enviado por SMS.
- Garantir que todas as operações de inicio de sessão de utilizadores com permissões elevadas exijam MFA.

3.3.2 Secção V4.3.2

Descrição

"Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders."

O ponto V4.3.2 exige que o sistema desative a navegação de diretórios, exceto quando intencionalmente necessária, e previna a exposição de metadados ou conteúdos de diretórios que possam revelar informações sensíveis. Isso é fundamental para evitar ataques baseados na descoberta de ficheiros ou diretórios que possam conter dados confidenciais ou facilitar o reconhecimento por parte de atacantes.

Aplicabilidade

Este requisito é **aplicável** no nosso sistema porque ficheiros como **session.json**, **organization.json** e **documents.json** contêm informações confidenciais que poderiam ser exploradas por utilizadores maliciosos para comprometer a integridade do sistema ou aceder a dados sensíveis de outras organizações.

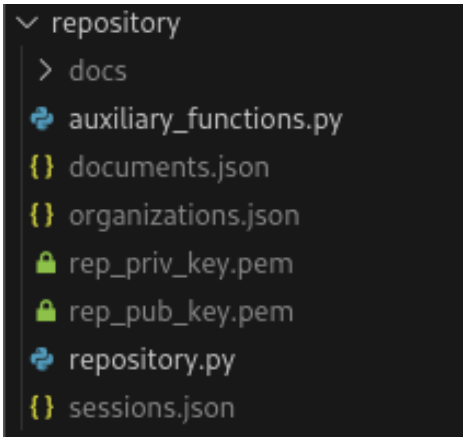


Figura 3.1: Pasta do repositório e ficheiros sensiveis.

Avaliação

De forma a implementar este ponto, o nosso sistema adota várias medidas para garantir a proteção contra acesso não autorizado a ficheiros sensíveis:

- **Armazenamento server-side:** Ficheiros confidenciais como session.json, organization.json e documents.json são armazenados exclusivamente no lado do servidor, fora do alcance de utilizadores finais.
- **Comandos limitados:** Os utilizadores interagem com a API através de uma lista de comandos pré-definidos, eliminando a possibilidade de realizar operações não autorizadas.
- **Rotas estáticas:** Todas as rotas da API são estáticas, prevenindo alterações dinâmicas de URL que poderiam ser usadas para tentar aceder a ficheiros sensíveis.

3.3.3 Secção V4.3.3

Descrição

"Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems, and / or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud."

O ponto V4.3.3 exige que aplicações implementem mecanismos adicionais de autorização, como **autenticação adaptativa ou escalonada**, para sistemas de menor valor. Para sistemas de alto valor, o requisito sugere a **segregação de funções** como medida de controle antifraude, considerando o risco associado à aplicação e fraudes anteriores. Esses controles são essenciais para minimizar o impacto de acessos indevidos e mitigar riscos de fraude.

Autenticação escalonada: É um mecanismo que exige uma autenticação adicional para operações ou acessos mais sensíveis.

Segregação de Funções: Refere-se à divisão de responsabilidades para garantir que nenhuma pessoa ou conta tenha controle total sobre operações críticas.

Aplicabilidade

Este requisito é **aplicável** ao nosso sistema, uma vez que contém dados sensíveis associados a organizações e sessões, além de permitir operações administrativas que podem impactar diretamente a integridade dos dados armazenados.

Avaliação

No nosso sistema, foram implementadas as seguintes medidas para atender ao requisito:

- **Segregação de funções:** Os utilizadores são atribuídos a roles específicas com permissões limitadas, sendo que apenas o utilizador com a role manager, tem acesso irrestrito a todas as operações críticas da sua organização.
- **Controle de permissões:** Operações administrativas e de alto impacto requerem permissões específicas, que são validadas para cada comando enviado à API.
- **Sessões autenticadas:** Cada operação crítica exige que o utilizador esteja autenticado e autorizado, garantindo que apenas utilizadores válidos possam executar comandos.

Melhorias Propostas

Embora o sistema já implemente medidas básicas de controle, algumas melhorias podem ser consideradas para reforçar a segurança e melhor atender ao exigido:

- **Autenticação escalonada:** Introduzir um segundo fator de autenticação (MFA) para operações de alto impacto ou ao acessar dados sensíveis.
- **Segregação de funções:** Ao criar uma organização, alterar o role manager default por várias outras roles com permissões diferentes, de forma a garantir que nenhuma role individual tenha acesso irrestrito a todas as operações críticas.
- **Auditoria de operações:** Registrar todas as operações críticas em logs detalhados, permitindo a detecção e análise de atividades suspeitas.

