Projeto Segurança Informática

CONSTRUINDO BARREIRAS DIGITAIS: IMPLEMENTAÇÃO DE MODELOS DE SEGURANÇA

A segurança de sistemas informáticos é uma preocupação central na sociedade moderna, onde os dados se tornaram um dos recursos mais valiosos e vulneráveis. Organizações, governos e indivíduos enfrentam constantemente ameaças que podem comprometer a confidencialidade, integridade e disponibilidade de informações críticas. Neste contexto, os **modelos de controlo de acesso** emergem como ferramentas fundamentais para regular quem pode acessar, modificar ou visualizar dados e recursos em sistemas computacionais. Estes modelos formam a base das políticas de segurança e garantem que os sistemas se comportem de forma previsível e controlada.

Entre os modelos de controlo de acesso mais relevantes destacam-se o **Bell-LaPadula** e o **Biba**, amplamente utilizados em cenários onde a segurança é uma prioridade máxima. O modelo **Bell-LaPadula**, criado com foco na proteção da **confidencialidade**, garante que informações classificadas sejam acessíveis apenas por utilizadores devidamente autorizados, prevenindo a divulgação não intencional de dados sensíveis. Já o modelo **Biba**, centrado na **integridade**, assegura que os dados sejam manipulados exclusivamente por fontes confiáveis, prevenindo alterações indevidas que possam comprometer a consistência do sistema.

O presente projeto propõe a implementação prática de um dos dois modelos, Bell-LaPadula ou Biba, utilizando um sistema operativo Linux. A escolha do Linux é particularmente relevante, uma vez que este oferece uma ampla gama de ferramentas nativas para configuração de permissões, grupos e utilizadores, permitindo uma experiência prática enriquecedora para os participantes. O desafio não se limita à aplicação das regras teóricas dos modelos, mas também à integração de mecanismos complementares, como **chaves assimétricas** para encriptação de dados e a criação de um **programa de consulta restrita** configurado com permissões especiais (`setuid`).

Este projeto tem como objetivo principal proporcionar aos alunos uma compreensão aprofundada das complexidades envolvidas na configuração de políticas de segurança. Adicionalmente, busca desenvolver competências práticas, como a configuração de ambientes seguros, o uso de ferramentas criptográficas e a gestão de acessos. Ao final, espera-se que os participantes sejam capazes de aplicar os conceitos aprendidos em cenários reais, resolvendo problemas complexos de segurança com eficiência.

Assim, o projeto não apenas reforça os fundamentos teóricos da segurança informática, mas também capacita os alunos para enfrentar desafios práticos no mundo profissional, onde a proteção de sistemas e dados é uma necessidade constante.

AMBIENTE E DESAFIO

O projeto será desenvolvido em um ambiente Linux Ubuntu, fornecido no âmbito desta disciplina, e preparado para testar a aplicação prática de conceitos de segurança informática. Este ambiente contém os recursos necessários para implementar políticas de controlo de acesso e mecanismos de segurança, utilizando exclusivamente ferramentas nativas disponíveis no sistema operativo.

Os alunos terão acesso inicial ao sistema através de um único utilizador, **`seed`**, que possui privilégios de **sudo**. Este utilizador será o único responsável por criar e configurar o ambiente, incluindo a criação de outros utilizadores, grupos, diretórios e a configuração de permissões de acesso. Qualquer modificação no sistema deve ser realizada a partir desse utilizador, garantindo controle centralizado e alinhamento com os princípios de segurança estabelecidos.

O sistema Ubuntu disponibiliza ferramentas robustas para a gestão de utilizadores e permissões, como `useradd`, `groupadd`, `chmod`, e `chown`. Estas ferramentas serão utilizadas para implementar as regras de segurança baseadas nos modelos **Bell-LaPadula** ou **Biba**, conforme a escolha do grupo. Além disso, os alunos deverão estruturar e proteger diretórios e ficheiros de acordo com os níveis de segurança estabelecidos, aplicando restrições de leitura, escrita e execução.

