Hackers do Bem – Fundamental
Prof. Fábio Carneiro de Castro
27/02/2026

Atividade Prática – Módulo 5

Aulas 3 e 4

Gabriel dos Santos Schmitz

Este documento foi criado usando LaTeX
https://github.com/gabrielzschmitz/uni/tree/main/hackers-do-bem/fundamental/modulo5/aulas3-4.

1 Introdução

Este documento apresenta as evidências práticas das atividades do Módulo 5 (Aulas 3 e 4) do Programa Hackers do Bem – Nível Fundamental, por meio dos prints solicitados, demonstrando a correta execução das tarefas propostas.

Conforme as orientações do curso, este documento reúne, em um único arquivo PDF, os seguintes registros obrigatórios: Atividade 5.6 (passo 18), Atividade 5.7 (passo 9), Atividade 5.8 (passo 7), Atividade 5.9 (passo 9) e Atividade 5.10 (passo 5), todos acompanhados de breve descrição explicativa, com o objetivo de facilitar a análise e avaliação por parte do instrutor.

2 Atividades

Atividade 5.6. Implementando o Controle de Acesso Discricionário (DAC) no Windows Server 2022

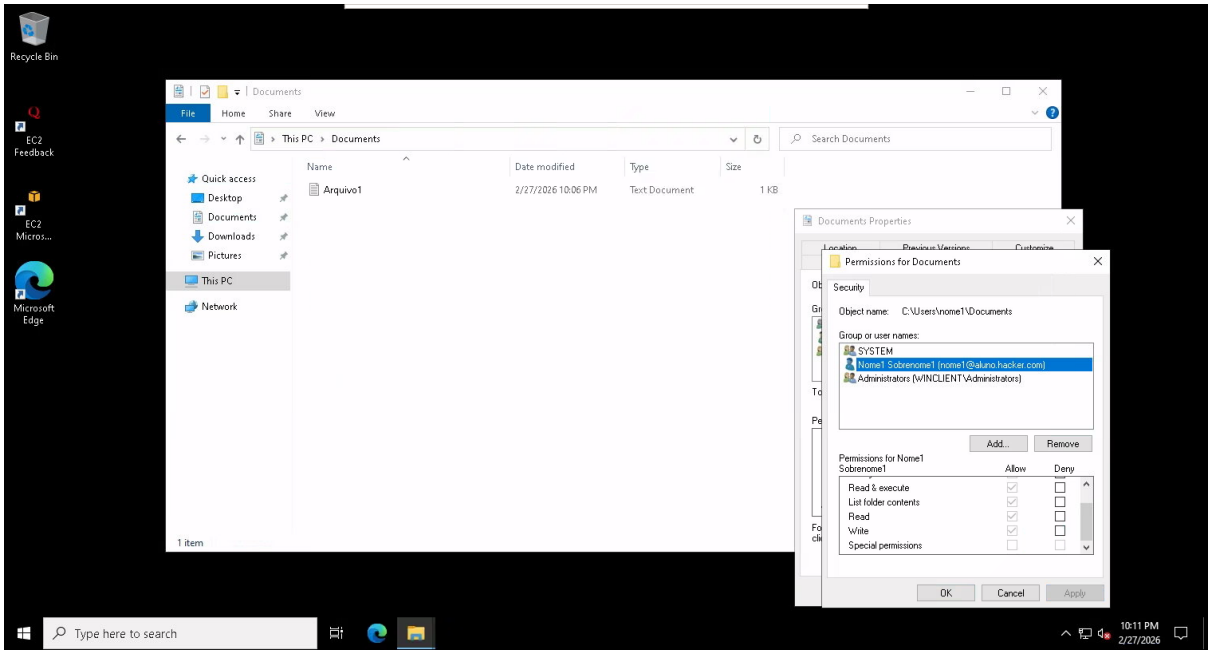

Fig. 1: Configuração de permissões de acesso em arquivos no Windows Server 2022

Sobre:

Nesta atividade, foi realizado o processo de implementação do Controle de Acesso Discricionário (DAC) no Windows Server 2022, demonstrando a aplicação de permissões de segurança em arquivos e diretórios.