Capítulo 4

Implementação das Funcionalidades e Estrutura de Dados

Neste capítulo, descrevemos a implementação das funcionalidades do sistema e a estrutura de dados utilizada no repositório. O sistema permite a gestão de **organizações, utilizadores, permissões e documentos**, garantindo a segurança e o controlo de acessos definidos na análise anterior.

4.1 Comandos Implementados

O sistema disponibiliza **comandos locais** e **comandos que interagem com a API** para a gestão do repositório.

4.1.1 Comandos Locais

Os comandos locais podem ser executados sem necessidade de conexão ao repositório. Foram implementados os seguintes:

- `rep_subject_credentials <password> <credentials file>`
  Exemplo de uso:

  ```
  ./rep_subject_credentials 1234 my_keys.json
  ```

  Comportamento esperado: Cria um ficheiro `my_keys.json` contendo um par de chaves pública e privada, com a chave privada cifrada.

- `rep_decrypt_file <encrypted file> <encryption metadata>`
  Função: Desencripta um ficheiro encriptado, conhecendo os seus metadados de encriptação.

4.1.2 Comandos com a API

Estes comandos exigem que o repositório esteja em execução. Para iniciar a API:

```
python3 repository/repository.py
```

Uma vez ativa, podemos executar comandos como:

- `rep_create_org <organization> <username> <name> <email> <public key file>`
  Exemplo de uso:

  ```
  ./rep_create_org my_org tigs Tiago tigs@ua.pt my_keys.json
  ```

  Comportamento esperado: Cria uma organização chamada `my_org` com o subject `tigs`.

- `rep_list_orgs`
  Função: Lista todas as organizações existentes no repositório.

  ```
  ./rep_list_orgs
  ```

- `rep_create_session <organization> <username> <password> <credentials file> <session file>`
  Exemplo de uso:

  ```
  ./rep_create_session my_org tigs 1234 my_keys.json my_session.json
  ```

  Comportamento esperado: Cria uma sessão autenticada e guarda a informação em `my_session.json`.

4.2 Estrutura de Dados do Repositório

O repositório armazena as entidades em dicionários estruturados. Abaixo estão os formatos principais.

4.2.1 Dicionário de Organizações

```
organizations = {
    "org1": {
        "SUBJECTS": {
            "user1": {
                "FULL_NAME": "Tiago Costa",
```

```
                "EMAIL": "tigs@ua.pt",
                "PUBLIC_KEY": "<chave_publica>",
                "STATUS": "ACTIVE",
                "ROLES": ["MANAGER"]
            }
        },
        "ACL": {
            "MANAGER": {
                "PERMISSIONS": ["ROLE_ACL", "SUBJECT_NEW", "DOC_NEW"]
            }
        }
    }
}
```

4.2.2 Dicionário de Sessões

```
sessions = {
    "session_id": {
        "SUBJECT": "user1",
        "ORG_NAME": "org1",
        "ROLES": ["MANAGER"],
        "EXPIRY_TIME": "ISO-8601-timestamp"
    }
}
```

4.2.3 Dicionário de Documentos

```
documents = {
    "doc1": {
        "NAME": "my_doc",
        "CREATION_DATE": "10-01-2025",
        "CREATOR": "user1",
        "ACL": {"MANAGER": ["DOC_READ", "DOC_DELETE"]}
    }
}
```

4.3 Considerações Finais

O sistema implementa um modelo seguro para a gestão de documentos e permissões, seguindo as normas definidas no *Access Control* do ASVS. O formato estruturado dos dados e a implementação de comandos facilitam a gestão e garantem a segurança dos acessos.

