Plano de Formação

Linux - Serviços de Redes

Formador: [Seu Nome]

Data de Elaboração: [Data]

Gerado com LaTeX

[

Guia do Formador]Conteúdo

[

Guia do Formador]Conteúdos

Objetivos: O aluno será capaz de identificar e configurar endereços IP e sub-redes, e utilizar ferramentas de diagnóstico de rede. **Pré-requisitos:** Conhecimentos básicos de linha de comando Linux. **Material de Apoio:** Diagrama de endereçamento IPv4 e IPv6, slides explicativos sobre máscaras de rede.
Visão Geral: Esta secção é o alicerce do curso. Sem a compreensão básica de endereços IP e como os computadores se comunicam, os alunos terão dificuldade em entender os serviços de rede. O objetivo é desmistificar o networking com analogias simples e ferramentas práticas.
Analogia para o Aluno: "Pensem na internet como uma grande cidade. Os endereços IP são como os endereços de rua das casas. Sem eles, as encomendas (dados) não saberiam para onde ir. A máscara de rede é o código postal, que define a 'vizinhança' (a sub-rede)."
Roteiro da Aula: - Comece com a analogia da cidade. - Apresente a estrutura do endereço IPv4 (quatro octetos) e IPv6 (formato mais complexo). - Explique a máscara de rede em binário para mostrar como a rede e o host são separados (exemplo: 192.168.1.10 com 255.255.255.0). - Apresente 'ping' como um teste de conectividade simples. - Apresente 'traceroute' como uma forma de ver os "saltos"(routers) no caminho. - Mostre a saída do 'ifconfig' ou 'ip addr' para que os alunos identifiquem o seu próprio endereço. - No exercício, peça-lhes para 'ping' o router local e depois um site externo.

0. Introdução a Fundamentos de Redes (10 horas)

Antes de explorarmos os serviços de rede, é fundamental entender a base: os endereços IP e a comunicação na rede.

- **Endereços IP e Máscaras de Rede** (4 horas)

  O que é um Endereço IP? Um endereço de Protocolo de Internet (IP) é um rótulo numérico atribuído a cada dispositivo conectado a uma rede de computadores que usa o Protocolo de Internet para comunicação. Funciona de forma análoga a um endereço postal, permitindo que os dados sejam enviados para o destino correto. Existem duas versões principais em uso:

  - **IPv4**: Utiliza um formato de 32 bits, geralmente representado por quatro octetos (números de 0 a 255) separados por pontos, como `192.168.1.1`. Cada octeto é um grupo de 8 bits. Este formato está a esgotar-se devido ao número limitado de combinações (aproximadamente 4.3 mil milhões).
  - **IPv6**: Utiliza um formato de 128 bits, representado por oito grupos de quatro dígitos hexadecimais, como `2001:0db8:85a3:0000:0000:8a2e:0370:7334`. Foi criado para fornecer um número vastamente maior de endereços (2^{128}) e inclui características de autoconfiguração.

O Papel da Máscara de Rede A máscara de rede é um valor de 32 bits que, juntamente com o endereço IP, define a sub-rede. A sua função é separar a porção de **rede** da porção de **host** de um endereço IP. A máscara de rede é composta por uma série de uns (1) binários contíguos, seguidos por uma série de zeros (0).

- Os bits que correspondem a '1' na máscara de rede definem a porção de rede do endereço IP. Todos os dispositivos na mesma sub-rede devem ter a mesma porção de rede.
- Os bits que correspondem a '0' na máscara de rede definem a porção de host, que é única para cada dispositivo dentro dessa sub-rede.

A notação **CIDR (Classless Inter-Domain Routing)**, como '/24', é uma forma compacta de representar a máscara de rede. O número após a barra indica a quantidade de bits que compõem a porção de rede do endereço. Por exemplo, '/24' significa que os primeiros 24 bits são a parte da rede, e os restantes 8 bits são para os hosts.

```
1                    Endereço IP: 192.168.1.50
2                    Máscara de Rede: 255.255.255.0  (ou /24)
```

Listing 1: Exemplo de IP e Máscara

- **Ferramentas de Diagnóstico de Rede** (3 horas)

Ping O comando 'ping' é uma ferramenta fundamental para testar a conectividade entre dois hosts. Ele envia pacotes de requisição de eco **ICMP (Internet Control Message Protocol)** para um destino. Se o destino estiver ativo e acessível, ele responderá com um pacote de resposta de eco. O 'ping' mede o tempo de ida e volta dos pacotes ('RTT - Round-Trip Time') e a percentagem de perda de pacotes, sendo uma forma rápida de diagnosticar problemas básicos de rede.

Traceroute O 'traceroute' mostra o caminho completo que um pacote de dados percorre para chegar ao seu destino. Ele utiliza a mesma técnica do 'ping' mas com um valor de **TTL (Time To Live)** crescente para cada pacote. O TTL é um contador de "saltos"(hops), ou seja, routers que o pacote pode atravessar antes de ser descartado. Ao enviar pacotes com TTLs de 1, 2, 3, etc., o 'traceroute' recebe uma mensagem de erro de cada router no caminho, permitindo-lhe mapear a rota completa.

ifconfig / ip addr Estes comandos são essenciais para ver a configuração das interfaces de rede do seu próprio computador. Eles mostram o seu endereço IP, a máscara de rede, o endereço MAC e o estado da interface (se está ativa ou inativa). O 'ip addr' é a ferramenta moderna recomendada para substituição do 'ifconfig', que se tornou obsoleto em muitas distribuições Linux. O 'ip' é mais abrangente, permitindo gerir e visualizar todas as aspetos da configuração de rede.

```
1                    ping google.com       # Testar a
                         conectividade com um host
2                    traceroute google.com   # Seguir o caminho
                         até um destino
3                    ifconfig / ip addr       # Ver os detalhes
                         das interfaces de rede
```

Listing 2: Comandos de Diagnóstico de Rede

- **Exercício de Consolidação** (3 horas)

1. Abra o terminal e use 'ifconfig' ou 'ip addr' para encontrar o endereço IP da sua máquina.

2. Use o comando ‘ping‘ para testar a conectividade com o router da rede (geralmente 192.168.1.1 ou 10.0.0.1).
3. Use o ‘traceroute‘ para seguir o caminho até um site conhecido e explique o que vê em cada "salto".

