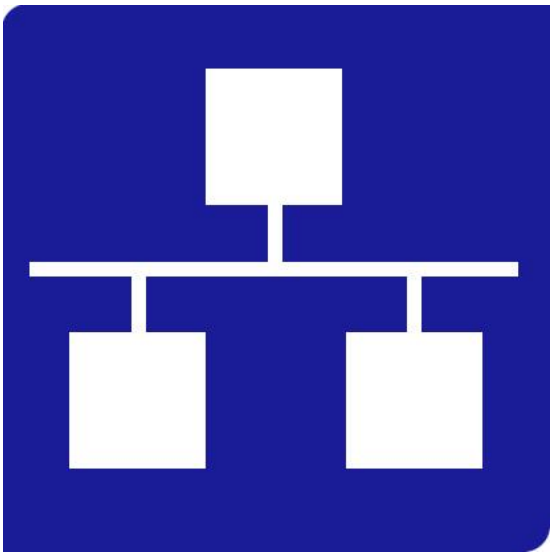
MANUAL DE FORMAÇÃO



Curso	Técnico/a de Informática – Instalação e Gestão de Redes
Duração	2045 Horas
Módulo/Unidade	**0840 – Servidores web**
Duração	**50**
Forma de Organização	☒ Presencial ☐ eLearning ☐ bLearning
Autoria (Formador/a)	**João Silva**
Direitos Reservados	Este manual é da autoria do Formador referido, o qual assume todos os direitos de autor relativos aos conteúdos aqui desenvolvidos. Foi entregue à Talentus para sua utilização como Recurso Técnico-Pedagógico no âmbito deste curso.

Cofinanciado por:

POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

Formação com Garantia

Índice

Objetivos do Manual

Os servidores web são componentes essenciais para o funcionamento da Internet, pois são responsáveis por armazenar, processar e entregar os arquivos que compõem os sites para os navegadores dos usuários. Neste manual, aprenderá o que são servidores web, como funcionam, quais são os tipos de servidores web (estáticos e dinâmicos), quais são os principais recursos de servidores web e quais são os servidores web mais utilizados e suas aplicações práticas.

Destinatários

Este manual é destinado a formandos de nível médio que desejam ampliar seus conhecimentos sobre os servidores web e suas funcionalidades. O manual não requer conhecimentos prévios sobre o assunto, mas é recomendável ter noções básicas de informática e Internet. O manual também oferece exercícios e atividades.

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

Pré-requisitos

Ter um Virtualbox com ubuntu instalado e configurado
Ter acesso à Internet e a um navegador web
Saber como usar o terminal do ubuntu e os comandos básicos de linux
Ter interesse em aprender sobre os servidores web e suas aplicações

Introdução

Neste manual, aprenderá como instalar e configurar um servidor web no Linux Ubuntu. Neste manual, aprenderá sobre os principais protocolos e serviços de computação remota, que permitem a comunicação e a troca de dados entre diferentes computadores conectados em rede. A computação remota é uma forma de aceder e utilizar recursos de um computador à distância, como se estivesse localmente. O manual está dividido em quatro capítulos:

• No primeiro capítulo, aprenderá sobre os protocolos TELNET, RLOGIN e SSH, que permitem o acesso remoto ao terminal de um computador. aprenderá como instalar e configurar os programas necessários para cada protocolo, como criar e administrar usuários e senhas para o acesso remoto, como estabelecer uma conexão segura com o computador remoto e como executar comandos e programas no terminal remoto.

• No segundo capítulo, aprenderá sobre o protocolo FTP, que permite a transferência de arquivos entre computadores remotos. aprenderá como instalar e configurar o programa FTP, como criar e administrar usuários e senhas para o FTP, como configurar as permissões de acesso aos arquivos remotos e como transferir arquivos entre computadores remotos utilizando um cliente FTP.

• No terceiro capítulo, aprenderá sobre o serviço TALK, que permite a comunicação em tempo real entre usuários de computadores remotos. aprenderá como instalar e configurar o programa TALK, como iniciar e encerrar uma conversa com um usuário remoto, como enviar e receber mensagens de texto e voz no TALK e como administrar as solicitações de conversa.

• No quarto capítulo, aprenderá sobre o serviço NFS, que permite a partilha de arquivos e diretórios entre computadores remotos. aprenderá como instalar e configurar o programa NFS, como criar e administrar os pontos de montagem dos arquivos e diretórios compartilhados, como aceder e modificar os arquivos e diretórios compartilhados e como controlar o acesso aos arquivos e diretórios compartilhados.

Ao final deste manual, você terá os conhecimentos necessários para utilizar os protocolos e serviços de computação remota de forma eficiente e segura. Você também terá uma visão geral sobre as vantagens e desvantagens da computação remota em diferentes cenários.

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

O login remoto é um recurso que permite aos utilizadores acederem ao sistema ou servidor remotamente, ou seja, de um local diferente onde o sistema está fisicamente localizado. Isso proporciona a conveniência de administrar, administrar e executar comandos num sistema sem a necessidade de estar fisicamente presente nele.

O Telnet é um serviço cliente amplamente utilizado para realizar login remoto em servidores. Permite que os utilizadores se conectem a um servidor Telnet usando um cliente Telnet em seu próprio sistema. O cliente Telnet estabelece uma conexão com o servidor Telnet e permite ao utilizador interagir com o sistema remoto através de um terminal virtual.

O Telnet é um protocolo de rede que permite aos utilizadores acederem remotamente um servidor e interagirem com ele como se estivessem fisicamente presentes no sistema. No entanto, é importante destacar que o Telnet não fornece nenhum tipo de criptografia ou autenticação segura, o que o torna um método de login remoto não recomendado em ambientes de produção.

Apesar de sua ampla utilização no passado, o Telnet é considerado inseguro devido ao fato de que todas as informações transmitidas, incluindo senhas e comandos, são enviadas em texto simples, o que possibilita a interceção e a leitura não autorizada dos dados por terceiros.

Devido a suas vulnerabilidades de segurança, o Telnet foi amplamente substituído por protocolos mais seguros, como o SSH (Secure Shell), que oferece criptografia forte e autenticação segura.

Recomenda-se evitar o uso do Telnet em ambientes de produção e, em vez disso, utilizar métodos de login remoto mais seguros, como o SSH, que protege as informações transmitidas durante a sessão de acesso remoto.

É importante priorizar a segurança ao aceder aos sistemas remotamente. Ao optar por soluções de login remoto, escolha métodos que ofereçam criptografia robusta, autenticação segura e proteção das informações confidenciais transmitidas durante a sessão remota.

O RLOGIN é um protocolo semelhante ao Telnet, que também permite o acesso remoto a um servidor usando uma conexão de terminal. No entanto, assim como o Telnet, o RLOGIN não é seguro, pois os dados são transmitidos em texto simples, o que representa riscos de segurança.

O SSH é um protocolo seguro para acesso remoto a um servidor. Oferece criptografia e autenticação robustas, tornando a conexão remota segura. O SSH substituiu amplamente o Telnet e o RLOGIN em ambientes de produção devido à sua segurança aprimorada. Ele permite o acesso remoto a um servidor com uma interface de linha de comando segura e a capacidade de transferir ficheiros de forma segura (usando o utilitário SCP).

O FTP é um protocolo usado para transferir ficheiros entre sistemas. Ele permite que um utilizador se conecte a um servidor remoto e transfira ficheiros de e para o servidor. No entanto, o FTP também não é um protocolo seguro, pois as informações são transmitidas em texto simples. É recomendado usar variantes seguras do FTP, como o FTPS (FTP com SSL/TLS) ou o SFTP (SSH File Transfer Protocol), que fornecem criptografia e autenticação adicionais.
É importante destacar que, para garantir a segurança dos sistemas e dados, é altamente recomendável utilizar o SSH e suas variantes seguras (FTPS, SFTP) para o acesso e transferência de ficheiros remotos. O Telnet e o RLOGIN devem ser evitados em ambientes de produção devido às suas vulnerabilidades de segurança.

Certifique-se de seguir as melhores práticas de segurança ao configurar e utilizar os serviços de acesso remoto, como a configuração adequada de senhas fortes, a utilização de chaves de autenticação e a proteção do servidor contra acesso não autorizado.

1.1.1 Exercícios práticos

Instale o servidor SSH num sistema Linux.
Configure o arquivo de configuração do SSH para permitir apenas a autenticação baseada em chave e desabilitar o acesso root remoto.
Gere um par de chaves SSH (chave pública e privada) num cliente.
Copie a chave pública para o servidor SSH e configure a autenticação baseada em chave para um cliente específico.
Tente se conectar ao servidor SSH usando a chave privada do cliente, autenticando-se sem a necessidade de digitar uma senha.
Transferência segura de ficheiros:

Instale um servidor FTP seguro (FTPS ou SFTP) num sistema Linux.
Configure o servidor FTP para usar criptografia SSL/TLS ou SSH para garantir a segurança das transferências de ficheiros.
A partir de um cliente, conecte-se ao servidor FTP seguro e realize transferências de ficheiros para o servidor e vice-versa.
Verifique se as transferências de ficheiros estão criptografadas e protegidas contra interceção.

Configurando restrições de acesso remoto:

Aceda o arquivo "/etc/securetty" num sistema Linux.
Edite o arquivo para restringir as conexões remotas apenas a determinadas portas.
Reinicie o serviço de acesso remoto correspondente (por exemplo, SSH) para aplicar as alterações.
Tente se conectar remotamente ao sistema usando uma porta não permitida e verifique se a conexão é rejeitada.

Desabilitando o Telnet e o RLOGIN:

Aceda às configurações do sistema Linux que controlam os serviços Telnet e RLOGIN. Desabilite os serviços Telnet e RLOGIN para aumentar a segurança do sistema. Verifique se os serviços estão desabilitados tentando fazer a conexão remota pelo Telnet ou RLOGIN.

2 Telnet

O Telnet é um protocolo de acesso remoto que permite a conexão a um servidor através de um terminal remoto. O nome Telnet vem da junção teletype e network, significa rede de teletipos, permite a comunicação em texto plano e comunicação bididirecional através de uma conexão em terminal virtual. Foi criado por volta de 1969 pelas forças armadas americanas para transmissão de dados entre bases militares.

A **configuração do serviço Telnet** envolve ajustar as configurações no servidor Telnet para permitir ou restringir o acesso remoto. Isso inclui definir permissões de acesso, configurar autenticação, definir portas de escuta, entre outros parâmetros relevantes. Consulte a documentação específica do servidor Telnet em uso para obter instruções detalhadas sobre como configurar o serviço.

Instale o pacote do servidor Telnet usando o gerenciador de pacotes do seu sistema operacional. Por exemplo, no Ubuntu, pode usar o seguinte comando: **sudo apt-get install telnetd**
Após a instalação, o serviço Telnet será iniciado automaticamente.
Verifique se o serviço está em execução digitando o comando: **sudo service telnet status**
Configure as permissões e restrições de acesso do serviço Telnet de acordo com suas necessidades. Por exemplo, pode definir limites de conexão por IP ou restringir o acesso a determinados utilizadores.

Aceda o arquivo de configuração do servidor Telnet, geralmente localizado em **/etc/inetd.conf.**

Localize a linha que contém a configuração do Telnet, geralmente começa com telnet seguido de um espaço.

Verifique se a linha não está comentada (sem o caractere # no início da linha). Se estiver comentada, remova o caractere # para ativá-la.

Defina as opções de configuração conforme necessário, como definir a porta de escuta, as permissões de acesso e a autenticação.

Salve as alterações e reinicie o serviço Telnet para aplicar as configurações.

Cofinanciado por:

2.1 Uso do serviço Telnet para administração remota:

O Telnet permite que se conecte remotamente a um servidor e execute comandos administrativos diretamente através de um terminal remoto. Para usar o serviço Telnet para administração remota, siga estes passos:

Com o cliente Telnet (como o PuTTY), insira o endereço IP do servidor e o número da porta Telnet.
Conecte-se ao servidor Telnet digitando as credenciais de autenticação necessárias.
Uma vez conectado, poderá executar comandos no servidor remotamente como se estivesse usando um terminal local.

