Capítulo 7 - Servidor Firewall e Aplicações

Wesley Ferreira, Henrique Moura, Gabriel Ornelas, Mateus Silva - IFG Campus Formosa

1 Ficha Técnica

Serviço	**Descrição**
Método de Comunicação	Protocolo TCP/UDP, portas configuráveis (22, 80, 443, etc.)
Funções	Controle de tráfego de rede, filtragem de pacotes, NAT, balanceamento de carga
Pacote para Instalação	iptables (já incluído no kernel Linux), iptables-persistent
Script de controle	`systemctl status netfilter-persistent`, `netfilter-persistent save`

Tabela 1: Ficha Técnica do servidor netfilter/iptables

2 Descrição do Servidor

2.1 Netfilter: O framework de firewall

O **netfilter** é o framework de firewall do kernel Linux que permite controlar o tráfego de rede através de regras de filtragem de pacotes. É o sistema fundamental que implementa as funcionalidades de firewall no Linux, permitindo:

- **Filtragem de pacotes**: controlar quais pacotes podem entrar ou sair do sistema
- **NAT (Network Address Translation)**: tradução de endereços de rede
- **Mangle**: modificação de cabeçalhos de pacotes
- **Logging**: registro de atividades de rede para auditoria

2.2 iptables: A ferramenta de Gerenciamento

O **iptables** é uma ferramenta de linha de comando utilizada para gerenciar o netfilter. É a interface mais comum para configurar as regras do firewall, permitindo:

- criar, modificar e deletar regras de firewall
- definir políticas padrão para as cadeias
- configurar NAT e port forwarding
- implementar proteções contra ataques

O iptables trabalha com três tabelas principais:

- **filter**: tabela padrão para filtragem de pacotes
- **nat**: para tradução de endereços de rede
- **mangle**: para modificação de pacotes

E cinco cadeias (chains) principais:

- **INPUT**: pacotes destinados ao sistema local
- **OUTPUT**: pacotes originados do sistema local
- **FORWARD**: pacotes que passam pelo sistema (roteamento)
- **PREROUTING**: pacotes que chegam (NAT)
- **POSTROUTING**: pacotes que saem (NAT)

3 iptables-legacy vs. nftables: A transição no Debian

3.1 Contexto Histórico

Desde o Debian 10 ("Buster"), o framework padrão para firewall no Linux é o **nftables**, que substitui o iptables como ferramenta nativa. No Debian 12, quando você executa comandos `iptables`, está na verdade usando uma camada de compatibilidade (`iptables-nft`) que traduz a sintaxe do iptables para regras de nftables.

3.2 Verificando qual versão está em uso

Listing 1: Verificação da versão do iptables

```
# verificar qual versão do iptables está ativa
sudo update-alternatives --config iptables

# verificar se está usando nftables ou legacy
```

```
sudo iptables --version

# verificar as regras nativas do nftables
sudo nft list ruleset
```

3.3 Implicações práticas

- **Compatibilidade**: os comandos iptables continuam funcionando normalmente
- **Performance**: nftables oferece melhor performance e flexibilidade
- **Debugging**: as regras podem aparecer diferentes quando visualizadas com `nft list ruleset`
- **Futuro**: recomenda-se aprender nftables para novos projetos

3.4 Usando iptables-legacy (Opcional)

Se você precisar usar o iptables "clássico"para compatibilidade com scripts antigos:

Listing 2: Configuração do iptables-legacy

```
# instalar iptables-legacy
sudo apt install iptables-legacy

# configurar para usar legacy
sudo update-alternatives --set iptables /usr/sbin/iptables-legacy
sudo update-alternatives --set ip6tables /usr/sbin/ip6tables-
   legacy
```

4 Instalação

O iptables já vem incluído no kernel Linux, mas para uma configuração completa, é necessário instalar alguns pacotes adicionais:

Listing 3: Instalação do iptables

```
# atualizar repositórios
sudo apt update

# instalar iptables e ferramentas relacionadas
sudo apt install iptables iptables-persistent netfilter-
   persistent
```

```
6
7 # verificar se o iptables está funcionando
8 sudo iptables -L -v
```

Para verificar se o módulo do kernel está carregado:

Listing 4: Verificação de módulos

```
1 # verificar módulos carregados
2 lsmod | grep iptable
3
4 # carregar módulos se necessário
5 sudo modprobe iptable_filter
6 sudo modprobe iptable_nat
7 sudo modprobe iptable_mangle
```