Inicialmente, foi estabelecida uma conexão remota ao Windows Server 2022 (cliente) por meio do protocolo RDP, utilizando um usuário de domínio com permissões padrão. Em seguida, foi criado um arquivo de teste por meio do *Notepad*, denominado `Arquivo1.txt`, armazenado na pasta `Documents` do usuário.

Posteriormente, por meio do *File Explorer*, foram acessadas as propriedades da pasta `Documents`, especificamente a aba de segurança (*Security*), onde são exibidos os usuários e grupos com permissões associadas ao recurso.

Na etapa seguinte, foi realizada a análise das permissões existentes, verificando que o grupo *Administrators* possui controle total sobre o diretório. Tentou-se remover esse grupo, porém o sistema impediu a ação, evidenciando a proteção aplicada às contas administrativas.

Em seguida, foram modificadas as permissões do usuário *Nome1*, aplicando-se a negação (*Deny*) de controle total sobre a pasta. Após a confirmação das alterações, o acesso ao diretório foi imediatamente restringido, impedindo a visualização e manipulação dos arquivos.

Ao tentar acessar novamente a pasta `Documents`, o sistema apresentou mensagens de negação de acesso, sendo necessário fornecer credenciais administrativas para prosseguir, o que demonstra a hierarquia de privilégios no sistema operacional.

Por fim, as permissões de negação foram removidas, restaurando o acesso do usuário ao diretório. A validação foi realizada ao acessar novamente a pasta, confirmando que o conteúdo estava disponível.

Atividade 5.7. Criando uma Organizational Unit (OU) no Windows Server 2022

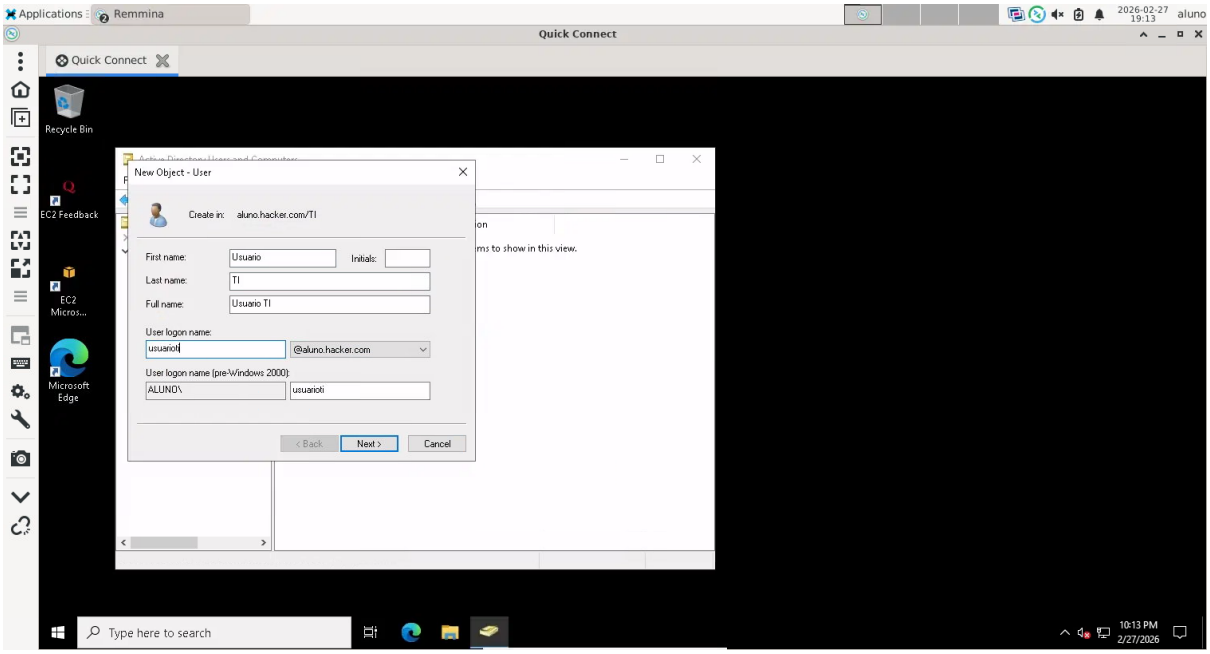

Fig. 2: Criação de uma Unidade Organizacional (OU) e de um usuário no Active Directory

Sobre:

Nesta atividade, foi realizado o processo de criação de uma Unidade Organizacional (*Organizational Unit – OU*) no Windows Server 2022, bem como a adição de um usuário ao domínio, evidenciando a organização hierárquica de objetos no Active Directory.