2.2 Execução de aplicativos remotamente via Telnet:

Além de executar comandos administrativos, o Telnet também permite a execução de aplicativos remotamente. Para isso, siga estes passos:

Exemplo:

Conecte-se ao servidor Telnet usando um cliente Telnet.
Navegue até o diretório onde o aplicativo está localizado no servidor.
Execute o aplicativo digitando seu nome ou caminho completo no terminal remoto.
O aplicativo será iniciado e poderá interagir com ele remotamente.

2.3 Shutdown remoto:

O Telnet também pode ser usado para desligar um servidor remotamente. Para realizar um shutdown remoto usando o Telnet, siga estes passos:

Conecte-se ao servidor Telnet usando um cliente Telnet.
Digite o comando de desligamento específico do sistema operacional. Por exemplo, no Linux, pode usar o comando **shutdown -h now** para desligar o servidor imediatamente.

Desabilitação do serviço Telnet:

Por motivos de segurança, é recomendável desabilitar o serviço Telnet, pois ele transmite informações em texto simples. Para desabilitar o serviço Telnet num servidor Linux, siga estes passos:

Aceda o servidor Linux através de um terminal.
Execute o comando para parar o serviço Telnet. Por exemplo, no Ubuntu, pode usar o seguinte comando: **sudo service telnet stop**
Desative o serviço Telnet para que ele não seja iniciado

durante a inicialização do sistema. Por exemplo, no Ubuntu, pode usar o seguinte comando: **sudo systemctl disable telnet**

2.4 Impedimento do servidor de uso do serviço Telnet:

Para impedir o servidor de permitir o uso do serviço Telnet, pode bloquear a porta padrão do Telnet na firewall ou configurar uma política de segurança que negue o acesso ao serviço Telnet.

2.5 Outras formas de administração remota:

Além do Telnet, existem outras opções mais seguras para a administração remota de servidores, como o SSH (Secure Shell) e ferramentas de gerenciamento remoto baseadas na web, como o cPanel, o Webmin ou o Plesk. Essas ferramentas oferecem recursos avançados de segurança e administração remota.

2.6 Servidor Linux - Acesso a ficheiros do servidor mesmo sem Telnet:

Mesmo sem o uso do serviço Telnet, é possível aceder os ficheiros do servidor Linux remotamente usando outras ferramentas, como o SSH, FTP (File Transfer Protocol) ou SCP (Secure Copy). Essas ferramentas permitem transferir ficheiros entre o servidor e o cliente de forma segura.

3 RLOGIN

O **rlogin é um serviço** de acesso remoto que permite que os utilizadores se conectem a um servidor usando autenticação baseada em senha. No entanto, esse serviço não é seguro e é recomendado desabilitá-lo para evitar riscos de segurança.

O nome rlogin vem da junção de remote e login, que significa login remoto. O login é o processo de se conectar a um sistema informático restrito usando credenciais como nome de utilizador e senha.

O rlogin permite que as credenciais do cliente sejam armazenadas num arquivo rhosts local, para que o utilizador se possa conectar. O Telnet não tem essa funcionalidade e requer que o utilizador digite seu nome de utilizador e senha a cada vez que se conecta. Além disso, o telnet permite ao utilizador passar variáveis de ambiente como parâmetros, enquanto o rlogin não. No entanto, ambos os protocolos são inseguros, pois enviam as mensagens em texto sem formatação, que podem ser facilmente interceptadas e lidas por alguém na rede. Por isso, é recomendado usar o SSH, que é um protocolo mais seguro e criptografado.

Para desabilitar o rlogin, Aceda o arquivo de configuração relevante no sistema operacional (geralmente **/etc/xinetd.d/rlogin**) e altere a linha "**disable = no**" para "**disable = yes**". Salve as alterações e reinicie o serviço xinetd para que as configurações tenham efeito.

Por padrão, os serviços de acesso remoto como SSH, Telnet e FTP estão **associados a portas específicas**. No entanto, é possível configurar esses serviços para usar portas diferentes, o que pode ajudar a aumentar a segurança e dificultar ataques automatizados. Para realizar a ligação a outras portas, aceda os ficheiros de configuração dos respectivos serviços (por exemplo, **/etc/ssh/sshd_config** para o SSH) e procure a linha que define a porta. Altere o número da porta para o desejado e salve as alterações. Reinicie o serviço para aplicar as configurações.

O **arquivo /etc/securetty** lista os terminais nos quais o acesso remoto é permitido para a conta root. É importante revisar e configurar corretamente esse arquivo para limitar o acesso remoto ao root apenas a terminais específicos.
Abra o arquivo /etc/securetty e remova quaisquer linhas que não sejam necessárias para restringir o acesso remoto do root. Por exemplo, podem ser removidas as entradas relacionadas a terminais de acesso físico ou virtuais não utilizados para acesso remoto. Salve as alterações.

3.1 Desabilitação do acesso remoto do root:

É altamente recomendado desabilitar o acesso remoto direto à conta root, pois isso reduz o risco de ataques bem-sucedidos e ajuda a evitar problemas de segurança.
No arquivo de configuração do serviço relevante (por exemplo, /etc/ssh/sshd_config para o SSH), encontre a linha que define a opção "PermitRootLogin". Mude o valor para "no" para desabilitar o acesso remoto direto à conta root. Salve as alterações e reinicie o serviço.

3.2 Configuração dos serviços em xinetd.d

O xinetd é um daemon do sistema que gerencia serviços de rede no Linux. Ele permite configurar e controlar serviços de acesso remoto de forma centralizada.
Para configurar serviços em xinetd.d, Aceda o diretório /etc/xinetd.d/ e localize o arquivo de configuração correspondente ao serviço desejado (por exemplo, telnet, ftp). Abra o arquivo e edite as opções conforme necessário, como restrições de acesso, limites de conexão, autenticação, entre outros. Salve as alterações e reinicie o serviço xinetd.

3.2.1 Exercícios práticos

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

Descreva os passos necessários para desabilitar o serviço rlogin num sistema Linux.

4 SSH

O nome SSH vem da junção de Secure Shell, que significa cápsula segura. O SSH é um protocolo que permite se conectar a um servidor remoto pela internet, de maneira segura e criptografada. O SSH foi criado em 1995 por Tatu Ylönen, cientista finlandês, como alternativa mais segura ao Telnet e ao rlogin.

4.1 Comparação do SSH ao telnet/ftp e o rlogin

O SSH é um protocolo que se diferencia do Telnet, do FTP e do rlogin pela segurança. O SSH usa criptografia para proteger os dados que são enviados e recebidos pela rede, enquanto os outros protocolos usam texto simples. Significa que SSH evita que alguém intercepte e leia as informações trocadas entre cliente e o servidor. Além disso, o SSH permite fazer transferências seguras de ficheiros, usando os protocolos SCP ou SFTP. O SSH usa a porta TCP 22 por padrão, Telnet usa a porta TCP 23, o FTP a porta TCP 21 e o rlogin usa a porta TCP 513.

4.2 Logon em máquinas remotas com o SSH

Para fazer login em máquinas remotas com o SSH, é necessário ter um par de chaves SSH, que consiste numa chave privada e uma chave pública. A chave privada fica no seu computador local e deve ser protegida com uma senha. A chave pública deve ser copiada para o servidor remoto e adicionada ao arquivo **~/.ssh/authorized_keys** na conta de utilizador que você deseja aceder. Depois de configurar as chaves SSH, pode usar o comando ssh para se conectar ao servidor remoto, especificando o nome de utilizador e o endereço IP ou o domínio do servidor. Por exemplo:

ssh utilizador@192.168.1.192

O SSH solicita a senha da sua chave privada e depois abre a Shell remota onde poderá executar comandos no servidor remoto como se estivesse no terminal local. Para encerrar a sessão SSH, digite exit ou pressione Ctrl+D .

4.3 Copia de ficheiros pelo SSH(scp)

A cópia de ficheiros é uma tarefa comum na administração de sistemas. O SSH (Secure Shell) fornece uma maneira segura de copiar ficheiros entre um cliente e um servidor remoto por meio do comando "scp" (Secure Copy). Muito útil para fazer backup, sincronizar ou compartilhar ficheiros entre diferentes sistemas.

O comando SCP tem a seguinte sintaxe geral:

scp [opções] origem destino

Onde:

• opções são parâmetros opcionais que modificam o comportamento do comando. Algumas opções comuns são:

-r para copiar recursivamente os diretórios e seus conteúdos.

-p para preservar as permissões e os tempos dos ficheiros.

-v para aumentar a verbosidade da saída.

-q para suprimir a saída.

-P para especificar uma porta diferente da porta padrão (22).

-i para especificar um arquivo de identidade (chave SSH) para a autenticação.

origem é o arquivo ou diretório que será copiado. Pode ser um caminho local ou remoto. Um caminho remoto tem a seguinte forma:

user@host:/caminho

Onde:

user é o nome do utilizador no sistema remoto.

host é o endereço IP ou o nome de domínio do sistema remoto.

/caminho é o caminho absoluto ou relativo do arquivo ou diretório no sistema remoto.

destino é o arquivo ou diretório onde a cópia será salva. Pode ser um caminho local ou remoto, seguindo a mesma forma da origem.

Exemplos de uso do comando SCP
A seguir, vamos ver alguns exemplos práticos de como usar o comando SCP para transferir ficheiros entre diferentes sistemas.

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

Para copiar um arquivo local para um sistema remoto, basta especificar o caminho local do arquivo como origem e o caminho remoto como destino. Por exemplo, para copiar o arquivo relatorio.pdf da pasta atual para a pasta **/home/user/**documentos no sistema remoto com o **IP 192.168.0.10** e o utilizador user, podemos usar o seguinte comando:

scp relatorio.pdf user@192.168.0.10:/home/user/documentos

O comando vai pedir a senha do utilizador remoto e depois iniciar a transferência do arquivo. Se tudo correr bem, ele vai mostrar uma mensagem como esta:

relatorio.pdf 100% 1MB 1.0MB/s 00:01

Copiar um arquivo remoto para um sistema local
Para copiar um arquivo remoto para um sistema local, basta inverter a ordem dos argumentos de origem e destino. Por exemplo, para copiar o arquivo **/home/user/fotos/foto.jpg** do sistema remoto com o IP 192.168.0.10 e o utilizador user para a pasta atual do sistema local, podemos usar o seguinte comando:

scp user@192.168.0.10:/home/user/fotos/foto.jpg .

O ponto final indica a pasta atual do sistema local. O comando vai pedir a senha do utilizador remoto e depois iniciar a transferência do arquivo. Se tudo correr bem, ele vai mostrar uma mensagem como esta:

foto.jpg 100% 500KB 500KB/s 00:01

Para copiar vários ficheiros, podemos usar curingas (*) ou listar os nomes dos ficheiros separados por espaços. Por exemplo, para copiar todos os ficheiros com extensão .txt da pasta atual do sistema local para a pasta /home/user/textos do sistema remoto com o IP 192.168.0.10 e o utilizador user, podemos usar o seguinte comando:

scp *.txt user@192.168.0.10:/home/user/textos

O comando vai pedir a senha do utilizador remoto e depois iniciar a transferência dos ficheiros. Se tudo correr bem, ele vai mostrar uma mensagem como esta:

a.txt 100% 10KB 10KB/s 00:01
b.txt 100% 20KB 20KB/s 00:01
c.txt 100% 30KB 30KB/s 00:01

Para copiar pastas e os seus conteúdos, precisamos usar a opção -r para copiar recursivamente os ficheiros e sub-diretorias. Por exemplo, para copiar o diretório /home/user/projetos do sistema local para o diretório /home/user/backup do sistema remoto com o IP 192.168.0.10 e o utilizador user, podemos usar o seguinte comando:

scp -r /home/user/projetos user@192.168.0.10:/home/user/backup

O comando vai pedir a senha do utilizador remoto e depois iniciar a transferência dos diretórios e ficheiros. Se tudo correr bem, ele vai mostrar uma mensagem como esta:

projetos/a/a.txt 100% 10KB 10KB/s 00:01
projetos/a/b.txt 100% 20KB 20KB/s 00:01
projetos/b/c.txt 100% 30KB 30KB/s 00:01
projetos/b/d.txt 100% 40KB 40KB/s 00:01

Para copiar ficheiros usando uma chave SSH para autenticar a conexão, precisamos usar a opção -i para especificar o arquivo da chave privada que será usada pelo comando SCP. Por exemplo, se temos uma chave privada chamada id_rsa na pasta .ssh do nosso diretório pessoal, podemos usá-la para copiar o arquivo relatorio.pdf da pasta atual do sistema local para a pasta /home/user/documentos do sistema remoto com o IP 192.168.0.10 e o utilizador user, usando o seguinte comando:

scp -i ~/.ssh/id_rsa relatorio.pdf user@192.168.0.10:/home/user/documentos

O comando não vai pedir a senha do utilizador remoto, mas pode pedir a frase secreta da chave privada se ela tiver uma. Depois iniciar a transferência do arquivo normalmente.

