LINUX
Utilitários de Redes

Eduardo Maroñas Monks

YOUTUBE.COM/EMMONKS

Este livro tem como objetivo disponibilizar um guia de referência rápido em utilitários de rede para auxiliar administradores de sistemas operacionais Linux nas rotinas de administração.

1ª edição (revisão 1), Junho de 2023

- Primeira publicação

1ª edição (revisão 2), Julho de 2023

Comandos Básicos

- arp-scan

Utilitários Avançados

- tcpump
- socat
- sshpass
- Python Simple HTTP Server
- comandos r (rcp, rexec, rlogin, rsh)
- net
- snmp*
- nfs*
- sshfs

Iptables

1ª edição (revisão 3), Setembro de 2023

2 Comandos Básicos

- netcat (novos exemplos)
- ftp
- whois
- mutt
- mail/mailx

4 Utilitários Avançados

- nmap
- ngrep

Anexo

- systemctl
- journalctl

Sumário

1. Introdução

Neste capítulo são apresentadas as motivações para o conhecimento das ferramentas de administração de redes em sistemas operacionais Linux.

1.1 Administração de Sistemas Linux

A administração de servidores Linux é a rotina da maioria dos administradores de redes e sistemas. Desta forma, dominar os comandos e utilitários essenciais do Linux agilizam o diagnóstico e a resolução dos problemas. Portanto, o primeiro momento é ter conhecimento de quais são os comandos e utilitários essenciais. Em um segundo momento é explorá-los para adaptá-los as necessidades de cada problema.

1.2 Ambiente para Testes

Para realizar testes com as ferramentas é importante o uso com cenários mais completos. Uma das alternativas é o uso do emulador CORE. Neste emulador é possível criar cenários complexos com centenas de hosts Linux e com o uso das ferramentas apresentadas neste livro. Uma máquina virtual com o emulador CORE está disponível (link da VM).

1.3 Escolha da Distribuição

As principais distribuições são baseadas em Redhat (CentOS, Oracle Linux) ou em Debian (Ubuntu, Mint). As diferenças principais entre as distribuições estão na forma de gerenciamento de pacotes, na organização do sistema de arquivos com caminhos diferentes e na política de uso. Por exemplo, uma distribuição tal como a Debian possui políticas restritas para o uso de pacotes que não sigam o licenciamento GPLv3 e os não disponibiliza por padrão. Entretanto, a escolha da distribuição a ser usada é regida muito mais por uma questão de gosto pessoal ou por necessidade de homologação devido a alguma aplicação a ser instalada ou algum hardware específico.
Os utilitários tratados neste livro estão disponíveis em qualquer distribuição atual e os parâmetros dos exemplos deverão funcionar sem problemas.

2. Comandos Básicos

Neste capítulo são apresentados os comandos básicos para configuração de interfaces de rede. Uma das exceções aos comandos básicos é o comando **ip** que engloba diversas funcionalidades para administração de redes em Linux.

2.1 ifconfig

Descrição 2.1 Comando para realizar configurações na interface de rede e obter estatísticas diversas. Em distribuições atuais vem sendo substituído pelo comando **ip**.

- Para mostrar todas as interfaces disponíveis
  - **Exemplo 2.1** ifconfig -a
- Para mostrar informações sobre a interface eth0
  - **Exemplo 2.2** ifconfig eth0
- Para configurar um IP na interface eth0
  - **Exemplo 2.3** ifconfig eth0 10.0.0.2 netmask 255.255.255.0
  - **Exemplo 2.4** ifconfig eth0 10.0.0.2/24
- Trocar o endereço físico da interface eth0
  - **Exemplo 2.5** ifconfig eth0 hw ether 00:cc:00:ff:ff:ee
- Para criar um outro IP na interface eth0 (IP alias)
  - **Exemplo 2.6** ifconfig eth0:1 10.10.0.2/24
- Para modificar o MTU da interface eth0
  - **Exemplo 2.7** ifconfig eth0 mtu 9000
- Para remover um IP da interface eth0
  - **Exemplo 2.8** ifconfig eth0 0.0.0.0
- Para desativar a interface eth0
  - **Exemplo 2.9** ifconfig eth0 down
- Para desativar um IP alias
  - **Exemplo 2.10** ifconfig eth0:1 down
- Para ativar a interface eth0
  - **Exemplo 2.11** ifconfig eth0 up
- Para adicionar um endereço IPv6 na interface eth0
  - **Exemplo 2.12** ifconfig eth0 inet6 add 2001:0db8:0:200::3/64
- Para remover um endereço IPv6 na interface eth0

- **Exemplo 2.13** ifconfig eth0 inet6 del 2001:0db8:0:200::3/64

2.2 ip

Descrição 2.2 Comando para realizar configurações na interface de rede, roteamento, tunelamento, obter estatísticas e outras diversas funcionalidades. Em distribuições atuais está disponibilizado como principal ferramenta de configuração de interfaces de rede.

- Para listar todas as interfaces
  - **Exemplo 2.14** ip link show
- Para ativar a interface eth0
  - **Exemplo 2.15** ip link set eth0 up
- Para desativar a interface eth0
  - **Exemplo 2.16** ip link set eth0 down
- Para mostrar o endereçamento das interfaces
  - **Exemplo 2.17** ip addr show
- Para mostrar o endereçamento da interface eth0
  - **Exemplo 2.18** ip addr show dev eth0
- Para mostrar os hosts vizinhos (conectados na mesma rede física que tenham de comunicado como host, tabela ARP)
  - **Exemplo 2.19** ip neigh show
  - **Exemplo 2.20** ip neigh show dev eth0
- Para adicionar uma entrada na tabela ARP
  - **Exemplo 2.21** ip neigh add 192.168.1.1 lladdr 00:cc:00:ff:ff:ee dev eth0
- Para remover uma entrada da tabela ARP
  - **Exemplo 2.22** ip neigh del 192.168.1.1 dev eth0
- Limpar toda a tabela ARP de uma VLAN
  - **Exemplo 2.23** ip -s neigh flush dev eth1.212
- Para adicionar mais um IP na interface eth0 (similar ao IP alias do ifconfig)
  - **Exemplo 2.24** ip addr add 192.168.1.2/24 dev eth0
- Para remover endereços adicionais no eth0
  - **Exemplo 2.25** ip addr del 192.168.1.5/24 dev eth0
- Para configurar um endereço IP na interface eth0
  - **Exemplo 2.26** ip addr add 1.2.3.4/24 broadcast 1.2.3.255 dev eth0
- Para remover um endereço IP na interface eth0
  - **Exemplo 2.27** ip addr del 1.2.3.4/24 broadcast 1.2.3.255 dev eth0
- Para trocar o endereço físico da interface eth0

- **Exemplo 2.28** ip link set dev eth0 down
- **Exemplo 2.29** ip link set dev eth0 address 00:cc:00:ff:ff:ee
- **Exemplo 2.30** ip link set dev eth0 up

- Para modificar o MTU da interface para o valor 1476
  - **Exemplo 2.31** ip link set mtu 1476 dev eth0
- Para listar a tabela de roteamento
  - **Exemplo 2.32** ip route show
- Para adicionar uma rota padrão IPv4
  - **Exemplo 2.33** ip route add default via 192.168.1.254
- Para remover uma rota padrão IPv4
  - **Exemplo 2.34** ip route del default via 192.168.1.254
- Para adicionar uma rota estática IPv4
  - **Exemplo 2.35** ip route add 192.168.1.0/24 dev eth0
  - **Exemplo 2.36** ip route add 192.168.8.0/24 via 192.168.254.254
- Para remover uma rota estática IPv4
  - **Exemplo 2.37** ip route del 192.168.1.0/24 dev eth0
  - **Exemplo 2.38** ip route del 192.168.8.0/24 via 192.168.254.254
- Para mostrar os endereços IPv6 de todas as interfaces
  - **Exemplo 2.39** ip -6 a
- Para adicionar um endereço IPv6 na interface eth0
  - **Exemplo 2.40** ip -6 addr add 2001:0db8:0:200::3/64 dev eth0
- Para remover um endereço IPv6 na interface eth0
  - **Exemplo 2.41** ip -6 addr del 2001:0db8:0:200::3/64 dev eth0
- Para adicionar uma rota default IPv6
  - **Exemplo 2.42** ip -6 route add default via 2001:0db8:0:200::1
- Para remover uma rota default IPv6
  - **Exemplo 2.43** ip -6 route del default via 2001:0db8:0:200::1
- Para adicionar uma rota estática IPv6
  - **Exemplo 2.44** ip -6 route add 2001:0db8:0:201::/64 via 2001:0db8:0:200::1
- Para remover uma rota estática IPv6
  - **Exemplo 2.45** ip -6 route del 2001:0db8:0:201::/64 via 2001:0db8:0:200::1
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.46** ip -6 route show

- Para listar a tabela de vizinhos
  - **Exemplo 2.47** ip -6 neigh show
- Para adicionar as marcações de VLANs. No exemplo é adicionada a VLAN ID 100 na interface eth1, criando a interface eth1.100. Para criar múltiplas interfaces marcadadas basta repetir o comando variando o name (nome) e o vlan id.
  - **Exemplo 2.48** ip link add link eth1 name eth1.100 type vlan id 100
- Para remover uma interface marcada para uso de VLANs. No exemplo
  - **Exemplo 2.49** ip link delete eth1.100

2.3 arp

Descrição 2.3 O protocolo ARP (*Address Resolution Protocol*) tem como principal função a tradução de endereços de forma dinâmica na rede. O uso mais comum é prover o processo da tradução de endereços IP e de endereços físicos (MAC address). O Comando **arp** permite gerenciar as tabelas de tradução de endereços e é bastante útil para descoberta de *hosts* e na adminstração de serviços de rede.

- Para listar a tabela ARP
  - **Exemplo 2.50** arp -an
- Para remover uma entrada na tabela ARP
  - **Exemplo 2.51** arp -i eth0 -d 192.168.1.1
- Para adicionar uma entrada na tabela ARP de forma estática
  - **Exemplo 2.52** arp -s 192.168.1.1 00:cc:00:ff:ff:ee

2.4 netstat

Descrição 2.4 Comando para realizar análises e amostragens das conexões de rede, tabelas de roteamento e estatísticas gerais. O comando **ss** é a opção em distribuições mais atuais.

- Para mostrar todas as portas TCP e UDP em escuta e o processos responsáveis
  - **Exemplo 2.53** netstat -nutlp
- Para listar a tabela de roteamento
  - **Exemplo 2.54** netstat -rn
- Para mostrar todas as conexões TCP abertas
  - **Exemplo 2.55** netstat -tlnp
- Para mostrar todas as conexões TCP abertas, de forma contínua
  - **Exemplo 2.56** netstat -tlnpc
- Para mostrar todas as conexões TCP abertas, modo estendido
  - **Exemplo 2.57** netstat -tulpen
- Para listar todas as conexões

- **Exemplo 2.58** netstat -a

- Para listar estatísticas por protocolo
  - **Exemplo 2.59** netstat -s
- Para listar estatísticas somente do protocolo UDP
  - **Exemplo 2.60** netstat -su
- Para listar estatísticas somente do protocolo TCP
  - **Exemplo 2.61** netstat -st
- Para listar estatísticas das interfaces
  - **Exemplo 2.62** netstat -i
- Para obter informações sobre os temporizadores das conexões
  - **Exemplo 2.63** netstat -o
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.64** netstat -rn -A inet6

2.5 route

Descrição 2.5 Comando para mostrar informações e gerenciar tabelas de roteamento em IPv4 e IPv6. Este comando vem sendo substituído pelo **ip** em distribuições mais atuais.

- Para listar a tabela de roteamento
  - **Exemplo 2.65** route
- Para adicionar a rota padrão
  - **Exemplo 2.66** route add default gw 192.168.1.1
- Para adicionar uma rota estática
  - **Exemplo 2.67** route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
  - **Exemplo 2.68** route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254
- Para remover rotas estáticas
  - **Exemplo 2.69** route del -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.254
- Para remover a rota padrão
  - **Exemplo 2.70** route del default gw 192.168.1.1
- Para adicionar uma rota IPv6 padrão
  - **Exemplo 2.71** route -A inet6 add default gw 2001:0db8:0:200::1
- Para remover uma rota IPv6 padrão
  - **Exemplo 2.72** route -A inet6 del default gw 2001:0db8:0:200::1
- Para listar a tabela de roteamento IPv6
  - **Exemplo 2.73** route -A inet6

2.6 ethtool

Descrição 2.6 Comando para gerenciar funções de mais baixo nível da interface de rede, tais como auto-negociação, velocidade, duplex e estatísticas de uso.