Inicialmente, foi estabelecida a conexão remota ao servidor por meio do protocolo RDP, utilizando credenciais administrativas. Em seguida, foi acessada a ferramenta *Active Directory Users and Computers*, responsável pelo gerenciamento de objetos no diretório.

Posteriormente, no domínio previamente configurado `aluno.hacker.com`, foi criada uma nova Unidade Organizacional denominada `TI`, com o objetivo de estruturar logicamente os recursos e facilitar a aplicação de políticas e administração de usuários.

Na sequência, foi realizada a criação de um novo usuário dentro da OU `TI`, por meio da opção de inclusão de novos objetos do tipo *User*. Foram definidos os atributos do usuário, incluindo nome, sobrenome e o *User logon name*, que corresponde ao identificador de login (também conhecido como *Common Name*).

Em seguida, foi configurada uma senha de acesso para o usuário, sendo desabilitada a obrigatoriedade de alteração no primeiro logon e ativada a opção de senha sem expiração, garantindo a continuidade do acesso para fins de laboratório.

Após a finalização do assistente, foi possível verificar a criação do novo usuário dentro da OU `TI`, confirmando o sucesso da operação.

Atividade 5.8. Implementando o Controle de Acesso Discricionário (DAC) no Kali Linux

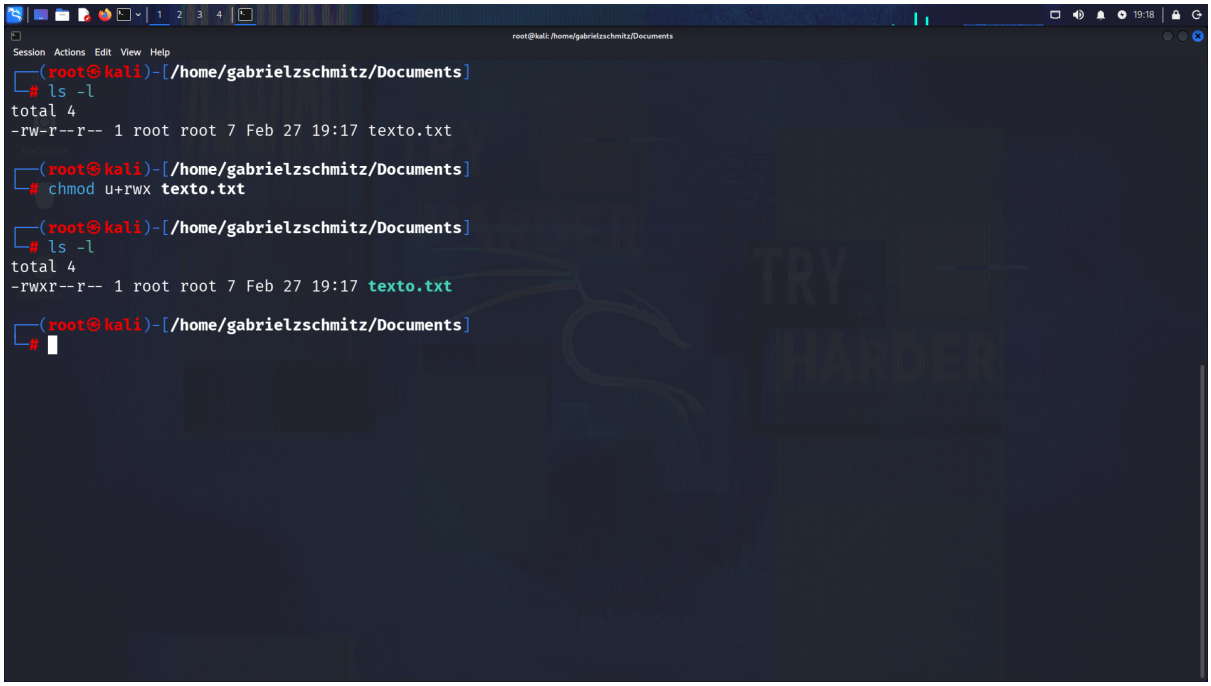

Fig. 3: Alteração das permissões do arquivo texto.txt utilizando o comando chmod no Kali Linux