—

Objetivos: O aluno será capaz de gerir serviços de rede usando ‘systemd‘ e ‘systemctl‘, e entender a diferença entre gestão de estado e de arranque. **Pré-requisitos:** Conhecimento da secção anterior. **Material de Apoio:** Guia de referência rápida para ‘systemctl‘, exemplos de ficheiros de serviço (‘.service‘).
Visão Geral: O ‘systemd‘ é o coração dos sistemas Linux modernos. O professor deve focar-se na diferença entre o estado do serviço (ligado/desligado) e o estado de arranque (habilitado/desabilitado).
Analogia para o Aluno: "O ‘systemd‘ é o maestro de uma orquestra. Ele garante que cada músico (serviço) entre na altura certa. O ‘systemctl‘ é o ’comando de voz’ do maestro. Com ele, pedimos ao serviço para ’começar a tocar’ (‘start‘) ou ’calar-se’ (‘stop‘). O ‘enable‘ é como dizer ao músico: ’Quando a orquestra começar a tocar no próximo concerto, entra também!’. O ‘disable‘ é o oposto."
Roteiro da Aula: - Explique o papel do ‘systemd‘ como o ‘PID 1‘. - Demonstre ‘systemctl status <serviço>‘ e mostre o que significa ‘active (running)‘, ‘inactive‘ e ‘enabled‘. - Execute ‘systemctl start sshd‘ e depois ‘systemctl status sshd‘ para mostrar a transição de estado. - Explique a diferença crucial entre ‘start/stop‘ (gestão do momento) e ‘enable/disable‘ (gestão do arranque). - Introduza os conceitos de ‘Timers‘ e ‘Sockets‘ como alternativas inteligentes para ‘cron‘ e ‘xinetd‘. - No exercício, guie os alunos na criação de um ‘.service‘ e um ‘.timer‘ para reforçar a matéria.

1. Serviços de rede (15 horas)

Nesta secção, vamos explorar como os serviços de rede são geridos no Linux, desde o seu início até ao encerramento, e os principais ficheiros e diretórios envolvidos neste processo.

- **O Gestor de Tarefas: O Papel do `systemd`** (3 horas)

  O que é o `systemd`? O `systemd` é um sistema de inicialização e gestor de serviços para sistemas operativos Linux. Ele é o primeiro processo a ser iniciado pelo kernel (com o **PID 1**) e é responsável por gerir todo o ciclo de vida dos outros processos, incluindo serviços de rede. Ao contrário de sistemas de inicialização mais antigos (como o SysVinit), o ‘systemd‘ utiliza uma abordagem paralela, o que resulta num arranque de sistema mais rápido.

  A filosofia do ‘systemd‘ baseia-se em "unidades"(‘units‘), que são ficheiros de configuração simples que descrevem recursos do sistema e como devem ser geridos. Estas unidades incluem:

  - `.service`: Para daemons e serviços de rede.
  - `.timer`: Para agendar a execução de tarefas.
  - `.socket`: Para ativar serviços "on-demand".
  - `.target`: Para agrupar várias unidades, como o ‘multi-user.target‘.

- **Controlar Serviços com `systemctl`** (5 horas)

  Gerir o estado do serviço O comando ‘systemctl‘ é a principal ferramenta para interagir com o ‘systemd‘. A gestão de um serviço envolve a compreensão de duas dimensões de estado:

  - **Estado do momento (‘runtime‘):** O serviço pode estar **ativo** (‘active (running)‘) ou **inativo** (‘inactive‘). Os comandos ‘systemctl start‘, ‘stop‘, ‘restart‘ e ‘status‘ gerem e mostram este estado.
  - **Estado de arranque (‘boot time‘):** O serviço pode estar **habilitado** (‘enabled‘) ou **desabilitado** (‘disabled‘). Um serviço habilitado é configurado para ser iniciado automaticamente durante o arranque do sistema. Os comandos ‘systemctl enable‘ e ‘disable‘ modificam esta configuração, criando ou removendo um ‘symlink‘ (ligação simbólica) nos diretórios de arranque do sistema.

```
# Ver o estado de um serviço (Exemplo: SSH)
systemctl status sshd

# Ligar, desligar e reiniciar um serviço
systemctl start sshd
systemctl stop sshd
systemctl restart sshd
```

Listing 3: Comandos de gestão de serviços

- **Tarefas Programadas e Serviços On-Demand** (5 horas)

  Systemd Timers (‘.timer‘) São a alternativa moderna e recomendada ao ‘cron‘ para agendar a execução de tarefas. A sua principal vantagem é a robustez: um ‘timer‘ pode ser configurado para ser executado no próximo arranque, mesmo que o sistema tenha estado desligado durante a sua hora de execução. A configuração é feita em dois ficheiros: um ‘.service‘ que define a tarefa a ser executada e um ‘.timer‘ que define o agendamento.