- Para mostrar informações gerais sobre a interface eth0
  - **Exemplo 2.74** ethtool eth0
- Para mostrar informações sobre o módulo do kernel (driver) da interface eth0
  - **Exemplo 2.75** ethtool -i eth0
- Para mostrar estatísticas de tráfego da interface eth0
  - **Exemplo 2.76** ethtool -S eth0
- Para mostrar informações sobre configurações de TX, RX e auto-negociação
  - **Exemplo 2.77** ethtool -a eth0
- Para fazer piscar o led da placa (ajudar na identificação física da placa)
  - **Exemplo 2.78** ethtool -p eth0
- Para forçar 100Mbit/s na interface eth0
  - **Exemplo 2.79** ethtool -s eth0 speed 100
- Para desabilitar a auto-negociação na interface eth0
  - **Exemplo 2.80** ethtool -s eth0 autoneg off
- Para forçar o modo full-duplex na interface eth0
  - **Exemplo 2.81** ethtool -s eth0 duplex full
- Para ativar a opção Wake-on-LAN na interface eth0
  - **Exemplo 2.82** ethtool -s eth0 wol g

2.7 dig

Descrição 2.7 Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **nslookup** e e **host**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.83** dig www.senacrs.com.br
- Para descobrir informações sobre um domínio, com saída reduzida
  - **Exemplo 2.84** dig www.senacrs.com.br +noall +answer
- Para descobrir informações sobre um domínio, com saída mínima
  - **Exemplo 2.85** dig www.senacrs.com.br +short
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.86** dig MX senacrs.com.br
- Para descobrir informações sobre os servidores de DNS de um domínio

- **Exemplo 2.87** dig NS senacrs.com.br

- Para receber informações sobre registros de IPv6 de um domínio
  - **Exemplo 2.88** dig AAAA senacrs.com.br
- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.89** dig ANY senacrs.com.br
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.90** dig -x 177.1.214.233
- Para fazer consulta sobre o DNS reversos para determinado IP, com saída reduzida
  - **Exemplo 2.91** dig -x 177.1.214.233 +short
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.92** dig @8.8.8.8 www.senacrs.com.br
- Para realizar uma consulta completa para determinado domínio, passando pelos servidores raiz
  - **Exemplo 2.93** dig www.senacrs.com.br +trace
- Para obter informações sobre o SOA (*Start of Authority*) de um domínio
  - **Exemplo 2.94** dig SOA senacrs.com.br
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.95** dig TXT senacrs.com.br

2.8 host

Descrição 2.8 Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **nslookup** e **dig**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.96** dig www.senacrs.com.br
- Para descobrir informações sobre um domínio
  - **Exemplo 2.97** host www.senacrs.com.br
- Para descobrir informações sobre um domínio, com saída aumentada
  - **Exemplo 2.98** host -v www.senacrs.com.br
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.99** host -t MX senacrs.com.br
- Para descobrir informações sobre os servidores de DNS de um domínio
  - **Exemplo 2.100** host -t NS senacrs.com.br
- Para descobrir informações sobre o IPv6 de um domínio
  - **Exemplo 2.101** host -t AAAA senacrs.com.br

- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.102** host -t ANY senacrs.com.br
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.103** host 177.1.214.233
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.104** host www.senacrs.com.br 8.8.8.8
- Para obter informações sobre o SOA (*Start of Authority*) de um domínio
  - **Exemplo 2.105** host -t SOA senacrs.com.br
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.106** host -t TXT senacrs.com.br

2.9 nslookup

Descrição 2.9 Comando para realizar consultas servidores DNS (*Domain Naming System*), alternativa aos comandos **host** e **dig**.

- Para descobrir informações sobre um domínio
  - **Exemplo 2.107** nslookup www.senacrs.com.br
- Para descobrir informações sobre os servidores de e-mail de um domínio
  - **Exemplo 2.108** nslookup -query=MX senacrs.com.br
- Para descobrir informações sobre os servidores de DNS de um domínio
  - **Exemplo 2.109** nslookup -query=NS senacrs.com.br
- Para receber o maior número de informações sobre um domínio
  - **Exemplo 2.110** nslookup -query=ANY senacrs.com.br
- Para receber informações sobre IPv6 de um domínio
  - **Exemplo 2.111** nslookup -query=AAAA senacrs.com.br
- Para fazer consulta sobre o DNS reversos para determinado IP
  - **Exemplo 2.112** nslookup 177.1.214.233
- Para descobrir informações sobre um domínio, consultando outro servidor, no exemplo o servidor 8.8.8.8
  - **Exemplo 2.113** nslookup www.senacrs.com.br 8.8.8.8
- Para obter informações sobre o SOA (Start of Authority) de um domínio
  - **Exemplo 2.114** nslookup -query=SOA senacrs.com.br
- Para obter informações sobre configurações e validações de SPF de um domínio
  - **Exemplo 2.115** nslookup -query=TXT senacrs.com.br

2.10 ping

Descrição 2.10 Comando para realizar testes de conectividade e condições da rede. Funciona baseado no protocolo ICMP e possibilita obter diagnósticos de perdas, atraso e alcance entre hosts.

- Para realizar teste contínuo
  - **Exemplo 2.116** ping www.senacrs.com.br
- Para realizar teste com 20 pacotes
  - **Exemplo 2.117** ping -c 20 www.senacrs.com.br
- Para realizar teste com pacotes de 1000 Bytes
  - **Exemplo 2.118** ping -s 1000 www.senacrs.com.br
- Para realizar teste com pacotes enviados no intervalo de 0,5 segundos
  - **Exemplo 2.119** ping -i 0.5 www.senacrs.com.br
- Para realizar teste com pacotes no modo flood (inundação)
  - **Exemplo 2.120** ping -f www.senacrs.com.br
- Para utilizar IPv6
  - **Exemplo 2.121** ping6 www.senacrs.com.br

2.11 traceroute

Descrição 2.11 Comando para realizar testes de conectividade, condições da rede e traçar o caminho (rota) entre uma origem e um destino. O **traceroute** é baseado no protocolo ICMP e no campo TTL (*Time to Live*) do protocolo IP que é decrementado a cada passagem por um roteador e ao chegar a zero retorna uma mensagem ICMP para a origem.

- Para realizar testes de rota básico para determinado destino
  - **Exemplo 2.122** traceroute www.senacrs.com.br
- Para realizar testes de rota básico para determinado destino, em IPv6
  - **Exemplo 2.123** traceroute6 www.senacrs.com.br
- Para realizar testes de rota básico para determinado destino, sem resolução de nomes (DNS)
  - **Exemplo 2.124** traceroute -n www.senacrs.com.br
- Para realizar testes de rota básico para determinado destino, com o protocolo ICMP ao invés do UDP padrão
  - **Exemplo 2.125** traceroute -I www.senacrs.com.br

2.12 iperf

Descrição 2.12 Ferramenta para realizar testes de vazão entre dois ou mais hosts. O **iperf** permite a utilização do protocolo TCP e UDP para realizar os testes de vazão que tem como objetivo medir o desempenho da rede.

- Para colocar em modo servidor, com o protocolo TCP e a porta padrão 5001

  ■ **Exemplo 2.126** iperf -s

- Para colocar em modo servidor, com o protocolo TCP e a porta padrão 5001, com IPv6

  ■ **Exemplo 2.127** iperf -s -V

- Para executar o cliente, com o protocolo TCP, 10 segundos de teste e a porta padrão 5001

  ■ **Exemplo 2.128** iperf -c IP_Servidor

- Para executar o cliente, com o protocolo TCP, 10 segundos de teste e a porta padrão 5001, com IPv6

  ■ **Exemplo 2.129** iperf –V -c IP_Servidor

- Para executar o cliente, com o protocolo TCP, 30 segundos de teste, relatórios a cada 1s e a porta padrão 5001

  ■ **Exemplo 2.130** iperf -c IP_Servidor -i 1 -t 30

- Para colocar em modo servidor, com o protocolo TCP e a porta 15001

  ■ **Exemplo 2.131** iperf -s -p 15001

- Para colocar em modo servidor, com o protocolo UDP e a porta padrão 5001

  ■ **Exemplo 2.132** iperf -s -u

- Para executar o cliente, com o protocolo UDP, 30 segundos de teste, relatórios a cada 1s e a porta 15001

  ■ **Exemplo 2.133** iperf -c IP_Servidor -i 1 -t 30 -u -p 15001

- Para executar o cliente, com o protocolo TCP, 30 segundos de teste, relatórios a cada 1s, a porta padrão 5001 e com 10 conexões em paralelo

  ■ **Exemplo 2.134** iperf -c IP_Servidor -i 1 -t 30 -P 10

2.13 arping

Descrição 2.13 Comando para realizar consultas por meio do protocolo ARP (*Address Resolution Protocol*), similar a ferramenta **ping**. O objetivo do uso é descobrir se um determinado endereço físico está ativo em rede.

- Para enviar requisições ARP para um host vizinho pela interface eth0. Caso o host que tenha o IP 192.168.1.1 esteja ativo na rede, haverá o retorno com a resposta e o tempo de latência.

  ■ **Exemplo 2.135** arping -I eth0 192.168.1.1

- Para procurar endereços IP duplicados. Este comando envia mensagens ARP solicitando quem teria o endereço físico do endereço IP 192.168.1.1. O IP 192.168.1.1 seria um endereço IP conhecido e se estaria buscando mais alguma interface com o mesmo IP. Em caso de IPs duplicados seriam informados os endereços físicos dos hosts.

- **Exemplo 2.136** arping -D -I eth0 192.168.1.1

2.14 wget

Descrição 2.14 É um cliente em linha de comando do protocolo HTTP e HTTPS. Permite que sejam feitos acessos a servidores HTTP e downloads de arquivos (uso mais comum).

- Para fazer o download de uma URL
  - **Exemplo 2.137** wget http://192.168.200.3/arquivo.iso
- Para fazer o download de uma URL que possua usuário e senha
  - **Exemplo 2.138** wget –http-user=aluno –http-password=senha http://192.168.200.3/arquivo.iso
- Para fazer o download de uma URL por meio de um proxy
  - **Exemplo 2.139** wget -e use_proxy=yes -e http_proxy=192.168.200.253:8080 http://192.168.200.3/arquivo.iso
- Exportar as variáveis do shell http_proxy e https_proxy para uso com web proxy.
  - **Exemplo 2.140** export http_proxy="http://192.168.200.253:8080"
    export https_proxy="http://192.168.200.253:8080"

2.15 ssh e scp

Descrição 2.15 Comandos para realizar acesso remoto e cópia de arquivos por meio do protocolo SSH. Estas ferramentas são fundamentais para o gerenciamento de servidores Linux.

- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno.
  - **Exemplo 2.141** ssh aluno@192.168.200.3
- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno e obter o modo de depuração.
  - **Exemplo 2.142** ssh -vvvv aluno@192.168.200.3
- Para acessar um servidor SSH (192.168.200.3), na porta 34000, com o usuário aluno.
  - **Exemplo 2.143** ssh -p34000 aluno@192.168.200.3
- Para acessar um servidor SSH (192.168.200.3), na porta padrão, com o usuário aluno e executar o comando "dig @8.8.8.8 www.senacrs.com.br".
  - **Exemplo 2.144** ssh aluno@192.168.200.3 "dig @8.8.8.8 www.senacrs.com.br"
- Para copiar o diretório /opt/arquivos do servidor remoto para o diretório local /var/opt, com a porta padrão e com o usuário aluno, mantendo as permissões dos arquivos e diretórios remotos.
  - **Exemplo 2.145** scp -p -r aluno@192.168.200.3:/opt/arquivos /var/opt
- Para copiar o diretório local /tmp/relatorio para o servidor remoto no diretório /home-/aluno, com a porta 34000 e com o usuário aluno, mantendo as permissões dos arquivos e diretórios locais.

■ **Exemplo 2.146** scp -P34000 -p -r /tmp/relatorio aluno@192.168.200.3:

2.16 ss

Descrição 2.16 Comando para realizar análises e amostragens das conexões de rede, tabelas de roteamento e estatísticas gerais. O comando **netstat** é a opção em distribuições mais antigas.