4.4 Criação de uma nova assinatura digital

Uma forma de criar uma assinatura digital é usando uma chave SSH, que é um par de chaves criptográficas que permitem se autenticar num sistema remoto usando o protocolo SSH.

Para gerar uma nova chave SSH no sistema local, execute o seguinte comando

ssh-keygen -t ed25519 -C "seu_email@exemplo.com"

Onde:

-t ed25519 especifica o tipo de algoritmo usado para gerar a chave, que neste caso é o Ed25519, considerado mais seguro e eficiente que outros algoritmos.

-C "seu_email@exemplo.com" adiciona um comentário à chave, que pode ser usado para identificar o proprietário da chave.

O comando irá solicitar um arquivo para salvar a chave e uma frase secreta (opcional) para proteger a chave.

A chave gerada consiste em dois ficheiros: um arquivo privado (por padrão, ~/.ssh/id_ed25519) e um arquivo público (por padrão, ~/.ssh/id_ed25519.pub). O arquivo privado deve ser mantido em segredo e não deve ser compartilhado com ninguém, enquanto o arquivo público pode ser distribuído para os sistemas remotos que se deseja aceder ou enviar mensagens assinadas.

Adicionando a chave SSH ao agente SSH

O agente SSH é um programa que gere as chaves SSH e lembra as frases secretas das chaves, evitando que se tenha que digitá-las toda vez que se usa a chave.

Para adicionar a chave SSH ao agente SSH no sistema local, execute os seguintes comandos:

eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519

Onde:

eval "$(ssh-agent -s)" inicia o agente SSH e configura as variáveis de ambiente necessárias para se comunicar com ele.

ssh-add ~/.ssh/id_ed25519 adiciona a chave privada ao agente SSH.

Para assinar uma mensagem com a chave SSH no sistema local, execute o seguinte comando https://blog.dlncloud.com.br/linux/copiar-ficheiros-com-seguranca-pelo-ssh/:

echo "Mensagem" | ssh-keygen -Y sign -n contexto -f ~/.ssh/id_ed25519

Onde:

"Mensagem" é a mensagem que se deseja assinar.

-Y sign especifica que se quer assinar a mensagem.

-n contexto especifica um contexto para a assinatura, que pode ser qualquer string arbitrária que identifique o propósito da assinatura.

-f ~/.ssh/id_ed25519 especifica o arquivo da chave privada usada para assinar.

O comando irá gerar uma saída como esta:

SIGNATURE CONTEXT:context
SIGNATURE:AAAAHWNvbnRleHRlZDI1NTE5LXNoYTItbmlzdHAyNTYAAABhB
AAAAAEAAACLAQAAAYwAAAAEAAAAIAAAAACBAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABkAAAACAAAABgAAAAEA
AAAMbWVzc2FnZQAAAAEAAAAGc3NoOi8vAAAAAQAAABIAAAABAAAAF
HNzaC1lZDI1NTE5AAAAIAAAAACBAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA==

Onde:

A primeira linha contém o contexto da assinatura.

A segunda linha contém a assinatura em si, codificada em base64.

Verificando uma mensagem assinada com a chave SSH
Para verificar uma mensagem assinada com a chave SSH no sistema local ou remoto, execute o seguinte comando https://blog.dlncloud.com.br/linux/copiar-ficheiros-com-seguranca-pelo-ssh/:

echo "Mensagem" | ssh-keygen -Y verify -n contexto -f ~/.ssh/id_ed25519.pub -s SIGNATURE

Onde:

"Mensagem" é a mensagem original que foi assinada.

-Y verify especifica que se quer verificar a mensagem.

-n contexto especifica o mesmo contexto usado na assinatura.

-f ~/.ssh/id_ed25519.pub especifica o arquivo da chave pública usada para verificar.

-s SIGNATURE especifica a assinatura em base64 obtida na etapa anterior.

O comando irá gerar uma saída como esta:

Good signature for context "context" with public key at "/home/uttilizador/.ssh/id_ed25519.pub"

A primeira linha indica que a assinatura é válida para o contexto e a chave pública fornecidos.

Se a mensagem, o contexto ou a chave pública forem alterados, o comando irá gerar uma saída como esta:

Bad signature for context "context" with public key at "/home/uttilizador/.ssh/id_ed25519.pub"
Signature verification failed for context "context" with public key at "/home/uttilizador/.ssh/id_ed25519.pub"

Onde:

A primeira linha indica que a assinatura é inválida para o contexto ou a chave pública fornecida.

A segunda linha indica que a verificação da assinatura falhou para o contexto ou a chave pública fornecida.

4.4.1 Exercícios práticos

Passo1: Criação da nova assinatura digital (chave SSH)

Abra o terminal no seu sistema local.
Digite o comando **ssh-keygen** e pressione Enter.
Será solicitado que você escolha o local e o nome do arquivo onde a chave será armazenada. Pressione Enter para aceitar o local e o nome padrão ou forneça um local e nome personalizados.
Em seguida, você será solicitado a fornecer uma senha para proteger a chave. pode optar por inserir uma senha ou deixá-la em branco para uma autenticação sem senha. Pressione Enter para continuar.
Uma nova chave SSH será gerada nos formatos público e privado e será armazenada no local especificado.

Passo 2: Configuração da chave pública no servidor remoto

Aceda o servidor remoto usando o comando **ssh** seguido pelo nome de utilizador e endereço IP do servidor. Por exemplo:

ssh user@servidor

Digite a senha do utilizador para se autenticar no servidor.
No servidor remoto, abra o arquivo **~/.ssh/authorized_keys** num editor de texto.
Copie o conteúdo da chave pública gerada no Passo 1 (arquivo com a extensão **.pub** localizado no mesmo diretório da chave privada) e cole-o no arquivo **authorized_keys** do servidor remoto.
Salve o arquivo e feche o editor de texto.

Passo 3: Logon usando a assinatura

No terminal do sistema local, digite o comando **ssh user@servidor** (substitua "user" pelo seu nome de utilizador e "servidor" pelo endereço IP ou nome de domínio do servidor remoto) e pressione Enter.

Se você configurou uma senha para a chave, será solicitado que você insira a senha.

Após inserir a senha correta, você será autenticado com sucesso no servidor remoto e terá acesso ao prompt do servidor.

Criou uma assinatura digital (chave SSH) e logon usando essa assinatura. Neste momento poderá usar a chave para se autenticar de forma segura em servidores remotos, sem a necessidade de ter de inserir sempre a sua senha. A assinatura digital é uma maneira eficiente e segura para aceder aos sistemas remotos através do SSH.

O SSH também oferece recursos avançados, como a execução de programas remotos, a utilização do SSH para clientes X localmente e a criação de túneis SSH para encaminhamento de portas. Neste manual, exploraremos esses recursos do SSH e aprenderemos como utilizá-los de maneira eficaz.

4.5 *Utilização do SSH para Execução de Programas Remotos*

O SSH permite que você execute programas num servidor remoto através da linha de comando local.

Exemplo de execução de um programa remoto:

ssh user@servidor comando

O programa será executado no servidor remoto e qualquer saída será exibida localmente.

4.6 *Utilização do SSH para Clientes X Localmente*

O SSH suporta o redirecionamento de exibição X11, permitindo que você execute aplicativos gráficos de um servidor remoto e os exiba em sua máquina local.

Para habilitar o redirecionamento X11, você precisa adicionar a opção "-X" ao usar o comando "ssh":

ssh -X user@servidor

Depois de estabelecer a conexão SSH, poderá executar aplicativos gráficos remotos que serão exibidos em sua máquina local.

4.7 *Túneis SSH*

Os túneis SSH permitem que você encaminhe o tráfego de rede entre duas máquinas através de uma conexão SSH segura.

4.7.1 Túnel de encaminhamento de porta local:

ssh -L porta_local:destino:porta_destino user@servidor

4.7.2 Túnel de encaminhamento de porta remota:

ssh -R porta_remota:destino:porta_destino user@servidor

Os túneis SSH são úteis para aceder serviços numa rede remota de forma segura ou para contornar restrições de firewall.

4.7.3 A habilitação e desabilitação do acesso remoto do utilizador de "root"

O acesso remoto ao utilizador "root" é uma funcionalidade importante em sistemas Unix/Linux, mas também pode representar um risco de segurança se não for configurado corretamente. Neste manual, abordaremos os procedimentos para habilitar e desabilitar o acesso remoto ao utilizador "root", fornecendo diretrizes para garantir a segurança do seu sistema.

4.7.4 Habilitação do Acesso Remoto do utilizador "root"

aceder o servidor remotamente usando um utilizador com privilégios de administrador.

Editar o arquivo de configuração SSH localizado em /etc/ssh/sshd_config.

Localizar a linha que contém a diretiva PermitRootLogin e modificar seu valor para "yes".

Salvar as alterações e reiniciar o serviço SSH para que as alterações entrem em vigor.

Testar o acesso remoto usando o utilizador "root" para verificar se a habilitação foi bem-sucedida.

4.7.5 Desabilitação do Acesso Remoto do utilizador "root"

aceder o servidor remotamente usando um utilizador com privilégios de administrador.

Editar o arquivo de configuração SSH localizado em /etc/ssh/sshd_config.

Localizar a linha que contém a diretiva PermitRootLogin e modificar seu valor para "no".

Salvar as alterações e reiniciar o serviço SSH para que as alterações entrem em vigor.

Testar o acesso remoto usando o utilizador "root" para verificar se a desabilitação foi bem-sucedida.

Cofinanciado por:

4.7.6 Alternativas para Acesso Remoto

É recomendado criar um utilizador com privilégios de administrador e utilizar esse utilizador para aceder o sistema remotamente.

Utilizar chaves de autenticação pública/privada para acesso remoto seguro, em vez de autenticação baseada em senha.

Configurar medidas adicionais de segurança, como limitar o acesso remoto a partir de endereços IP específicos ou utilizar VPN para conexões remotas.

4.7.7 Considerações de Segurança

Habilitar o acesso remoto do utilizador "root" pode aumentar o risco de ataques de força bruta e comprometimento do sistema.

Desabilitar o acesso remoto do utilizador "root" é uma prática recomendada para mitigar riscos de segurança.

Sempre siga as melhores práticas de segurança ao habilitar ou desabilitar o acesso remoto ao utilizador "root" em seu ambiente.

5 FTP

O **FTP (File Transfer Protocol)** é um protocolo muito utilizado para transferência de ficheiros entre cliente e servidor remoto. Foi desenvolvido na década de 1970 sendo desde então uma das formas mais comuns de compartilha de ficheiros na Internet.

O FTP permite que os utilizadores enviem, baixem, editem e excluam ficheiros num servidor remoto por meio de uma conexão TCP/IP. Fornece uma estrutura simples e padronizada para transferir dados de forma eficiente e confiável.

A conexão pode ser ativa ou passiva, dependendo de quem inicia o canal de dados. O modo é ativo quando o servidor FTP cria a conexão de dados com o cliente FTP, e o modo passivo é quando o cliente FTP cria a conexão de dados com o servidor FTP.

