MÓDULO 3: VPN (VIRTUAL PRIVATE NETWORK)

ÍNDICE

---

1. CONCEITOS DE VPN

O que é VPN?

VPN (Virtual Private Network) cria uma conexão segura e criptografada sobre uma rede insegura (Internet), permitindo acesso remoto seguro.

Tipos de VPN

1. Remote Access VPN (Acesso Remoto)

• Usuários remotos → Rede corporativa

• Exemplo: Home office acessando servidor da empresa

2. Site-to-Site VPN (Site a Site)

• Matriz ↔ Filial

• Conecta redes inteiras

3. Client-to-Site VPN

• Cliente individual → Servidor VPN

Protocolos VPN

Benefícios

Segurança: Criptografia end-to-end

Privacidade: Oculta tráfego e IP

Acesso Remoto: Trabalho de qualquer lugar

Bypass Geo-restrição: Acesso a conteúdo regional

---

2. OPENVPN - INSTALAÇÃO

Servidor Ubuntu/Debian

```
# Atualizar sistema
sudo apt update && sudo apt upgrade -y

# Instalar OpenVPN e Easy-RSA
sudo apt install openvpn easy-rsa -y

# Criar diretório PKI
make-cadir ~/openvpn-ca
cd ~/openvpn-ca
```

Configurar PKI (Infraestrutura de Chaves)

```
# Editar variáveis
nano vars
```

```
# Configurações personalizadas
set_var EASYRSA_REQ_COUNTRY    "BR"
set_var EASYRSA_REQ_PROVINCE   "SP"
set_var EASYRSA_REQ_CITY       "Sao Paulo"
set_var EASYRSA_REQ_ORG        "MinhaEmpresa"
set_var EASYRSA_REQ_EMAIL      "admin@empresa.com.br"
set_var EASYRSA_REQ_OU         "TI"
```

```
# Inicializar PKI
./easyrsa init-pki

# Criar CA (Certificate Authority)
./easyrsa build-ca nopass

# Gerar chave do servidor
./easyrsa gen-req server nopass

# Assinar certificado do servidor
./easyrsa sign-req server server

# Gerar parâmetros Diffie-Hellman
./easyrsa gen-dh
```

```
# Gerar chave TLS
openvpn --genkey secret ta.key

# Copiar arquivos para OpenVPN
sudo cp pki/ca.crt /etc/openvpn/server/
sudo cp pki/issued/server.crt /etc/openvpn/server/
sudo cp pki/private/server.key /etc/openvpn/server/
sudo cp pki/dh.pem /etc/openvpn/server/
sudo cp ta.key /etc/openvpn/server/
```

---

3. CONFIGURAÇÃO DO SERVIDOR OPENVPN

Arquivo de Configuração

```
# Criar configuração
sudo nano /etc/openvpn/server/server.conf



# PORTA E PROTOCOLO
port 1194
proto udp

# DISPOSITIVO DE REDE
dev tun

# CERTIFICADOS E CHAVES
ca ca.crt
cert server.crt
key server.key
dh dh.pem
tls-auth ta.key 0

# REDE VPN
server 10.8.0.0 255.255.255.0

# CONFIGURAÇÕES DE REDE
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"

# ROTAS (opcional - acesso à rede local)
push "route 192.168.1.0 255.255.255.0"

# SEGURANÇA
cipher AES-256-GCM
auth SHA256
tls-version-min 1.2

# PERSISTÊNCIA
```

```
keepalive 10 120
persist-key
persist-tun

# COMPRESSÃO
compress lz4-v2
push "compress lz4-v2"

# USUÁRIO E GRUPO
user nobody
group nogroup

# LOGS
status /var/log/openvpn/openvpn-status.log
log-append /var/log/openvpn/openvpn.log
verb 3

# MÚLTIPLOS CLIENTES
client-to-client
duplicate-cn
```

IP Forwarding

```
# Habilitar IP forwarding
sudo nano /etc/sysctl.conf



# Descomentar ou adicionar
net.ipv4.ip_forward=1



# Aplicar
sudo sysctl -p
```

Firewall (iptables)

```
# Obter interface de rede
ip route | grep default

# Configurar NAT (substitua eth0 pela sua interface)
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE

# Permitir tráfego VPN
sudo iptables -A INPUT -i tun0 -j ACCEPT
sudo iptables -A FORWARD -i tun0 -j ACCEPT
sudo iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT

# Salvar regras
sudo apt install iptables-persistent -y
sudo netfilter-persistent save
```