Sobre:

Nesta atividade, foi realizado o processo de implementação do Controle de Acesso Discricionário (DAC) no Kali Linux, por meio da manipulação de permissões de arquivos no sistema de arquivos.

Inicialmente, foi estabelecida a conexão com a máquina virtual Kali Linux via RDP, utilizando as credenciais fornecidas. Em seguida, foi realizada a elevação de privilégios com o comando `sudo -i`, permitindo a execução de operações administrativas no sistema.

Posteriormente, foi acessado o diretório `/home/aluno/Documentos`, onde foi criado um arquivo de texto denominado `texto.txt` por meio do editor `nano`. No arquivo, foi inserido um conteúdo simples para fins de teste, e em seguida o arquivo foi salvo.

Na sequência, foi utilizado o comando `ls` para listar os arquivos presentes no diretório, confirmando a criação do arquivo. Em seguida, foi executado o comando `ls -l`, que permite visualizar as permissões associadas ao arquivo, bem como informações como proprietário, grupo, tamanho e data de modificação.

Foi observado que, inicialmente, o arquivo possuía permissões `-rw-r--r--`, indicando que o proprietário possuía permissões de leitura e escrita, enquanto o grupo e os demais usuários possuíam apenas permissão de leitura.

Posteriormente, foi aplicado o comando `chmod u+rwx texto.txt`, com o objetivo de conceder ao usuário proprietário permissões completas de leitura, escrita e execução sobre o arquivo. Esse comando é fundamental no modelo DAC, pois permite que o proprietário do arquivo determine quem pode acessá-lo e de que forma.

Após a alteração, foi novamente utilizado o comando `ls -l` para verificar as mudanças, sendo possível observar que as permissões do arquivo foram atualizadas para `-rwxr--r--`, indicando que o proprietário agora possui controle total sobre o arquivo.

