Aula 19 — Comandos, Arquivos, Usuários e Permissões em Windows e Linux

UC01 • Planejar e executar a instalação de redes locais | Encontro 19 de 27 | 3h30 | Turma TR5

Competências que desenvolveremos

- Organizar arquivos e diretórios de um laboratório de redes sem alterar recursos de produção.
- Identificar contas de usuário, grupos e o princípio do menor privilégio.
- Consultar permissões NTFS no Windows e interpretar permissões rwx no Linux.
- Aplicar permissão de leitura no Linux a um arquivo próprio, mantendo outros usuários sem acesso.
- Documentar comandos, resultados e justificativas para o projeto Escola Horizonte.

Roteiro de 210 minutos

Etapa	Duração
Abertura e problema da escola	15 min
Arquivos, caminhos, usuários e grupos	35 min
Permissões Windows x Linux	35 min
Demonstração comparativa orientada	25 min
Laboratório prático em duplas	70 min
Desafio e registro de evidências	20 min
Síntese e Teams	10 min

Situação-problema — Escola Horizonte

Após preparar as estações da Aula 18, a equipe precisa organizar a pasta de documentação da rede. Os técnicos devem editar o inventário, os docentes poderão consultá-lo e usuários visitantes não devem ter acesso. Como demonstrar esse controle com evidências, sem modificar contas ou permissões institucionais?

1 | Arquivos, diretórios, contas e grupos

Arquivo é a unidade lógica usada para armazenar dados; diretório organiza arquivos e outros diretórios. Um caminho absoluto indica a localização desde a raiz ou unidade; um caminho relativo é interpretado a partir do diretório de trabalho. O usuário autenticado executa processos com uma identidade e direitos definidos. Grupos ajudam a atribuir permissões sem administrar cada conta individualmente.

Conceito	Windows	Linux
Caminho típico	C:\Users\Aluno\Documents	/home/aluno/documentos
Listar arquivos	Get-ChildItem	ls -l
Identidade atual	whoami	whoami / id

Grupos	whoami /groups	groups / id
Permissões de arquivo	Get-Acl / icacls	ls -l / chmod

ATENÇÃO Nos notebooks do Senac, faça alterações somente em uma pasta de testes autorizada. Não crie contas reais, não use contas administrativas, não altere ACLs de pastas institucionais e não realize operações destrutivas.

2 | Windows: ACLs e permissões NTFS

Em volumes NTFS, as permissões são representadas por listas de controle de acesso (ACLs), com entradas para usuários e grupos. Os direitos podem incluir leitura, gravação, modificação ou controle total, e muitas permissões são herdadas das pastas superiores. A leitura com Get-Acl é uma operação de observação; a mudança de ACL não é necessária nesta aula.

3 | Linux: proprietário, grupo e outros

No modelo POSIX clássico, o arquivo apresenta proprietário, grupo e demais usuários. Cada classe recebe r (leitura), w (escrita) e x (execução). Em diretórios, x permite atravessar o caminho; r permite listar nomes e w, normalmente junto com x, permite criar ou remover entradas. Sistemas Linux também podem ter ACLs adicionais; rwx não esgota todas as possibilidades.

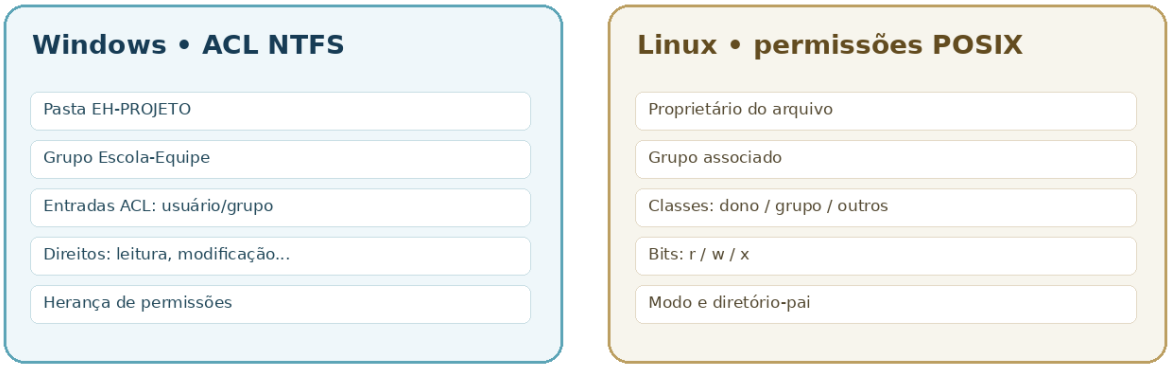


Figura 19.1 — Comparação conceitual. ACL NTFS e bits POSIX não são mecanismos idênticos; ambos ajudam a controlar o acesso a arquivos.

