
Revisão do Tópico 212 – Segurança do Sistema

212.1 – Configurando um Roteador

Classes de IPs / IPs Privados

Classe	Primeiro Octeto	Range	IPs Privados
A	1-126	1.0.0.0 – 126.255.255.255	10.0.0.0 – 10.255.255.255
B	128-191	128.0.0.0 – 191.255.255.255	172.16.0.0 – 172.31.255.255
C	192-223	192.0.0.0 – 223.255.255.255	192.168.0.0 – 192.168.255.255

Configurando um Roteador

Para habilitar o roteamento entre redes dentro de um servidor Linux, os seguintes parâmetros devem ser habilitados:

- IPv4: /proc/sys/net/ipv4/ip_forward (1)
- IPv6: /proc/sys/net/ipv6/conf/all/forwarding (1)

Internamente as rotas podem ser gerenciadas de forma estática ou dinâmica.

Para rotas estáticas, é utilizado principalmente o comando “route”. Por exemplo:

- # route -n
- # route add -net 172.16.32.0/24 gw 192.168.1.100:80

Para rotas dinâmicas, utilizados em roteadores que lidam com muitas redes e conexões, são usados alguns serviços como:

- routed
- gated
- quagga
- bird

Iptables

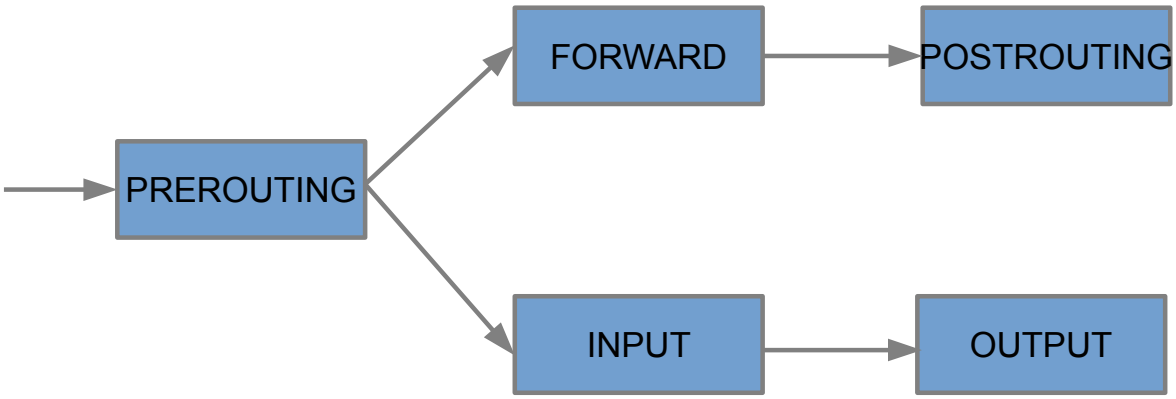
O iptables é a aplicação utilizada para gerenciar o recurso netfilter do kernel do Linux.

Basicamente, ele é administrado através de 3 **tabelas**:

- **filter**: Tabela padrão do iptables. Utilizada para criar filtros de pacotes, bloqueando e/ou liberando o tráfego.
- **nat**: Manipulação de pacotes que criam novas conexões, mudando endereços dos pacotes, origem, destino e etc.
- **mangle**: Regras para alterações nos pacotes

Essas tabelas são utilizadas para administrar regras nas seguintes **chains**:

- **INPUT**: Pacotes com destino ao host local
- **OUTPUT**: Pacotes saindo do host local
- **FORWARD**: Pacotes sendo encaminhados entre redes dentro de um host
- **PREROUTING**: Pacotes antes de se decidir o roteamento
- **POSTROUTING**: Pacotes sendo enviados a uma rede remota após o forward



Principais Opções do IPTABLES:

- -A : Adicionar regra na chain
- -I : Inserir regra em uma posição específica da chain
- -R : Substituir regra na chain
- -D : Apagar chain
- -P : Definir política padrão para uma chain
- -L : Listar as regras de uma chain
- -F : Apagar todas as regras de uma chain

Criação das Regras no IPTABLES:

- -s endereço (--source) : IP ou Rede Origem do pacote
- -d endereço (--destination) : IP ou Rede Destino do pacote
- -p protocolo (--protocol) : Define o protocolo: tcp, udp, icmp ou all
- -i interface (--in-interface) : Interface de Entrada
- -o interface (--out-interface) : Interface de Saída
- -j ação (--jump) : Ação para o pacote. Mais comuns ACCEPT, DROP, REJECT

NAT (Network Address Translation)

O iptables pode ser utilizado para criar regras que alteram a origem ou destino dos pacotes, o que é chamado de nat.