  Systemd Sockets (‘.socket‘) Os ‘sockets‘ permitem a ativação de serviços "on-demand". O ‘systemd‘ pode "escutar"numa porta de rede e, em vez de iniciar o serviço imediatamente, ele aguarda por uma ligação. Apenas quando uma ligação é recebida é que o serviço é iniciado e a ligação é passada para o daemon. Isto é ideal para serviços que não são usados com frequência, pois poupa recursos do sistema ao não os manter em execução constante.

- **Exercício de Consolidação** (2 horas)

  1. Crie uma unidade de serviço (‘.service‘) e uma unidade de temporizador (‘.timer‘) para um script simples que escreve a data e hora num ficheiro a cada minuto.
  2. Verifique o estado do temporizador e do serviço com `systemctl list-timers`.

—

Objetivos: O aluno será capaz de configurar e gerir serviços "on-demand"usando ‘xinetd‘. **Pré-requisitos:** Conhecimento da secção sobre ‘systemd‘. **Material de Apoio:** Ficheiro de configuração de exemplo ‘/etc/xinet.d/ftp‘.
Visão Geral: O ‘xinetd‘ é um "super-servidor de internet estendido". A sua principal função é poupar recursos ao gerir serviços que não precisam de estar sempre em execução. É um conceito mais antigo, mas ainda importante para entender.
Analogia para o Aluno: "O ‘xinetd‘ é um rececionista. Ele fica na entrada do escritório à espera de clientes. Em vez de ter o funcionário do serviço de FTP a trabalhar o dia todo e a gastar energia, o rececionista (‘xinetd‘) fica de guarda. Quando um cliente chega e pede FTP, o rececionista acorda o funcionário do FTP para o atender. Assim que a tarefa termina, o funcionário volta a ’dormir’. É uma forma de poupar recursos."
Roteiro da Aula: - Explique a diferença entre um serviço standalone (sempre ativo) e um gerido pelo ‘xinetd‘ (ativado por pedido). - Mostre que os ficheiros de configuração estão em ‘/etc/xinet.d/‘. - Abra o ficheiro de configuração de um serviço (como o ‘ftp‘) e explique as diretivas mais comuns (‘server‘, ‘port‘, ‘disable‘). - Para o exercício, altere a diretiva ‘disable‘ de ‘yes‘ para ‘no‘ para ativar o serviço e reinicie o ‘xinetd‘.

2. XINET,d (5 horas)

O `xinetd` é como um "rececionista"que só acorda um funcionário (serviço) quando alguém aparece para o ver.

- **Configuração e Gestão** (3 horas)

  O que é o ‘xinetd‘? O ‘xinetd‘ (Extended Internet Daemon) é um super-servidor que gere serviços de rede que não necessitam de estar permanentemente em execução. Ele "escuta"em portas de rede específicas. Quando uma ligação chega a uma dessas portas, o ‘xinetd‘ **avalia as regras de segurança** e, se a ligação for permitida, ele inicia o serviço correspondente e encaminha a ligação para o mesmo. Após a conclusão do serviço, este é terminado, libertando os recursos que estava a consumir. Esta abordagem é particularmente útil para serviços que não têm uma alta frequência de uso, como o ‘telnet‘ ou o ‘tftp‘.

  Ficheiros de Configuração A configuração do ‘xinetd‘ é modular e baseia-se em ficheiros de configuração individuais para cada serviço, localizados no diretório `/etc/xinet.d/`. Cada ficheiro define um serviço e contém as seguintes diretivas importantes:

  - `service <nome_serviço>`: Define o início de uma nova configuração de serviço.
  - `server`: O caminho completo para o executável do servidor (ex: ‘/usr/sbin/in.tftpd‘).
  - `port`: O número da porta de rede que o ‘xinetd‘ vai escutar para este serviço.
  - `type`: Especifica o tipo de socket (‘stream‘, ‘dgram‘).
  - `protocol`: O protocolo de transporte (‘tcp‘, ‘udp‘).
  - `disable`: A diretiva de ativação/desativação. O valor ‘yes‘ desativa o serviço.

- **Exercício de Consolidação** (2 horas)

  1. Encontre o ficheiro de configuração do serviço `ftp` no diretório `/etc/xinet.d/`.
  2. Altere o valor da opção `disable` para habilitá-lo e reinicie o serviço ‘xinetd‘.
  3. (Opcional) Crie uma nova configuração para um serviço simples.

—

Objetivos: O aluno será capaz de configurar ‘TCPWrappers‘ para controlar o acesso a serviços de rede com base no endereço IP. **Pré-requisitos:** Compreensão de endereços IP. **Material de Apoio:** Diagrama do fluxo de processamento dos ficheiros ‘hosts.allow‘ e ‘hosts.deny‘.
Visão Geral: Os ‘TCPWrappers‘ oferecem um controlo de acesso simples, mas poderoso, para serviços de rede. A regra de ouro é: os ficheiros de permissão (‘hosts.allow‘) são lidos antes dos ficheiros de negação (‘hosts.deny‘).
Analogia para o Aluno: "Pensem nos ‘TCPWrappers‘ como os seguranças à porta de um clube. Eles têm duas listas: a dos ’convidados VIP’ (‘hosts.allow‘) e a ’lista negra’ (‘hosts.deny‘). Quando alguém tenta entrar, o segurança primeiro olha para a lista de convidados. Se estiver lá, a pessoa entra imediatamente. Só se não estiver na primeira lista é que o segurança olha para a ’lista negra’ para ver se a pessoa deve ser barrada. Por isso, a ordem é tão importante!"
Roteiro da Aula: - Explique a função de ‘hosts.allow‘ e ‘hosts.deny‘ e a ordem de processamento. - Mostre a sintaxe das regras: ‘serviço: hosts‘. - Apresente exemplos práticos com ‘ALL‘, ‘EXCEPT‘ e sub-redes (‘192.168.1.0/24‘). - No exercício, comece por negar o acesso a todos (‘ALL: ALL‘) em ‘hosts.deny‘. - Depois, mostre como reverter a negação para um IP específico (‘sshd: 192.168.1.100‘) no ‘hosts.allow‘.