Outro desafio central é a implementação de **chaves assimétricas**, utilizando ferramentas nativas como `openssl`, `ssh-keygen` ou `gpg`, para encriptação e desencriptação de ficheiros. Este mecanismo deverá assegurar que os dados armazenados nos níveis superiores de segurança estejam protegidos e sejam acessíveis apenas por utilizadores autorizados. A utilização de chaves assimétricas irá introduzir os alunos a um conceito avançado de criptografia, essencial em cenários reais de segurança.

Por fim, os alunos deverão desenvolver um **programa de consulta restrita**, configurado com permissões especiais (`setuid`). Este programa deverá ser capaz de validar e permitir consultas a ficheiros específicos, garantindo que apenas utilizadores autorizados possam acessar os dados, mesmo em situações de tentativas de exploração, como ataques de **path traversal**.

O ambiente e os desafios propostos buscam reproduzir situações reais, desenvolvendo nos alunos habilidades práticas e a capacidade de pensar criticamente sobre segurança informática. A execução correta deste projeto exigirá planeamento, colaboração e uma abordagem rigorosa para atender às exigências de segurança estabelecidas.

POLÍTICA DE SEGURANÇA

A política de segurança deste projeto estabelece um conjunto de normas e diretrizes que os alunos deverão seguir para configurar um sistema seguro, funcional e alinhado aos princípios dos modelos Bell-LaPadula (confidencialidade) ou Biba (integridade). Esta política desempenha um papel central na definição de permissões, acessos e regras que simulam cenários reais de segurança informática, promovendo a aplicação prática dos conceitos teóricos aprendidos.

O sistema deverá ser organizado em três níveis hierárquicos de segurança ou integridade — **alto**, **médio** e **baixo** —, representados por diretórios, utilizadores e grupos. Cada nível reflete diferentes graus de confiança ou criticidade dos dados, replicando situações reais em que o acesso é restrito com base em classificações de segurança. A configuração de permissões deve respeitar as regras específicas do modelo escolhido.

Além da configuração de permissões, a política de segurança exige a aplicação de mecanismos de **criptografia assimétrica**. Os alunos deverão gerar pares de chaves pública e privada para proteger os ficheiros mais críticos, encriptando-os com chaves públicas e permitindo a desencriptação apenas por utilizadores autorizados com acesso às chaves privadas. Este requisito não apenas reforça a segurança do sistema, mas também introduz os alunos à utilização prática de criptografia, que é uma componente essencial em sistemas modernos de proteção de dados.

Um dos aspetos mais desafiadores da política é o desenvolvimento de um **programa de consulta restrita**, configurado com permissões especiais (`setuid`). Este programa deverá permitir que utilizadores de níveis inferiores acessem apenas ficheiros autorizados em níveis superiores, garantindo que o acesso seja controlado e validado. O programa também deverá incluir proteções contra explorações, como ataques de **path traversal**, que tentam manipular o sistema para acessar ficheiros fora do escopo permitido. A implementação deste programa exigirá atenção ao detalhe e compreensão prática dos conceitos de segurança.

Além de abordar as necessidades específicas dos modelos de segurança, a política enfatiza a importância de um processo bem documentado. Os alunos deverão justificar cada configuração, demonstrar como as regras foram aplicadas e fornecer evidências de que as permissões, chaves criptográficas e o programa de consulta respeitam as diretrizes estabelecidas. Essa abordagem não só assegura a conformidade com a política de segurança, mas também prepara os alunos para enfrentar desafios reais em ambientes profissionais.

Em resumo, a política de segurança serve como um guia abrangente para a criação de um sistema seguro, enfatizando a aplicação de regras de acesso, criptografia e desenvolvimento de ferramentas para controle avançado de acessos. Esta política visa não apenas proteger os dados, mas também promover a reflexão crítica e a capacidade de resolver problemas complexos de segurança informática.o e funcional, que respeite as regras teóricas e as traduza para configurações práticas.