- Para mostrar todas as portas TCP e UDP em escuta e o processos responsáveis.
  ■ **Exemplo 2.147** ss -tupl
- Para mostrar todas as conexões TCP abertas.
  ■ **Exemplo 2.148** ss -t -a
- Para mostrar todas as conexões UDP abertas.
  ■ **Exemplo 2.149** ss -u -a
- Para mostrar todas as conexões TCP abertas, de forma contínua a cada 5s.
  ■ **Exemplo 2.150** watch -n 5 "ss -t -a"
- Para listar todas as conexões.
  ■ **Exemplo 2.151** watch -n 5 "ss -t -a"
- Para listar todas as conexões.
  ■ **Exemplo 2.152** ss -an
- Para listar estatísticas por protocolo.
  ■ **Exemplo 2.153** ss -sa
- Para obter informações sobre os temporizadores das conexões.
  ■ **Exemplo 2.154** ss -o

2.17 mtr

Descrição 2.17 Comando para realizar consultas a roteadores e traçar a rota entre dois hosts. O **mtr** é uma versão do utilitário **traceroute** com mais opções.

- Para realizar testes de forma conínua para determinado endereço.
  ■ **Exemplo 2.155** mtr 8.8.8.8
- Para realizar testes de forma conínua para determinado endereço, com IPv6.
  ■ **Exemplo 2.156** mtr -6 www.google.com
- Para realizar testes por 10 vezes e gerar um relatório.
  ■ **Exemplo 2.157** mtr -r -c 10 8.8.8.8
- Para realizar testes de forma conínua para determinado endereço, sem resolução de nomes (DNS).
  ■ **Exemplo 2.158** mtr -n 8.8.8.8

2.18 dhclient

Descrição 2.18 Comando que implementa um cliente do protocolo DHCP e permite gerenciar o empréstimo de endereços IP.

- Para renovar (renew) o IP por DHCP na interface eth0.
  - **Exemplo 2.159** dhclient eth0
- Para liberar (release) o IP por DHCP na interface eth0.
  - **Exemplo 2.160** dhclient -r eth0

2.19 telnet

Descrição 2.19 Comando que implementa um cliente do protocolo Telnet. É usado para realizar testes em portas de serviços que utilizam o protocolo TCP.

- Para fazer uma conexão a porta 80 de um endereço IP.
  - **Exemplo 2.161** telnet 19.168.200.3 80
- Obs.: para cancelar a conexão, utilizar a combinação de teclas CTRL+Ç (’Ĵ’)

2.20 netcat (nc)

Descrição 2.20 Comando que provê diversas funcionalidades relacionadas a conexões de rede e para diagnóstico de serviços.

- Para criar um servidor, com o protocolo TCP, na porta 8000.
  - **Exemplo 2.162** nc -l 8000
- Para conectar em um servidor, com o protocolo TCP, na porta 8000.
  - **Exemplo 2.163** nc 192.168.200.3 8000
- Para criar um servidor, com o protocolo UDP, na porta 8000.
  - **Exemplo 2.164** nc -u -l 8000
- Para conectar em um servidor, com o protocolo UDP, na porta 8000.
  - **Exemplo 2.165** nc -u 192.168.200.3 8000
- Para criar um servidor, com o protocolo TCP, na porta 8000, e manter o socket aberto depois da primeira conexão.
  - **Exemplo 2.166** nc -k -l 8000
- Para transmitir um arquivo do lado do cliente para o lado do servidor.
  - **Exemplo 2.167** Servidor: nc -l 8000 > /tmp/arquivo.dat
  - **Exemplo 2.168** Cliente: nc 192.168.200.3 8000 < arquivo.dat
- Para transmitir um arquivo do lado do servidor para o lado do cliente.
  - **Exemplo 2.169** Servidor: nc -l 8000 < /tmp/arquivo.dat
  - **Exemplo 2.170** Cliente: nc 192.168.200.3 8000 > arquivo.dat

- Para copiar um diretório **com** compactação, diretório corrente no cliente para o diretório corrente no servidor com o IP 10.20.12.254 na porta 4444
  - **Exemplo 2.171** Servidor: nc -l -p 4444| tar xzv
  - **Exemplo 2.172** Cliente: tar czv . | nc -q 0 10.20.12.254 4444
- Para copiar um diretório **sem** compactação, diretório corrente no cliente para o diretório corrente no servidor com o IP 10.20.12.254 na porta 4444
  - **Exemplo 2.173** Servidor: nc -l -p 4444| tar xv
  - **Exemplo 2.174** Cliente: tar cv . | nc -q 0 10.20.12.254 4444

2.21 curl

Descrição 2.21 É um cliente em linha de comando do protocolo HTTP e HTTPS. Permite que sejam feitos acessos a servidores HTTP e downloads de arquivos (uso mais comum).

- Realizar o download do arquivo.zip.
  - **Exemplo 2.175** curl https://192.168.254.95/arquivo.zip
- Realizar o download do arquivo.zip e salvar com o nome de arquivo.zip_bk.
  - **Exemplo 2.176** curl -o arquivo.zip_bk https://192.168.254.95/arquivo.zip
- Realizar o download do arquivo.zip, com o limite de 1 Mbit/s.
  - **Exemplo 2.177** curl –limit-rate 1m -O https://192.168.254.95/arquivo.zip
- Realizar o download do arquivo "arquivo.zip" utilizando protocolo FTP no servidor 192.168.254.90, por meio do dados de autenticação aluno e senha teste.
  - **Exemplo 2.178** curl -u aluno:teste ftp://192.168.254.90/arquivo.zip
- Realizar o download do arquivo "arquivo.zip" na URL http://www.meusite.com.br utilizando um proxy de endereço 192.168.254.1 na porta 3128.
  - **Exemplo 2.179** curl -x 192.168.254.1:3128 http://www.meusite.com.br/arquivo.zip
- Realizar o download do arquivo "arquivo.zip" utilizando uma conexão HTTPS e ignorando os erros de certificados SSL.
  - **Exemplo 2.180** curl –insecure https://www.meusite.com.br/arquivo.zip

2.22 rysnc

Descrição 2.22 Utilitário e protocolo para realizar a sincronização de arquivos e diretórios de forma local ou remota.

- Sincroniza arquivos locais .pdf em /dados para o host 192.168.254.29 em /backup utilizando o usuário root.
  - **Exemplo 2.181** rsync -avz –include '*.pdf' /dados/ root@192.168.254.29:/backup
- Sincroniza o diretório atual, mostrando o progresso da cópia e possibilitando o resumo da cópia em caso de falhas, limitando em 600 Kbytes/s para o host remoto 192.168.254.95 em /root/teste.

■ **Exemplo 2.182** rsync -avh –partial –progress –bwlimit=600 . root@192.168.254.95:/root/teste

- Sincroniza o diretório local, /dados/arquivos, com o diretório remoto /dados no host 192.168.254.95. O uso do rsync será por meio do protocolo SSH, com o usuário "aluno" no host remoto.

  ■ **Exemplo 2.183** rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHosts-File=/dev/null" –progress /dados/arquivos/ aluno@192.168.254.95:/dados/

- Obs.: o diretório arquivos será sincronizado como um subdiretório em /dados

  ■ **Exemplo 2.184** rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHosts-File=/dev/null" –progress /dados/arquivos/* aluno@192.168.254.95:/dados/

- Obs.: o conteúdo do diretório arquivos será sincronizado no diretório em /dados

2.23 iwconfig

Descrição 2.23 Comando para realizar configurações na interface de rede sem fios (wireless) e obter estatísticas diversas.

- Para listar todas as interfaces sem fios disponíveis.

  ■ **Exemplo 2.185** iwconfig -a

- Para ativar a interface ath0.

  ■ **Exemplo 2.186** iwconfig ath0 up

- Para desativar a interface ath0.

  ■ **Exemplo 2.187** iwconfig ath0 down

- Para trocar para o canal 11 na interface ath0.

  ■ **Exemplo 2.188** iwconfig ath0 interface channel 11

- Para ativar o modo monitor na interface ath0.

  ■ **Exemplo 2.189** iwconfig ath0 mode monitor

- Para ativar o modo managed (padrão) na interface ath0.

  ■ **Exemplo 2.190** iwconfig ath0 mode managed

2.24 iwlist

Descrição 2.24 Comando que provê funcionalidades de gerenciamento relacionadas a conexões de rede sem fios.

- Para listar os clientes associados ao access point.

  ■ **Exemplo 2.191** iwlist peers

- Para realizar varredura de canais (site survey) utilizando a interface ath0.

  ■ **Exemplo 2.192** iwlist ath0 scan

- Para listar os canais disponíveis.

  ■ **Exemplo 2.193** iwlist channel

2.25 ntpdate

Descrição 2.25 Cliente do protocolo NTP (Network Time Protocol) para ajuste de dia e horário. Para o ajuste ser correto a zona de tempo (timezone) do host deverá estar correta. No horário de Brasília a zona de tempo é a GMT-3.

- Para ajustar o horário do sistema conforme o servidor de NTP 200.132.0.132.
  - **Exemplo 2.194** ntpdate 200.132.0.132
- Para apenas solicitar informações sobre dia e horário ao servidor NTP 200.132.0.132.
  - **Exemplo 2.195** ntpdate –q 200.132.0.132

2.26 iftop

Descrição 2.26 Comando para realizar o monitoramento do tráfego em interfaces de rede. Esta ferramenta permite o uso de filtros compatíveis com a biblioteca Libpcap (tcpdump).

- Para monitorar o tráfego de rede na interface eth0.
  - **Exemplo 2.196** iftop –i eth0
- Para monitorar o tráfego de rede na interface eth0 que seja relacionado ao endereço IP 192.168.254.254.
  - **Exemplo 2.197** iftop –i eth0 –f “host 192.168.254.254”
- Para monitorar o tráfego de rede na interface eth2 que seja relacionado a rede 192.168.254.0/24 e na porta TCP 8080.
  - **Exemplo 2.198** iftop –i eth0 –f “net 192.168.254 and tcp port 8080”

2.27 arp-scan

Descrição 2.27 Comando para realizar descobertas de IPs e hosts ativos na rede forçando mensagens ARP.

- Realiza uma varredura na rede da interface eth1, 10.10.16.0/24 e filtra pelo IP 10.10.16.71
  - **Exemplo 2.199** arp-scan -I eth1 10.10.16.0/24 | grep 10.10.16.71
- Realiza uma varredura na rede da primeira interface do sistema)(*Address Resolution Protocol*
  - **Exemplo 2.200** arp-scan -l

2.28 ftp

Descrição 2.28 Cliente do protocolo FTP (File Transfer Protocol) utilizado para a transferência de arquivos. O cliente de FTP possui um interpretador de comandos que possibilita gerenciar as transferências e a organização de arquivos e pastas locais e remotas durante a conexão.

- Acesso ao servidor de FTP ftp.scene.org . Em caso de acesso a um servidor de FTP público é padrão utilizar o usuário anonymous (anônimo) e a senha o e-mail do usuário.

Para acessar o diretório público (/pub) e listar o conteúdo do mesmo (ls) e encerra a conexão (quit)

■ **Exemplo 2.201** ftp ftp.scene.org
ftp> cd /pub
ftp> ls
ftp> quit

- Para realizar o download de um arquivo binário (bin) de nome poweramiga1.zip (get) mostrando a progressão do download (hash)

  ■ **Exemplo 2.202** ftp>bin
  ftp>hash
  ftp> get poweramiga1.zip

- Para realizar o upload de todos os arquivos com a extensão .zip (mput *.zip) do diretório corrente para o diretório remoto corrente no servidor de FTP

  ■ **Exemplo 2.203** ftp>bin
  ftp> mput *.zip

2.29 whois

Descrição 2.29 Comando para realizar pesquisas sobre registros das propriedades de domínios e endereços IP. O resultado do comando mostra informações que pode ser úteis na identificação da origem de conexões.

- Realiza busca de informações sobre o domínio youtube.com

  ■ **Exemplo 2.204** whois youtube.com

- Realiza busca de informações sobre o endereço IP 200.18.78.15

  ■ **Exemplo 2.205** whois 200.18.78.15

2.30 mutt

Descrição 2.30 Cliente de e-mail em linha de comando para utilizar de forma interativa ou em scripts.

- Envia um e-mail para admin@empresa.com.br com o assunto "Backup Iniciado" e no corpo da mensagem o texto "Iniciando o backup".

  ■ **Exemplo 2.206** echo "Iniciando o backup" |mutt -s "Backup Iniciado" admin@empresa.com.br

- Envia mensagem de monitor@empresa.com.br, nome do emissor Monitor, para admin@empresa.com.br com o assunto "Torrent", no corpo da mensagem "Uso de Torrent" e o arquivo captura.zip em anexo.