3. TCPWrappers (5 horas)

Pense nos `TCPWrappers` como um porteiro. Ele decide quem pode entrar (permitir) e quem não pode (negar) num serviço, com base no endereço IP.

- **As Duas Listas: ‘hosts.allow‘ e ‘hosts.deny‘** (2 horas)

  O que são os ‘TCPWrappers‘? Os ‘TCPWrappers‘ (também conhecidos como ‘libwrap‘) são uma camada de segurança adicional para serviços de rede. Eles fornecem um controlo de acesso simples, mas eficaz, a nível de host, antes que o daemon do serviço seja executado. Um serviço que é compilado com suporte a ‘libwrap‘ verifica automaticamente os ficheiros `/etc/hosts.allow` e `/etc/hosts.deny` para determinar se um pedido de conexão deve ser permitido ou negado.

  Fluxo de Processamento A ordem de leitura dos ficheiros é o aspeto mais importante dos ‘TCPWrappers‘:

  1. O serviço consulta o ficheiro **‘/etc/hosts.allow‘**.
  2. Se uma regra que **permite** o acesso for encontrada, a ligação é estabelecida e a verificação termina.
  3. Se nenhuma regra que permita o acesso for encontrada, o serviço consulta o ficheiro **‘/etc/hosts.deny‘**.
  4. Se uma regra que **nega** o acesso for encontrada, a ligação é rejeitada.
  5. Se nenhuma regra for encontrada em nenhum dos ficheiros, a ligação é **permitida** por padrão.

  Isto significa que uma regra em ‘hosts.allow‘ tem precedência sobre uma regra em ‘hosts.deny‘.

- **Sintaxe e Exemplos Avançados** (2 horas)

  Sintaxe das Regras A sintaxe das regras é: `serviços: hosts`.

  – `serviços`: Uma lista separada por vírgulas de nomes de daemons. Pode-se usar a palavra-chave ‘ALL‘ para todos os serviços.

- `hosts`: Uma lista de endereços IP, nomes de host, sub-redes (em formato ‘x.y.z.w/-mask‘) ou as palavras-chave ‘ALL‘ ou ‘LOCAL‘.

Exemplos Práticos

- ‘sshd, vsftpd: 192.168.1.0/24‘: Permite acesso SSH e FTP para toda a sub-rede ‘192.168.1.x‘.
- ‘ALL: ALL‘: Uma regra muito comum em ‘hosts.deny‘ para negar o acesso a todos os serviços, a partir de qualquer host.
- ‘sshd: 192.168.1.10 EXCEPT 192.168.1.100‘: Permite acesso SSH a toda a rede ‘192.168.1.x‘ exceto para o host ‘192.168.1.100‘. Esta regra não funciona diretamente no ‘hosts.allow‘, pois a palavra-chave ‘EXCEPT‘ é interpretada de forma diferente em cada ficheiro. A melhor prática é usar regras separadas.

```
# No ficheiro /etc/hosts.allow
sshd: 192.168.1.100 EXCEPT 192.168.1.101
```

Listing 4: Exemplo de configuração

- **Exercício de Consolidação** (1 hora)
  1. Configure o seu ficheiro `hosts.deny` para negar o acesso SSH a todos.
  2. No ficheiro `hosts.allow`, adicione o IP do seu computador para poder aceder.

—

Objetivos: O aluno será capaz de instalar e configurar um ambiente NIS (‘ypserv‘ e ‘ypbind‘) para centralizar a gestão de utilizadores. **Pré-requisitos:** Conhecimentos sobre ficheiros de utilizador (‘/etc/passwd‘, ‘/etc/group‘). **Material de Apoio:** Diagrama de arquitetura NIS (mestre-cliente).
Visão Geral: O NIS (Network Information Service) é um dos primeiros serviços de diretório, usado para centralizar contas de utilizador e grupos. É um sistema cliente-servidor.
Analogia para o Aluno: "O NIS é como um cartão de estudante que funciona em toda a escola. Em vez de ter um cartão diferente para a biblioteca, para o refeitório e para o laboratório, vocês têm um só cartão que é gerido por um escritório central. O servidor NIS é o escritório central e os clientes são as máquinas onde usam o vosso cartão."
Roteiro da Aula: - Explique a arquitetura mestre-cliente. - Mostre os ficheiros que o NIS centraliza (‘/etc/passwd‘, ‘/etc/group‘). - Explique os comandos ‘ypserv‘ (servidor) e ‘ypbind‘ (cliente). - No exercício, os alunos devem: 1. Configurar um servidor ‘ypserv‘. 2. Ligar as máquinas clientes a ele. 3. Usar ‘ypcat‘ para ver as contas de utilizador a partir do cliente.

4. NIS (10 horas)

O NIS é como ter uma única "identidade"para toda uma rede de computadores. Em vez de ter uma conta de utilizador em cada máquina, você tem uma conta num servidor central que funciona em todas as máquinas NIS.

- **Arquitetura NIS (Servidor Mestre e Cliente)** (3 horas)

O que é o NIS? O NIS (anteriormente conhecido como Yellow Pages, daí o prefixo ‘yp‘) é um protocolo de cliente-servidor para distribuição de dados de configuração do sistema, como nomes de utilizador, senhas e grupos, através de uma rede. Ele centraliza informações que normalmente seriam armazenadas localmente em ficheiros como ‘/etc/passwd‘ e ‘/etc/group‘. A sua arquitetura é baseada num modelo de **domínio**, onde um servidor mestre gere os "mapas"de dados e os distribui para os clientes ou para servidores subordinados.