1. CONFIGURAÇÃO GERAL DO SISTEMA

1. **Utilizador Inicial:**
   - O utilizador inicial `seed` será o único responsável por todas as configurações e implementações.
2. **Diretórios:**
   - Devem ser criados os seguintes diretórios para simular os níveis de segurança/integridade:
   - `/dados/alto_grupo_<X>`
   - `/dados/médio_grupo_<X>`
   - `/dados/baixo_grupo_<X>`
   - O valor de `<X>` será criado concatenando os dois últimos dígitos dos números de identificação académica dos membros do grupo, separados por "_".
   - Por exemplo, para os números 23345, 3345 e 5678:
   - Diretório: `/dados/alto_grupo_45_45_78`
3. **Ficheiros:**
   - Cada diretório deverá conter um ficheiro para representar os dados no respetivo nível:
   - `/dados/alto_grupo_<X>/confidencial_<X>.txt`
   - `/dados/médio_grupo_<X>/intermediario_<X>.txt`
   - `/dados/baixo_grupo_<X>/publico_<X>.txt`
   - Os ficheiros devem incluir no conteúdo:
   - Os nomes completos dos membros do grupo.
   - Os números de identificação académica dos membros.
4. **Utilizadores:**
   - Criar os seguintes utilizadores para representar diferentes níveis de acesso:
   - `user_grupo_<X>_alto`
   - `user_grupo_<X>_médio`
   - `user_grupo_<X>_baixo`
5. **Grupos:**
   - Associar utilizadores a grupos que representem os níveis de acesso:
   - Grupo `grupo_<X>_alto`: Para `user_grupo_<X>_alto`.

- Grupo `grupo_<X>_médio`: Para `user_grupo_<X>_médio`.
- Grupo `grupo_<X>_baixo`: Para `user_grupo_<X>_baixo`.

REGRAS DE ACESSO BASEADAS NO MODELO ESCOLHIDO

As **regras de acesso** são o núcleo da política de segurança e refletem as características específicas do modelo de segurança escolhido, Bell-LaPadula ou Biba. Estas regras determinam como utilizadores, grupos e processos interagem com os diferentes níveis de dados no sistema, garantindo a aplicação prática dos princípios de **confidencialidade** ou **integridade**, conforme o modelo adotado.

No contexto deste projeto, o sistema é estruturado em três níveis hierárquicos — **alto**, **médio** e **baixo** —, cada um representando diferentes níveis de confiança ou criticidade dos dados. As regras de acesso definem quais operações, como leitura, escrita ou execução, são permitidas ou restritas entre esses níveis, dependendo das propriedades do modelo escolhido. O objetivo é implementar controles que previnam acessos indevidos e modifiquem o comportamento do sistema para respeitar as diretrizes teóricas.

Se o grupo optar pelo **modelo Bell-LaPadula**, as regras deverão garantir que a **confidencialidade** das informações seja preservada. Neste caso, os utilizadores de níveis inferiores não poderão ler dados de níveis superiores (**No Read Up**), e os utilizadores de níveis superiores não poderão escrever dados em níveis inferiores (**No Write Down**). Este modelo é particularmente relevante em cenários como sistemas militares ou governamentais, onde informações classificadas devem ser protegidas contra vazamentos.

Por outro lado, se o grupo escolher o **modelo Biba**, o foco será na **integridade** dos dados. O acesso será regulado de forma a garantir que dados de níveis superiores não sejam comprometidos por interações com níveis inferiores. Assim, os utilizadores de níveis inferiores não poderão escrever dados em níveis superiores (**No Write Up**), e os utilizadores de níveis superiores não poderão confiar em dados de níveis inferiores para leitura (**No Read Down**). Este modelo é ideal para sistemas críticos, como financeiros ou industriais, onde a consistência dos dados é essencial.

Esta secção descreve detalhadamente as regras de acesso que os alunos devem implementar, assegurando que o sistema atenda aos princípios do modelo escolhido. A configuração correta destas regras será essencial para a avaliação do projeto e para o sucesso na criação de um sistema seguro e funcional.