Capítulo 5

Testes realizados

Introdução

Esta secção foi elaborada com o suporte de um *script* desenvolvido pela equipa, que automatiza a execução de testes a todos os comandos implementados no projeto. O *script*, denominado `run test`, é executado com o seguinte comando na linha de comandos:

```
./run_test
```

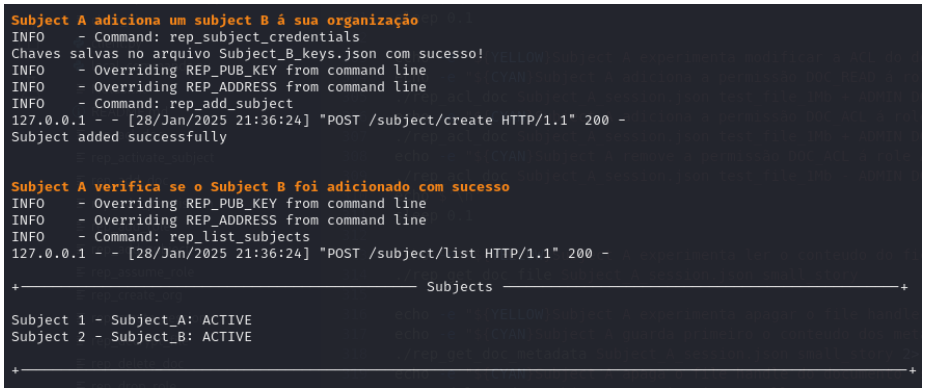
Este *script* realiza uma validação abrangente das funcionalidades, desde a criação de credenciais até operações mais complexas, como a gestão de permissões e a interação com a API. O objetivo principal é garantir que todas as funcionalidades implementadas funcionam como esperado, cumprindo os requisitos definidos e garantindo a segurança e robustez do sistema. Os resultados de cada teste são registados automaticamente, facilitando a análise e correção de eventuais problemas.

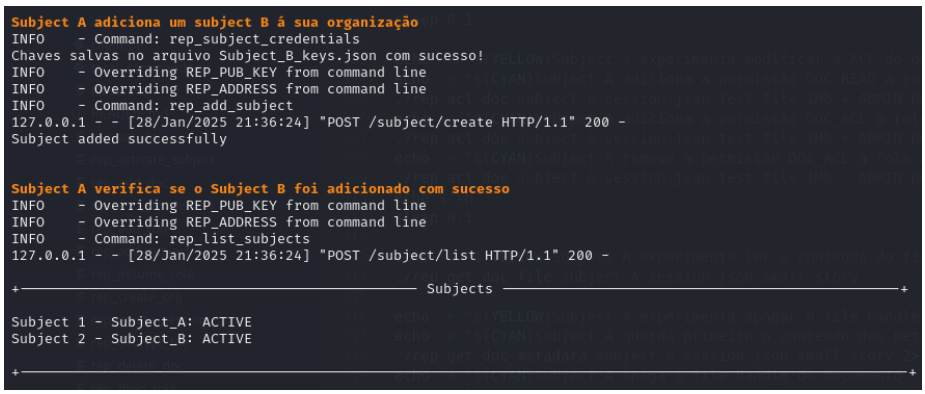
5.1 Primeiros passos de um subject novo

Estes primeiros testes mostram o funcionamento inicial de um *subject*. Entre as ações realizadas estão a criação de credenciais (chaves assimétricas), criação de uma organização, início de sessão dentro da organização e a assunção de uma *role* (neste caso, a `manager`, uma vez que é o primeiro *subject* da organização).

```
Subject A cria a suas credenciais
INFO     - Command: rep_subject_credentials
Chaves salvas no arquivo Subject_A_keys.json com sucesso!


Subject A cria uma nova organização
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_create_org
127.0.0.1 - - [28/Jan/2025 21:36:23] "POST /organization/create HTTP/1.1" 200 -
Organization created successfully


Subject A verifica se a sua organização foi criado com sucesso
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_orgs
127.0.0.1 - - [28/Jan/2025 21:36:23] "GET /organization/list HTTP/1.1" 200 -
Operation successful

+──────────────────────────────────────────── Organizations ────────────────────────────────────────────+

Organization 1: Subject_A_ORG

+──────────────────────────────────────────────────────────────────────────────────────────────────────+



Subject A inicia uma sessão dentro da organização que acabou de criar
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_create_session
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /session/create HTTP/1.1" 200 -
Operation successful


Subject A assume a role MANAGER (role inicial default)
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_assume_role
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /role/assume HTTP/1.1" 200 -
Role assumed successfully
```