Para usar o FTP, é necessário o programa cliente de FTP, por exemplo FileZilla que é um software livre e gratuito que suporta FTP, FTPS e SFTP. O FTPS é uma versão do FTP que usa SSL/TLS para proteger a transmissão dos dados, e o SFTP é uma versão do FTP que usa SSH para proteger a transmissão dos dados.

5.1 Wu-FTPd

Wu-FTPd é um software de servidor FTP gratuito para sistemas operacionais Unix/Linux. Foi originalmente escrito por Chris Myers e Bryan D. O'Connor na Washington University em St. Louis como um substituto do daemon FTP do BSD, para uso na rede da universidade, principalmente no grande site wuarchive .O software evoluiu para se tornar um substituto em outros sistemas operacionais comerciais da época, como o Ultrix da DEC, o AIX da IBM e o SunOS e Solaris da Sun. Foi também portado para outros

sistemas operacionais baseados em código aberto, como FreeBSD e Linux https://en.wikipedia.org/wiki/WU-FTPD.

Wu-FTPd tnum grande número de opções configuráveis que o tornam superior ao daemon FTP "clássico" ou BSD que ainda é usado por alguns sabores de Unix, mas não é tão flexível ou limpo quanto o ProFTPD, coberto no capítulo 40. Esse capítulo também tem uma breve introdução ao protocolo FTP, que deve ler antes de continuar se não estiver familiarizado com conceitos como clientes e servidores FTP.

Em sua configuração padrão normal, Wu-FTPd permite que qualquer usuário Unix (exceto os utilizadores do sistema) faça login com seus nomes de usuário e senhas padrão e faça upload, download e manipule arquivos no sistema do servidor com as mesmas permissões que teriam se estivessem conectados via telnet ou SSH. Foi também pode ser configurado para suportar logins anónimos, para que qualquer pessoa possa se conectar sem precisar de uma conta Unix válida - embora os clientes anónimos normalmente sejam restritos a um determinado diretório e impedidos de fazer upload de arquivos.

O arquivo de configuração principal do Wu-FTPd é chamado **/etc/ftpaccess**, mas também usa vários outros arquivos como /etc/ftpusers e /etc/ftphosts. O arquivo ftpaccess contém uma série de diretivas, uma por linha, cada uma com um nome e vários valores. Cada diretiva define uma única opção, como o caminho para um arquivo de mensagem ou um alias de diretório.

Como o ProFTPD, Wu-FTPd pode ser executado como um processo daemon permanente independente ou a partir de um super-servidor como inetd ou xinetd. Normalmente a segunda opção é usada, pois isso elimina a necessidade de um processo de servidor adicional estar executando o tempo todo esperando pora conexão FTP. Quanto aos clientes e ao arquivo de configuração, não há diferença entre os dois modos além do desempenho.

5.2 Construção de um servidor FTP

Para construir um **servidor FTP no Windows** é necessário configurar o servidor FTP através do Administrador de **Serviços de Informações da Internet (IIS)**, configurar a firewall para permitir as conexões FTP, criar um utilizador e definir uma pasta de partilha.

Para construir um **servidor FTP no Linux** , proceda do seguinte modo:
Abra o terminal (CTRL + ALT + T) e execute o comando para a instalação do servidor FTP:

sudo apt-get install vsftpd

Configure o servidor FTP:

Abra o arquivo de configuração do servidor FTP com um editor de texto. Por exemplo:

sudo nano /etc/vsftpd.conf

Faça as alterações necessárias nas opções de configuração:

Definir a pasta raiz para os utilizadores do FTP.
Permitir ou restringir o acesso anônimo.
Configurar o modo de transferência (ativo ou passivo).
Definir permissões de escrita e leitura.
Configurar o uso de TLS/SSL para conexões seguras (opcional).

Reiniciar o servidor:

sudo service vsftpd restart

Verifique se as permissões corretas estão configuradas na pasta raiz definida no passo anterior. Certifique-se de que os utilizadores do FTP tenham permissão de gravação/leitura, conforme necessário. Certifique-se de abrir as portas necessárias para permitir conexões ao servidor FTP. Por padrão, a porta 21.

Os clientes poderão conectar-se a Usando um programa de FTP, como o FileZilla, fornecendo o endereço IP do servidor FTP, nome de usuário e senha.

5.3 FTP público vs. FTP de utilizadores

Um **servidor FTP público** é um servidor FTP que não requer credenciais para aceder seus arquivos. O FTP público normalmente usa o nome de utilizador “anonymous" e não requer uma senha. Um FTP público pode ser usado para disponibilizar arquivos para download ou upload de forma livre e gratuita. O ***directório pub*** é um diretório público que contém arquivos que podem ser acedidos por qualquer utilizador num servidor FTP. É muito comum que servidores FTP tenham um diretório pub para distribuição de arquivos sem restrições pela Internet, como por exemplo de servidor FTP público que contém os arquivos do sistema operacional FreeBSD.

Um **servidor FTP de utilizadores** é um servidor FTP que requer que os utilizadores se autentiquem com uma conta e uma senha válida durante a conexão. Um FTP de utilizadores normalmente usa o nome de usuário e a senha do sistema operacional onde o servidor FTP está instalado ou usa contas virtuais criadas pelo administrador do servidor FTP. Um FTP de utilizadores pode ser usado para restringir o acesso aos arquivos e controlar as permissões dos utilizadores. Terá de ter o endereço do servidor, um nome de utilizador e a senha. Poderá utilizar um cliente FTP e inserir essas informações na interface do programa, ou usar a linha de comando, por exemplo:

ftp user@ftp.exemplo.com

Ou através de um navegador:

ftp://user:senha@ftp.exemplo.com

Depois de se conectar poderá navegar pelas diretorias e transferir arquivos usando os comandos do cliente FTP ou do navegador web.

5.4 Comandos do cliente FTP

Um **cliente FTP** é um programa que permite se conectar a um servidor FTP e transferir arquivos. Como tínhamos já visto existem atualmente vários clientes FTP disponíveis, FileZilla, WinSCP etc. Também pode usar a linha de comando no Linux.

Comandos mais comuns:

ftp - Inicia o cliente FTP e se conecta a um servidor FTP. pode especificar o nome ou o endereço IP do servidor como um parâmetro. Por exemplo:

ftp ftp.exemplo.com

open - Abre uma conexão com um servidor FTP. pode usar este comando se você já iniciou o cliente FTP sem especificar um servidor. Por exemplo:

open ftp.exemplo.com

user - Envia o nome de usuário e a senha para o servidor FTP. pode usar este comando se o servidor requer autenticação. Por exemplo:

user uttilizador senha

close - Fecha a conexão com o servidor FTP atual. pode usar este comando se você quiser se conectar a outro servidor sem sair do cliente FTP.

quit - Sai do cliente FTP e fecha a conexão com o servidor FTP atual.

cd - Muda o diretório atual no servidor FTP. pode especificar o nome do diretório como um parâmetro. Por exemplo:

cd public_html

lcd - Muda o diretório atual no seu computador local. pode especificar o nome do diretório como um parâmetro. Por exemplo:

lcd C:\Users\uttilizador\Documents

pwd - Mostra o nome do diretório atual no servidor FTP.

lpwd - Mostra o nome do diretório atual no seu computador local.

dir - Lista os arquivos e diretórios no diretório atual do servidor FTP.

ls - Lista os nomes dos arquivos e diretórios no diretório atual do servidor FTP.

get - Copia um arquivo do servidor FTP para o seu computador local. pode especificar o nome do arquivo como um parâmetro. Por exemplo:

get index.html

put - Copia um arquivo do seu computador local para o servidor FTP. pode especificar o nome do arquivo como um parâmetro. Por exemplo:

put index.html

mget - Copia vários arquivos do servidor FTP para o seu computador local. pode usar caracteres curinga (*) para especificar os arquivos que deseja copiar. Por exemplo:

mget *.jpg

mput - Copia vários arquivos do seu computador local para o servidor FTP. pode usar caracteres curinga (*) para especificar os arquivos que deseja copiar. Por exemplo:

mput *.jpg

delete - Exclui um arquivo no servidor FTP. pode especificar o nome do arquivo como um parâmetro. Por exemplo:

delete index.html

mdelete - Exclui vários arquivos no servidor FTP. pode usar caracteres curinga (*) para especificar os arquivos que deseja excluir. Por exemplo:

mdelete *.jpg

rename - Renomeia um arquivo ou diretório no servidor FTP. pode especificar o nome antigo e o novo como parâmetros. Por exemplo:

rename index.html home.html

mkdir - Cria um diretório no servidor FTP. pode especificar o nome do diretório como um parâmetro. Por exemplo:

mkdir images

rmdir - Remove um diretório vazio no servidor FTP. pode especificar o nome do diretório como um parâmetro. Por exemplo:

rmdir images

ascii - Define o modo de transferência de arquivo como ASCII. Este modo é usado para transferir arquivos de texto.

binary - Define o modo de transferência de arquivo como binário. Este modo é usado para transferir arquivos que não são de texto, como imagens, vídeos, etc.

hash - Ativa ou desativa a exibição de símbolos (#) para cada bloco de dados transferido.

prompt - Ativa ou desativa o modo prompt. No modo prompt, o cliente FTP solicita confirmação antes de cada transferência de arquivo.

help - Mostra uma lista de comandos disponíveis ou informações sobre um comando específico.

5.4.1 Exercícios práticos

5.4.2 Configuração de um *diretório pub* num servidor ftp linux

Para configurar um diretório pub num servidor ftp em linux, você precisa ter um software de servidor FTP instalado, vamos usar o **vsftpd** (Very Secure FTP Daemon), que é um dos servidores FTP mais populares e seguros para Linux.

Instalação do vsftpd:

Para instalar o vsftpd no linux, Debian/Ubuntu execute o comando:

sudo apt-get install vsftpd

Depois de instalar o vsftpd, você precisa iniciar e habilitar o serviço para que ele funcione automaticamente na inicialização do sistema. pode fazer isso com os seguintes comandos:

sudo systemctl start vsftpd

sudo systemctl enable vsftpd

Verifique o estado do serviço com o seguinte comando:

sudo systemctl status vsftpd

Configuração do vsftpd:

A configuração do vsftpd é feita através do arquivo **/etc/vsftpd.conf**, que contém várias opções e comentários explicativos. Antes de editar esse arquivo, é recomendável fazer uma cópia de backup do original com o seguinte comando:

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig

Abra o arquivo de configuração com vi:

sudo vi /etc/vsftpd.conf

Para configurar um diretório pub no seu servidor FTP, você precisa alterar ou adicionar as seguintes opções no arquivo de configuração:

anonymous_enable=YES - Esta opção permite o acesso anônimo ao servidor FTP, que é necessário para o diretório pub.

local_enable=NO - Esta opção desabilita o acesso local ao servidor FTP, ou seja, os utilizadores do sistema não se podem conetar ao servidor FTP com suas credenciais. pode habilitar esta opção se quiser permitir também o acesso local.

write_enable=NO - Esta opção desabilita a escrita no servidor FTP, ou seja, os utilizadores não podem fazer upload ou modificar arquivos no servidor FTP. pode habilitar esta opção se quiser permitir também a escrita.

anon_root=/srv/ftp - Esta opção define o diretório raiz para os utilizadores anónimos. pode alterar este valor para o diretório que quiser usar como diretório pub. Por padrão, o vsftpd usa /srv/ftp como diretório pub de acordo com o Filesystem Hierarchy Standard.

no_anon_password=YES - Esta opção desabilita a solicitação de senha para os utilizadores anónimos. pode habilitar esta opção se quiser solicitar uma senha para os utilizadores anónimos.

anon_max_rate=0 - Esta opção define a taxa máxima de transferência para os utilizadores anónimos em bytes por segundo. pode alterar este valor para limitar a largura de banda usada pelos utilizadores anónimos. O valor 0 significa ilimitado.