Mecanismo de funcionamento O NIS opera através da partilha de "mapas", que são bases de dados ‘dbm‘ (formato de base de dados de Berkeley) geradas a partir de ficheiros de texto do sistema. O servidor NIS mantém estes mapas atualizados e os clientes NIS solicitam as informações sempre que precisam. Este sistema simplifica a gestão de identidades em redes pequenas e médias. O cliente NIS utiliza o serviço **‘ypbind‘** para encontrar o servidor NIS mestre (ou subordinado) e se ligar a ele, enquanto o servidor NIS executa o daemon **‘ypserv‘**.

- **Configuração de um Servidor NIS Mestre** (5 horas) A configuração de um servidor NIS envolve os seguintes passos, que devem ser seguidos de forma rigorosa para evitar falhas de autenticação:
  1. Instalar os pacotes necessários: ‘ypserv‘ (servidor), ‘ypbind‘ (cliente) e ‘yp-tools‘.
  2. Definir o nome do domínio NIS no servidor, que deve ser único.
  3. Editar o ficheiro de configuração do ‘ypserv‘ (‘/etc/ypserv.conf‘) para definir quais hosts têm permissão para aceder aos mapas.
  4. Executar o comando ‘ypinit -m‘ para criar os mapas iniciais do NIS a partir dos ficheiros do sistema.
  5. Iniciar e habilitar o serviço ‘ypserv‘ com ‘systemctl enable ypserv systemctl start ypserv‘.
- **Exercício de Consolidação** (2 horas)
  1. Numa máquina cliente, use o comando `ypwhich` para ver qual é o servidor NIS.
  2. Use `ypcat passwd` para listar as contas de utilizador.
  3. Crie uma nova conta no servidor NIS e verifique se consegue iniciar sessão com essa conta a partir do computador cliente.

—

Objetivos: O aluno será capaz de configurar um servidor DHCP para atribuir endereços IP de forma dinâmica e estática. **Pré-requisitos:** Sólidos conhecimentos sobre endereçamento de rede (IP e máscaras). **Material de Apoio:** Diagrama do processo DORA, exemplos de ficheiro ‘dhcpd.conf‘.
Visão Geral: O DHCP (Dynamic Host Configuration Protocol) é um protocolo essencial para a gestão automática de endereços IP. Destaque a diferença entre atribuição dinâmica (para novos dispositivos) e estática (para servidores).
Analogia para o Aluno: "O DHCP é como um rececionista de hotel. Quando um novo hóspede (dispositivo) chega, ele não precisa de procurar um quarto. O rececionista (servidor DHCP) atribui-lhe um quarto disponível (endereço IP). O processo DORA é a conversa passo a passo: o cliente ‘D‘iscover (descobre) o servidor, o servidor ‘O‘ffer (oferece) um IP, o cliente ‘R‘equest (pede) esse IP, e o servidor ‘A‘cknowledge (confirma)."
Roteiro da Aula: - Apresente o processo DORA e o seu significado. - Mostre o ficheiro de configuração ‘dhcpd.conf‘ e explique as diretivas: ‘subnet‘, ‘range‘, ‘option routers‘, ‘host‘ e ‘hardware ethernet‘. - Explique a lógica por trás da atribuição de IP: 1. Endereços estáticos. 2. Endereços dinâmicos da ‘range‘. - Apresente o conceito de ‘DHCP Relay Agent‘. - No exercício, os alunos devem criar uma configuração DHCP para uma sub-rede e uma atribuição estática para um dispositivo específico usando o seu MAC address.

5. DHCP (15 horas)

O DHCP é um protocolo que automaticamente dá um "endereço de rua"(o endereço IP) a cada dispositivo que se liga à sua rede.

- **O Processo DORA: Como funciona?** (4 horas)

O que é o DHCP? O Protocolo de Configuração Dinâmica de Host (DHCP) é um protocolo de rede que permite a um servidor atribuir automaticamente um endereço IP e outras configurações de rede (como a máscara de sub-rede, gateway padrão e DNS) a um cliente. Isto elimina a necessidade de configurar manualmente cada dispositivo na rede, simplificando a gestão. O DHCP utiliza o protocolo de transporte **UDP (User Datagram Protocol)**, nas portas '67' (servidor) e '68' (cliente).

O Processo DORA O processo de atribuição de um endereço IP via DHCP é conhecido como **DORA**, um acrónimo para os quatro passos de troca de mensagens. As mensagens são transmitidas em *broadcast* na fase inicial para que o cliente possa encontrar o servidor, independentemente de ter um endereço IP pré-atribuído.

1. **D**iscover (**Descoberta**): O cliente envia uma mensagem de transmissão ('DHCPDISCOVER') na rede para encontrar servidores DHCP.
2. **O**ffer (**Oferta**): Um ou mais servidores DHCP respondem com uma mensagem de oferta ('DHCPOFFER'), propondo um endereço IP disponível.
3. **R**equest (**Pedido**): O cliente escolhe uma das ofertas e envia uma mensagem de pedido ('DHCPREQUEST') para o servidor selecionado, formalizando a sua intenção de usar o endereço IP oferecido.
4. **A**cknowledge (**Confirmação**): O servidor confirma a atribuição do endereço IP com uma mensagem de confirmação ('DHCPACK'). É neste momento que a concessão (lease) do IP é ativada.