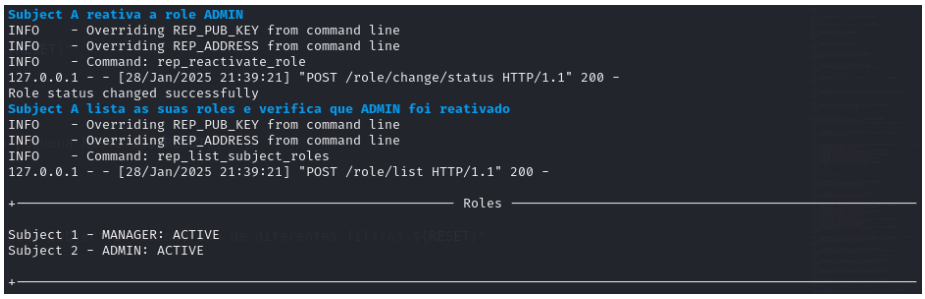
```
Subject A quer que o Subject B tenha a role ADMIN e consiga adicionar documentos.
Para isso primeiro adiciona a permissão DOC_NEW á role ADMIN
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_add_permission
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/add/permission HTTP/1.1" 200 -
Operation successful: Permission added to role


Subject B inicia sessão e assume a role ADMIN
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_create_session
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /session/create HTTP/1.1" 200 -
Operation successful
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_assume_role
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/assume HTTP/1.1" 200 -
Role assumed successfully
```

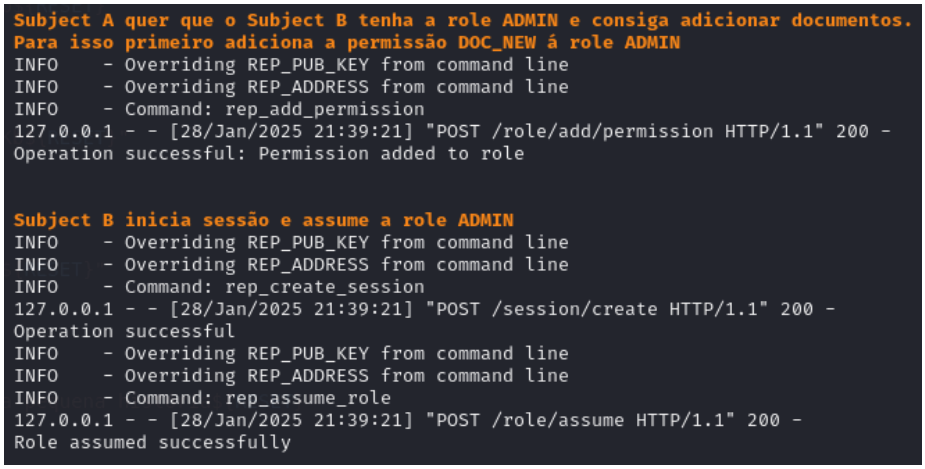

Subject A adiciona um subject B á sua organização
INFO - Command: rep_subject_credentials
Chaves salvas no arquivo Subject_B_keys.json com sucesso!
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_add_subject
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /subject/create HTTP/1.1" 200 -
Subject added successfully
Subject A verifica se o Subject B foi adicionado com sucesso
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_list_subjects
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /subject/list HTTP/1.1" 200 -
Subjects
Subject 1 - Subject_A: ACTIVE
Subject 2 - Subject_B: ACTIVE