MODELO BELL-LAPADULA

O **Modelo Bell-LaPadula** é um dos principais modelos de segurança utilizados para proteger a **confidencialidade** de informações em sistemas computacionais. Desenvolvido na década de 1970, este modelo foi projetado para atender às necessidades de sistemas militares e governamentais, onde a proteção contra acessos não autorizados a dados classificados é essencial. Sua abordagem baseia-se em garantir que informações sensíveis permaneçam acessíveis apenas a utilizadores com os níveis de permissão apropriados.

O modelo organiza os dados em **níveis de classificação**, como **Alto**, **Médio** e **Baixo**, que representam diferentes graus de sensibilidade ou criticidade. Os utilizadores, por sua vez, são associados a níveis de segurança que determinam o que podem acessar ou modificar. Esta estrutura hierárquica é governada por duas regras fundamentais:

1. **"No Read Up" (NRU):** Um utilizador não pode ler dados classificados em um nível superior ao seu. Esta regra garante que indivíduos sem a devida autorização não acessem informações confidenciais.
2. **"No Write Down" (NWD):** Um utilizador não pode gravar ou transferir dados para um nível inferior ao seu. Esta regra previne a divulgação acidental ou maliciosa de informações sensíveis para ambientes menos seguros.

O objetivo principal do Bell-LaPadula é impedir o **vazamento de informações confidenciais**, seja por erros humanos ou por ações intencionais. É um modelo altamente eficaz em sistemas onde a confidencialidade é mais importante que outros aspectos da segurança, como a integridade ou disponibilidade dos dados.

No entanto, o modelo apresenta algumas limitações. Por exemplo, ele não trata da **integridade** dos dados, ou seja, não impede que informações de níveis inferiores sejam modificadas por níveis superiores. Além disso, pode ser difícil de implementar em sistemas dinâmicos, onde os níveis de classificação mudam frequentemente ou onde há necessidade de grande flexibilidade.

Ao longo do projeto, os alunos que escolherem o modelo Bell-LaPadula terão o desafio de traduzir estas regras teóricas em configurações práticas no sistema operativo Linux. Isso inclui a criação de utilizadores, grupos, diretórios e ficheiros, bem como a definição de permissões que reflitam os princípios de **"No Read Up"** e **"No Write Down"**, garantindo a confidencialidade dos dados armazenados.

MODELO BIBA

O **Modelo Biba** é um modelo de segurança desenvolvido para proteger a **integridade** dos dados em sistemas computacionais. Introduzido em 1977 como uma resposta às limitações do Modelo Bell-LaPadula, o Biba concentra-se em garantir que os dados sejam mantidos corretos, confiáveis e consistentes, prevenindo alterações indevidas que possam comprometer sua validade. Este modelo é amplamente utilizado em sistemas críticos, como financeiros, industriais ou de saúde, onde a integridade dos dados é tão ou mais importante do que a confidencialidade.

No Biba, os dados e utilizadores são organizados em **níveis de integridade**, como **Alto**, **Médio** e **Baixo**, que representam diferentes graus de confiança e criticidade. O acesso é regulado por duas regras principais:

1. **"No Write Up" (NWU):** Um utilizador não pode escrever ou modificar dados em um nível superior ao seu. Esta regra impede que dados críticos sejam corrompidos por fontes menos confiáveis.
2. **"No Read Down" (NRD):** Um utilizador não pode ler dados de um nível inferior ao seu. Esta regra assegura que as decisões em níveis mais críticos não sejam baseadas em informações de fontes potencialmente imprecisas ou não confiáveis.

O principal objetivo do modelo Biba é evitar a **corrupção de dados**, tanto acidental quanto maliciosa, garantindo que apenas utilizadores ou processos confiáveis possam modificar informações sensíveis. Isso o torna ideal para sistemas onde a integridade dos dados é fundamental para a operação segura e confiável, como no processamento de transações financeiras, gestão de infraestruturas críticas ou aplicações médicas.