  ■ **Exemplo 2.207**
  ```
  echo "Uso de Torrent"|mutt -a captura.zip \
  -e 'set from=monitor@empresa.com.br realname="Monitor"' \
  -s "Torrent" -- admin@empresa.com.br
  ```

- Acessa a caixa de entrada do usuário johnfoo localizada em /var/spool/mail/johnfoo em linha de comando.

■ **Exemplo 2.208** mutt -f /var/spool/mail/johnfoo

2.31 mail / mailx

Descrição 2.31 Cliente de e-mail em linha de comando para utilizar de forma interativa ou em scripts. Para instalar o utilitário no CentOS "yum install mailx" e no Ubuntu/Debian/LinuxMint "apt install mailutils" . Os comandos mail e mailx apontam para o mesmo utilitário e são aliases.

- Envia mensagem para o usuário johnfoo@mail.com utilizando o envio pelo servidor de SMTPS "smtps.empresa.com.br" na porta 587, com autenticação (usuário no-reply e senha aq123456@). O assunto do e-mail é "Host Infectado".

  ■ **Exemplo 2.209**
  ```
  echo -e "Host Infectado com Malware" | mailx -v \
  -r "no-reply@empresa.com.br" \
  -s "Aviso: Host Infectado " \
  -S smtp="smtps.empresa.com.br:587" \
  -S smtp-use-starttls \
  -S smtp-auth=login \
  -S smtp-auth-user="no-reply" \
  -S smtp-auth-password="aq123456@" \
  -S ssl-verify=ignore \
  johnfoo@mail.com
  ```

- Envia e-mail com o assunto "Manual" para johnfoo@mail.com com o conteúdo do arquivo "corpo-mail.txt" como texto da mensagem.

  ■ **Exemplo 2.210** mail -s "Manual" johnfoo@mail.com < corpo-mail.txt

- Envia e-mail com a mensagem redirecionada com o assunto "Aviso"" para os receptores johnfoo@mail.com e para alice@mail.com. O redirecionador com 3 "<" faz com que a mensagem seja enviada direto, sem interação com o usuário.

  ■ **Exemplo 2.211** `mail -s "Aviso!" johnfoo@mail.com,alice@mail.com <<< "Por favor, não esquecer de desligar a cafeteira."`

3. Procedimentos

Neste capítulo são apresentados procedimentos comuns para configurações de rede no sistema operacional Linux.

3.1 Renomear interface

Para trocar o nome da interface **eth0** para **externo** em tempo de execução

Procedimento 3.1.1

ip link set **eth0** down

ip link set **eth0** name **externo**

ip link set **externo** up

Para trocar o nome da interface **eth0** para **externo** de forma permanente, editar o arquivo **/etc/udev/rules.d/70-persistent-net.rules** e trocar o nome da interface que corresponde ao endereço MAC:

Procedimento 3.1.2

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:1d:86:fd", ATTR{type}=="1", KERNEL=="eth*", NAME="**eth0**"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:1d:86:fd", ATTR{type}=="1", KERNEL=="eth*", NAME="**externo**"

3.2 Uso de VLANs

Para ativar o uso de VLANs no Linux, deve ser usado o módulo **8021q**. Por exemplo:

Procedimento 3.2.1

modprobe 8021q

Para criar uma interface virtual deve-se utilizar o comando **vconfig**. Por exemplo, para criar a interface eth1.100 com a marcação 100

Procedimento 3.2.2

vconfig add eth1 100

Para evitar problemas com a filtragem de pacotes, deve-se ativar a **flag** para tornar o **dump** (uso em capturas de tráfego de rede) da interface tal como se não houvesse VLANs. Por exemplo:

Procedimento 3.2.3

vconfig set_flag eth1.100 1

Para remover uma VLAN, usa-se o comando **vconfig** com o parâmetro **rem**. Por exemplo, para remover a interface eth1.100:

Procedimento 3.2.4

vconfig rem eth1.100

Para definir um endereço IP de uma interface com VLAN, usa-se o comando **ifconfig** padrão. Por exemplo, para definir o IP 10.0.0.100 na interface eth1.100:

Procedimento 3.2.5

ifconfig eth1.100 10.0.0.100/24

3.3 Interface em Bridge

Para criar uma bridge de nome br0

Procedimento 3.3.1

ip link add br0 type bridge
brctl addbr br0

Para adicionar a interface eth0 na bridge br0

Procedimento 3.3.2

brctl addif br0 eth0
ip link set eth0 master br0

Para mostrar informações sobre a bridge br0

Procedimento 3.3.3

brctl show

Para mostrar a tabela MAC da bridge br0

Procedimento 3.3.4

brctl showmacs br0

Para ativar a bridge de nome br0

Procedimento 3.3.5

ip link set up dev br0

Para desativar a bridge de nome br0

Procedimento 3.3.6

ip link set dev br0 down

Para remover a interface eth0 de uma bridge de nome br0

Procedimento 3.3.7

ip link set eth0 nomaster

ip link set eth0 down

Para remover uma bridge de nome br0

Procedimento 3.3.8

ip link delete br0 type bridge

brctl delbr br0

Para configurar a interface bridge na inicialização

Procedimento 3.3.9

Debian

Arquivo /etc/network/interfaces

auto lo br0

iface lo inet loopback

iface eth0 inet manual

iface eth1 inet manual

Bridge br0

iface br0 inet static

bridge_ports eth0 eth1

address 192.168.200.3

broadcast 192.168.200.255

netmask 255.255.255.0

gateway 192.168.200.1

Procedimento 3.3.10

Centos

Arquivo /etc/sysconfig/network-scripts/ifcfg-br0

DEVICE=br0

TYPE=Bridge

IPADDR=192.168.200.3

NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
DELAY=0

Arquivo /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FF
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0

Arquivo /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FE
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0

Obs.: é necessária a instalação do pacote bridge-utils para o utilitário brctl

3.4 Interface em Bonding

Para definir qual o modo de operação da interface bonding de nome bond0, editar o arquivo **/etc/modprobe.d/bonding.conf**

Procedimento 3.4.1
alias bond0 bonding
options bond0 miimon=80 mode=0
Obs.: mode=1 (Active-Passive), mode=0 (Round-Robin)

Para listar sobre a interface bond0

Procedimento 3.4.2

cat /proc/net/bonding/bond0

Para criar a interface bond0, com as interfaces físicas eth0 e eth1, no modo round-robin

Procedimento 3.4.3

modprobe bonding

ifenslave bond0 eth0 eth1

ip link set bond0 up

Para trocar a interface ativa para eth1, no modo de operação Active-Passive

Procedimento 3.4.4

ifenslave -c bond0 eth1

Para remover a inerface eth0 do bonding

Procedimento 3.4.5

ifenslave -d bond0 eth0

Ativação da interface bond0 na inicialização

Procedimento 3.4.6

CentOS

Arquivo /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE="eth0"

BOOTPROTO="none"

ONBOOT="yes"

TYPE="Ethernet"

MASTER=bond0

SLAVE=yes

Arquivo /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE="eth1"

BOOTPROTO="none"

ONBOOT="yes"

TYPE="Ethernet"

MASTER=bond0

SLAVE=yes

Arquivo /etc/sysconfig/network-scripts/ifcfg-bond0

DEVICE=bond0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.200.3
PREFIX=24
NETWORK=192.168.200.0
GATEWAY=192.168.200.1

Procedimento 3.4.7

Debian

Arquivo /etc/network/interfaces

auto bond0
iface bond0 inet static
address 192.168.200.3
netmask 255.255.255.0
network 192.168.200.0
gateway 192.168.200.1
slaves eth0 eth1
bond_mode active-backup
bond_mode 0 para Round-Robin
bond_miimon 100
bond_downdelay 200
bond_updelay 200

3.5 Configurações de interface em arquivos

Configurações de interfaces de rede em sistemas padrão **Debian**.

Procedimento 3.5.1

Arquivo de configuração: /etc/network/interfaces

 # Exemplo de IPv4

auto eth0

```
iface eth0 inet static
address 192.168.200.3
netmask 255.255.255.0
gateway 192.168.200.254
broadcast 192.168.200.255
dns-nameservers 192.168.200.1 8.8.8.8
dns-search empresa.local

# Exemplo de IPv6
iface eth0 inet6 static
address fc00:0:2010:60::190
netmask 64
gateway fc00:0:2010:60::191

# Exemplo com o uso de VLANs
# Primeira opção: chamar um script para a criação das interfaces marcadas
auto eth1
iface eth1 inet manual
up ifconfig eth1 0.0.0.0 up
up /root/vlans/vlan.sh

# Segunda opção: adicionar as interfaces com a nomeação para uso de VLANs
auto eth0.2
iface eth0.2 inet static
address 192.168.2.1
netmask 255.255.255.0

auto eth0.3
iface eth0.3 inet static
address 192.168.3.1
netmask 255.255.255.0

# IP alias
auto eth0:1
```

```
iface eth0:1 inet static
address 192.168.1.7
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0
```

Configurações de interfaces de rede em sistemas padrão **Redhat/CentOS**).

Procedimento 3.5.2

Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth0

```
# Exemplo de IPv4 e IPv6
DEVICE="eth0"
BOOTPROTO="static"
BROADCAST="192.168.200.255"
DNS1="8.8.8.8"
GATEWAY="192.168.200.254"
HWADDR="00:50:56:A8:6F:66"
IPADDR="192.168.200.3"
IPV6ADDR="fc00:0:2010:60::116/64"
IPV6INIT="yes"
IPV6_AUTOCONF="no"
NETMASK="255.255.255.0"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
UUID="849a1bcf-9d10-4fca-a910-6b0e9af18aba"
IPV6_DEFAULTGW=fc00:0:2010:60::191

# VLAN (um arquivo para cada interface)
# Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth1.192
DEVICE=eth1.192
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
```

```
PREFIX=24
NETWORK=192.168.1.0
VLAN=yes

# IP alias
# Arquivo de configuração: /etc/sysconfig/network-scripts/ifcfg-eth1:0
DEVICE=eth1:0
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
PREFIX=24
NETWORK=192.168.1.0
NAME=eth0:0
```

3.6 Análise de Logs

Configurações de interfaces de rede em sistemas padrão **Debian**.

Procedimento 3.6.1

Arquivo de configuração: /etc/network/interfaces

```
# Exemplo de IPv4
auto eth0
iface eth0 inet static
address 192.168.200.3
netmask 255.255.255.0
gateway 192.168.200.254
broadcast 192.168.200.255
dns-nameservers 192.168.200.1 8.8.8.8
dns-search empresa.local

# Exemplo de IPv6
iface eth0 inet6 static
address fc00:0:2010:60::190
netmask 64
gateway fc00:0:2010:60::191
```

```
# Exemplo com o uso de VLANs
# Primeira opção: chamar um script para a criação das interfaces marcadas
auto eth1
iface eth1 inet manual
up ifconfig eth1 0.0.0.0 up
up /root/vlans/vlan.sh

# Segunda opção: adicionar as interfaces com a nomeação para uso de VLANs
auto eth0.2
iface eth0.2 inet static
address 192.168.2.1
netmask 255.255.255.0

auto eth0.3
iface eth0.3 inet static
address 192.168.3.1
netmask 255.255.255.0

# IP alias
auto eth0:1
iface eth0:1 inet static
address 192.168.1.7
netmask 255.255.255.0
broadcast 192.168.1.255
network 192.168.1.0
```

4. Utilitários Avançados

Neste capítulo são apresentados utilitários avançados de redes para administração de sistemas operacionais Linux. Estas ferramentas proveem poderosas funcionalidades que demandam uma base sólida de conhecimentos na área de redes para poderem ser aproveitas da forma correta.