A **alteração da origem** é normalmente feita para permitir que uma rede local/privada tenha acesso à Internet.

Nesse caso, as regras podem ser criadas da seguinte maneira:

- # iptables -t nat -A POSTROUTING -s 172.16.32.0/24 -o eth1 -j **SNAT** --to-source 200.201.202.203
- # iptables -t nat -A POSTROUTING -s 172.16.32.0/24 -o eth1 -j **MASQUERADE**

A **alteração do destino** é normalmente feita para realizar o redirecionamento de portas, por exemplo:

- # iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.1.50:8080
- # iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080

Salvando e Restaurando Regras do Iptables

- iptables-save : Cria uma saída com as atuais regras de firewall, possibilitando que sejam salvos em arquivo.
  - # iptables-save > /etc/iptables.conf
- iptables-restore : A partir de um arquivo, gerado pelo iptables-save, faz um restore das regras.
  - # iptables-restore < /etc/iptables.conf

IPv6

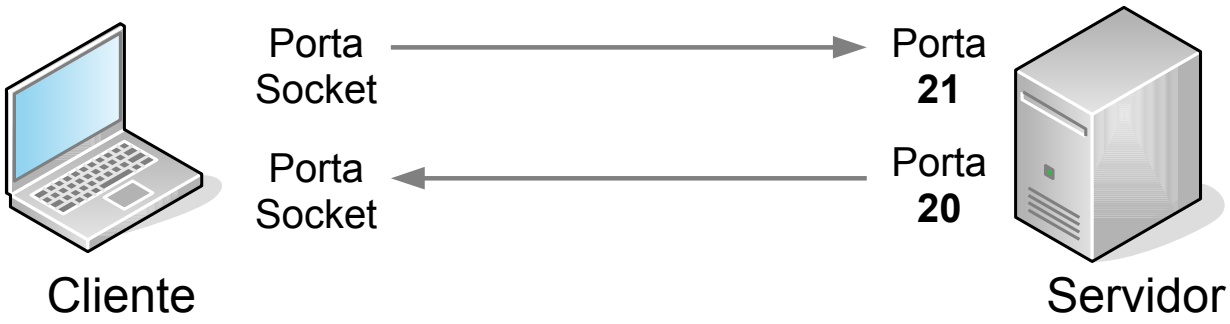
Para o gerenciamento das regras em pacotes IPv6 são utilizados os seguintes comandos:

- ip6tables
- ip6tables-save
- ip6tables-restore

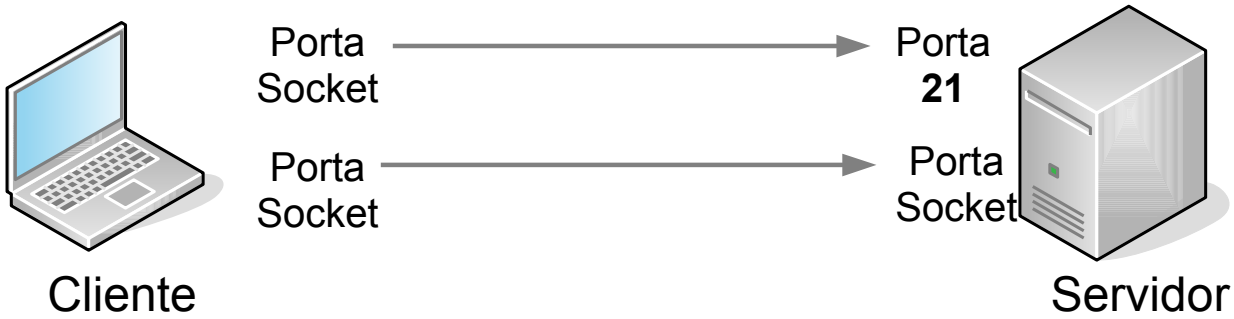
212.2 – Protegendo Servidores FTP

Modos de Conexão Ativo x Passivo

Modo Ativo



Modo Passivo



---

vsFTPd (Very Secure FTP)

Arquivo de Configuração: /etc/vsftpd.conf ou /etc/vsftpd/vsftpd.conf

Principais parâmetros de configuração:

- local_enable : Habilita/Desabilita o login de usuários locais do sistema
- anonymous_enable : Habilita/Desabilita o login de modo anônimo
- write_enable : Habilita/Desabilita a alteração de dados via FTP
- anon_upload_enable: Habilita/Desabilita o upload de arquivos por usuários anônimos
- chroot_local_users: Determina se após o login o usuário ficará ou não preso ao seu diretório

O usuário que executa o processo vsftpd é por padrão o usuário **ftp**.

Para impedir que os usuários visualizem os arquivos presentes no diretório, basta remover a permissão de leitra.