- **Configuração e Atribuição de Endereços** (7 horas) A configuração do servidor DHCP é feita no ficheiro de configuração principal, geralmente localizado em `/etc/dhcp/dhcpd.conf`. Os blocos de configuração mais importantes são:
  - **Subnet**: O bloco 'subnet' define as configurações de rede para uma sub-rede específica, incluindo o endereço da sub-rede e a máscara.
  - **Range**: A diretiva 'range' define o conjunto de endereços IP que o servidor pode atribuir dinamicamente. É importante que este intervalo não inclua endereços já em uso estaticamente.
  - **Options**: As diretivas 'option' fornecem configurações adicionais ao cliente, como o endereço do router ('option routers'), o servidor DNS ('option domain-name-servers') e o nome do domínio ('option domain-name').
  - **Static Leases**: A diretiva 'host' é utilizada para atribuir um endereço IP fixo a um dispositivo específico com base no seu endereço MAC ('hardware ethernet'). Isto é crucial para servidores ou impressoras, que necessitam de um IP consistente.
- **Conceitos Avançados: DHCP Relay Agent** (2 horas)

DHCP Relay Agent Em redes complexas com várias sub-redes separadas por routers, um cliente numa sub-rede pode não conseguir ver as mensagens de 'broadcast' de um servidor DHCP noutra sub-rede. Nesses casos, um **DHCP Relay Agent** (agente de retransmissão DHCP) é configurado num router. O agente de retransmissão recebe as mensagens de 'DHCPDISCOVER' de 'broadcast' e as encaminha como mensagens de 'unicast' diretamente para o servidor DHCP, permitindo que um único servidor sirva múltiplos segmentos de rede.

- **Exercício de Consolidação** (2 horas)
  1. Configure um servidor DHCP para a sua rede de treino com um 'range' de IPs dinâmicos.
  2. Adicione uma entrada estática para o computador do instrutor.
  3. Configure um cliente Linux para obter o IP do servidor e verifique se as configurações (IP, gateway, DNS) estão corretas.

—

Objetivos: O aluno será capaz de configurar um servidor DNS primário (autoritativo) usando BIND, incluindo zonas de pesquisa direta e inversa. **Pré-requisitos:** Sólidos conhecimentos sobre endereçamento de rede e portas de serviços. **Material de Apoio:** Diagrama de hierarquia DNS, exemplos de ficheiros de zona.
Visão Geral: O DNS é um dos serviços mais importantes da internet. O professor deve dominar a hierarquia e o processo de consulta recursiva. A prática deve focar-se na configuração de zonas de pesquisa direta (‘A‘, ‘CNAME‘) e inversa (‘PTR‘) no BIND.
Analogia para o Aluno: "O DNS é a ’lista telefónica’ da internet. Em vez de decorar números de telefone (endereços IP), nós só precisamos de saber os nomes das pessoas (nomes de sites). A hierarquia de DNS é como se tivéssemos várias listas telefónicas: uma para nomes de família (‘.com‘), outra para nomes próprios (‘google‘), e um ’livro de contactos’ pessoal."
Roteiro da Aula: - Explique a hierarquia de DNS: Raiz, TLD (.com, .org), e servidores autoritativos. - Descreva o processo de consulta recursiva: cliente -> servidor local -> servidores de hierarquia. - Detalhe os tipos de registos mais comuns (‘A‘, ‘AAAA‘, ‘CNAME‘, ‘MX‘, ‘PTR‘) com exemplos práticos. - No laboratório, guie a configuração do BIND: ‘named.conf‘, ficheiro de zona ‘forward‘ (‘db.exemplo.com‘) e ‘reverse‘ (‘db.1.168.192‘). - Enfatize a sintaxe dos ficheiros de zona (o ponto final, ‘TTL‘). $-$ $Use o comando ‘dig‘ para testar as configuraes e veja o resultado.$

6. DNS (20 horas)

O DNS é como a "lista telefónica"da internet. Ele traduz nomes fáceis de lembrar (como ‘google.com‘) em endereços IP que os computadores entendem.

- **Como a Lista Telefónica Funciona: A Hierarquia** (6 horas)

  Hierarquia e Consulta O Sistema de Nomes de Domínio (DNS) é uma base de dados hierárquica e descentralizada, distribuída por milhares de servidores no mundo. A sua hierarquia funciona de forma invertida, com o **domínio raiz** (‘.‘) no topo. Abaixo da raiz, estão os **TLDs (Top-Level Domains)**, como ‘.com‘ ou ‘.org‘. Por baixo dos TLDs, estão os domínios de segundo nível, como ‘google.com‘.

  Processo de Consulta Recursiva Quando um utilizador digita um URL, o seu computador executa uma consulta DNS. Se o servidor DNS local não tiver a resposta em cache, ele inicia um processo de consulta recursiva:

  1. O cliente envia uma consulta para o seu servidor DNS local.
  2. O servidor local, se não souber a resposta, contacta um servidor de raiz para obter o endereço do servidor TLD (‘.com‘).
  3. O servidor TLD aponta para o servidor autoritativo de ‘google.com‘.
  4. O servidor autoritativo fornece o endereço IP final.
  5. O servidor DNS local armazena a resposta em cache e a envia de volta ao cliente.