4.1 tcpdump

Descrição 4.1 Ferramenta para captura de tráfego de rede padrão em sistemas Linux. O Tcpdump é composto por uma ferramenta para interação com o usuário, por uma linguagem de filtros denominada BPF (BSD packet filter) e pela biblioteca Libpcap que possibilita o desenvolvimento de outras ferramentas de captura compatíveis com o formato de arquivos pcap, por exemplo o Wireshark. Parâmetros de captura:

- "-e" - Mostra informações da camada de enlace (VLAN, endereço físico)
- "-F" - Utiliza um arquivo com os filtros para captura (os filtros em linha de comando serão ignorados)
- "-i" - Define qual interface será utilizada para captura. Por padrão, será escolhida a interface com menor número de identificação. No Linux, pode-se utilizar o parâmetro "any" para capturar em qualquer interface, em modo NÃO promíscuo. Para capturar pacotes na interface de loopback, utilizar a interface "lo".
- "-A" – mostra o conteúdo dos pacotes em formato texto (ASCII).
- "-n" - Não resolve os endereços IP para o nome (DNS). Importante para melhorar o desempenho das capturas e evitar perdas de pacotes.
- "-nn" - Não resolve endereços IP para o nome (DNS) e não converte o número da porta para o nome do serviço (Ex.: porta 80 para http)
- "-vv" e "-vvv" - Modo verbose, mostra mais informações sobre os pacotes. Um "v" a mais, aumenta o nível de detalhamento dos pacotes
- "-w" - Salva a captura em arquivo no formato da biblioteca Libpcap (poderá ser aberto no Wireshark)
- "-s" – Define o tamanho do pacote a ser capturado. Por padrão é 64KB (Exemplos: -s120 (captura os primeiros 120 Bytes de cada pacote, -s0 corresponde ao pacote inteiro)

Procedimento 4.1.1 **Opções:**

Para capturar pacotes na interface padrão, basta executar a ferramenta sem parâmetros

■ **Exemplo 4.1** tcpdump

Outros parâmetros são relacionados ao gerenciamento da captura Pacotes da interface eth1, sem resolver o nome dos hosts ou as portas de comunicação e salvar a captura no arquivo /tmp/salvo.cap)

■ **Exemplo 4.2** tcpdump –i eth1 –nn –w /tmp/salvo.cap

A filtragem de pacotes é feita por meio de parâmetros da linguagem BPF Pacotes com

origem ou destino do IP 10.0.5.10 e que sejam do protocolo UDP)

- **Exemplo 4.3** tcpdump host 10.0.5.10 and udp

Procedimento 4.1.2 Filtros Básicos:

Por endereço IP, origem ou destino

- **Exemplo 4.4** tcpdump host 10.0.89.15

Por endereço IP, como origem dos pacotes

- **Exemplo 4.5** tcpdump src host 10.0.89.15

Por endereço IP, como destino dos pacotes

- **Exemplo 4.6** tcpdump dst host 10.0.89.15

Por porta da camada de transporte, qualquer protocolo

- **Exemplo 4.7** tcpdump port 80

Por porta de origem da camada de transporte, qualquer protocolo

- **Exemplo 4.8** tcpdump src port 80

Por porta de destino da camada de transporte, qualquer protocolo

- **Exemplo 4.9** tcpdump dst port 80

Por porta do protocolo TCP, origem ou destino

- **Exemplo 4.10** tcpdump tcp port 80

Por porta do protocolo TCP, destino

- **Exemplo 4.11** tcpdump tcp dst port 80

Por porta do protocolo TCP, origem

- **Exemplo 4.12** tcpdump tcp src port 80

Por porta do protocolo UDP, origem ou destino

- **Exemplo 4.13** tcpdump udp port 53

Procedimento 4.1.3 Filtros Diversos:

Filtrar todos os pacotes com destino ou origem de determinada rede

- **Exemplo 4.14** tcpdump net 192.168.254.0/24

Filtrar todos os pacotes com destino a determinada rede

- **Exemplo 4.15** tcpdump dst net 192.168.254.0/24

Filtrar todos os pacotes com origem de determinada rede

- **Exemplo 4.16** tcpdump src net 192.168.254.0/24

Filtrar todos os pacotes que não sejam originados ou destinados a determinada rede

■ **Exemplo 4.17** tcpdump not net 192.168.254.0/24

Filtrar todos os pacotes do protocolo ICMP

■ **Exemplo 4.18** tcpdump icmp

Filtrar todos os pacotes do protocolo ARP

■ **Exemplo 4.19** tcpdump arp

Filtrar todos os pacotes do protocolo IPv6

■ **Exemplo 4.20** tcpdump ip6

Filtrar todos os pacotes multicast ou broadcast

■ **Exemplo 4.21** tcpdump multicast or broadcast

Filtrar pacotes de determinado endereço físico, origem ou destino

■ **Exemplo 4.22** tcpdump ether host 10:BF:48:89:67:1B

Filtrar pacotes de determinado endereço físico, origem

■ **Exemplo 4.23** tcpdump ether src host 10:BF:48:89:67:1B

Filtrar pacotes de determinado endereço físico, destino

■ **Exemplo 4.24** tcpdump ether dst host 10:BF:48:89:67:1B

Filtrar pacotes de determinada vlan

■ **Exemplo 4.25** tcpdump vlan 200

Filtrar os pacotes de multicast ou broadcast que não sejam originados do endereço físco 10:BF:48:89:67:1B

■ **Exemplo 4.26** tcpdump multicast or broadcast and not ether host 10:BF:48:89:67:1B

Filtrar os pacotes da VLAN com ID 200, que sejam nas portas 80 ou 443 TCP ou na porta 53 UDP

■ **Exemplo 4.27** tcpdump -nn -i eth1 vlan 200 and tcp port 80 or tcp port 443 or udp port 53

Filtrar pacotes do IP 192.178.15.17 que sejam com a porta 80 TCP

■ **Exemplo 4.28** tcpdump -nn -i eth1 tcp port 80 and host 192.178.15.17

Filtrar pacotes do protocolo ICMP que sejam originados ou destinados aos endereços da rede 192.168.254.0/24

■ **Exemplo 4.29** tcpdump -nn -i eth1 icmp and net 192.168.254.0/24

4.2 socat

Descrição 4.2 Comando que pode funcionar com um repassador de conexões genérico e um cliente de conexões TCP e UDP.

Procedimento 4.2.1 **Exemplos de uso:**

Redireciona as conexões locais na porta 80 para o IP 192.168.10.15 na porta 8080

■ **Exemplo 4.30** socat TCP-LISTEN:80,fork TCP:192.168.10.15:8080

Conecta na porta 80 do endereço IP 10.1.1.1

■ **Exemplo 4.31** socat - TCP:10.1.1.1:80

4.3 sshpass

Descrição 4.3 Comando que atua como um cliente não interativo de SSH, possibilitando o uso em scripts e, principalmente, em equipamentos onde não é possível usar autenticação por certificados, tais como switches, APs e roteadores.

Procedimento 4.3.1 Exemplos de uso:

A acessa o servidor de SSH no IP 192.168.254.15 com o usuário "aluno" e a senha "senha123"

■ **Exemplo 4.32** sshpass -p 'senha123' ssh aluno@192.168.254.15

Acessa o servidor de SSH no IP 192.168.254.15 com o usuário "aluno" e a senha "senha123" e não checa a chave do host.

■ **Exemplo 4.33** sshpass -p 'senha123' ssh -o StrictHostKeyChecking=no aluno@192.168.254.15

Acessa o servidor SSH 192.168.10.10 e executa o comando "iwconfig ath0", salvando o resultado em /tmp/saida.txt. A conexão terá o limite de 10s, antes de encerrar por timeout.

■ **Exemplo 4.34** sshpass -p "senha" ssh -o ConnectTimeout=10 -o StrictHostKeyChecking=no ubnt@192.168.10.10 "iwconfig ath0" > /tmp/saida.txt

4.4 Python Simple HTTP Server

Descrição 4.4 Módulo disponível na linguagem Python que disponibiliza um servidor HTTP simplificado no diretório atual para copiar arquivos de forma facilitada.

Procedimento 4.4.1 Exemplos de uso:

Disponibiliza um servidor HTTP na porta 8000 no diretório atual (Python 2.x).

■ **Exemplo 4.35** python -m SimpleHTTPServer 8000

Disponibiliza um servidor HTTP na porta 9500 no diretório atual (Python 3.x).

■ **Exemplo 4.36** python3 -m http.server 9500

4.5 comandos r (rcp, rexec, rlogin, and rsh)

Descrição 4.5 Realizar login, execução de comandos e cópias remotas entre hosts. Os comandos "r" são legados, não são recomendados para uso em produção. Entretanto, podem ser úteis em casos onde existam versões de Unix antigas. Utilizam a porta TCP 514

Procedimento 4.5.1 Exemplos de uso:

Copia de forma recursiva e mantendo as permissões dos arquivos do diretório local /dados para o diretório /backup no servidor1 (o servidor1 é um nome de host disponível no arquivo /etc/hosts da máquina local).

- **Exemplo 4.37** rcp -r -p /dados/ servidor1:backup/

Realiza a execução remota no servidor de IP 192.168.254.95 como usuário root do comando "uptime".

- **Exemplo 4.38** rsh -l root 192.168.254.95 uptime

4.6 net

Descrição 4.6 Comandos para realizar a administração remota de servidores Samba ou Microsoft Windows.

Procedimento 4.6.1 Exemplos de uso:

Solicita ao host 192.168.254.17 a lista de serviços disponíveis utilizando o usuário teste123 do domínio nomedomain. Será solicitada a senha.

- **Exemplo 4.39** net rpc service list -I 192.168.254.17 -U teste123 –W nomedomain

Realiza o reboot do host 192.168.254.5 utilizando o usuário administrador e a senha senhaadmin

- **Exemplo 4.40** net rpc shutdown -r -I 192.168.254.5 -U administrador%senhaadmin

Verifica o status do serviço Spooler (impressão) do host 192.168.254.17 utilizando o usuário teste123

- **Exemplo 4.41** net rpc service status Spooler -I 192.168.254.17 -U teste123

4.7 snmp*

Descrição 4.7 Comandos para realizar consultas e alterações em agentes do protocolo SNMP (Simple Network Management Protocol). Dois comandos básicos são o snmpget e o snmpwalk. Para relizar o uso dos comandos é necessário conhecer a MIB (Management Information Base) para saber qual OID (Objetc Identifier) a ser consultado.

Procedimento 4.7.1 Exemplos de uso:

Acessa o agente SNMP no IP 192.168.254.254, com a comunidade “public”, versão “2c” do protocolo e solicita informações sobre o OID “System”

- **Exemplo 4.42** snmpwalk –c public –v2c 192.168.254.254 system

Mostra informações sobre a OID sysContact.0 do agente SNMP em localhost

- **Exemplo 4.43** snmpget -c public -v2c localhost SNMPv2-MIB::sysContact.0

4.8 nfs*

Descrição 4.8 O protocolo NFS (Network File System) tem como objetivo o compartilhamento em rede de área de armazenaemnto. O protocolo pode utilizar o transporte em UDP e/ou em TCP, sendo que a versão 4 utiliza somente porta 2049 (TCP). O protocolo deverá estar ativado no kernel para funcionar. O serviço no lado cliente é composto pelo serviço do NFS e pelo arquivo **/etc/exports**. O arquivo **/etc/exports** define quais áreas de armazenamento serão compartilhadas e as configurações correspondentes. Exemplos de entradas no arquivo **/etc/exports**:

- Compartilhar o diretório /home para qualquer dentro da faixa 10.10.10.0/24, com direitos de leitura e escrita, com o método assíncrono e evita que usuários conectados no compartilhamento possam ter privlégios de root e aplica o ID do usuário nfsnobody
- **/home 10.10.10.0/24(rw,async,root_squash)**
- Compartilhar o diretório /home/ftp para qualquer endereço IP com direito de leitura e escrita
- **/home/ftp *(rw)**
- Comparilhar o diretório /dados para qualquer IP dento da faixa 172.26.0.0/16 com direitos de somente leitura.
- **/dados 172.26.0.0/16(ro)**

Após alterar o arquivo **/etc/exports** deve-se executar o comando **exportfs -a** para aplicar as alterações.

Procedimento 4.8.1 Montagem de compartilhamentos NFS

Montagem no /home local do /home compartilhado do servidor com o IP 10.0.0.1

■ **Exemplo 4.44** mount 10.0.0.1:/home /home

Montagem no /arquivos-remotos local do /dados compartilhado do servidor com o IP 10.0.0.1

■ **Exemplo 4.45** mount 10.0.0.1:/dados /arquivos-remotos

Montagem no /backup local do /dados compartilhado do servidor com o IP 10.0.0.1 usando a versão 4 do NFS

■ **Exemplo 4.46** mount -t nfs4 10.0.0.1:/dados /backup

Para desmontar o compartilhamento, o comando umount padrão é utilizado. No exemplo está sendo desmontado o /home local

■ **Exemplo 4.47** umount /home

Para mostrar os compartilhamentos disponíveis em um servidor com o endereço IP 192.168.67.12

■ **Exemplo 4.48** showmount -e 192.168.67.12

Procedimento 4.8.2 Exemplos de configurações do arquivo /etc/exports

Diretório: /dados
Endereços IP com acesso: 192.168.254.0/24
Parâmetros:
rw - Liberado acesso de leitura e escrita

sync - Modo síncrono (só após confirmada a operação é feita a escrita em disco), configuração padrão
no_root_squash – ignora qual o usuário remoto, basta ser um dos endereços liberados para ter acesso como root.