Apesar de sua robustez na proteção da integridade, o Biba não aborda a **confidencialidade** dos dados, o que significa que ele não impede que informações sensíveis sejam lidas por utilizadores não autorizados. Além disso, como no modelo Bell-LaPadula, a implementação prática pode ser complexa em ambientes dinâmicos que exigem maior flexibilidade.

No projeto, os alunos que optarem pelo modelo Biba terão o desafio de configurar um sistema Linux de forma a aplicar as regras de **"No Write Up"** e **"No Read Down"**, assegurando que as permissões e acessos respeitem as hierarquias de integridade definidas. A criação de utilizadores, grupos, diretórios e ficheiros, juntamente com a configuração de permissões rigorosas, será essencial para demonstrar a eficácia do modelo em prevenir alterações indevidas e preservar a confiabilidade dos dados.

CHAVES ASSIMÉTRICAS

A utilização de chaves assimétricas no projeto tem como principal objetivo proteger e partilhar de forma segura os ficheiros armazenados em diretórios do sistema, garantindo confidencialidade e exclusividade de acesso. Este método baseia-se na utilização de um par de chaves: uma chave pública, que pode ser amplamente distribuída, e uma chave privada, que deve permanecer exclusivamente em posse do utilizador autorizado. Esta abordagem elimina a necessidade de partilha direta de uma chave secreta, reduzindo significativamente os riscos de exposição e garantindo uma camada adicional de segurança.

Com a implementação de chaves assimétricas, os ficheiros sensíveis dos níveis de maior criticidade, como os diretórios de segurança **Alto** e **Médio**, podem ser encriptados de forma a que apenas utilizadores autorizados possam acessá-los.

Para garantir a implementação eficaz e funcional dos objetivos descritos, os seguintes requisitos devem ser cumpridos. Estes requisitos detalham as ações necessárias para que a gestão e utilização de chaves assimétricas sejam realizadas de forma segura e alinhada aos princípios do projeto.

Requisitos de Implementação

1. **Encriptação de Ficheiros Sensíveis:** Os ficheiros armazenados nos diretórios `/dados/alto_grupo_<X>` e `/dados/médio_grupo_<X>` devem ser encriptados utilizando chaves assimétricas. Isso garantirá que os ficheiros sejam inacessíveis a utilizadores não autorizados, mesmo que consigam acessar fisicamente os dados.
2. **Gestão de Chaves:** Cada utilizador deve gerar um par de chaves pública e privada; A chave privada deverá ser armazenada de forma segura e protegida com uma senha robusta. Ela não pode, em hipótese alguma, ser partilhada ou exposta; A chave pública deverá ser distribuída apenas para utilizadores autorizados, garantindo que esses possam encriptar ficheiros destinados ao proprietário da chave privada.
3. **Desencriptação de Ficheiros:** Apenas os utilizadores com a chave privada correspondente devem ser capazes de desencriptar os ficheiros encriptados. A implementação deve demonstrar claramente este processo, assegurando que os dados permanecem protegidos.
4. **Partilha Segura de Dados:** Deve ser demonstrado como um ficheiro sensível pode ser encriptado e partilhado entre utilizadores com diferentes níveis de segurança, respeitando as restrições impostas pelas chaves assimétricas e pelos modelos de segurança escolhidos.
5. **Evidências no Relatório:** O relatório deverá incluir uma descrição detalhada das etapas seguidas para implementar as chaves assimétricas, incluindo a geração de chaves, encriptação, desencriptação e partilha de ficheiros. Testes que comprovem a eficácia do sistema também devem ser documentados.

PROGRAMA DE CONSULTA RESTRITA

Em sistemas de segurança, garantir o acesso controlado a dados sensíveis é essencial para prevenir violações de políticas de acesso e proteger a integridade e a confidencialidade dos dados. No contexto deste projeto, o **programa de consulta restrita** será uma solução prática para validar os princípios de controlo de acesso, assegurando que utilizadores de níveis inferiores possam consultar apenas ficheiros autorizados localizados em diretórios de níveis superiores.