5 Arquivos de Configuração e Principais Características

5.1 Arquivos de Configuração

- **/etc/iptables/rules.v4**: arquivo principal de regras IPv4
- **/etc/iptables/rules.v6**: arquivo principal de regras IPv6
- **/etc/default/iptables**: configurações padrão
- **/proc/net/ip_tables_names**: tabelas ativas

5.2 Comandos Principais

Listing 5: Comandos básicos do iptables

```
1 # listar todas as regras
2 sudo iptables -L -v -n
3
4 # listar regras com números de linha
5 sudo iptables -L -v -n --line-numbers
6
7 # limpar todas as regras
8 sudo iptables -F
9
10 # definir políticas padrão
```

```
sudo iptables -P INPUT DROP
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD DROP
```

6 Glossário de Comandos e Opções Comuns

Para construir e entender as regras do iptables, é fundamental conhecer o significado de suas principais opções (flags) e alvos (targets). Abaixo estão os mais utilizados neste guia.

6.0.1 Opções de Gerenciamento de Regras

- **-A, –append**: Adiciona uma nova regra ao final de uma cadeia (chain).

  Exemplo: `iptables -A INPUT ...`

- **-D, –delete**: Deleta uma regra de uma cadeia, seja pelo seu número ou pela sua especificação exata.

  Exemplo: `iptables -D INPUT 1`

- **-P, –policy**: Define a política padrão para uma cadeia. Esta é a ação a ser tomada caso nenhum pacote corresponda a uma regra específica.

  Exemplo: `iptables -P INPUT DROP`

- **-L, –list**: Lista todas as regras de uma cadeia.

- **-F, –flush**: Limpa (deleta) todas as regras de uma cadeia.

- **-X, –delete-chain**: Deleta uma cadeia personalizada que esteja vazia.

- **-C, –check**: Verifica se uma regra existe, sem aplicá-la.

6.0.2 Parâmetros de Correspondência (Matching)

- **-p, –protocol**: Especifica o protocolo do pacote (ex: tcp, udp, icmp).

- **–dport, –destination-port**: Especifica a porta de destino para protocolos como TCP e UDP.

- **-i, –in-interface**: Especifica a interface de entrada pela qual um pacote foi recebido (ex: eth0, lo).

- **-o, –out-interface**: Especifica a interface de saída pela qual um pacote será enviado.

- **-m, –match**: Carrega um módulo de correspondência estendido. Essencial para regras mais complexas.

  Exemplo: `-m state` para usar o módulo de estado da conexão.

- –**state**: Usado com o módulo `-m state`, verifica o estado de uma conexão (NEW, ESTABLISHED, RELATED).

- –**ctstate**: Usado com o módulo `-m conntrack`, é a forma moderna de verificar o estado da conexão.

6.0.3 Alvos (Targets)

O alvo é especificado pela opção **-j, –jump** e define o que fazer com um pacote que corresponde a uma regra.

- **ACCEPT**: Aceita o pacote, permitindo que ele continue seu fluxo normal.

- **DROP**: Descarta o pacote silenciosamente. Nenhuma resposta é enviada à origem, o que é ideal para segurança, pois não revela a presença do firewall.

- **REJECT**: Rejeita o pacote, bloqueando-o, mas enviando uma mensagem de erro à origem (ex: "port unreachable").

- **LOG**: Registra o pacote nos logs do sistema (geralmente em `/var/log/syslog`). É um alvo que não interrompe o fluxo; após o log, o pacote continua para a próxima regra na cadeia.

- **MASQUERADE**: Usado na tabela nat, "camufla"os endereços de IP de uma rede interna, fazendo com que todos pareçam vir do endereço IP do firewall. Essencial para compartilhar uma conexão com a internet.

- **DNAT**: (Destination NAT) Altera o endereço de IP de destino do pacote. Usado para redirecionar portas (port forwarding).

6.1 Exemplo de Configuração Básica

NOTA: Para executar este script, primeiro salve-o em um arquivo (por exemplo, `firewall-rules.sh`) e depois conceda permissão de execução com o comando: `chmod +x firewall-rules.sh`. Em seguida, execute-o com `sudo ./firewall-rules.sh`.