■ **Exemplo 4.49** /dados 192.168.254.0/24(rw,sync,no_root_squash)

Diretório /pub
Endereço: * (qualquer um)
Parâmetros:
ro - somente leitura
insecure – libera portas maiores que 1024 como origem no cliente (usuário remoto não precisa ser root)
all_squash – todos os acessos serão mapeados para um usuário comum, normalmente nobody.

■ **Exemplo 4.50** /pub *(ro,insecure,all_squash)

Diretório: /pub2
Endereço IP com acesso: 192.168.254.219
Parâmetros:
rw - Liberado acesso de leitura e escrita
all_squash – faz o mapeamento de usuário e grupo para o UID e GID definidos no parâmetros anonuid e anongid. Neste caso, o UID e GID 99 são do usuário nobody.

■ **Exemplo 4.51** /pub2 192.168.254.219(rw,all_squash,anonuid=99,anongid=99)

4.9 sshfs

Descrição 4.9 sshfs (SSH Filesystem) se comporta como um sistema de arquivos para uso remoto baseado no protocolo SSH (SFTP) e com suporte da biblioteca FUSE. O sshfs possibilita a montagem de diretórios remotos e o uso tal como fossem locais, de forma simples. Desta forma, simplifica o compartilhamento devido ao usar somente a porta de SSH, com autenticação e criptografia já existentes. Para instalar o serviço:

- Debian - apt install sshfs
- CentOS - yum –enablerepo=powertools install fuse-sshfs

Procedimento 4.9.1

Neste exemplo será montado o "/home/aluno" originado no servidor remoto (endereço IP_Servidor) no diretório local "/arq_remotos".

Para este procedimento funcionar, o diretório "arq_remotos" já deverá existir no cliente local que está executando o comando sshfs.

O usuário "aluno" deverá ter uma conta no servidor remoto e permissões para acessar o diretório "/home/aluno".

No cliente local deverá estar montado o "/home/aluno" remoto no diretório local "arq_remotos".

■ **Exemplo 4.52** sshfs aluno@IP_Servidor:/home/aluno /arq_remotos

4.10 nmap

Descrição 4.10 A ferramenta nmap possibilita a exploração de rede e segurança e um excelente scanner de portas. A ferramenta possui um repositório de scripts que podem ser usados em atividades de pentest de forma simples e prática.

Procedimento 4.10.1 Um dos usos mais comuns da ferramenta é realizar varreduras de endereços de hosts, Existem diversas formas de definir a faixa de endereços a serem buscados.

Faz uma varredura básica no endereço www.empresa.com.br

■ **Exemplo 4.53** nmap www.empresa.com.br

Faz uma varredura básica, em modo verbose, no endereço www.empresa.com.br

■ **Exemplo 4.54** nmap -v www.empresa.com.br

Faz uma varredura básica nos endereços 10.0.0.10, 10.0.0.50 e 172.16.7.10

■ **Exemplo 4.55** nmap 10.0.0.10, 10.0.0.50 e 172.16.7.10

Faz uma varredura básica nos endereços 192.168.0.0 até 192.168.0.255

■ **Exemplo 4.56** nmap 192.168.0.*

■ **Exemplo 4.57** nmap 192.168.0.0/24

Faz uma varredura básica nos endereços contidos no arquivo lista.txt. Os endereços devem estar um em cada linha do arquivo.

■ **Exemplo 4.58** nmap -iL lista.txt

Faz uma varredura básica nos endereços da faixa 172.26.10.50 até 172.26.10.78

■ **Exemplo 4.59** nmap 172.26.10.50-78

Faz uma varredura básica nos endereços 192.168.0.0 até 192.168.0.255, excluindo o endereço 192.168.0.50.

■ **Exemplo 4.60** nmap 192.168.0.* –exclude 192.168.0.50

Procedimento 4.10.2 A varredura de portas é outra funcionalidade comum da ferramenta nmap.

Faz uma varredura de portas TCP 80 no endereço www.empresa.com.br

■ **Exemplo 4.61** nmap -p 80 www.empresa.com.br

Faz uma varredura de portas TCP 80 e 443 no endereço www.empresa.com.br

■ **Exemplo 4.62** nmap -p T:8888,80 www.empresa.com.br

Faz uma varredura de portas UDP 53 e 161 no endereço 172.17.90.7

■ **Exemplo 4.63** nmap -sU 53,161 172.17.90.7

Faz uma varredura de faixa de portas TCP 1000 a 1500 no endereço 172.17.90.7

■ **Exemplo 4.64** nmap -p 1000-1500 172.17.90.7

Faz uma varredura para identificar as versões dos serviços em execução no endereço 172.17.90.7

■ **Exemplo 4.65** nmap -sV 1000-1500 172.17.90.7

Faz uma varredura para identificar as 1000 portas mais comuns, com o método TCP Syn (não completa a conexão) no endereço 172.17.90.7

■ **Exemplo 4.66** nmap -sT 172.17.90.7

Faz uma varredura para 50 portas mais utilizadas, no endereço 172.17.90.7

■ **Exemplo 4.67** nmap –top-ports 50 172.17.90.7

Realiza a varredura em modo stealthy (mínimo ruído gerado) no endereço 10.100.5.1

■ **Exemplo 4.68** nmap -sS 10.100.5.1

Procedimento 4.10.3 O uso de funcionalidades específicas para burlar firewalls e detectar vulnerabilidades em sistemas operacionais e aplicações são opções avançadas que o nmap provê e são de importante ajuda nas rotinas de administração de redes.

Para atualizar o repositório de scripts

■ **Exemplo 4.69** nmap --script-updatedb

Para executar varredura de vulnerabilidades nos endereços da rede 172.27.1.0/24

■ **Exemplo 4.70** nmap -v --script vuln 172.27.1.0/24

Para executar scripts de descoberta de vulnerabilidades no serviço de Samba/Domínio na porta TCP 445 nos endereços da rede 172.27.1.0/24

■ **Exemplo 4.71** nmap -p 445 --script smb-os-discovery 172.27.1.0/24

Para executar testes de força-bruta contra o serviço de SSH (porta TCP 22) no endereço 172.27.1.1. Os parâmetros usuarios.txt e senhas.txt devem ser preenchidos com um item por linha. Caso não sejam usados arquivos de usuários e senhas personalizados, o script usará uma lista padrão.

■ **Exemplo 4.72** nmap -p 22 --script ssh-brute --script-args userdb=usuarios.txt,passdb=senhas.txt --script-args ssh-brute.timeout=4s 172.27.1.1

Realiza a detecção de do sistema operacional e dos serviços em execução (parâmetro "-A") e utiliza o parâmetro "-T4" para execução mais rápida para o endereço 172.27.1.1 .

■ **Exemplo 4.73** nmap -A -T4 172.27.1.1

Para executar testes de força-bruta contra o serviço MS-SQL Server no endereço 172.27.1.1. Os arquivos usuarios.txt e senhas.txt devem ser criados com um item por linha.

■ **Exemplo 4.74** nmap -p 1433 --script ms-sql-brute --script-args userdb=usuarios.txt,passdb=senhas.txt 172.27.1.1

Realiza varredura para busca de malware no endereço 172.27.1.1

■ **Exemplo 4.75** nmap -sV --script=http-malware-host 172.27.1.1

Realiza varredura para busca de malware, com o uso da lista do Google, no endereço 172.27.1.1

■ **Exemplo 4.76** nmap -p80 --script http-google-malware 172.27.1.1

Procedimento 4.10.4 Para realizar uma varredura de hosts e portas com endereços IPv6 deve-se usar previamente a detecção dos hosts ativos. Em caso contrário, uma varredura em um prefixo /64 poderia levar semanas para terminar. Uma das formas mais simples é com o uso de shell script com encadeamento de comandos. Outra forma mais elegante é com o script v6disc, disponivel em HTTPS://GITHUB.COM/CVMILLER/V6DISC. Procedimento para obter o script v6disc:

- git clone https://github.com/cvmiller/v6disc.git

O script "targets-ipv6-multicast-echo.nse" descobre hosts com endereços IPv6 ativos na rede. A partir desta descoberta poderão ser executados os testes diretamente em cada host. No exemplo, está sendo usada a interface ens33 para poder executar o script.

■ **Exemplo 4.77** `ping -6 -c 2 ff02::1|awk '{ print $4 }'|grep ^fe80 \`
`|awk -F"%" '{ print $1 }'|sort|uniq|xargs -Iativo nmap -v -6 ativo`

■ **Exemplo 4.78** v6disc.sh -N

■ **Exemplo 4.79** nmap -6 –script=targets-ipv6-multicast-echo.nse
–script-args 'newtargets,interface=ens33' -sL

4.11 ngrep

Descrição 4.11 Ferramenta baseada na biblioteca Libpcap, a mesma do Tcpdump, que permite utilizar filtros do Tcpdump e o mecanismos de busca do grep em pacotes de rede. Esta ferramenta pode ser útil para identificar padrões de texto em pacotes e realizar ações.

Procedimento 4.11.1

Neste exemplo serão analisados 60000 pacotes ou um tempo de 30 segundos, o que chegar primeiro, capturados na interface eth1 e registrados no arquivo captura.txt os pacotes que possuam as palavras "GNUTELLA ou Bittorrent" na área de dados.

■ **Exemplo 4.80** ngrep -d eth1 -i -p -n 60000 -i "GNUTELLA|BitTorrent" > captura.txt &PID=$!
sleep 30
kill -1 $PID

5. Iptables

Neste capítulo serão apresentadas as funcionalidades do utilitário **iptables**. Este utilitário tem a função de manipular o sistema netfilter que é o firewall de pacotes do Linux. Algumas das funcionalidades mais comuns do Iptables são a construção de firewalls de rede, uso de NAT e PAT, recursos para implementação de proxies transparentes, manipulação de pacotes (tabela mangle) e edição de cabeçalhos. A base da filtragem de pacotes no Linux está em dois módulos. O módulo Netfilter possui funcionalidades de rede internas ao kernel para permitir a filtragem e manipulação de pacotes. O módulo Iptables é a ferramenta em nível de usuário para acesso ao Netfilter.

5.1 Histórico

- Primeira geração: ipfw (BSD)
- Segunda geração: ipfwadm (Linux 2.0)
- Terceira geração: ipchains (Linux 2.2)
- Quarta geração: iptables (Linux 2.4, 2.6, 3.x. 4.x)
- Quinta geração: nftables (Linux 3.13 em diante)

5.2 Componentes

- Chains
- Tabelas
- Políticas
- Ações

5.2.1 Chains

- INPUT: pacotes com destino ao host
- OUTPUT: pacotes originados do host
- FORWARD: pacotes passando pelo host
- POSTROUTING: pacotes que já passaram pelo processo de roteamento
- PREROUTING: pacotes que ainda não passaram pelo processo de roteamento

■ **Exemplo 5.1** iptables –A INPUT –i eth0 –p tcp —dport 80 –j ACCEPT

■ **Exemplo 5.2** iptables –A INPUT –i eth1 –p tcp –s 192.168.10.15/32 –dport 5001 –j REJECT

■ **Exemplo 5.3** iptables –A OUTPUT –p udp —dport 53 –j DROP

■ **Exemplo 5.4** iptables –A FORWARD –i eth0 –s 200.18.79.0/24 –d 189.78.10.45/32 —dport 3389 –j ACCEPT

■ **Exemplo 5.5** iptables -t nat -A POSTROUTING -o eth0 -j SNAT -s 192.168.200.0/24 –to-source 200.67.10.5

■ **Exemplo 5.6** iptables -t nat -A PREROUTING -i eth1 -p tcp -s 192.168.200.0/24 –dport 80 -j REDIRECT –to-port 3128

5.2.2 Tabelas

- Filter: padrão, onde é realizada a filtragem de pacotes
- NAT: onde acontecem as traduções de endereçamentos e portas dos pacotes
- Mangle: onde são modificados os cabeçalhos dos pacotes

■ **Exemplo 5.7** iptables –A INPUT –t filter –i eth0 –p tcp —dport 80 –j ACCEPT

■ **Exemplo 5.8** iptables -t nat -A POSTROUTING -o eth0 -j SNAT -s 192.168.200.0/24 –to-source 200.67.10.5