Este programa será configurado com permissões especiais, utilizando o mecanismo de **`setuid`**. O `setuid` permite que um programa seja executado com os privilégios do proprietário do ficheiro (geralmente o utilizador `seed`), mesmo quando o programa é invocado por um utilizador com permissões limitadas. Este mecanismo será utilizado para encapsular um comando nativo do Linux num script ou programa que implemente validações rigorosas antes de permitir o acesso aos ficheiros.

A funcionalidade do programa deve garantir que:

1. Apenas ficheiros previamente autorizados possam ser consultados.
2. Qualquer tentativa de acesso não autorizado, como a exploração de **path traversal** (e.g., manipulação de caminhos para acessar ficheiros fora do escopo permitido), seja bloqueada.

Este componente é um exemplo prático da aplicação de permissões avançadas e será um dos aspectos avaliados para validar o domínio dos alunos sobre segurança no sistema Linux.

Descrição

O **programa de consulta restrita** é uma ferramenta que permitirá utilizadores de níveis inferiores acessar ficheiros autorizados localizados no diretório `/dados/médio_grupo_<X>`. Este programa não só fornece um acesso controlado, mas também introduz os alunos à criação de scripts ou programas que integrem validação, permissões e execução segura.

Ao encapsular um comando nativo, o programa atuará como um intermediário que valida as permissões e as regras de acesso antes de permitir que o ficheiro seja consultado. As permissões especiais `setuid` atribuirão ao programa privilégios do utilizador responsável pelo script, permitindo a execução de ações que estariam fora do alcance dos utilizadores que o invocam diretamente.

Requisitos de Implementação

1. **Criação do Programa:**
    - Criar um script ou programa que encapsule um comando nativo, como a consulta de ficheiros.

- O script deverá incluir:
- Validação do ficheiro solicitado para garantir que ele está no diretório `/dados/médio_grupo_<X>`.
- Verificação de que o ficheiro solicitado está numa lista de permissões pré-definida (whitelist).
- Bloqueio de tentativas de manipulação de caminhos (e.g., uso de `../` para acessar ficheiros fora do diretório permitido).

2. **Configuração de Permissões:**
   - Configurar o programa com permissões especiais `setuid`, de modo que ele execute com privilégios elevados, independentemente do utilizador que o invoca.
   - Garantir que apenas utilizadores específicos possam executar o programa, aplicando permissões de leitura, escrita e execução adequadas.
3. **Proteções contra Explorações:**
   - Implementar validações no programa para evitar vulnerabilidades como:
   - **Path traversal**: Manipulação de caminhos para acessar ficheiros fora do escopo.
   - **Input malicioso**: Utilização de entradas inesperadas para contornar as verificações de segurança.
4. **Testes de Validação:**
   - Realizar testes rigorosos para garantir que o programa:
   - Permite apenas o acesso aos ficheiros autorizados.
   - Bloqueia acessos não autorizados ou fora do diretório permitido.
   - Funciona corretamente mesmo sob tentativas de exploração.
5. **Documentação e Relatório:**
   - Incluir no relatório a descrição do script/programa, os testes realizados e as validações implementadas para garantir a segurança do sistema.

CONSTITUIÇÃO DOS GRUPOS

- **Grupos de 3 alunos** são preferenciais.
- **Grupos de 2 alunos** podem ser formados, mas são menos recomendados.
- Todos os alunos do grupo devem estar **inscritos no mesmo laboratório**.
- Quaisquer cenários que não cumpram estas condições (e.g., grupos de 1 aluno ou alunos de laboratórios diferentes) deverão ser aprovados pelo professor do laboratório.