Salve as alterações e reinicie o serviço vsftpd :

sudo systemctl restart vsftpd

Teste do diretório pub usando o comando ftp

ftp [endereço_ip_do_servidor]

Deverá ver apresentada uma mensagem de boas-vindas do seu servidor FTP e a solicitação de um nome de utilizador. Digite ***anonymous*** ou outro nome ao seu gosto. Não deverá ser solicitado a digitar uma senha se tiver desabilitado a opção no_anon_password.
Depois de se conectar ao servidor FTP, deverá estar no diretório raiz dos utilizadores anónimos, que é o seu *diretório pub.* Nesta fase poderá listar os arquivos e diretórios com o comando ls ou dir. Você também pode navegar pelos diretórios com o comando cd ou baixar arquivos com o comando get. Para sair do servidor FTP, basta digitar **quit** ou **bye**.

5.5 *Configuração de mensagens para os utilizadores no servidor vsftpd*

O vsftpd permite configuração personalizada de mensagens personalizadas a utilizadores que se conectam ao servidor. Estas são usadas para fornecer informações úteis, avisos ou saudações.

Existem dois tipos de mensagens: **mensagens banner** e de **diretório**.

5.5.1 Mensagens de banner

As mensagens de banner são exibidas aos utilizadores antes ou depois do login. Elas podem ser configuradas através das seguintes opções no arquivo **/etc/vsftpd.conf**:

ftpd_banner - Esta opção define a mensagem que é exibida aos utilizadores antes do login. pode usar esta opção para informar os utilizadores sobre as regras ou políticas do servidor FTP. Por exemplo:

ftpd_banner=Bem-vindo ao servidor FTP. Por favor, respeite os direitos autorais e não faça upload de arquivos ilegais.

banner_file - Esta opção define o nome do arquivo que contém a mensagem que é exibida aos utilizadores depois do login. pode usar esta opção para fornecer informações adicionais ou saudações aos utilizadores. Por exemplo:

banner_file=/etc/vsftpd.banner

Neste caso, você precisa criar o arquivo **/etc/vsftpd.banner** com o conteúdo da mensagem que deseja exibir. Por exemplo:

Olá, {USER}. Você está conectado ao servidor FTP. O seu endereço IP é {REMOTE_IP}. O seu diretório atual é {CWD}.

Pode-se usar as seguintes ***variáveis*** na mensagem:

{USER} - O nome do utilizador.

{LOCAL_IP} - O endereço IP local do servidor FTP.

{REMOTE_IP} - O endereço IP remoto do utilizador.

{CWD} - O diretório atual do utilizador.

5.5.2 Mensagens de diretório

As mensagens de diretório são exibidas aos utilizadores quando eles entram num diretório. São usadas para fornecer

informações específicas sobre o conteúdo ou o propósito do diretório.

Para configurar as mensagens de diretório, você precisa habilitar a seguinte opção no arquivo /etc/vsftpd.conf:

dirmessage_enable=YES - Esta opção habilita a exibição das mensagens de diretório.

Em seguida, você precisa criar um arquivo chamado. message em cada diretório que deseja exibir uma mensagem. O conteúdo do arquivo. message será a mensagem que será exibida aos utilizadores quando eles entrarem nesse diretório. Por exemplo:

Este é o diretório pub. Aqui pode encontrar arquivos gratuitos para download. Por favor, não faça upload de arquivos neste diretório.

Depois de configurar as mensagens de banner e de diretório, você precisa reiniciar o serviço vsftpd para que as alterações entrem em vigor. pode fazer isso com o seguinte comando:

sudo systemctl restart vsftpd

5.5.3 Pasta /var/ftp

A pasta **/var/ftp** é utilizada para armazenar os arquivos acedidos pelos utilizadores anónimos. Essa pasta é definida pela opção anon_root no arquivo /etc/vsftpd.conf. pode alterar essa opção para usar outro diretório como diretório pub.

Para que os utilizadores anónimos possam aceder a pasta **/var/ftp**, você precisa habilitar a opção anonymous_enable no arquivo **/etc/vsftpd.conf** e definir as permissões adequadas para a pasta. Por exemplo:

sudo chmod 755 /var/ftp

sudo chown nobody:nogroup /var/ftp

Você também pode criar sub-diretorias dentro da pasta /var/ftp para organizar os arquivos por categorias ou temas. Por exemplo:

sudo mkdir /var/ftp/music

sudo mkdir /var/ftp/documents

sudo mkdir /var/ftp/images

Você deve definir as permissões e a propriedade desses sub-diretorias de acordo com o nível de acesso que deseja conceder aos utilizadores anónimos. Por exemplo, se pretender que os utilizadores anónimos façam upload de arquivos na pasta /var/ftp/music, faça o seguinte:

sudo chmod 777 /var/ftp/music

sudo chown nobody:nogroup /var/ftp/music

Se você quiser restringir o acesso dos utilizadores anónimos a apenas alguns sub-diretorias, pode usar a opção hide_file no arquivo /etc/vsftpd.conf para ocultar os arquivos ou diretórios que começam com um determinado caractere. Por exemplo, se você quiser ocultar todos os arquivos ou diretórios que começam com um ponto (.), pode fazer o seguinte:

hide_file=. *

5.5.4 Arquivo /etc/ftpaccess

O arquivo **/etc/ftpaccess** é um arquivo de configuração usado pelo **vsftpd** para definir regras de acesso e controle para os utilizadores do servidor FTP. Esse arquivo permite especificar as seguintes opções:

class - Define uma classe de utilizadores com base em seus endereços IP ou nomes de domínio. pode usar essa opção para aplicar diferentes restrições ou limites para diferentes classes de utilizadores. Por exemplo:

class local 192.168.* 127.*

class external *.com *.net *.org

limit - Define um limite para o número de utilizadores simultâneos de uma determinada classe ou grupo. pode usar essa opção para evitar a sobrecarga do servidor FTP ou garantir uma distribuição justa dos recursos. Por exemplo:

limit local 10 Any /etc/msgs/msg.local

limit external 5 Any /etc/msgs/msg.external

message - Define uma mensagem que é exibida aos utilizadores quando eles entram num determinado diretório. pode usar essa opção para fornecer informações específicas sobre o conteúdo ou o propósito do diretório. Por exemplo:

message /var/ftp Welcome to the public FTP server.

message /var/ftp/music Esta diretoria contém ficheiros de musica para download.

readme - Define um arquivo que é exibido aos utilizadores quando eles entram num determinado diretório. pode usar essa opção para fornecer informações mais detalhadas ou atualizadas sobre o conteúdo ou o propósito do diretório. Por exemplo:

readme README* login

readme README* cwd=*

compress - Define os tipos de arquivos que podem ser compactados antes da transferência. pode usar essa opção para economizar largura de banda e tempo de transferência. Por exemplo:

compress yes *.txt *.html *.doc *.pdf

tar - Define os tipos de arquivos que podem ser agrupados num arquivo tar antes da transferência. pode usar essa opção para facilitar o download de vários arquivos num único arquivo. Por exemplo:

tar yes *.jpg *.png *.gif

chmod - Define as permissões que podem ser alteradas pelos utilizadores nos arquivos ou diretórios do servidor FTP. pode usar essa opção para permitir ou negar aos utilizadores a capacidade de modificar as permissões dos arquivos ou diretórios que eles possuem ou aos quais têm acesso. Por exemplo:

chmod no anonymous

chmod yes localuser staff

delete - Define se os utilizadores podem apagar arquivos ou diretórios do servidor FTP. pode usar essa opção para permitir ou negar aos utilizadores a capacidade de remover arquivos ou diretórios que eles possuem ou aos quais têm acesso. Por exemplo:

delete no anonymous

delete yes localuser staff

rename - Define se os utilizadores podem renomear arquivos ou diretórios do servidor FTP. pode usar essa opção para permitir ou negar aos utilizadores a capacidade de alterar os nomes dos arquivos ou diretórios que eles possuem ou aos quais têm acesso. Por exemplo:

rename no anonymous

rename yes localuser staff

overwrite - Define se os utilizadores podem sobrescrever arquivos existentes no servidor FTP. pode usar essa opção para permitir ou negar aos utilizadores a capacidade de substituir arquivos que já existem no servidor FTP com novos arquivos com o mesmo nome. Por exemplo:

overwrite no anonymous

overwrite yes localuser staff

umask - Define a máscara de permissão padrão para os novos arquivos ou diretórios criados pelos utilizadores no servidor FTP. pode usar essa opção para definir as permissões iniciais dos novos arquivos ou diretórios criados pelos utilizadores no servidor FTP. Por exemplo:

umask anonymous 077

umask localuser 022 staff 002

5.5.5 Definição de ícones de arquivo

A definição de ícones de arquivo é uma forma de personalizar a aparência dos arquivos e diretórios no servidor FTP quando são visualizados por cliente FTP gráfico, como um navegador web. pode definir ícones de arquivo usando a opção icon no arquivo /etc/vsftpd.conf.

A opção icon permite especificar um ícone para cada tipo de arquivo com base na sua extensão ou nome. O ícone deve ser um arquivo GIF, JPEG ou PNG armazenado no mesmo diretório do arquivo ao qual se refere. Por exemplo:

icon .txt text.gif

icon .html html.gif

icon .pdf pdf.gif

icon README readme.gif

icon * default.gif

Essas linhas significam que os arquivos com extensão .txt terão o ícone text.gif, os arquivos com extensão .html terão o ícone html.gif, os arquivos com extensão .pdf terão o ícone pdf.gif, os arquivos chamados README terão o ícone readme.gif e todos os outros arquivos terão o ícone default.gif.

Pode definir um ícone padrão para todos os diretórios usando a opção diricon no arquivo /etc/vsftpd.conf. O ícone deve ser um arquivo GIF, JPEG ou PNG armazenado no mesmo diretório do diretório ao qual se refere. Por exemplo:

diricon folder.gif

Esta linha significa que todos os diretórios terão o ícone folder.gif.

5.5.6 Limitação do número de utilizadores

A limitação do número de utilizadores é uma forma de controlar o acesso ao servidor FTP e evitar a sobrecarga do sistema ou da rede. pode limitar o número de utilizadores usando as seguintes opções no arquivo **/etc/vsftpd.conf**:

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

max_clients - Esta opção define o número máximo total de utilizadores simultâneos que se podem conetar ao servidor FTP. Se esse limite for atingido, novas conexões serão recusadas até que algum utilizador se desconecte. O valor padrão é 0, que significa ilimitado. Por exemplo:

max_clients=50

Esta linha significa que apenas 50 utilizadores se podem conetar ao servidor FTP ao mesmo tempo.

max_per_ip - Esta opção define o número máximo de utilizadores simultâneos que podem se conectar ao servidor FTP a partir do mesmo endereço IP. Se esse limite for atingido, novas conexões a partir desse endereço IP serão recusadas até que algum utilizador se desconecte. O valor padrão é 0, que significa ilimitado. Por exemplo:

max_per_ip=5

Esta linha significa que apenas 5 utilizadores podem se conectar ao servidor FTP a partir do mesmo endereço IP ao mesmo tempo.

max_login_fails - Esta opção define o número máximo de tentativas de login falhadas que um utilizador pode fazer antes de ser desconectado do servidor FTP. Se esse limite for atingido, o utilizador será desconectado e não poderá tentar se conectar novamente por período de tempo definido pela opção delay_failed_login. O valor padrão é 3. Por exemplo:

max_login_fails=3

Esta linha significa que um utilizador pode tentar se conectar ao servidor FTP no máximo 3 vezes com credenciais inválidas antes de ser desconectado.

delay_failed_login - Esta opção define o tempo em segundos que um utilizador deve esperar antes de tentar se conectar novamente ao servidor FTP após atingir o limite de tentativas de login falhadas definido pela opção max_login_fails. Se esse valor for 0, não haverá atraso. O valor padrão é 1. Por exemplo:

delay_failed_login=10

Esta linha significa que um utilizador deve esperar 10 segundos antes de tentar se conectar novamente ao servidor FTP após falhar 3 vezes no login.

local_max_rate - Esta opção define a taxa máxima de transferência em bytes por segundo para os utilizadores locais do servidor FTP. Se esse valor for 0, não haverá limite. O valor padrão é 0. Por exemplo:

local_max_rate=1000000

Esta linha significa que os utilizadores locais podem transferir dados no máximo a 1 MB/s.

anon_max_rate - Esta opção define a taxa máxima de transferência em bytes por segundo para os utilizadores anônimos do servidor FTP. Se esse valor for 0, não haverá limite. O valor padrão é 0. Por exemplo:

anon_max_rate=500000

Esta linha significa que os utilizadores anónimos podem transferir dados no máximo a 500 KB/s.