■ **Exemplo 5.9** iptables -t nat -A PREROUTING -i eth1 -p tcp -s 192.168.200.0/24 –dport 80 -j REDIRECT –to-port 3128

■ **Exemplo 5.10** iptables -A PREROUTING -t nat -p tcp -d 177.90.17.45 –dport 18001 -j DNAT –to 192.168.200.10:80

■ **Exemplo 5.11** iptables -t mangle -A PREROUTING -i eth1 -p udp –dport 53 -j DSCP –set-dscp-class CS2

■ **Exemplo 5.12** iptables -t mangle -A POSTROUTING -o eth0 -p udp –dport 53 -j DSCP –set-dscp-class CS2

5.2.3 Políticas

- ACCEPT: todos os pacotes que não são explicitamente bloqueados, serão aceitos
- DROP: todos os pacotes que não são explicitamente liberados, serão bloqueados

■ **Exemplo 5.13** iptables –P INPUT DROP

■ **Exemplo 5.14** iptables –P OUTPUT DROP

■ **Exemplo 5.15** iptables –P FORWARD ACCEPT

5.2.4 Ações

- ACCEPT: se ocorrer uma combinação com a regra, o pacote é liberado
- DROP: se ocorrer uma combinação com a regra, o pacote é bloqueado sem aviso ao remetente
- REJECT: se ocorrer uma combinação com a regra, o pacote é bloqueado e o remetente receberá um aviso
  - UDP: ICMP host ou porta inalcançável
  - TCP: segmento com a flag Reset ativada
- LOG: se ocorrer uma combinação com a regra, o pacote será registrado em arquivo Normalmente, em /var/log/messages
- REDIRECT: se ocorrer uma combinação com a regra, o pacote será redirecionando para uma porta definida na regra
- MASQUERADE: se ocorrer uma combinação com a regra, o pacote terá o endereço de origem traduzido para o endereço da interface de saída

■ **Exemplo 5.16** iptables -A INPUT -p tcp -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

■ **Exemplo 5.17** iptables -A FORWARD -i eth1.100 -p icmp -s 192.168.200.0/24 -d 0/0 -j DROP

■ **Exemplo 5.18** iptables -A FORWARD -i eth0 -p udp -s 10.10.20.0/24 -d 10.10.30.0/24 -j REJECT

■ **Exemplo 5.19** iptables -A FORWARD -p tcp -s 10.10.20.15/32 -d 0/0 -m limit –limit 20/min –limit-burst 5 -j LOG –log-level 7

■ **Exemplo 5.20** iptables -t nat -A PREROUTING -i eth1.50 -p tcp -s 192.168.200.0/24 –dport 80 -j REDIRECT –to-port 3128

■ **Exemplo 5.21** iptables -A POSTROUTING -t nat -o eth0 -s 192.168.200.0/24 -j MASQUERADE

5.3 Manipulação de Regras

Parâmetros -A, -I e -D:

- "-A" insere a regra abaixo da última regra adicionada.
- "-I" posiciona a regra no topo da lista de regras atual
- "-D" remove a regra

■ **Exemplo 5.22** iptables -A FORWARD -p tcp -s 10.10.20.15/32 -d 0/0 -j ACCEPT

■ **Exemplo 5.23** iptables –I FORWARD -p tcp -s 10.10.20.15/32 -d 0/0 -j ACCEPT

■ **Exemplo 5.24** iptables –D FORWARD -p tcp -s 10.10.20.15/32 -d 0/0 -j ACCEPT

- Parâmetro –L –n –v : para listar as regras, sem tradução de nomes e com os contadores de Bytes e pacotes em cada regra
- Parâmetro –F: para limpar as regras das tabelas filter, mangle e nat

■ **Exemplo 5.25** iptables –L –n –v

■ **Exemplo 5.26** iptables -F -t filter

■ **Exemplo 5.27** iptables -F -t nat

■ **Exemplo 5.28** iptables -F -t mangle

5.4 Valores padrão

Todo o parâmetro não atribuído, é assumido o valor qualquer um (any) ou o valor todos Se não for definido um endereço de origem, destino, interface, protocolo ou porta, será assumido qualquer endereço de origem e destino, porta, protocolo e todas as interfaces.

■ **Exemplo 5.29** iptables –A INPUT –j ACCEPT

■ **Exemplo 5.30** iptables –A FORWARD –p tcp –j DROP

■ **Exemplo 5.31** iptables -A FORWARD -p udp –s 0/0 —dport 161 –j ACCEPT

5.5 Módulos

- Multiport: para colocar múltiplas portas na regra
- Limit: para limitar o número de combinações da regra
- State: para criar combinações conforme o estado do fluxo de pacotes Ativa o modo stateful, onde o estado do fluxo de pacotes é mantido pelo firewall

■ **Exemplo 5.32** iptables -A FORWARD -p tcp -m multiport -s 10.0.0.10/32 -d 0/0 –dports 80,443,465,993,995,587,8080,8081 -j ACCEPT

■ **Exemplo 5.33** iptables -A FORWARD -p tcp -s 10.0.0.10/32 -d 0/0 -m limit –limit 20/min –limit-burst 5 -j LOG –log-level 7

■ **Exemplo 5.34** iptables -I INPUT -p udp -s 10.15.0.0/16 -d 0/0 –dport 53 -m state –state NEW,ESTABLISHED,RELATED -j ACCEPT

- Connlimit: para limitar o número de conexões de determinada regra
- Mac: para criar regras baseadas no endereço físico (mac address)
- Iprange: para usar em faixas de IPs
- String: para usar em regras que combinem palavras nos pacotes

■ **Exemplo 5.35** iptables -A INPUT -p tcp –syn –dport 22 -m connlimit –connlimit-above 3 -j REJECT

■ **Exemplo 5.36** iptables -A FORWARD -m mac –mac-source 00:0F:EA:91:04:08 -j ACCEPT

■ **Exemplo 5.37** iptables -A FORWARD -p tcp -m iprange –src-range 192.168.1.5-192.168.1.100 –dport 143 -j ACCEPT

■ **Exemplo 5.38** iptables -A FORWARD -m string –algo bm –string "facebook" –p udp –dport 53 -j DROP

5.6 Outros Exemplos

NAT 1:1: para traduzir um endereço IP interno para um endereço IP externo, quando se tem mais de um IP público

■ **Exemplo 5.39** # NAT 1:1 (10.15.160.25 -> 200.18.16.13)
iptables -A FORWARD -s 10.15.160.25/32 -j ACCEPT
iptables -A FORWARD -d 10.15.160.25/32 -j ACCEPT
iptables -A INPUT -s 10.15.160.25/32 -j ACCEPT
iptables -A POSTROUTING -t nat -s 10.15.160.25/32 -o eth0 -j SNAT –to 200.18.16.13
iptables -A PREROUTING -t nat -d 200.18.16.13 -j DNAT –to 10.15.160.25

Redirecionamento de portas para outro host: a ação REDIRECT permite apenas o redirecionamento de portas para o próprio firewall. Com o uso de ferramentas tais como redir ou socat junto ao iptables, é possível fazer o redirecionamento para uma porta em um host remoto

■ **Exemplo 5.40** # Redirecionamento de portas e hosts
Instalar o utilitário redir
redir –lport=8000 –caddr=192.168.200.6 –cport=80
iptables -t nat -I PREROUTING -i eth1 -p tcp -d 192.168.200.3 –dport 80 -j REDIRECT –to-port 8000

5.7 Exemplo de Script de Firewall Completo

Exemplo de script de firewall para o cenário da Figura 5.1. Neste cenário são apresentados dois firewalls que atendem as empresas Empresa1 e Empresa2.

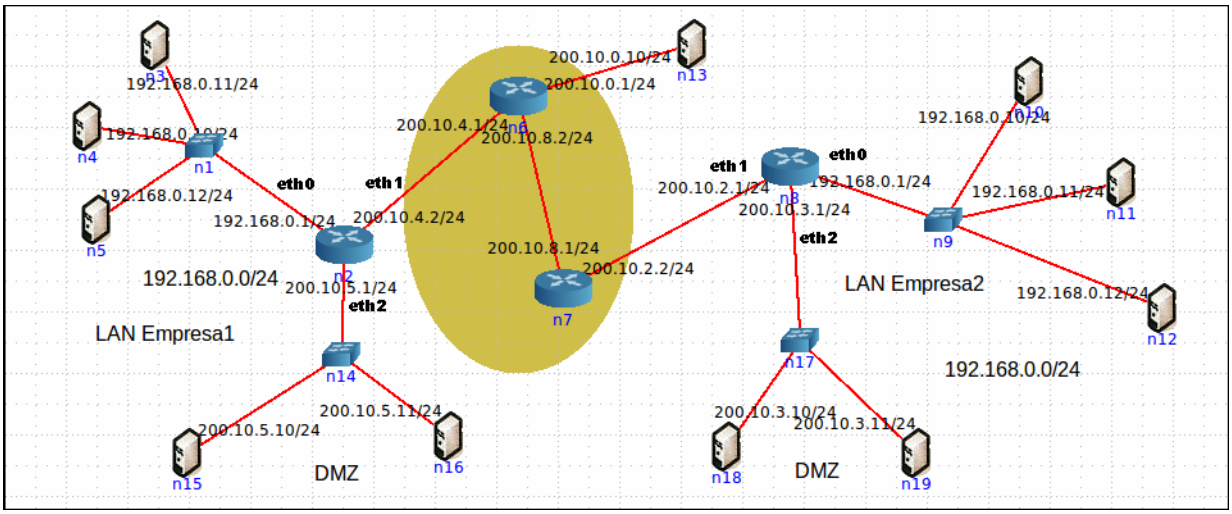

Figure 5.1: Cenário exemplo para script de firewall com Iptables.