Pure-FTPd

Configurações em:

- /etc/pre-ftpd/pure-ftpd.conf (RH/CentOS)
- /etc/default/pure-ftpd-common (Debian)
- /etc/pure-ftpd/conf (Debian)

O processo também pode ser executado via linha de comando.

As principais opções são:

- ChrootEveryone / -A / --chrooteveryone : Determina se após o login o usuário ficará ou não preso ao seu diretório
- AnonymousOnly / -e / --anonymousonly : Determina se o servidor apenas aceitará login de usuários anônimos
- NoAnonymous / -E / --noanonymous : Determina se será bloqueado o login de usuários anônimos
- AnonymousCantUpload / -i / --anonymouscantupload : Determina se será bloqueado o upload de arquivos por usuários anônimos

---

ProFTPd (Noções)

Arquivos de Configuração em /etc/proftdp/proftpd.conf ou /etc/proftpd.conf

212.3 – Secure Shell (SSH)

Configurações do Servidor SSH

As configurações do servidor SSH são implementadas pelo arquivo /etc/ssh/sshd_config, as principais são:

- Port : Porta utilizada pelo servidor
- PermitRootLogin : Define se o login pode ser feito diretamente pelo usuário root
- PubkeyAuthentication : Define se o uso de chaves públicas será uma opção de login
- PasswordAuthentication : Define se o uso de senhas será uma opção de login
- ChallengeResponseAuthentication : Define se será possível o uso do método de múltiplas questões para o login
- UsePAM : Define se será possível a autenticação através do uso do PAM
- PermitEmptyPassword: Define se serão aceitas senhas em branco
- Protocol : Define o protocolo utilizado. Para segurança apenas utilizar o 2.
- AllowUsers : Se utilizado, define a lista de usuários que podem fazer login
- AllowGroupd : Se utilizado, define a lista de grupos que podem fazer login
- DenyUsers : Se utilizado, define a lista de usuários que não podem fazer login
- DenyGroups : Se utilizado, define a lista de grupos que não podem fazer login
- X11Forwarding : Define se será aceito o acesso a recursos do X11 via SSH
- AllowTcpForwarding: Define se será aceito o uso de túneis SSH

Autenticação via Chaves

As chaves públicas dos **servidores conhecidos**, em que já foi feita alguma conexão, são armazenadas localmente no arquivo **~usuario/.ssh/known_hosts ou /etc/known_hosts**

Para habilitar o acesso via chave a um servidor deve-se seguir os seguintes passos:

- Criação das chaves público/privada do usuário local:
  - # **ssh-keygen** -t <tipo> -b <tamaho>
    - Serão criados 2 arquivos
      - ~usuario/.ssh/**id_rsa – Chave privada**
      - ~usuario/.ssh/**id_rsa.pub – Chave pública**
- Enviar a chave pública ao destino:
  - # **ssh-copy-id** usuario@IP
  - A chave ficará armazenada no arquivo **~usuario/.ssh/authorized_keys**

Também pode ser utilizado o login via chaves através do ssh-agent, através dos seguintes passos:

- Criação das chaves público/privada do usuário local:
  - # **ssh-keygen** -t <tipo> -b <tamaho>
  - Definir uma senha para a chave
- Iniciar o ssh-agent
  - # ssh-agent bash
- Incluir a chave no ssh-agent
  - # ssh-add

Túneis Criptográficos com SSH

Com o SSH é possível criar um túnel criptografado entre 2 hosts de forma que outro serviço não seguro, o utilize para o tráfego de dados. Abaixo a sintaxe:

ssh -N -f -L Porta-Local:IP-Remoto:Porta-Remota usuario@IP-Remoto

ssh -N -f -L 4321:192.168.1.210:1234 ricardo@192.168.1.210

SSH e X11

Acessar via SSH um recurso gráfico via X11:

ssh -X usuario@IP “aplicação”

ssh -X ricardo@192.168.1.210 “mousepad”

TCP Wrapper

Os recursos do TCP Wrapper podem ser utilizados para proteger ainda mais o acesso ao serviço SSH.

Por exemplo:
/etc/hosts.allow
sshd: 192.168.1.* 10.0.0.10 localhost

/etc/hosts.deny
sshd: ALL

212.4 – Tarefas de Segurança

Scan e Testes de Portas

Os seguintes comandos podem ser utilizados para se realizar a varredura e o teste de portas de um host local ou remoto:

- telnet
  - # telnet 192.168.1.1 25
- netcat / nc
  - # nc 192.168.1.1 25
  - # nc -l -p 1234
- netstat
  - # netstat --tcp --listening
  - # netstat -tunl
- nmap
  - # nmap -sT localhost : Conexões TCP abertas
  - # nmap -SU localhost : Conexões UDP abertas
  - # nmap -O : Testes para identificação do Sistema Operacional

* Revisar o sub-tópico 205.2

Fail2Ban (IPS – Intrusion Prevention System)

O fail2ban monitora arquivos de logs e pode bloquear através do Iptables o acesso de IPs que sejam tentando um acesso indevido a algum serviço.