5.5.7 Realização FTP para conta de utilizador

A realização FTP para conta de utilizador é uma forma de permitir que os utilizadores do sistema se conectem ao servidor FTP usando suas credenciais locais e tenham acesso aos seus diretórios pessoais. pode habilitar essa funcionalidade usando as seguintes opções no arquivo /etc/vsftpd.conf:

local_enable=YES - Esta opção habilita o acesso local ao servidor FTP, ou seja, os utilizadores do sistema podem se conectar ao servidor FTP com suas credenciais locais. O valor padrão é NO. Por exemplo:

local_enable=YES

chroot_local_user=YES - Esta opção habilita o chroot para os utilizadores locais, ou seja, os utilizadores locais ficam restritos aos seus diretórios pessoais e não podem aceder outros diretórios fora do seu chroot. É uma **limitação do utilizador à sua pasta raiz** no servidor FTP.O valor padrão é NO. Por exemplo:

chroot_local_user=YES

write_enable=YES - Esta opção habilita a escrita no servidor FTP, ou seja, os utilizadores podem fazer upload ou modificar arquivos no servidor FTP. O valor padrão é NO. Por exemplo:

write_enable=YES

Você também pode usar outras opções para configurar o acesso e o controle dos utilizadores locais, como as opções **chmod**, **delete**, **rename**, **overwrite** e **umask** descritas anteriormente.

5.5.8 Exercícios práticos

Preparação de uma pasta para um cliente atualizar o site via FTP
Se você quiser permitir que um cliente atualize o site via FTP é necessária uma pasta no servidor FTP e fornecer ao cliente as credenciais de acesso a essa pasta. Assim,o cliente poderá fazer upload ou download de arquivos para essa pasta.

Crie um utilizador no sistema para o cliente:

sudo adduser cliente

Este comandovai criar um utilizador chamado cliente e pedir para você definir uma senha e algumas informações opcionais.

Crie uma pasta no servidor FTP para o cliente:

sudo mkdir /var/ftp/cliente

Altere a propriedade e as permissões da pasta do cliente:

sudo chown cliente:cliente /var/ftp/cliente

sudo chmod 755 /var/ftp/cliente

Altere o diretório pessoal do utilizador do cliente para a pasta do cliente:

sudo usermod -d /var/ftp/cliente cliente

Este comandovai alterar o diretório pessoal do utilizador cliente para /var/ftp/cliente.

Altere a opção **chroot_local_user** no arquivo **/etc/vsftpd.conf** para **YES.** Isso vai fazer com que os utilizadores locais fiquem restritos aos seus diretórios pessoais e não possam aceder outros diretórios fora do seu chroot. Por exemplo:

chroot_local_user=YES

Reinicie o serviço vsftpd para que as alterações sejam feitas:

sudo systemctl restart vsftpd

Comunique ao cliente as suas credenciais de acesso ao servidor FTP, o endereço do servidor, o nome de utilizador, a senha e a porta (normalmente 21). Você também poderá realizar um pequeno manual com os passos todos. Por exemplo:

Servidor FTP: ftp.example.com
Nome de utilizador: cliente
Senha: ******
Porta: 21

Cliente FTP: FileZilla
Configuração: https://www.hostinger.com.br/tutoriais/como-configurar-o-cliente-filezilla

Ok, eu posso tentar adicionar um capítulo sobre a desactivação do FTP. Por favor, note que este é um texto gerado automaticamente e pode conter erros ou imprecisões.

5.5.9 Desactivação do FTP

Acesso ao servidor e utlize o comando systemctl :

sudo systemctl stop vsftpd

Este comandovai para o serviço vsftpd e interrompe as conexões FTP ativas.

Desabilite o serviço vsftpd para que ele não seja iniciado automaticamente:

sudo systemctl disable vsftpd

Remova o pacote vsftpd do servidor:

sudo apt-get remove vsftpd

Capitulo 5: Servidor Web - computação remota, TALK e NFS

6 Computação Remota

A **computação remota** é a capacidade de acesso e poder controlar um computador à distância por meio de uma rede ou Internet. A computação remota pode ser útil para diversas finalidades, como administração de sistemas, suporte técnico, trabalho colaborativo ou acesso a recursos compartilhados.

Existem diferentes ferramentas que permitem a realização da computação remota, vistos anteriormente como por exemplo TELNET, o RLOGIN, o SSH, o FTP, o VNC e o NFS.

6.1 VNC

VNC (Virtual Network Computing) é uma ferramenta de computação remota que permite aceder e controlar a interface gráfica de um computador à distância por meio de uma rede ou da internet. O VNC pode ser útil para diversas finalidades, como administração de sistemas, suporte técnico, trabalho colaborativo ou acesso a recursos compartilhados.

Para usar o VNC, você precisa ter um servidor VNC instalado e executando no computador remoto e um cliente VNC instalado no computador local.

No Windows, use TightVNC Viewer, utilize o endereço IP ou o nome de domínio do computador remoto seguido pelo número da porta (normalmente 5900). Por exemplo:

192.168.0.10:5900

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

Insira os dados solicitados. Controle a interface gráfica do computador remoto como se estivesse a utilizá-lo localmente. Feche a janela do cliente VNC para encerrar a conexão.
O VNC é uma ferramenta versátil **multiplataforma** que permite aceder e controlar o computador com um servidor VNC instalado. Permite a **computação móvel**, acesso a um computador remoto por meio de um dispositivo móvel, smartphone ou tablet.

6.2 TALK

O programa Talk é uma ferramenta de comunicação que permite estabelecer uma conversa em tempo real com outro utilizador por meio de uma interface de texto dividida em duas partes. O programa Talk pode ser usado para trocar mensagens instantâneas com outro utilizador que esteja online e que tenha um terminal aberto.

Para usar o programa Talk terá de o instalar no computador e no computador de outro utilizador. Terá de saber o nome de utilizador e o endereço IP ou o nome de domínio do outro utilizador.

O programa Talk é uma ferramenta simples e rápida de comunicação que não requer nenhum servidor intermediário ou de ter registo prévio. Possuí algumas limitações, por exemplo da disponibilidade do outro utilizador, a falta de criptografia dos dados transmitidos e a incompatibilidade com alguns sistemas operacionais.

6.2.1 Configuração dos serviços necessários para Talk

Para usar o programa Talk é necessário antes **configurar os serviços necessários** para Talk no seu computador e no computador do outro utilizador. Os serviços são:

O daemon talkd, que é o responsável por receber e enviar os convites e as mensagens do programa Talk.

O daemon inetd ou xinetd, que é o responsável por iniciar o talkd quando uma solicitação de conexão é recebida na porta 517 ou 518.

O arquivo **/etc/hosts.equiv** ou .**rhosts**, que é o responsável por definir quais hosts ou utilizadores podem se conectar ao seu computador por meio do programa Talk.

Instale o pacote talk e o pacote talkd no seu computador e assegure-se que o o outro utilizador também o tenha instalado.

sudo apt-get install talk talkd

Edite o arquivo **/etc/inetd.conf** ou **/etc/xinetd.d/talk** para habilitar o serviço talkd no seu computador e no computador do outro utilizador. Utilize o editor de texto.

sudo nano /etc/inetd.conf

ou

sudo nano /etc/xinetd.d/talk

Descomente ou adicione as seguintes linhas no arquivo /etc/inetd.conf ou /etc/xinetd.d/talk:

talk dgram udp wait root /usr/sbin/in.talkd in.talkd

ntalk dgram udp wait root /usr/sbin/in.ntalkd in.ntalkd

Essas linhas indicam que o daemon inetd ou xinetd deve iniciar quando solicitação de conexão é recebida na porta 517 ou 518.

Edite o arquivo **/etc/hosts.equiv** ou **.rhosts** para definir quais hosts ou utilizadores tem autorização para conectar ao seu computador pelo Talk.

sudo nano /etc/hosts.equiv

ou

nano .rhosts

Adicione os nomes das hosts ou dos utilizadores com permissão de conexão ao arquivo /etc/hosts.equiv ou .rhosts:

192.168.0.10 user

Esta linha indica que o utilizador user do host 192.168.0.10 pode se conectar ao seu computador por meio do programa Talk.

Reinicie o serviço inetd ou xinetd:

sudo systemctl restart inetd

ou

sudo systemctl restart xinetd

6.3 NFS

O **Network File System (NFS)** é um sistema de arquivos distribuídos que permite a montagem de sistemas de arquivos remotos numa rede TCP/IP. O NFS foi desenvolvido pela Sun Microsystems nos anos 80 e é baseado na especificação RFC 1094

O NFS permite que os utilizadores acedam arquivos armazenados em servidores NFS como se estivessem localmente, usando os mesmos comandos e interfaces. O NFS também facilita a partilha de arquivos e diretórios entre computadores com diferentes sistemas operacionais, como Windows, Linux e UNIX.

6.3.1 Utilidades do NFS

O NFS pode ser útil para diversas finalidades, como:
Fornecer acesso de múltiplos protocolos à mesma partilha de arquivo em protocolos SMB e NFS em clientes multiplataforma.

Implantar um servidor de arquivos NFS num ambiente de sistema operacional predominantemente não Windows para fornecer aos computadores cliente não Windows acesso aos compartilhamentos de arquivos NFS.

Migrar aplicativos de um sistema operacional para outro armazenando os dados em compartilhamentos de arquivos acessíveis por meio de protocolos SMB e NFS.

Centralizar o armazenamento e o backup dos dados dos utilizadores num único servidor NFS.

Reduzir o espaço local e o consumo de energia dos clientes NFS.

6.3.2 Daemons do NFS

Para que o NFS funcione corretamente, é necessário que alguns daemons estejam executando nos servidores e nos clientes NFS. Esses daemons são:

rpcd: daemon que fornece serviços básicos de chamada remota de procedimento (RPC) para os outros daemons do NFS.

statd: daemon que monitora o status dos servidores e clientes NFS e notifica sobre falhas ou recuperações.
mountd: daemon que atende às solicitações de montagem dos clientes NFS e verifica as permissões de acesso aos compartilhamentos de arquivos NFS.

nfsd: daemon que atende às solicitações dos clientes NFS e realiza as operações de leitura e escrita nos arquivos NFS.

6.3.3 Configuração do arquivo exports

Para que um servidor NFS possa compartilhar seus sistemas de arquivos com os clientes NFS, é necessário configurar o arquivo /etc/exports. Esse arquivo contém as informações sobre quais sistemas de arquivos serão exportados, para quais clientes e com quais opções.

A sintaxe básica do arquivo /etc/exports é:

/diretorio cliente1(opcoes) cliente2(opcoes) ...

Onde:

/diretorio é o caminho absoluto do sistema de arquivos que será exportado pelo servidor NFS.

cliente1, cliente2, ... são os nomes ou endereços IP dos clientes que terão acesso ao sistema de arquivos exportado.

opcoes são as opções que definem as características da partilha, como permissões, sincronização, versão do protocolo, etc.

Por exemplo:

/home 192.168.0.10(rw,sync,no_root_squash) 192.168.0.11(ro,sync)

Esse exemplo indica que o servidor NFS vai exportar o sistema de arquivos /home para os clientes 192.168.0.10 e 192.168.0.11, sendo que o primeiro terá permissão de leitura e escrita (rw), o segundo terá permissão somente de leitura (ro), ambos terão sincronização dos dados (sync) e o primeiro não terá restrição para o usuário root (no_root_squash).