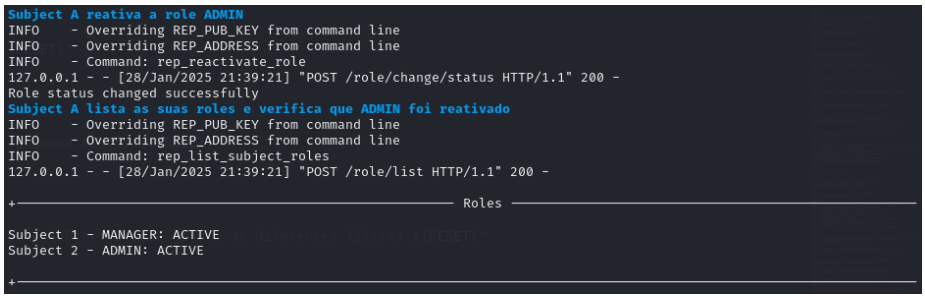

Subject A adiciona um subject B á sua organização
INFO - Command: rep_subject_credentials
Chaves salvas no arquivo Subject_B_keys.json com sucesso!
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_add_subject
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /subject/create HTTP/1.1" 200 -
Subject added successfully
Subject A verifica se o Subject B foi adicionado com sucesso
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_list_subjects
127.0.0.1 - - [28/Jan/2025 21:36:24] "POST /subject/list HTTP/1.1" 200 -
Subjects
Subject 1 - Subject_A: ACTIVE
Subject 2 - Subject_B: ACTIVE

```
Subject A experimenta suspender o Subject B
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_suspend_subject
127.0.0.1 - - [28/Jan/2025 21:39:19] "POST /subjects/suspend HTTP/1.1" 200 -
Operation successful
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_subjects
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /subject/list HTTP/1.1" 200 -

+---------- Subjects ----------

Subject 1 - Subject_B: SUSPENDED

+------------------------------

Subject A volta a ativar o Subject B
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_activate_subject
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /subjects/activate HTTP/1.1" 200 -
Operation successful
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_subjects
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /subject/list HTTP/1.1" 200 -
+---------- Subjects ----------

Subject 1 - Subject_B: ACTIVE

+------------------------------
```

```
Subject A experimenta com o gerenciamento de permissões e roles
Subject A lista as permissões da role ADMIN
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_role_permissions
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /role/list/permissions HTTP/1.1" 200 -
+---------- ADMIN Permissions ----------

Permission 1: SUBJECT_DOWN
Permission 2: SUBJECT_UP

+---------------------------------------

Subject A lista as roles com a permissão SUBJECT_UP
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_permission_roles
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /permission/list/role HTTP/1.1" 200 -

+---------- Organization roles with SUBJECT_UP permission ----------

Role 1: MANAGER
Role 2: ADMIN

+-------------------------------------------------------------------

Subject A suspende a role ADMIN
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_suspend_role
127.0.0.1 - - [28/Jan/2025 21:39:20] "POST /role/change/status HTTP/1.1" 200 -
Role status changed successfully
Subject A lista as suas roles e verifica que ADMIN foi suspenso
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_subject_roles
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/list HTTP/1.1" 200 -
+---------- Roles ----------

Subject 1 - MANAGER: ACTIVE
Subject 2 - ADMIN: SUSPENDED

+---------------------------
```


Subject A reativa a role ADMIN
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_reactivate_role
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/change/status HTTP/1.1" 200 -
Role status changed successfully
Subject A lista as suas roles e verifica que ADMIN foi reativado
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_list_subject_roles
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/list HTTP/1.1" 200 -
Roles
Subject 1 - MANAGER: ACTIVE
Subject 2 - ADMIN: ACTIVE


Subject A quer que o Subject B tenha a role ADMIN e consiga adicionar documentos.
Para isso primeiro adiciona a permissão DOC_NEW á role ADMIN
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_add_permission
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/add/permission HTTP/1.1" 200 -
Operation successful: Permission added to role
Subject B inicia sessão e assume a role ADMIN
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_create_session
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /session/create HTTP/1.1" 200 -
Operation successful
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_assume_role
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/assume HTTP/1.1" 200 -
Role assumed successfully