Principais configurações em **/etc/fail2ban/jail.conf**.

Logs em: /var/log/fail2ban.log

Principais configurações:

- bantime – Tempo de banimento do IP bloqueado
- ignoreip – Lista de IPs que não serão banidos
- maxretry – Quantidade de tentativas de login aceitas antes do banimento
- enabled – Habilita o controle de cada serviço (ssh/telnet/ftp/apache/etc)
- port – Determina o serviço que será monitorado
- logpath – Determina o log que será monitorado para determinado serviço

---

Snort (NIDS – Network Intrusion Detection System)

Analisa padrões de tráfego que indicam uma possível tentativa de invasão, gerando alertas.

Configurações em /etc/snort/snort.conf
Regras/Assinaturas em /etc/snort/rules

OpenVAS (Open Vulnerability Assessment System)

- Voltado para Pentests (Testes de Invasão)
- Trabalha no Modelo Cliente Servidor
- Servidor armazena um conjunto de testes de vulnerabilidade e os executa regularmente nos clientes
- NVT = Network Vulnerability Tests

Boletins de Segurança

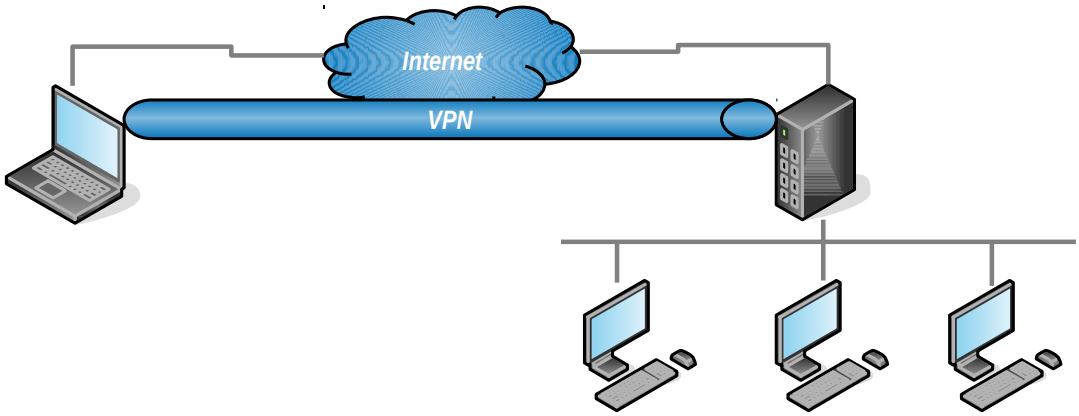
As principais fontes de informações de incidentes de segurança são:

- **Bugtraq**: www.securityfocus.com / https://www.securityfocus.com/archive/1
- **CERT**: https://www.us-cert.gov/, www.cert.br

212.5 – OpenVPN

VPN – Virtual Private Network

- Túnel Criptografado entre dois ou mais elementos de rede, normalmente conectados por uma rede pública.
- Em ambientes GNU/Linux, a aplicação mais popular é o OpenVPN



OpenVPN

- Utiliza port UDP 1194
- Tipos de Encriptação:
  - Static Key – Cliente e Servidor usam a mesma chave
  - Public Key – São gerados certificados e chaves que são compartilhadas
- Tanto na origem quanto no destino haverá uma interface virtual para a criptografia dos dados, referenciadas como tun ou tap.
  - tap – Nível ethernet (layer 2). Usado para Bridges.
  - tun – Nível de rede (layer 3). Roteamento IP.

A chave estática pode ser gerada pelo próprio openvpn pelo coamndo:
openvpn --genkey --secret static.key

Configurações presentes em /etc/openvpn. Principais parâmetros:

- dev : Define a interface como tap ou tun
- ifconfig : Define as configurações de IP da rede criptografada
- secret : Define a localização do arquivo que contém a chave estática utilizada na conexão
- push (apenas servidor) : Envia comandos DHCP após uma conexão de cliente
- status : Define um arquivo em que serão registradas estatísticas das conexões
- remote (apenas cliente) : IP do Servidor VPN em que o cliente se conectará
- nobind (apenas cliente) : Alocar uma porta dinâmica ao invés da 1194 na conexão do cliente

O processo pode ser iniciado das seguintes maneiras:

- # openvpn server.conf &
- # openvpn --config client.conf –daemon