5.7.1 Firewall da Empresa 1

Link para download: HTTPS://PASTE.EE/P/4OBUE

```
#!/bin/bash
# Firewall da empresa 1
# NAT - rede interna para a interface externa
# eth1 - interface externa (WAN)
# eth0 - interface interna (LAN)
# eth2 - interface DMZ
# Ativa o roteamento
echo 1 > /proc/sys/net/ipv4/ip_forward
#### Ativa Modulos
        /sbin/modprobe ip_conntrack
        /sbin/modprobe ip_conntrack_ftp
        /sbin/modprobe ip_nat_ftp
        /sbin/modprobe ipt_LOG
#### Limpa tabelas e configura defaults
        iptables -F -t filter
        iptables -F -t nat
        iptables -F -t mangle
        ## Delete chains nao defaults
        iptables -X
        iptables -X -t nat
        iptables -X -t mangle
        # Política DROP, chains INPUT e FORWARD
        iptables -P INPUT  DROP -t filter
        iptables -P OUTPUT ACCEPT -t filter
        iptables -P FORWARD DROP -t filter
# para manter as conexoes existentes com o proprio firewall (entrada)
        iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
        # para manter as conexoes existentes com o proprio firewall (saida)
```

```
        iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# para manter as conexoes passando pelo firewall
        iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT


# Libera acesso ao ICMP (Ping) nas interfaces LAN, DMZ e WAN
iptables -A INPUT -i eth0 -p icmp -j ACCEPT
iptables -A INPUT -i eth1 -p icmp -j ACCEPT
iptables -A INPUT -i eth2 -p icmp -j ACCEPT
# Libera SSH na porta 22
iptables -A INPUT -i eth1 -p tcp  -s 0/0 --dport 22 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp  -s 0/0 --dport 22 -j ACCEPT
iptables -A INPUT -i eth2 -p tcp  -s 0/0 --dport 22 -j ACCEPT
# Libera a resolucao de DNS no servidor local  (LAN e DMZ)
iptables -A INPUT -i eth0 -p udp -s 192.168.0.0/24 -d 0/0 --dport 53 -m state
--state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i eth2 -p udp -s 200.10.5.0/24 -d 0/0 --dport 53 -m state
--state NEW,ESTABLISHED,RELATED -j ACCEPT
# Mantem o estado das conexoes da interface de loopback
iptables -A INPUT  -s 127.0.0.1  -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT  -s 127.0.0.1  -m state --state RELATED,ESTABLISHED -j ACCEPT
# Libera todos os acesso originados de localhost para localhost
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1  -j ACCEPT
# Ativa mascaramento (rede interna para o IP da WAN) (IP Fixo)
iptables -t nat -A POSTROUTING -o eth1 -j SNAT -s 192.168.0.0/24 --to-source 200.10.4.2

### Liberacoes (rede interna (LAN) para fora)
# Servicos comuns
iptables -A FORWARD -i eth0 -p tcp -m multiport -s 192.168.0.0/24 -d 0/0 --dports
22,80,443,5001 -j ACCEPT
# Host 192.168.0.12 bloqueado ICMP para fora
iptables -A FORWARD -i eth0 -p icmp -s 192.168.0.12/32 -d 0/0 -j DROP
# ICMP
iptables -A FORWARD -i eth0 -p icmp -s 192.168.0.0/24 -d 0/0 -j ACCEPT
# DNS, NTP
iptables -A FORWARD -i eth0 -p udp -s 192.168.0.0/24 -d 0/0 -m multiport --dports 53,143
-m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
# Redireciona portas (porta publica: 5001, IP interno: 192.168.0.10, Porta interna: 5001)
iptables -A FORWARD -i eth1 -d 192.168.0.10/32 -j ACCEPT
iptables -A PREROUTING -t nat -p tcp -d 200.10.4.2 --dport 5001 -j DNAT
--to 192.168.0.10:5001


### Liberacoes (rede DMZ para fora)
# Servicos comuns
iptables -A FORWARD -i eth2 -p tcp -m multiport -s 200.10.5.0/24 -d 0/0 --dports
22,80,443,5001 -j ACCEPT
# ICMP
iptables -A FORWARD -i eth2 -p icmp -s 200.10.5.0/24 -d 0/0 -j ACCEPT
# DNS, NTP
iptables -A FORWARD -i eth0 -p udp -s 200.10.5.0/24 -d 0/0 -m multiport --dports 53,143
-m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
### Liberacoes (rede externa para a DMZ)
# Host externo 200.10.0.10, bloqueado o acesso a porta 22 no servidor 200.10.5.10
iptables -A FORWARD -i eth1 -p tcp -s 200.10.0.10/32 -d 200.10.5.10/32 --dport 22 -j DROP
# Servicos comuns, somente para o servidor 200.10.5.10
iptables -A FORWARD -i eth1 -p tcp -m multiport -s 0/0 -d 200.10.5.10/32 --dports
22,80,443,5001 -j ACCEPT
# SSH e HTTP, somente para o servidor 200.10.5.11
iptables -A FORWARD -i eth1 -p tcp -m multiport -s 0/0 -d 200.10.5.11/32
--dports 22,80 -j ACCEPT
# ICMP
iptables -A FORWARD -i eth1 -p icmp -s 200.10.5.0/24 -d 0/0 -j ACCEPT
# Host externo 200.10.0.10, acesso a porta 5001 no servidor 200.10.5.11
iptables -A FORWARD -i eth1 -p tcp -s 200.10.0.10/32 -d 200.10.5.11/32
--dport 5001 -j ACCEPT
```

5.7.2 Firewall da Empresa 2

Link para download: HTTPS://PASTE.EE/P/HYIFI

```
#!/bin/bash
# Firewall da empresa 2
# NAT - rede interna para a interface externa
# eth1 - interface externa (WAN)
# eth0 - interface interna (LAN)
# eth2 - interface DMZ
# Ativa o roteamento
echo 1 > /proc/sys/net/ipv4/ip_forward
#### Ativa Modulos
        /sbin/modprobe ip_conntrack
        /sbin/modprobe ip_conntrack_ftp
        /sbin/modprobe ip_nat_ftp
        /sbin/modprobe ipt_LOG
#### Limpa tabelas e configura defaults
        iptables -F -t filter
        iptables -F -t nat
        iptables -F -t mangle
        ## Delete chains nao defaults
        iptables -X
        iptables -X -t nat
        iptables -X -t mangle
        # Política DROP, chains INPUT e FORWARD
        iptables -P INPUT  DROP -t filter
        iptables -P OUTPUT ACCEPT -t filter
        iptables -P FORWARD DROP -t filter
# para manter as conexoes existentes com o proprio firewall (entrada)
        iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
        # para manter as conexoes existentes com o proprio firewall (saida)
```

```
        iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# para manter as conexoes passando pelo firewall
        iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
# Libera acesso ao ICMP (Ping) nas interfaces LAN, DMZ e WAN
iptables -A INPUT -i eth0 -p icmp -j ACCEPT
iptables -A INPUT -i eth1 -p icmp -j ACCEPT
iptables -A INPUT -i eth2 -p icmp -j ACCEPT
# Libera SSH na porta 22
iptables -A INPUT -i eth1 -p tcp  -s 0/0 --dport 22 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp  -s 0/0 --dport 22 -j ACCEPT
iptables -A INPUT -i eth2 -p tcp  -s 0/0 --dport 22 -j ACCEPT
# Libera a resolucao de DNS no servidor local  (LAN e DMZ)
iptables -A INPUT -i eth0 -p udp -s 192.168.0.0/24 -d 0/0 --dport 53 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i eth2 -p udp -s 200.10.3.0/24 -d 0/0 --dport 53 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
# Mantem o estado das conexoes da interface de loopback
iptables -A INPUT  -s 127.0.0.1  -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT  -s 127.0.0.1  -m state --state RELATED,ESTABLISHED -j ACCEPT
# Libera todos os acesso originados de localhost para localhost
iptables -A INPUT  -s 127.0.0.1 -d 127.0.0.1  -j ACCEPT
# Ativa mascaramento (rede interna para o IP da WAN) (IP Fixo)
iptables -t nat -A POSTROUTING -o eth1 -j SNAT -s 192.168.0.0/24 --to-source 200.10.2.1

### Liberacoes (rede interna (LAN) para fora)
# Servicos comuns
iptables -A FORWARD -i eth0 -p tcp -m multiport -s 192.168.0.0/24 -d 0/0 --dports
22,80,443,5001 -j ACCEPT
# Host 192.168.0.12 bloqueado ICMP para fora
iptables -A FORWARD -i eth0 -p icmp -s 192.168.0.12/32 -d 0/0 -j DROP
# ICMP
iptables -A FORWARD -i eth0 -p icmp -s 192.168.0.0/24 -d 0/0 -j ACCEPT
# DNS, NTP
iptables -A FORWARD -i eth0 -p udp -s 192.168.0.0/24 -d 0/0 -m multiport --dports 53,143
-m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
# Redireciona portas (porta publica: 5001, IP interno: 192.168.0.10, Porta interna: 5001)
iptables -A FORWARD -i eth1 -d 192.168.0.10/32 -j ACCEPT
iptables -A PREROUTING -t nat -p tcp -d 200.10.2.1 --dport 5001 -j DNAT
--to 192.168.0.10:5001



### Liberacoes (rede DMZ para fora)
# Servicos comuns
iptables -A FORWARD -i eth2 -p tcp -m multiport -s 200.10.3.0/24 -d 0/0 --dports
22,80,443,5001 -j ACCEPT
# ICMP
iptables -A FORWARD -i eth2 -p icmp -s 200.10.3.0/24 -d 0/0 -j ACCEPT
# DNS, NTP
iptables -A FORWARD -i eth0 -p udp -s 200.10.3.0/24 -d 0/0 -m multiport --dports 53,143
-m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
### Liberacoes (rede externa para a DMZ)
# Host externo 200.10.0.10, bloqueado o acesso a porta 22 no servidor 200.10.3.10
iptables -A FORWARD -i eth1 -p tcp -s 200.10.0.10/32 -d 200.10.3.10/32 --dport 22 -j DROP
# Servicos comuns, somente para o servidor 200.10.3.10
iptables -A FORWARD -i eth1 -p tcp -m multiport -s 0/0 -d 200.10.3.10/32 --dports
22,80,443,5001 -j ACCEPT
# SSH e HTTP, somente para o servidor 200.10.3.11
iptables -A FORWARD -i eth1 -p tcp -m multiport -s 0/0 -d 200.10.3.11/32
--dports 22,80 -j ACCEPT
# ICMP
iptables -A FORWARD -i eth1 -p icmp -s 200.10.3.0/24 -d 0/0 -j ACCEPT
# Host externo 200.10.0.10, acesso a porta 5001 no servidor 200.10.3.11
iptables -A FORWARD -i eth1 -p tcp -s 200.10.0.10/32 -d 200.10.3.11/32 --dport 5001 -j ACCEPT
```

6. Anexo

Neste capítulo serão apresentados procedimentos e ferramentas auxiliares para a administração de sistemas operacionais Linux em rede.

6.1 systemctl

O gerenciamento da inicialização e dos serviços em Linux é feito pelo systemd Padrão nas distribuições atuais. A forma clássica de gerenciamento de serviços usava os scripts init.d e com a ferramenta service. Exemplos:

- /etc/init.d/apache2 restart
- services apache2 restart

Basicamente, são scripts que gerenciam a execução dos processos que ficam em segundo plano (daemon)

As opções padrão para gerenciar os serviços são:

- **start** – para iniciar um serviço
- **stop** – para interromper um serviço
- **restart** – para reinicar um serviço
- **reload** – para recarregar as configurações de um serviço
- **status** – para verificar o estado da execução de um processo

A opção de reload não termina o processo em execução, o objetivo é fazer um recarregamento para atualizar configurações. A opçõa restart encerra o processo e inicia outro processo.

O comando systemctl é usado para gerenciar os serviços em sistemas com systemd. Exemplos de uso para o serviço do Apache:

■ **Exemplo 6.1** systemctl start apache2 – para iniciar o serviço

■ **Exemplo 6.2** systemctl stop apache2 – para interromper o serviço

■ **Exemplo 6.3** systemctl restart apache2 – para reiniciar o serviço

■ **Exemplo 6.4** systemctl reload apache2 – para recarregar as configurações do serviço

■ **Exemplo 6.5** systemctl status apache2 – para verificar o estado da execução do processo

Não haverá saída na tela ao executar a inicialização de um serviço que aconteça com sucesso. Para verificar se o serviço está executando, utilizar a opção status, buscar os processos no sistema ou verificar nos logs.

A forma de gerenciamento com o systemctl permite outras opções que facilitam a administração dos serviços. Exemplos de uso:

■ **Exemplo 6.6** systemctl enable apache2 – para habilitar o serviço do Apache2 na inicialização do sistema

■ **Exemplo 6.7** systemctl disable apache2 – para desabilitar o serviço do Apache na inicialização do sistema

■ **Exemplo 6.8** systemctl list-units –all – para listar todas os scripts e dispositivos gerenciados pelo systemd

■ **Exemplo 6.9** systemctl– para listar os scripts dos serviços

O formato dos scripts de serviços seguem um padrão, possuem a extensão ".service" e ficam armazenados em /lib/systemd/system. Outros usos que podem ser feitos com o comando systemctl.

■ **Exemplo 6.10** systemctl daemon-reload– ao adicionar ou alterar um script de serviços deverá ser feito um reload

■ **Exemplo 6.11** systemctl set-default graphical.target – para habilitar o modo gráfico na inicialização do sistema

■ **Exemplo 6.12** systemctl set-default runlevel3.target – para habilitar o modo texto multiusuário na inicialização do sistema

■ **Exemplo 6.13** systemctl reboot – para reiniciar o sistema

O nível de execução (runlevel) determina qual o modo de interação com o sistema. O nível 5 é modo gráfico.

6.2 journalctl

O systemd possui uma forma particular (journald) para verificação dos logs de mensagens das execuções dos scripts de serviços. A ferramenta journalctl é usada para manipular os logs. Exemplos de uso:

■ **Exemplo 6.14** journalctl – mostra os logs em ordem cronológica

■ **Exemplo 6.15** journalctl –b –r – mostra os logs em ordem cronológica inversa desde o último boot do sistema

■ **Exemplo 6.16** journalctl –list-boots – lista os boots realizados pelo sistema

■ **Exemplo 6.17** journalctl –since "2023-01-10 13:45:00" – mostra os logs a partir de 10/01/2023 às 13:45:00 até agora

■ **Exemplo 6.18** journalctl –x –since yesterday – mostra os logs desde ontem até agora, com explicações das mensagens (quando disponível)

■ **Exemplo 6.19** journalctl –since 07:00 –until "1 hour ago"– mostra os logs desde às 7 horas e apenas até uma hora atrás

■ **Exemplo 6.20** journalctl -u apache2.service– mostra todos os logs relacionados ao serviço do Apache

■ **Exemplo 6.21** journalctl -n 20– mostra os últimos 20 logs

■ **Exemplo 6.22** journalctl -f – mostra os logs mais recentes de forma contínua