Listing 6: Configuração Básica de firewall

```
1 #!/bin/bash
2
3 # limpar todas as regras
```

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

# definir políticas padrão
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# permitir tráfego local
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# permitir conexões estabelecidas e relacionadas (essencial para o retorno de pacotes)
# ESTABLISHED: pacotes que fazem parte de uma conexão existente
# RELATED: pacotes de novas conexões que estão relacionadas a uma existente (ex: FTP)
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# permitir SSH (porta 22)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# permitir HTTP (porta 80)
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# permitir HTTPS (porta 443)
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# permitir ping (ICMP)
iptables -A INPUT -p icmp -j ACCEPT
```

6.2 Tornando as Regras Persistentes

Após executar o script `firewall-rules.sh`, as regras são aplicadas imediatamente, mas apenas na memória do kernel. Isso significa que, se o servidor for reiniciado, **essas regras serão perdidas**.

Para garantir que o seu firewall seja reativado com o mesmo conjunto de regras a cada inicialização, você deve salvá-las de forma permanente. O pacote `iptables-persistent`,

que foi instalado anteriormente, é a ferramenta correta para esta tarefa.

O processo é feito em duas etapas:

1. **Aplicar as regras** com o script, como já foi feito: `sudo ./firewall-rules.sh`.
2. **Salvar as regras ativas** no disco com o seguinte comando:

Listing 7: Salvando as regras para persistência

```
# Este comando salva as regras atuais do IPv4 e IPv6
sudo netfilter-persistent save
```

Ao executar este comando, o serviço `netfilter-persistent` irá ler as regras que estão ativas na memória e escrevê-las no arquivo `/etc/iptables/rules.v4`. A partir de agora, a cada boot do sistema, este serviço irá carregar automaticamente as regras salvas neste arquivo, garantindo que seu servidor esteja sempre protegido.

7 Outras Opções, Descritivas de Segurança, Boas Práticas e Exemplos

7.1 Configuração de NAT

Listing 8: Configuração de NAT

```
# habilitar NAT para rede interna
# nota: substitua 'eth0' pelo nome correto da sua interface de
   rede
# Use 'ip a' ou 'ifconfig' para identificar o nome da interface
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# redirecionar porta externa para interna
iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to-
   dest 192.168.1.100:80
```

NOTA sobre nomes de interfaces: Em sistemas modernos, as interfaces de rede raramente usam nomes como `eth0`. Use comandos como `ip a` ou `ifconfig` para identificar o nome correto da interface em seu sistema (ex: `enp3s0`, `ens18`).

7.2 Configuração de Logging

Listing 9: Configuração de logs

```
# Log de tentativas de conexão SSH
iptables -A INPUT -p tcp --dport 22 -j LOG --log-prefix "SSH_ATTEMPT: "

# Log de pacotes rejeitados (útil para debugging)
iptables -A INPUT -j LOG --log-prefix "DROP: "
```

7.3 Proteção Contra Ataques

Listing 10: Proteções de segurança

```
# Proteção contra SYN flood
iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 3 -j ACCEPT

# Proteção contra port scanning
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

# Proteção contra ataques de força bruta SSH (Versão corrigida)
# adicionar IPs que tentam novas conexões SSH a uma lista "recent"
iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --set --name SSH --rsource

# bloquear IPs na lista que fizerem mais de 4 tentativas em 60 segundos
iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 4 --name SSH --rsource -j DROP
```

7.4 Debugging de Regras

Uma das maiores dificuldades ao aprender iptables é descobrir por que um pacote está sendo bloqueado. Aqui estão técnicas úteis:

Listing 11: Técnicas de debugging

```
# adicionar logging antes da política DROP para diagnosticar regras
iptables -A INPUT -j LOG --log-prefix "PACOTE DESCARTADO: "

# verificar logs em tempo real
sudo tail -f /var/log/syslog | grep iptables
```

```
6
7  # Alternativa para sistemas com systemd-journald
8  # sudo journalctl -fu kernel | grep iptables
9
10 # testar regras específicas com contadores
11 iptables -A INPUT -p tcp --dport 80 -j ACCEPT
12 iptables -L INPUT -v -n  # ver contadores de pacotes (pkts) para
       confirmar que a regra foi atingida
13
14 # Usar iptables-save para ver todas as regras em formato legível
15 sudo iptables-save
```

7.5 Boas Práticas

1. **Sempre testar regras em ambiente de desenvolvimento**
2. **Manter backup das configurações atuais**
3. **Documentar todas as regras criadas**
4. **Monitorar logs regularmente**
5. **Usar políticas restritivas por padrão**
6. **Implementar rate limiting para serviços críticos**
7. **Manter o sistema atualizado**
8. **Usar `ss` em vez de `netstat` (obsoleto)**

7.6 Script de Backup e Restauração

Listing 12: Script de backup