Permissão POSIX	Proprietário	Grupo	Outros	Interpretação
640	rw-	r--	---	Dono edita, grupo lê, outros não acessam
600	rw-	---	---	Somente o dono pode ler/escrever
644	rw-	r--	r--	Outros podem ler; inadequado a conteúdo sigiloso
777	rwx	rwx	rwx	Excessivo; não utilizar na oficina

4 | Laboratório A — Windows PowerShell (somente pasta temporária)

Abra o PowerShell com sua conta comum. O roteiro abaixo cria arquivos dentro da pasta temporária do seu usuário. Se a política do laboratório impedir a execução de um comando, registre a restrição sem contorná-la.

```
$p = Join-Path $env:TEMP 'TR5_Aula19'
New-Item -ItemType Directory -Path $p -Force | Out-Null
'Escola Horizonte - inventario de teste' | Set-Content (Join-Path $p 'inventario.txt')
Get-ChildItem -Path $p
whoami
whoami /groups
Get-Acl -Path $p | Format-List Owner,Access
Get-Acl -Path (Join-Path $p 'inventario.txt') | Format-List Owner,Access
icacls $p
```

Interprete: quem aparece como proprietário? Há entradas herdadas? Qual é a diferença entre mostrar as permissões da pasta e as do arquivo? Não execute /grant, /deny ou /reset nesta aula.

5 | Laboratório B — Debian (arquivo do próprio aluno)

Use a VM Debian autorizada da Aula 18. Não utilize sudo. Crie uma pasta dentro de HOME, registre o modo inicial e altere somente o arquivo criado por você.

```
mkdir -p "$HOME/tr5_aula19"
printf '%s\n' 'Escola Horizonte - inventario de teste' > "$HOME/tr5_aula19/inventario.txt"
whoami
id
groups
ls -ld "$HOME/tr5_aula19"
ls -l "$HOME/tr5_aula19/inventario.txt"
chmod u=rw,g=r,o= "$HOME/tr5_aula19/inventario.txt"
ls -l "$HOME/tr5_aula19/inventario.txt"
cat "$HOME/tr5_aula19/inventario.txt"
```

Com a configuração u=rw,g=r,o=, o arquivo terá permissões equivalentes a 640 em um sistema POSIX usual. Observe que o acesso real também depende das permissões dos diretórios superiores, dos grupos de que a pessoa participa e de possíveis ACLs adicionais.

6 | Desafio em equipe — Política de acesso da escola

A direção da escola estabelece três perfis: equipe de TI (edição do inventário), docentes (consulta) e visitantes (sem acesso). Em dupla, desenhe uma matriz de acesso para a pasta “EH-PROJETO” e explique como representaria a política no Windows (grupos/ACL) e em um Linux dedicado (proprietário, grupo e permissões). Não é necessário implementar três usuários no laboratório.

Perfil	Pode consultar?	Pode alterar?	Justificativa
Equipe de TI	_____	_____	_____
Docentes	_____	_____	_____
Visitantes	_____	_____	_____

7 | Registro de evidências para o Teams

1. Identifique o SO e o local de testes (sem nomes de usuário pessoais nas capturas).

2. Anexe uma evidência do arquivo criado em cada SO, ou explique a limitação do laboratório.

3. Registre uma saída de Get-Acl no Windows e uma de ls -l após chmod no Debian.

4. Explique em 4 a 6 linhas por que 640 oferece menor exposição do que 777.

5. Preencha a matriz de acesso e indique uma dúvida a ser aprofundada na Aula 20.

IMPORTANTE Não anexe senhas, nomes de contas pessoais, dados da rede administrativa ou telas contendo informações de terceiros. Se houver conta institucional, resuma os resultados em vez de capturar dados sensíveis.

8 | Estudos de caso rápidos

CASO A — Uma equipe informa que o arquivo inventario.txt tem permissão rw-r-----, mas um docente não consegue abri-lo. Investigue: o docente está no grupo esperado? O diretório pai permite a travessia? Existe ACL adicional? Qual evidência faltou?

CASO B — No Windows, a ACL de uma pasta possui várias entradas herdadas. Um administrador propõe conceder Controle Total ao grupo Todos apenas para corrigir um acesso negado. Explique por que a ação seria inadequada e indique duas verificações de menor risco.

9 | Verificação rápida

1. Qual comando do Windows consulta ACL sem modificá-la?

2. O que significam os três grupos de bits em rw-r-----?

3. Por que conceder 777 a um arquivo de documentação é inadequado?

4. A permissão x em um diretório significa a mesma coisa que em um arquivo? Justifique.

5. Por que uma pasta compartilhada da Aula 20 exigirá avaliar permissões locais e de compartilhamento?

Referências para consulta

MICROSOFT. Get-Acl (PowerShell). https://learn.microsoft.com/pt-br/powershell/module/microsoft.powershell.security/get-acl

MICROSOFT. icacls (Windows Commands). https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/icacls

GNU. Coreutils Manual — chmod e estrutura de permissões. https://www.gnu.org/software/coreutils/manual/html_node/chmod-invocation.html

GNU. Coreutils Manual — Mode Structure. https://www.gnu.org/software/coreutils/manual/html_node/Mode-Structure.html