- **Tipos de Registos DNS (A, AAAA, CNAME, MX, PTR)** (4 horas)

  Registos de Recurso (RRs) Os registos de recurso são entradas na base de dados DNS. Cada registo tem um tipo, um tempo de vida (‘TTL‘), uma classe e dados específicos. Os tipos mais importantes são:

  - `A`: Mapeia um nome de domínio para um endereço **IPv4**. Essencial para a navegação web.
  - `AAAA`: Mapeia um nome de domínio para um endereço **IPv6**.
  - `CNAME`: Cria um "apelido"para um nome de domínio canónico. Permite que vários nomes de host (ex: ‘www.exemplo.com‘, ‘ftp.exemplo.com‘) apontem para o mesmo host (‘exemplo.com‘).
  - `MX`: Especifica o servidor de correio para um domínio.
  - `PTR`: Usado para pesquisa **inversa**, traduzindo um endereço IP para um nome de domínio. Essencial para serviços de correio eletrónico e segurança.

- **Configuração de um Servidor DNS com BIND** (8 horas)

  BIND (Berkeley Internet Name Domain) BIND é o servidor DNS mais utilizado no mundo. A sua configuração principal reside no ficheiro `/etc/named.conf` (ou ‘named.conf.local‘). Para se tornar um servidor DNS autoritativo para um domínio, é necessário configurar zonas.

– **Zonas de ‘forward lookup‘**: Contêm os registos que traduzem nomes para IPs (‘A‘, ‘CNAME‘, etc.). A diretiva ‘$ORIGIN$‘$define o nome de ded o domínio base da zona. O ponto final (‘.‘) no final de um nome de$‘$FullyQualifiedDomainName$‘$).$**$Zonas de ‘reverse lookup‘$**$: Contmos registos ‘PTR‘ que traduzem IPs para$‘$addr.arpa$‘$.

- **Exercício de Consolidação** (2 horas)

  1. Crie um servidor DNS com o BIND e configure-o para ser autoritativo para um domínio fictício (‘meu-curso.com‘).
  2. Adicione vários registos (A, CNAME) para máquinas na sua rede.
  3. Configure uma máquina cliente para usar este novo servidor DNS e use o ‘dig‘ para testar se as consultas estão a funcionar.

—

Objetivos: O aluno será capaz de estabelecer ligações seguras via SSH, gerir utilizadores e permissões, e configurar a autenticação por chave pública. **Pré-requisitos:** Conhecimento sobre a linha de comandos e gestão básica de utilizadores. **Material de Apoio:** Diagrama do processo de autenticação por chave SSH.
Visão Geral: O SSH (Secure Shell) é a espinha dorsal da administração remota de sistemas Linux. O professor deve focar-se na segurança e na eficiência do uso de chaves.
Analogia para o Aluno: "O SSH é como uma porta de entrada super-segura para o seu servidor. Em vez de uma chave simples (palavra-passe), que pode ser copiada, usamos uma ’chave’ digital muito mais robusta. O par de chaves, pública e privada, é como ter uma fechadura única na porta do servidor (chave pública) e a sua chave pessoal na sua mão (chave privada). Ninguém mais pode abrir a porta."
Roteiro da Aula: - Explique a função do SSH e a diferença entre o cliente (‘ssh‘) e o servidor (‘sshd‘). - Demonstre o acesso remoto básico com ‘ssh user@host‘. - Apresente o conceito de autenticação por chave pública, mostrando o processo de criação de chaves (‘ssh-keygen‘) e de cópia para o servidor (‘ssh-copy-id‘). - Explique as configurações de segurança mais comuns no ficheiro ‘sshd$_c onfig$‘, $como a desativa o do login por palavra-passe. - Demonstre como usar o ‘scp‘ para copiar ficheiros de forma segura.$

7. SSH (Secure Shell) (10 horas)

O SSH é a ferramenta mais usada para aceder e gerir servidores Linux de forma remota e segura.

- **Conceitos Fundamentais** (3 horas)

O que é o SSH? O Secure Shell (SSH) é um protocolo de rede criptografado utilizado para ligar e gerir um computador de forma segura através de uma rede insegura. Funciona na porta **TCP 22** por padrão. O SSH fornece um canal de comunicação seguro, onde todos os dados, incluindo a autenticação e os comandos, são criptografados.

Componentes e Sessões O SSH tem uma arquitetura cliente-servidor:

- **Servidor SSH ('sshd')**: O daemon que corre no servidor e aceita ligações. Ele é configurado no ficheiro `/etc/ssh/sshd_config`.
- **Cliente SSH ('ssh')**: O programa que o utilizador usa para se ligar ao servidor.
- **Canal SSH**: O canal de comunicação criptografado é criado após a autenticação do utilizador e a troca de chaves.

Para além de sessões de terminal, o SSH permite o **encaminhamento de portas ('port forwarding')** e a transferência de ficheiros segura através de 'scp' e 'sftp'.

- **Autenticação por Chave Pública** (5 horas)

Criptografia de Chave Pública A autenticação por chave pública é o método de segurança recomendado para o SSH. É baseada na criptografia assimétrica, que envolve um par de chaves criptográficas: uma chave **privada** (mantida em segredo pelo utilizador) e uma chave **pública** (que pode ser partilhada). O processo de autenticação é o seguinte:

1. O utilizador gera um par de chaves usando o comando 'ssh-keygen'.
2. O utilizador copia a chave pública para o servidor remoto, adicionando-a ao ficheiro ' /.ssh/authorized$_keys$'.$Quandooclientetentaligar-se, oservidorenviaumdesafiocriptogrfico.$

3. O cliente usa a sua chave privada para assinar o desafio e envia a resposta de volta.
4. O servidor usa a chave pública armazenada para verificar a assinatura. Se for válida, a autenticação é bem-sucedida.

```
# Criar um par de chaves (pública e privada)
ssh-keygen -t rsa

# Copiar a chave pública para o servidor
ssh-copy-id user@servidor

# Aceder ao servidor sem palavra-passe
ssh user@servidor
```

Listing 5: Geração e uso de Chaves SSH

- **Exercício de Consolidação** (2 horas)

1. Geração de chaves SSH numa máquina cliente.
2. Copie a chave pública para a máquina do instrutor.
3. Configure o servidor SSH para desativar a autenticação por palavra-passe.