Atividade 5.9. Implementando o Acesso Baseado em Função (RBAC) no Kali Linux

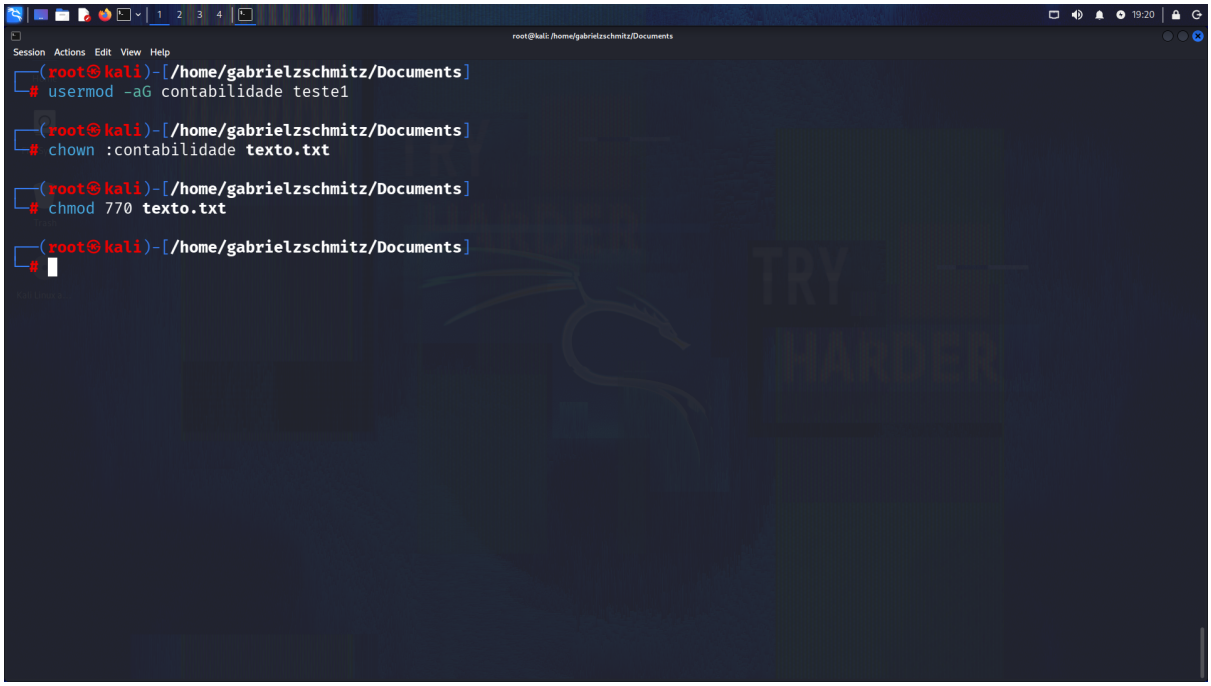

Fig. 4: Configuração de grupo e permissões do arquivo texto.txt para controle de acesso baseado em função no Kali Linux

Sobre:

Nesta atividade, foi realizado o processo de implementação do modelo de Acesso Baseado em Função (RBAC) no Kali Linux, por meio da utilização de grupos e permissões de arquivos.

Inicialmente, foi estabelecida a conexão com o sistema Kali Linux via RDP, seguida da elevação de privilégios com o comando `sudo -i`, permitindo a execução de tarefas administrativas.

Em seguida, foi acessado o diretório `/home/aluno/Documentos`, onde se encontrava o arquivo `texto.txt`, previamente criado. Foi realizada a listagem dos usuários e grupos existentes no sistema por meio dos comandos `getent passwd` e `getent group`, evidenciando a estrutura de contas e grupos do sistema operacional.

Posteriormente, foi criado um novo grupo denominado `contabilidade` utilizando o comando `groupadd`, representando uma função específica no modelo RBAC. Após a criação, foi confirmada a presença do grupo na lista de grupos do sistema.

Na sequência, o usuário `teste1` foi associado ao grupo `contabilidade` por meio do comando `usermod -aG`, atribuindo a esse usuário a função correspondente ao grupo criado.

Em seguida, foi alterado o grupo proprietário do arquivo `texto.txt` utilizando o comando `chown :contabilidade texto.txt`, vinculando o recurso ao grupo responsável pelo acesso.

Posteriormente, foram definidas as permissões do arquivo com o comando `chmod 770 texto.txt`, concedendo permissões completas de leitura, gravação e execução ao proprietário e ao grupo, enquanto restringe totalmente o acesso a outros usuários.

Essa configuração demonstra o modelo RBAC, no qual o acesso aos recursos é controlado com base em funções representadas por grupos, e não diretamente por usuários individuais, facilitando a administração de permissões em ambientes com múltiplos usuários.

Por fim, o arquivo `texto.txt` foi removido, concluindo a atividade. Este procedimento evidenciou como o uso de grupos e permissões no Linux pode ser empregado para implementar

controle de acesso baseado em função, promovendo maior organização e segurança no gerenciamento de recursos.