RELATÓRIO

Os grupos deverão apresentar um relatório detalhado e estruturado, acompanhado por todos os ficheiros necessários para a validação e avaliação do projeto. Este relatório será fundamental para demonstrar as decisões tomadas, os passos realizados e a eficácia da implementação. A seguir, são apresentados os requisitos específicos:

---

1. **Relatório em Formato PDF:**
    - O relatório deverá explicar detalhadamente todas as etapas do projeto, incluindo:
    - Escolha do modelo de segurança (Bell-LaPadula ou Biba) e justificativa da escolha.
    - Planeamento inicial e descrição das configurações aplicadas, como criação de utilizadores, grupos, permissões e diretórios.
    - Implementação das regras do modelo escolhido, com exemplos práticos.
    - Utilização de chaves assimétricas, explicando o processo de geração, encriptação e desencriptação de ficheiros.
    - Criação e configuração do programa de consulta restrita, detalhando sua funcionalidade e proteções implementadas.
    - Testes realizados para validar o cumprimento das políticas de segurança e dos requisitos de implementação.
    - Incluir capturas de ecrã, quando relevante, para ilustrar as configurações e resultados.

---

1. **Logs de Comandos (`comandos.txt`):**
    - Deve ser entregue um ficheiro de texto contendo uma listagem **cronológica e completa** de todos os comandos executados durante o projeto.
    - Cada comando deve ser acompanhado por um comentário que explique brevemente o seu propósito. Por exemplo:

```
    sudo useradd user_grupo_alto  # Criar o utilizador para o nível alto
    chmod 700 /dados/alto_grupo  # Configurar permissões de
leitura/escrita apenas para o proprietário
```

- Este log é essencial para verificar a exatidão das implementações e deve ser suficientemente detalhado para permitir que o processo seja replicado.

---

1. **Ficheiros Gerados:** Os grupos devem incluir todos os ficheiros relevantes que comprovem a execução do projeto, organizados de forma lógica e clara:
    - **Chaves públicas e ficheiros encriptados:**
    - As chaves públicas utilizadas devem estar incluídas, assim como os ficheiros que foram encriptados com elas.
    - **Scripts Criados:**
    - Scripts utilizados para implementar as configurações ou o programa de consulta restrita.
    - **Saídas dos Testes Realizados:**
    - Registos ou capturas de ecrã que demonstrem os resultados dos testes, evidenciando o cumprimento das regras do modelo e a proteção dos ficheiros.

---

1. **Organização em Pasta Compactada:**
    - Todos os ficheiros gerados e o relatório PDF devem ser organizados numa pasta estruturada. A estrutura mínima sugerida é:

```
projeto_grupo_X/
  -- relatório.pdf
  -- comandos.txt
  --  chaves/
       -- chave_publica_user1.asc
       -- chave_publica_user2.asc
       -- ficheiro_encriptado.gpg
  -- scripts/
     -- programa_consulta_restrita.sh
  -- testes/
     --  resultados_testes.txt
     -- capturas/
        -- exemplo_testes.png
  -- outros_ficheiros/
```

- A pasta final deve ser comprimida em `.zip` ou `.tar.gz` com o nome:

> **`projeto_grupo_<X>.zip`** ou **`projeto_grupo_<X>.tar.gz`**, onde `<X>` representa a identificação do grupo.

Esta organização e documentação claras facilitarão a análise, validação e avaliação do projeto, assegurando que todas as etapas sejam compreendidas e replicáveis. <<<

CRITÉRIOS DE AVALIAÇÃO

Critério	Subcritério	Peso (%)
Implementação do Modelo de Segurança	Configuração correta de permissões nos diretórios e ficheiros	15%
	Criação e associação adequada de utilizadores e grupos	10%
	Aplicação consistente das regras do modelo escolhido	10%
Implementação de Chaves Assimétricas	Geração e gestão correta de pares de chaves	5%
	Encriptação e desencriptação de ficheiros conforme as regras de segurança	10%
	Partilha segura de ficheiros utilizando chaves públicas e privadas	5%
Desenvolvimento do Programa de Consulta Restrita	Identificação e criação correta da cópia do comando nativo	5%
	Configuração segura do programa com `setuid`	5%
	Validação rigorosa do acesso a ficheiros autorizados	10%
	Testes de cenários (permitido/negado/path traversal)	5%
Qualidade do Relatório	Organização geral e clareza	5%
	Inclusão de documentação técnica, como comandos e logs	3%
	Reflexão crítica e análise de dificuldades encontradas	2%

PRAZO

Indicado no moodle na aba de submissão do projecto