—

Objetivos: O aluno será capaz de configurar regras de firewall para controlar o tráfego de rede e proteger serviços. **Pré-requisitos:** Conhecimentos sobre endereçamento de rede e portas de serviços. **Material de Apoio:** Diagrama de fluxo de pacotes do ‘iptables‘, exemplos de regras comuns.
Visão Geral: A firewall é a primeira linha de defesa de um servidor. O professor deve apresentar o ‘iptables‘ como a ferramenta de baixo nível e o ‘ufw‘ como uma interface simplificada, mais acessível para iniciantes.
Analogia para o Aluno: "A firewall é como o ’guarda da alfândega’ do seu servidor. Ela inspeciona todos os ’carros’ (pacotes de dados) que chegam e saem. A firewall pode decidir quais carros podem entrar ou sair, e por qual ’porta’ (porta de serviço). O ‘iptables‘ é o livro de regras complexo que o guarda usa, enquanto o ‘ufw‘ é uma versão simplificada e fácil de entender desse livro."
Roteiro da Aula: - Explique o papel de uma firewall e a diferença entre políticas padrão (‘ACCEPT‘ e ‘DROP‘). - Apresente o ‘iptables‘ e a sua sintaxe base (‘-A‘, ‘-I‘, ‘-D‘, ‘-P‘). Demonstre regras para permitir ou bloquear tráfego para portas específicas. - Apresente o ‘ufw‘ (Uncomplicated Firewall) como uma alternativa mais simples. Mostre comandos como ‘ufw allow 80/tcp‘ e ‘ufw status‘. - Compare os dois para que o aluno perceba quando usar um ou outro.

8. Firewall (iptables/ufw) (10 horas)

A firewall é um sistema que controla o tráfego de rede para proteger o seu computador.

- **Conceitos de Firewall** (3 horas)

  O que é uma Firewall? Uma firewall é um sistema de segurança de rede que monitoriza e controla o tráfego de rede de entrada e saída com base num conjunto de regras de segurança. O seu principal objetivo é proteger redes privadas de ataques externos e não autorizados. No Linux, a firewall é implementada no kernel e as regras são geridas por ferramentas como o ‘iptables‘.

  Fluxo de Pacotes e Cadeias (‘Chains‘) O ‘iptables‘ gere regras de filtro para pacotes. A sua operação baseia-se em tabelas, que contêm cadeias. As cadeias mais importantes são:

  - **‘INPUT‘**: Cadeia para pacotes que entram no sistema e são destinados a ele.
  - **‘OUTPUT‘**: Cadeia para pacotes que saem do sistema, originados por ele.
  - **‘FORWARD‘**: Cadeia para pacotes que estão a ser encaminhados através do sistema (que age como um router).

  As regras são lidas de cima para baixo. A primeira regra que corresponde ao pacote determina a sua ação. Se nenhuma regra corresponder, a **política padrão** da cadeia é aplicada. As políticas padrão mais comuns são ‘ACCEPT‘ (tudo é permitido por padrão) e ‘DROP‘ (tudo é negado por padrão, exceto o que for explicitamente permitido).

- **iptables: A Ferramenta de Poder** (4 horas)

  Sintaxe de Regras A sintaxe do ‘iptables‘ pode ser complexa. As opções principais são:

  - ‘-A <CHAIN>‘: Anexa a regra ao final da cadeia.
  - ‘-I <CHAIN>‘: Insere a regra no início da cadeia.
  - ‘-D <CHAIN>‘: Deleta uma regra.
  - ‘-P <CHAIN>‘: Define a política padrão da cadeia.

- '-p <protocolo>': Especifica o protocolo ('tcp', 'udp', 'icmp').
- '–dport <porta>': Especifica a porta de destino.
- '-s <IP>': Especifica o IP de origem.
- '-j <TARGET>': O que fazer com o pacote. 'ACCEPT', 'DROP' e 'REJECT' são alvos comuns.

É crucial guardar as regras com 'iptables-save' para que persistam após o reinício.

```
# Listar regras
sudo iptables -L

# Permitir tráfego SSH (porta 22)
sudo iptables -A INPUT -p tcp --dport 22 -j
    ACCEPT

# Bloquear todo o restante tráfego
sudo iptables -P INPUT DROP
```

Listing 6: Exemplo de regras com iptables

- **ufw: O Guia Simplificado** (3 horas)

O que é 'ufw'? 'ufw' (Uncomplicated Firewall) é uma interface simplificada para o 'iptables' que foi projetada para tornar a configuração da firewall mais fácil. Em vez de lidar com regras complexas, o 'ufw' utiliza comandos mais intuitivos, como 'ufw allow <serviço>'. Por baixo, o 'ufw' traduz estes comandos para as regras correspondentes do 'iptables'. O 'ufw' é ideal para a maioria das necessidades de firewall e é mais fácil de auditar e manter.

```
# Habilitar a firewall
sudo ufw enable

# Permitir acesso à porta 80 (HTTP)
sudo ufw allow 80

# Ver o estado
sudo ufw status verbose
```

Listing 7: Exemplo de regras com ufw

- **Exercício de Consolidação** (2 horas)

1. Instale o 'ufw'.
2. Configure o 'ufw' para permitir apenas o tráfego SSH (porta 22).
3. Tente aceder de um cliente em outra porta (por exemplo, a 80) e verifique se a conexão é rejeitada.