Subject A reativa a role ADMIN
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_reactivate_role
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/change/status HTTP/1.1" 200 -
Role status changed successfully
Subject A lista as suas roles e verifica que ADMIN foi reativado
INFO - Overriding REP_PUB_KEY from command line
INFO - Overriding REP_ADDRESS from command line
INFO - Command: rep_list_subject_roles
127.0.0.1 - - [28/Jan/2025 21:39:21] "POST /role/list HTTP/1.1" 200 -
Roles
Subject 1 - MANAGER: ACTIVE
Subject 2 - ADMIN: ACTIVE

```
Subject B experimenta adicionar um documento de 10Kb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_add_doc
127.0.0.1 - - [28/Jan/2025 21:39:22] "POST /doc/create HTTP/1.1" 200 -
Document added successfully


Subject B experimenta adicionar um documento de 100Kb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_add_doc
127.0.0.1 - - [28/Jan/2025 21:39:22] "POST /doc/create HTTP/1.1" 200 -
Document added successfully


Subject B experimenta adicionar um documento de 1Mb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_add_doc
127.0.0.1 - - [28/Jan/2025 21:39:22] "POST /doc/create HTTP/1.1" 200 -
Document added successfully


Subject A experimenta adicionar um documento com uma pequena história
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_add_doc
127.0.0.1 - - [28/Jan/2025 21:39:22] "POST /doc/create HTTP/1.1" 200 -
Document added successfully
```

```
Subject A verifica que documentos foram adicionados pelo Subject B fazendo uso de diferentes filtros.
Sem filtros (Todos os documentos)
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_docs
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/list HTTP/1.1" 200 -

+------------------------------------------------- Documents -------------------------------------------------+

Document 1 - Name: test_file_10Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 2 - Name: test_file_100Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 3 - Name: test_file_1Mb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 4 - Name: small_story - Creator: Subject_A - Creation date: 28-01-2025 21:39:22

+-------------------------------------------------------------------------------------------------------------+
Documentos do Subject B
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_docs
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/list HTTP/1.1" 200 -
+------------------------------------------------- Documents -------------------------------------------------+

Document 1 - Name: test_file_10Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 2 - Name: test_file_100Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 3 - Name: test_file_1Mb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
+-------------------------------------------------------------------------------------------------------------+

Documentos do Subject B mais recentes do que 18-12-20
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_docs
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/list HTTP/1.1" 200 -

+------------------------------------------------- Documents -------------------------------------------------+

Document 1 - Name: test_file_10Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 2 - Name: test_file_100Kb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22
Document 3 - Name: test_file_1Mb - Creator: Subject_B - Creation date: 28-01-2025 21:39:22

+-------------------------------------------------------------------------------------------------------------+

Documentos do Subject B mais antigos do que 18-12-20
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_docs
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/list HTTP/1.1" 200 -
No documents that meet the requirements
Documentos do Subject B do dia 18-12-20
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_list_docs
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/list HTTP/1.1" 200 -
No documents that meet the requirements
```

```
Subject A experimenta modificar a ACL do documento de 1Mb
Subject A adiciona a permissão DOC_READ á role ADMIN no ficheiro test_file_1Mb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_acl_doc
127.0.0.1 - - [28/Jan/2025 21:39:23] "POST /doc/change/acl HTTP/1.1" 200 -
Document ACL updated successfully
Subject A adiciona a permissão DOC_ACL á role ADMIN no ficheiro test_file_1Mb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_acl_doc
127.0.0.1 - - [28/Jan/2025 21:39:24] "POST /doc/change/acl HTTP/1.1" 200 -
Document ACL updated successfully
Subject A remove a permissão DOC_ACL á role ADMIN no ficheiro test_file_1Mb
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_acl_doc