```
1  #!/bin/bash
2
3  # Certifique-se de que o diretório de backup existe
4  # mkdir -p /root/Backups_Firewall
5
6  # backup das regras atuais em um diretório seguro (/root/
       backups_firewall)
7  iptables-save > /root/backups_firewall/iptables_rules_$(date +%Y%
       m%d_%H%M%S).bak
8
9  # restaurar regras
```

```
# iptables-restore < /root/backups_firewall/
   iptables_rules_recentes.bak
```

7.7 Monitoramento e Manutenção

Listing 13: Comandos de monitoramento

```
# verificar estatísticas
sudo iptables -L -v -n

# monitorar logs em tempo real
sudo tail -f /var/log/syslog | grep iptables

# Alternativa para sistemas com systemd-journald
# sudo journalctl -fu kernel | grep iptables

# verificar conexões ativas (recomendado: usar ss em vez de
   netstat)
sudo ss -tuln

# verificar interfaces de rede
sudo ip a
```

NOTA: O comando `netstat` é considerado obsoleto. Use `ss` (socket statistics) que é mais eficiente é moderno. Em sistemas que usam systemd-journald, uma alternativa moderna para monitoramento de logs é o comando `journalctl -fu kernel`.

8 Comandos Essenciais - Referência Rápida

Comando	Descrição
`iptables -L -v -n`	Listar todas as regras com estatísticas
`iptables -A INPUT -p tcp -dport 80 -j ACCEPT`	Adicionar regra para permitir HTTP
`iptables -D INPUT 1`	Deletar regra número 1 da cadeia INPUT
`iptables -F`	Limpar todas as regras
`iptables -P INPUT DROP`	Definir política padrão da cadeia INPUT
`iptables-save`	Salvar regras em arquivo
`iptables-restore < arquivo`	Restaurar regras de arquivo
`iptables -L -v -n -line-numbers`	Listar regras com números de linha
`iptables -C INPUT -p tcp -dport 22 -j ACCEPT`	Verificar se regra existe

Tabela 2: Comandos essenciais do iptables

9 Considerações Finais

O netfilter/iptables é um sistema poderoso e essencial para a segurança de redes Linux. O netfilter fornece o framework de firewall no kernel, enquanto o iptables oferece uma interface flexível para gerenciar as regras. Juntos, permitem implementar políticas de segurança complexas, desde configurações básicas até setups avançados de alta disponibilidade.

Principais pontos a considerar:

- **Complexidade**: A curva de aprendizado pode ser íngreme, mas o domínio do sistema é fundamental
- **Performance**: regras mal configuradas podem impactar o desempenho da rede
- **Manutenção**: configurações devem ser revisadas e atualizadas regularmente
- **Documentação**: manter documentação atualizada é crucial para troubleshooting
- **Testes**: sempre testar configurações em ambiente controlado antes da produção
- **Transição para nftables**: esteja ciente da transição em andamento para nftables

Para ambientes de produção, considere também:

- implementar failover para alta disponibilidade
- usar ferramentas de monitoramento como Nagios ou Zabbix
- implementar alertas automáticos para tentativas de intrusão
- manter procedimentos de recuperação de desastres
- treinar equipe de suporte nas configurações implementadas
- considerar a migração para nftables em novos projetos

10 Referências Bibliográficas

Referências

[1] Debian Project. *Debian 12 "Bookworm"Release Notes*. Disponível em: `https://www.debian.org/releases/bookworm/s390x/release-notes.pt-br.pdf`. Acesso em: 2024.

[2] Netfilter Project. *Netfilter Documentation*. Disponível em: https://www.netfilter.org/documentation/. Acesso em: 2024.

[3] Wikipedia. *Debian*. Disponível em: https://pt.wikipedia.org/wiki/Debian. Acesso em: 2024.

[4] OVHcloud. *Firewall iptables*. Disponível em: https://help.ovhcloud.com/csm/pt-dedicated-servers-firewall-iptables?id=kb_article_view&sysparm_article=KB0043443. Acesso em: 2024.

[5] Hostinger. *Tutorial iptables*. Disponível em: https://www.hostinger.com/br/tutoriais/tutorial-iptables. Acesso em: 2024.

[6] Red Hat. *iptables*. Disponível em: https://www.redhat.com/en/blog/iptables. Acesso em: 2024.