Atividade 5.10. Implementando a Rotação de Senha no Kali Linux

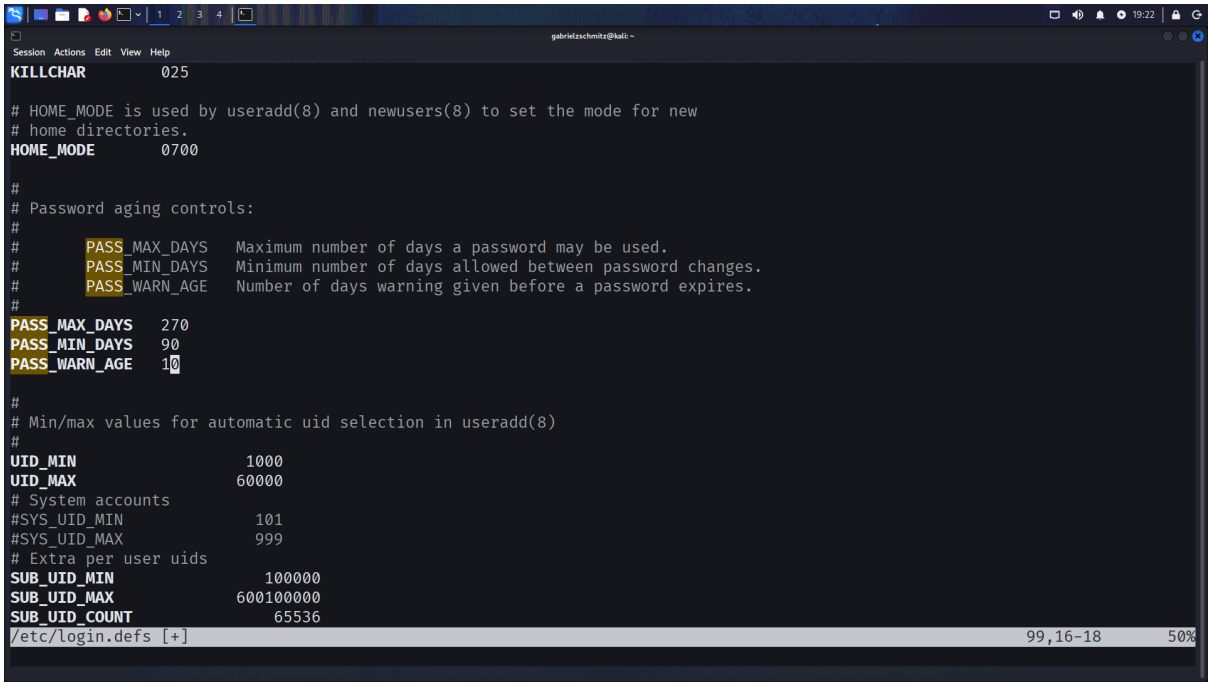

Fig. 5: Edição do arquivo /etc/login.defs para configuração de políticas de rotação de senha no Kali Linux

Sobre:

Nesta atividade, foi realizada a implementação de políticas de rotação de senhas no sistema Kali Linux, com o objetivo de aumentar a segurança das contas de usuário por meio da definição de regras de validade e troca periódica de senhas.

Inicialmente, foi estabelecida a conexão com o sistema via RDP, seguida da abertura do terminal e da elevação de privilégios utilizando o comando `sudo -i`, permitindo a execução de tarefas administrativas.

Em seguida, foram verificados os parâmetros atuais da senha do usuário `root` por meio do comando `passwd -S`, observando informações como status da senha, data da última alteração, período mínimo e máximo de validade, tempo de aviso antes do vencimento e política de desativação da conta. Foi possível identificar que a senha possuía um tempo de validade muito elevado, configurado em `99999` dias.

Posteriormente, foram analisados os parâmetros da senha do usuário `aluno`, utilizando o mesmo comando, verificando que as configurações também apresentavam valores padrão com validade extensa e pouca restrição para troca de senha.

Na sequência, foi realizada a edição do arquivo de configuração `/etc/login.defs` utilizando o editor `nano`, com o objetivo de ajustar as políticas globais de senha do sistema.

Os parâmetros foram modificados de:

- `PASS_MAX_DAYS 99999`
- `PASS_MIN_DAYS 0`
- `PASS_WARN_AGE 7`

para:

- `PASS_MAX_DAYS 270`

- `PASS_MIN_DAYS 90`
- `PASS_WARN_AGE 10`

Essas alterações estabelecem que a senha passa a ter validade máxima de 270 dias, não podendo ser alterada antes de 90 dias após a última mudança, além de notificar o usuário com 10 dias de antecedência sobre o vencimento da senha.

Após realizar as modificações, o arquivo foi salvo e o editor encerrado. Com isso, as novas políticas passam a ser aplicadas para novos usuários e para configurações futuras de senha.