127.0.0.1 - - [28/Jan/2025 21:39:24] "POST /doc/change/acl HTTP/1.1" 200 -
Document ACL updated successfully
```

```
Subject A experimenta ler o conteudo do ficheiro small_story.txt
INFO     - Overriding REP_PUB_KEY from command line
INFO     - Overriding REP_ADDRESS from command line
INFO     - Command: rep_get_doc_file
127.0.0.1 - - [28/Jan/2025 21:39:24] "POST /doc/get/metadata HTTP/1.1" 200 -
127.0.0.1 - - [28/Jan/2025 21:39:24] "POST /doc/get/file_by_handle HTTP/1.1" 200 -
File retrieved succesefully

+──────────────────────────────────────────────── File Content ────────────────────────────────────────────────

In a quiet village, a boy named Liam discovered a hidden key buried
in his garden. That night, he dreamt of a glowing door in the forest.
The next day, curiosity led him to the woods, where the key fit perfectly into
the mysterious door. As it creaked open, a dazzling world of floating islands and
golden skies lay before him. Liam stepped through, knowing his life would never be the same.


+──────────────────────────────────────────────────────────────────────────────────────────────────────────────
```

```
Subject A experimenta apagar o file handle do ficheiro small_story.txt e tenta ler o conteudo novamente
Subject A guarda primeiro o conteudo dos metadados
INFO    - Overriding REP_PUB_KEY from command line
INFO    - Overriding REP_ADDRESS from command line
INFO    - Command: rep_get_doc_metadata
127.0.0.1 - - [28/Jan/2025 23:18:41] "POST /doc/get/metadata HTTP/1.1" 200 -
Subject A apaga o file handle do documento 'small_story'
INFO    - Overriding REP_PUB_KEY from command line
INFO    - Overriding REP_ADDRESS from command line
INFO    - Command: rep_delete_doc
127.0.0.1 - - [28/Jan/2025 23:18:42] "POST /doc/clear/file-handle HTTP/1.1" 200 -
Document deleted successfully
Subject A experimenta ler o conteudo do ficheiro small_story.txt novamente
INFO    - Overriding REP_PUB_KEY from command line
INFO    - Overriding REP_ADDRESS from command line
INFO    - Command: rep_get_doc_file
127.0.0.1 - - [28/Jan/2025 23:18:42] "POST /doc/get/metadata HTTP/1.1" 200 -
127.0.0.1 - - [28/Jan/2025 23:18:42] "POST /doc/get/file_by_handle HTTP/1.1" 400 -
Como esperado, não funciona. O Subject A então usa o file handle guardado anteriormente para ler o ficheiro
INFO    - Overriding REP_PUB_KEY from command line
INFO    - Overriding REP_ADDRESS from command line
INFO    - Command: rep_get_file
127.0.0.1 - - [28/Jan/2025 23:18:42] "POST /doc/get/file_by_handle HTTP/1.1" 200 -
File retrieved successfully
b'8\xa2\xdb\xa0ß?0\xf5\xdb\xfaoe\xec\xca\x8e\x96}2j\x84\xb8n\x8aK\xb5^;M\xc9\xf9\xc3\x8c\xb0\x9cG\xd6AZd\xeb\xff\xe0
\xc6?\xbc\'8S\x94\x15a\xbeg\x05:\xecW\xdf\xfa\t\xf8\x98\xdd\xd3*_\xca\xf3\x13w7C\xaa\xf9\x19TO\xb6\x01\x18@\xb6Q"0/y
\xa0\xeax\xf1\xfb\xcc3\xd4\x8a\x92}\xc0R\xca\xc4\x91\xf5\x17f\x8d\xce\xc0(\x99<\x95\xa8 \x94M\x8e#\xc0\x17\xa9 \x7f\
xad\xbd50T\x7f\xb9;\xc6lW\r\x19My\x02\xac$\xb7s\xd9d\xc5\xb8\xc9i\x87O_\xe6\x98R\xe3R\x8d\xac\x00T\x9fB\xb7\x97\xce\
xe5-\x1c\x06\xe9\xb9F\x84,\xc6O\x8b\xcaIyB!\xac\xb3I>\x1bD\xa7\x99\x10t4|\xb4\xa6W\xd7|\xb0\x006\x9a\xba\xec\x02(\xf
0\xe68\x0cc!\xe6\xef[[\xb7\xe0\x96~=bPA\xae\xb1]\nDo\x8d,\xc7\xf4\x1e\xe4\xa9\xaa\'t\x01\x00\x81\xbbz\xb1y\xb4\xa3\x
92\x05T\x8f\x14=\xa8ll\xd9\xf6\x1ae\x9c\x13\x04\xfe\xa0]\x19?\xf0\xffUe\xf6T*\x8c\xdbr\xb8r\xc6|c\xb2\xdd\x18T\xffc\
xf8b"\xf5\x9a\xe7\xc6#\xfd\xfaL\x93\xe5\x11-b](#\xe2\x9e\xc8\xfb\x02\xe6\x08Dmr\xd8C\x0b\x82$\x93\x0cltD\xe0N,\xc5!6
YJy\xb6\x85\xa1\xf2\x99\'\x13\x05?|\x1e\x00h\xcf\x9b\x86\xe7\xe9\x02Xo\xc6\xb6\xf4\xba\x0e\xdf\x16\x98\x8a\xdc#jq\x0
2\t(\xb2\x930\xdf\t\x96\x01\x82V*\xa4\xac\xdbr\x97\xc2\xe1o\x16\xa2d'
No entanto, o conteudo vem encriptado, então o Subject A guarda o conteudo num ficheiro e desencripta com o comando
rep_decrypt_file
INFO    - Overriding REP_PUB_KEY from command line
INFO    - Overriding REP_ADDRESS from command line
INFO    - Command: rep_get_file
127.0.0.1 - - [28/Jan/2025 23:18:42] "POST /doc/get/file_by_handle HTTP/1.1" 200 -
File retrieved successfully
INFO    - Command: rep_decrypt_file

──────────────────────────────────────────── File Content ────────────────────────────────────────────

In a quiet village, a boy named Liam discovered a hidden key buried
in his garden. That night, he dreamt of a glowing door in the forest.
The next day, curiosity led him to the woods, where the key fit perfectly into
the mysterious door. As it creaked open, a dazzling world of floating islands and
golden skies lay before him. Liam stepped through, knowing his life would never be the same.

──────────────────────────────────────────────────────────────────────────────────────────────────────

═══════════ Fim da simulação ═══════════
```