6.3.4 Iniciação dos serviços de NFS

Para que o servidor NFS possa exportar seus sistemas de arquivos para os clientes NFS, é necessário iniciar os serviços necessários para o funcionamento do NFS. Esses serviços são:

portmap ou rpcbind: serviço que permite que os clientes NFS descubram qual porta o servidor NFS está utilizando.

nfs-server: serviço que inicia os daemons nfsd e mountd no servidor NFS.

nfs-lock: serviço que inicia o daemon statd no servidor NFS.

Para **iniciar** esses serviços, é possível usar o comando systemctl. Por exemplo, no Linux, é possível digitar:

sudo systemctl start portmap
sudo systemctl start nfs-server
sudo systemctl start nfs-lock

Para **verificar se os serviços estão ativos**, é possível usar o comando systemctl status. Por exemplo:

sudo systemctl status portmap
sudo systemctl status nfs-server
sudo systemctl status nfs-lock

Para **habilitar os serviços para que sejam iniciados automaticamente** na próxima vez que o servidor for reiniciado, é possível usar o comando systemctl enable. Por exemplo:

sudo systemctl enable portmap
sudo systemctl enable nfs-server
sudo systemctl enable nfs-lock

6.3.5 Definição de permissões de pastas exportados

Para garantir a segurança e a acessibilidade dos sistemas de arquivos exportados pelo servidor NFS, é necessário definir as permissões adequadas para as pastas exportadas. As permissões podem ser definidas usando os comandos chown e chmod.

O comando chown permite alterar a propriedade das pastas exportadas pelo servidor NFS. Por exemplo:

sudo chown -R user:group /home

Este comandoatribui a propriedade da pasta /home e todos os seus sub-diretorias ao usuário user e ao grupo group.

O comando chmod permite alterar as permissões das pastas exportadas pelo servidor NFS. Por exemplo:

sudo chmod -R 755 /home

Este comando define as permissões da pasta /home e todos os seus sub-diretorias como leitura, escrita e execução para o proprietário, leitura e execução para o grupo e leitura para os outros.

6.3.6 Definição de permissões de pastas exportados

Para que os clientes NFS possam aceder os sistemas de arquivos exportados pelo servidor NFS, é necessário definir as permissões adequadas para as pastas exportadas. As permissões podem ser definidas usando os comandos chown e chmod no servidor NFS, e também usando as opções no arquivo /etc/exports.

6.3.7 Acesso a pastas como root e utilizador

Por padrão, o servidor NFS não permite que o usuário root dos clientes NFS tenha acesso total às pastas exportadas. Isso significa que o usuário root não pode ler, escrever ou executar arquivos nas pastas exportadas, a menos que esses arquivos sejam públicos. Essa restrição visa aumentar a segurança e evitar que utilizadores mal-intencionados possam alterar ou apagar arquivos nas pastas exportadas.

Para permitir que o usuário root dos clientes NFS tenha acesso total às pastas exportadas, é necessário usar a opção no_root_squash no arquivo /etc/exports do servidor NFS. Por exemplo:

/home 192.168.0.10(rw,sync,no_root_squash)

Esta linha indica que o cliente 192.168.0.10 terá permissão de leitura e escrita (rw), sincronização dos dados (sync) e acesso total como usuário root (no_root_squash) à pasta /home do servidor NFS.

Por outro lado, se quiser restringir ainda mais o acesso do usuário root dos clientes NFS às pastas exportadas, é possível usar a opção root_squash no arquivo /etc/exports do servidor NFS. Por exemplo:

/home 192.168.0.10(rw,sync,root_squash)

Esta linha indica que o cliente 192.168.0.10 terá permissão de leitura e escrita (rw), sincronização dos dados (sync) e acesso limitado como usuário root (root_squash) à pasta /home do servidor NFS. Nesse caso, o usuário root será tratado como um usuário anônimo com UID e GID definidos pelo servidor NFS.

Além do usuário root, é possível definir as permissões de acesso para os demais utilizadores dos clientes NFS às pastas exportadas usando os comandos chown e chmod no servidor NFS. Por exemplo:

sudo chown -R user:group /home
sudo chmod -R 755 /home

Esses comandos atribuem a propriedade da pasta /home e todos os seus sub-diretorias ao usuário user e ao grupo group, e definem as permissões da pasta /home e todos os seus sub-diretorias como leitura, escrita e execução para o proprietário, leitura e execução para o grupo e leitura para os outros.

6.3.8 Importação de pastas num servidor

Para que um servidor possa importar as pastas exportadas por outro servidor NFS, é necessário montar os sistemas de arquivos remotos usando o comando mount ou o arquivo /etc/fstab.

6.3.9 Montagem de volumes NFS

A montagem de volumes NFS consiste em associar um sistema de arquivos remoto a um ponto de montagem local, permitindo assim o acesso aos arquivos do sistema de arquivos remoto como se estivessem localmente.

Para montar um volume NFS, é necessário conhecer o endereço IP ou o nome de domínio do servidor NFS, o caminho absoluto do sistema de arquivos exportado pelo servidor NFS e o caminho absoluto do ponto de montagem local.

Existem duas formas principais de montar um volume NFS: usando o comando mount ou usando o arquivo /etc/fstab.

6.3.10 Utilização do mount para aceder a um recurso remoto

O comando mount permite montar um volume NFS temporariamente, ou seja, até que o volume seja desmontado manualmente ou até que o sistema seja reiniciado.

A sintaxe básica do comando mount para montar um volume NFS é:

sudo mount -t nfs servidor:/diretorio /ponto_de_montagem

Onde:

-t nfs indica que o tipo do sistema de arquivos é nfs.

servidor é o endereço IP ou o nome de domínio do servidor NFS.

/diretorio é o caminho absoluto do sistema de arquivos exportado pelo servidor NFS.

/ponto_de_montagem é o caminho absoluto do ponto de montagem local.

Por exemplo:

sudo mount -t nfs 192.168.0.10:/home /mnt/nfs/home

Este comando monta temporariamente o sistema de arquivos /home do servidor 192.168.0.10 no ponto de montagem /mnt/nfs/home do cliente.

Para verificar se o volume foi montado corretamente, é possível usar o comando mount sem argumentos ou o comando df -h.

Para desmontar um volume NFS temporariamente montado, é possível usar o comando umount seguido do ponto de montagem local. Por exemplo:

sudo umount /mnt/nfs/home

Este comando desmonta temporariamente o volume NFS montado em /mnt/nfs/home.

6.3.11 Configuração do /etc/fstab para acesso

O arquivo /etc/fstab permite montar um volume NFS permanentemente, ou seja, sempre que o sistema for iniciado.

A sintaxe básica do arquivo /etc/fstab para montar um volume NFS é:

servidor:/diretorio /ponto_de_montagem nfs opcoes 0 0

Onde:

servidor é o endereço IP ou o nome de domínio do servidor NFS.

/diretorio é o caminho absoluto do sistema de arquivos exportado pelo servidor NFS.

/ponto_de_montagem é o caminho absoluto do ponto de montagem local.

Cofinanciado por:

nfs indica que o tipo do sistema de arquivos é nfs.

opcoes são as opções que definem as características da montagem, como versão do protocolo, tempo limite, etc.

0 0 são os valores para a frequência de backup e a ordem de verificação dos sistemas de arquivos.

Por exemplo:

192.168.0.10:/home /mnt/nfs/home nfs defaults 0 0

Esta linha indica que o sistema de arquivos /home do servidor 192.168.0.10 será montado permanentemente no ponto de montagem /mnt/nfs/home do cliente com as opções padrão.

Para aplicar as alterações feitas no arquivo /etc/fstab sem reiniciar o sistema, é possível usar o comando mount -a.

6.3.12 Exportação do CD-ROM e instalar num outro servidor

Para exportar um CD-ROM pelo servidor NFS e instalar num outro servidor, é necessário seguir os seguintes passos:

Insira o CD-ROM no drive do servidor NFS.Crie uma pasta para servir como ponto de montagem para o CD-ROM no servidor NFS. Por exemplo:
sudo mkdir /mnt/cdrom

Monte o CD-ROM na pasta criada usando o comando mount com a opção -o ro para indicar que a montagem será somente leitura. Por exemplo:
sudo mount -o ro /dev/cdrom /mnt/cdrom

Edite o arquivo /etc/exports do servidor NFS para adicionar a pasta onde o CD-ROM foi montado como uma pasta exportada para os servidores que vão instalar a partir dele. Por exemplo:

/mnt/cdrom 192.168.0.*(ro,sync)

Esta linha indica que a pasta /mnt/cdrom será exportada somente leitura (ro) e com sincronização dos dados (sync) para todos os servidores com endereço IP iniciado por 192.168.0.

Reinicie os serviços necessários para aplicar as alterações feitas no arquivo /etc/exports usando os comandos systemctl restart portmap e systemctl restart nfs-server.No outro servidor que vai instalar a partir do CD-ROM exportado pelo servidor NFS, crie uma pasta para servir como ponto de montagem para o CD-ROM remoto. Por exemplo:

sudo mkdir /mnt/cdrom

Monte o CD-ROM remoto na pasta criada usando o comando mount com a opção -t nfs para indicar que se trata de um sistema de arquivos nfs remoto. Por exemplo:
sudo mount -t nfs 192.168.0.10:/mnt/cdrom /mnt/cdrom

6.3.13 Configuração de um único site em múltiplos servidores usando NFS

Uma das vantagens do NFS é que ele permite compartilhar o mesmo sistema de arquivos entre vários servidores, facilitando assim a distribuição de carga e a redundância. Um exemplo disso é a configuração de um único site em múltiplos servidores usando NFS.

Para configurar um único site em múltiplos servidores usando NFS, é necessário seguir os seguintes passos:

Escolher um servidor para ser o servidor NFS, que vai exportar o sistema de arquivos onde o site está armazenado. Esse servidor deve ter instalado e configurado o serviço de servidor web, como apache ou Nginx, e o serviço de servidor NFS.Escolher os outros servidores para serem os clientes NFS, que vão importar o sistema de arquivos do servidor NFS e servir o site aos visitantes. Esses servidores devem ter instalado e configurado o serviço de cliente web, como Apache ou Nginx, e o serviço de cliente NFS.No servidor NFS, editar o arquivo /etc/exports para adicionar o sistema de arquivos onde o site está armazenado como uma pasta exportada para os clientes NFS. Por exemplo:

/var/www/html 192.168.0.*(ro,sync)

Esta linha indica que a pasta /var/www/html será exportada somente leitura (ro) e com sincronização dos dados (sync) para todos os clientes com endereço IP iniciado por 192.168.0.

No servidor NFS, reiniciar os serviços necessários para aplicar as alterações feitas no arquivo /etc/exports usando os comandos systemctl restart portmap e systemctl restart nfs-server.Nos clientes NFS, criar uma pasta para servir como ponto de montagem para o sistema de arquivos remoto. Por exemplo:
sudo mkdir /mnt/nfs/html

Nos clientes NFS, editar o arquivo /etc/fstab para adicionar uma linha para montar permanentemente o sistema de arquivos remoto na pasta criada:

192.168.0.10:/var/www/html /mnt/nfs/html nfs defaults 0 0

Esta linha indica que o sistema de arquivos /var/www/html do servidor 192.168.0.10 será montado permanentemente na pasta /mnt/nfs/html do cliente com as opções padrão.

Nos clientes NFS, aplicar as alterações feitas no arquivo /etc/fstab sem reiniciar o sistema usando o comando mount -a.Nos clientes NFS, editar o arquivo de configuração do serviço de cliente web para apontar para a pasta onde o sistema de arquivos remoto foi montado como a raiz do site. Por exemplo, se estiver usando Apache, editar o arquivo **/etc/apache2/sites-available/000-default.conf** e alterar a linha:

DocumentRoot /var/www/html

Para:

DocumentRoot /mnt/nfs/html

Nos clientes NFS, reiniciar o serviço de cliente web para aplicar as alterações feitas no arquivo de configuração usando o comando systemctl restart apache2.Testar se o site está funcionando corretamente acessando-o a partir de um navegador web usando o endereço IP ou o nome de domínio de qualquer um dos clientes NFS.
Verificação de partilhas locais e remotas (showmount)
Para verificar quais são as pastas exportadas pelo servidor NFS e quais são os clientes que têm acesso a elas, é possível usar o comando showmount.

A sintaxe básica do comando showmount é:

showmount -opcoes servidor

Onde:

-opcoes são as opções que definem o tipo de informação a ser mostrada, como -e para mostrar as pastas exportadas ou -a para mostrar os clientes ativos.

servidor é o endereço IP ou o nome de domínio do servidor NFS.

Por exemplo:

showmount -e 192.168.0.10

Este comando mostra as pastas exportadas pelo servidor 192.168.0.10 e os clientes que têm acesso a elas.

6.3.14 Configuração e execução do servidor NFS

Para configurar e executar um servidor NFS, é necessário instalar e configurar os serviços necessários para o funcionamento do NFS no servidor.

Os serviços necessários são:

portmap ou rpcbind: serviço que permite que os clientes NFS descubram qual porta o servidor NFS está utilizando.

nfs-server: serviço que inicia os daemons nfsd e mountd no servidor NFS.

nfs-lock: serviço que inicia o daemon statd no servidor NFS.

Para instalar estes serviços, utiliza-se o gestor de pacotes da distribuição Linux usada no servidor. Por exemplo, no Ubuntu ou Debian, difgite:

sudo apt install portmap nfs-server nfs-lock

Para configurar esses serviços, é necessário editar o arquivo /etc/exports do servidor NFS para adicionar as pastas que serão exportadas pelos clientes NFS e as opções correspondentes.

A sintaxe básica do arquivo /etc/exports é:

/diretorio cliente1(opcoes) cliente2(opcoes) ...

Onde:

/diretorio é o caminho absoluto do sistema de arquivos que será exportado pelo servidor NFS.

cliente1, cliente2, ... são os nomes ou endereços IP dos clientes que terão acesso ao sistema de arquivos exportado.

opcoes são as opções que definem as características da partilha, como permissões, sincronização, versão do protocolo, etc.

Por exemplo:

/home 192.168.0.*(rw,sync,no_root_squash)

Esta linha indica que a pasta /home será exportada com permissão de leitura e escrita (rw), sincronização dos dados (sync) e acesso total como usuário root (no_root_squash) para todos os clientes com endereço IP iniciado por 192.168.0.

Para iniciar esses serviços, é possível usar o comando systemctl. Por exemplo:

sudo systemctl start portmap
sudo systemctl start nfs-server
sudo systemctl start nfs-lock

Para verificar se os serviços estão ativos, é possível usar o comando systemctl status. Por exemplo:

sudo systemctl status portmap
sudo systemctl status nfs-server
sudo systemctl status nfs-lock

Para habilitar os serviços para que sejam iniciados automaticamente na próxima vez que o servidor for reiniciado, é possível usar o comando systemctl enable. Por exemplo:

sudo systemctl enable portmap
sudo systemctl enable nfs-server
sudo systemctl enable nfs-lock

Cofinanciado por: POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

6.3.15 RPC - conceito

RPC significa Remote Procedure Call (Chamada Remota de Procedimento) e é um protocolo que permite que um programa execute uma função ou procedimento em outro computador na rede sem precisar saber detalhes sobre esse computador.

O RPC funciona da seguinte forma: quando um programa quer executar uma função ou procedimento em outro computador na rede, ele envia uma mensagem ao computador remoto contendo os parâmetros da função ou procedimento desejado e espera pora resposta com o resultado da execução.

O computador remoto recebe a mensagem do programa e usa um programa intermediário chamado portmapper ou rpcbind para descobrir qual é o programa responsável por executar a função ou procedimento solicitado e qual é a porta que está utilizando.

O computador remoto então envia uma mensagem ao programa responsável pela função ou procedimento solicitado contendo os parâmetros recebidos do programa original e espera pora resposta com o resultado da execução.

O programa responsável pela função ou procedimento solicitado recebe a mensagem do computador remoto e executa a função ou procedimento com os parâmetros recebidos e envia uma mensagem ao computador remoto contendo o resultado da execução.

O computador remoto recebe a mensagem do programa responsável pela função ou procedimento solicitado contendo o resultado da execução e envia uma mensagem ao programa original contendo esse resultado.

O programa original recebe a mensagem do computador remoto contendo o resultado da execução da função ou procedimento solicitado e continua sua execução normalmente.

Um exemplo de uso do RPC é o protocolo NFS, que usa RPC para permitir que os programas nos clientes NFS executem funções ou procedimentos nos sistemas de arquivos exportados pelos servidores NFS. O NFS usa outro protocolo chamado RPC, que é uma forma de enviar e receber esses pedidos pela rede. O RPC faz com que o cliente e o servidor comuniquem como se estivessem executando as mesmas funções ou procedimentos localmente, mas na verdade eles estão fazendo isso remotamente.

6.3.16 Utilização do RPC para verificar se um servidor remoto esta executar o NFS

Para verificar se um servidor remoto está executando o NFS, usa-se o **comando rpcinfo**, que faz uma chamada de procedimento remoto (RPC) para um servidor e relata o que ele encontra. O comando rpcinfo tem várias opções, mas uma das mais úteis é a **opção -p**, que lista todos os programas registados com o mapeador de portas no servidor especificado. Por exemplo, para verificar se o servidor 192.168.1.100 está a executar o NFS:

rpcinfo -p
192.168.1.100

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia

Se o servidor estiver executando o NFS, terá uma saída semelhante a esta:

program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100003 3 udp 2049 nfs
100003 4 udp 2049 nfs
...

As linhas que contêm o número do programa 100003 indicam que o servidor está executando o NFS nas versões 3 e 4, nos protocolos TCP e UDP, na porta 2049 Se não existirem estas linhas, significa que o servidor não está em execução ou que está bloqueado por uma firewall ou outra configuração de rede.

6.3.17 Atomount e o fstab

Atomount é um programa que permite montar automaticamente partições ou dispositivos quando são acessados. Utiliza o arquivo **/etc/auto.master** para definir os pontos de montagem e as opções de montagem para cada dispositivo. Por exemplo, para montar automaticamente a partição /dev/sda2 no diretório /mnt/excess, pode adicionar esta linha ao arquivo /etc/auto.master:

/mnt/excess /etc/auto.misc --timeout=60

E depois adicionar esta linha ao arquivo **/etc/auto.misc**:

excess -fstype=ntfs :/dev/sda2

Fará com que o Atomount monte a partição /dev/sda2 como NTFS no diretório /mnt/excess quando ele for acessado, e desmonte após 60 segundos de inatividade
O arquivo **/etc/auto.misc** é um exemplo de arquivo de mapa que contém as informações sobre os dispositivos e os tipos de sistema de arquivos que devem ser montados.

O fstab é um arquivo que contém informações sobre as partições e dispositivos que devem ser montados no sistema. Usa o formato de colunas para especificar o dispositivo, o ponto de montagem, o tipo de sistema de arquivos, as opções de montagem, o dump e a ordem de verificação. Por exemplo, para montar a partição /dev/sda2 como NTFS no diretório /mnt/excess na inicialização do sistema, pode adicionar esta linha ao arquivo /etc/fstab:

/dev/sda2 /mnt/excess ntfs defaults 0 2

Fará com que o fstab monte a partição /dev/sda2 como NTFS no diretório /mnt/excess usando as opções padrão, sem fazer backup e com a segunda prioridade de verificação

6.3.18 Configuração do NFS no boot para iniciar um servidor com pastas criadas

Para configurar o NFS no boot para iniciar um servidor com pastas criadas, são necessários os seguintes passos:

No servidor host, instale o pacote nfs-kernel-server, que permite compartilhar seus diretórios

No servidor cliente, instale o pacote nfs-common, que fornece funcionalidade NFS sem incluir nenhum componente de servidor

No servidor host, crie os diretórios que deseja compartilhar e configure as permissões adequadas

No servidor host, edite o arquivo /etc/exports e adicione as pastas que deseja compartilhar, especificando os clientes e as opções de acesso

No servidor host, execute o comando exportfs -a para atualizar a lista de compartilhamentos

No servidor cliente, crie os pontos de montagem para os compartilhamentos remotos e configure as permissões adequadas

No servidor cliente, edite o arquivo /etc/fstab e adicione as entradas para os compartilhamentos remotos, especificando os endereços IP do host e as opções de montagem

No servidor cliente, execute o comando mount -a para montar todos os compartilhamentos especificados no arquivo /etc/fstab.

Esses passos devem garantir que o NFS seja iniciado no boot e que os compartilhamentos remotos sejam montados automaticamente.

6.3.19 Utilização do NFS para configurar pastas de utilizadores únicos num servidor

Para usar o NFS para configurar pastas de utilizadores únicos num servidor, você precisa seguir alguns passos:

No servidor host, instale o pacote nfs-kernel-server, que permite compartilhar seus diretórios
No servidor cliente, instale o pacote nfs-common, que fornece funcionalidade NFS sem incluir nenhum componente de servidor
No servidor host, crie as pastas de utilizadores que deseja compartilhar e configure as permissões adequadas

No servidor host, edite o arquivo /etc/exports e adicione as pastas de utilizadores que deseja compartilhar, especificando os clientes e as opções de acesso. pode usar a opção all_squash para mapear todos os utilizadores remotos para um usuário local https://tech-lib.wiki/nfs/.

No servidor host, execute o comando exportfs -a para atualizar a lista de compartilhamentos

No servidor cliente, crie os pontos de montagem para os compartilhamentos remotos e configure as permissões adequadas

No servidor cliente, edite o arquivo /etc/fstab e adicione as entradas para os compartilhamentos remotos, especificando os endereços IP do host e as opções de montagem

No servidor cliente, execute o comando mount -a para montar todos os compartilhamentos especificados no arquivo /etc/fstab

Esses passos devem permitir que você configure pastas de utilizadores únicos num servidor usando o NFS.

Webgrafia

RLOGIN e Telnet , https://pt.differkinome.com/articles/protocols—formats/difference-between-rlogin-and-telnet-3.html, 2023

Logon em máquinas remotas com o SSH, https://www.digitalocean.com/community/tutorials/how-to-use-ssh-to-connect-to-a-remote-server-pt, 2023
scp, https://www.hostinger.com.br/tutoriais/usar-comando-scp-linux-para-transferir-ficheiros, 2023
ssh, scp, https://blog.dlncloud.com.br/linux/copiar-ficheiros-com-seguranca-pelo-ssh/
scp, https://www.certificacaolinux.com.br/comando-linux-scp/, 2023
scp, https://www.hospedagemsegura.com.br/transferir-copiar-ficheiros-via-scp/, 2023

Nova assinatura Digital, https://docs.github.com/pt/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent, 2023
Para adicionar a chave SSH ao agente SSH no sistema local: https://docs.github.com/pt/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent, 2023
Como funciona o FTP, https://www.hostinger.com.br/tutoriais/ftp-o-que-e-como-funciona, 2023
FTP, conexões ativas e passivas, https://www.hostinger.com.br/tutoriais/ftp-o-que-e-como-funciona. https://askubuntu.com/questions/113733/how-to-mount-a-ntfs-partition-in-etc-fstab. 2023
WuFTP, https://en.wikipedia.org/wiki/WU-FTPD, 2023
NFS, https://pt.wikipedia.org/wiki/Network_File_System., 2023

RPC info, https://www.ibm.com/docs/pt-br/db2/11.1?topic=environment-verifying-that-nfs-is-running. 2023
Atomount e o fstab, https://diolinux.com.br/sistemas-operacionais/discos-particoes-linux-fstab.html. 2023
Configuração do NFS no Boot para iniciar um servidor com pastas criadas, https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-20-04-pt. 2023
Utilização do NFS para configurar pastas de utilizadores únicos num servidor, https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-20-04-pt., 2023

Cofinanciado por:
POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu
Formação com Garantia