Capítulo 6

Correções e melhorias

6.1 Correção da função add doc

Na implementação inicial da função `add_doc`, foi identificado um problema relacionado com o armazenamento do `file_handle`, que é o nome do ficheiro utilizado para guardar o documento. Este estava a ser guardado com codificação `base64`, que permite o uso do caracter `/`. No contexto de caminhos de ficheiros, o caracter `/` é interpretado como uma nova pasta, o que resultava na criação indevida de múltiplas pastas sempre que o `file_handle` continha este caracter.

Para resolver este problema, foi implementada uma solução simples mas eficaz: substituir o caracter `/` por `_` no `file_handle`. Com esta alteração, os documentos passaram a ser armazenados corretamente, sem a criação desnecessária de pastas, garantindo a integridade da estrutura do repositório.

Após a entrega do trabalho em época normal, o grupo propôs-se a melhorar o projeto com o *feedback* dado pelos professores. Ao longo deste capítulo serão exploradas as melhorias feitas, tendo cada uma delas a sua própria secção. Para implementar as melhorias, foi criado um *branch* "recurso"no *git*.

6.2 Melhoria do NONCE

Na defesa a ataques de *replay*, o sistema usa um NONCE que era guardado numa lista de NONCEs usados. No entanto, em grande escala esta lista resulta num consumo grande de memória desnecessário, uma vez que seria necessário guardar todos os números usados, para verificar se era único ou não. A melhoria feita passou pela eliminação dessa lista. O NONCE passou a ser um número incrementado a cada pacote enviado. Com isto, apenas é necessário fazer a verificação no sentido de confirmar que o NONCE do pacote atual é sempre superior ao NONCE anterior.

6.3 Correção da função add doc

Na implementação inicial da função `add_doc`, foi identificado um problema relacionado com o armazenamento do `file_handle`, que é o nome do ficheiro utilizado para guardar o documento. Este estava a ser guardado com codificação `base64`, que permite o uso do caracter `/`. No contexto de caminhos de ficheiros, o caracter `/` é interpretado como uma nova pasta, o que resultava na criação indevida de múltiplas pastas sempre que o `file_handle` continha este caracter.

Para resolver este problema, foi implementada uma solução simples mas eficaz: substituir o caracter `/` por `_` no `file_handle`. Com esta alteração, os documentos passaram a ser armazenados corretamente, sem a criação desnecessária de pastas, garantindo a integridade da estrutura do repositório.

6.4 Criação do script para a função rep get file

Na entrega em época normal, esta função foi feita na sua totalidade. No entanto, não foi criado um *script* de teste para a mesma, pelo que este foi um dos pontos a melhorar. O *script* foi totalmente criado e encontra-se funcional.

6.5 Criação de um script de teste geral

Uma vez que o grupo tinha criado vários *bash scripts* (um para cada função), este método de teste tornou-se demasiado extenso uma vez que para testar o sistema na sua completude era necessário correr todos os scripts individualmente e em sequência. Assim, o grupo identificou como melhoria a criação de um script *run test*. Este script automatiza o processo de teste, descrevendo com detalhe todo o processo no terminal.

6.6 Melhorias na entrega 1 e no relatório

Após a submissão da entrega 1, identificámos oportunidades para melhorar tanto o sistema implementado quanto a estrutura e conteúdo do relatório. As melhorias realizadas nesta fase foram motivadas por feedback recebido durante a avaliação e pela nossa própria análise crítica do trabalho.

6.6.1 Melhorias na entrega 1

Foram realizadas melhorias nesta primeira entrega, principalmente relacionadas ao código desenvolvido. Ao contrário da versão inicial, nesta etapa todos os códigos foram devidamente testados e ajustados, garantindo que as funcionalidades estão bem implementadas e em conformidade com os requisitos do projeto. Estas melhorias refletem o compromisso do grupo em entregar um sistema robusto e funcional, alinhado com as boas práticas de desenvolvimento e segurança.

6.6.2 Melhorias no relatório

Além das melhorias na implementação, o relatório foi aprimorado para refletir mais claramente as decisões técnicas e os resultados alcançados:

- **Adição de detalhes técnicos:** Foram incluídas explicações mais detalhadas sobre as funções implementadas, destacando os algoritmos e lógicas subjacentes.
- **Exemplos mais ilustrativos:** Prints e exemplos de código foram expandidos para incluir contextos mais claros, demonstrando como cada funcionalidade é usada no sistema.
- **Introdução de melhorias no relatório:** Descrevemos claramente as melhorias realizadas, justificando-as com base nas necessidades identificadas e no feedback recebido.