UFW (Firewall simplificado)

```
# Permitir OpenVPN
sudo ufw allow 1194/udp

# Configurar NAT no UFW
sudo nano /etc/ufw/before.rules



# Adicionar no início do arquivo
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
COMMIT



# Habilitar forwarding no UFW
sudo nano /etc/default/ufw



DEFAULT_FORWARD_POLICY="ACCEPT"



# Recarregar UFW
sudo ufw disable
sudo ufw enable
```

---

4. INICIAR SERVIDOR OPENVPN

```
# Criar diretório de logs
sudo mkdir -p /var/log/openvpn

# Iniciar OpenVPN
sudo systemctl start openvpn-server@server

# Habilitar no boot
sudo systemctl enable openvpn-server@server

# Verificar status
sudo systemctl status openvpn-server@server

# Ver logs
sudo journalctl -u openvpn-server@server -f
```

5. CRIAR CLIENTE OPENVPN

Gerar Certificado de Cliente

```
cd ~/openvpn-ca

# Gerar requisição do cliente
./easyrsa gen-req client1 nopass

# Assinar certificado
./easyrsa sign-req client client1

# Copiar arquivos
mkdir -p ~/client-configs/keys
cp pki/ca.crt ~/client-configs/keys/
cp pki/issued/client1.crt ~/client-configs/keys/
cp pki/private/client1.key ~/client-configs/keys/
cp ta.key ~/client-configs/keys/
```

Configuração do Cliente

```
# Criar configuração base
nano ~/client-configs/client.ovpn



client
dev tun
proto udp

# IP DO SERVIDOR (SUBSTITUIR)
remote SEU.IP.SERVIDOR.AQUI 1194

resolv-retry infinite
nobind
persist-key
persist-tun

remote-cert-tls server
cipher AES-256-GCM
auth SHA256
key-direction 1

compress lz4-v2
verb 3

# CERTIFICADOS INLINE
<ca>
# CONTEÚDO DO ca.crt
```

```
</ca>

<cert>
# CONTEÚDO DO client1.crt
</cert>

<key>
# CONTEÚDO DO client1.key
</key>

<tls-auth>
# CONTEÚDO DO ta.key
</tls-auth>
```

Script para Gerar Config Cliente

```
nano ~/client-configs/make_config.sh
```

```
#!/bin/bash

# Argumentos
CLIENT=$1
SERVER_IP=$2

# Verificar argumentos
if [ -z "$CLIENT" ] || [ -z "$SERVER_IP" ]; then
    echo "Uso: $0 <nome-cliente> <ip-servidor>"
    exit 1
fi

# Diretórios
KEY_DIR=~/client-configs/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/client.ovpn

# Criar diretório de saída
mkdir -p $OUTPUT_DIR

# Criar configuração
cat $BASE_CONFIG \
    <(echo -e '<ca>') \
    $KEY_DIR/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    $KEY_DIR/${CLIENT}.crt \
    <(echo -e '</cert>\n<key>') \
    $KEY_DIR/${CLIENT}.key \
    <(echo -e '</key>\n<tls-auth>') \
    $KEY_DIR/ta.key \
    <(echo -e '</tls-auth>') \
    | sed "s/remote .*/remote $SERVER_IP 1194/" \
    > $OUTPUT_DIR/${CLIENT}.ovpn

echo "Arquivo criado: $OUTPUT_DIR/${CLIENT}.ovpn"
```

```
# Dar permissão
chmod +x ~/client-configs/make_config.sh

# Executar
./make_config.sh client1 200.100.50.25
```

---

6. WIREGUARD - VPN MODERNA

Vantagens do WireGuard

• **Ultra-rápido:** Código enxuto (4.000 linhas vs 100.000 do OpenVPN)

• **Criptografia moderna:** ChaCha20, Poly1305

• **Simples:** Configuração mínima

• **Integrado:** Kernel Linux 5.6+

Instalação do WireGuard

```
# Ubuntu 20.04+
sudo apt update
sudo apt install wireguard -y

# Verificar módulo do kernel
sudo modprobe wireguard
lsmod | grep wireguard
```

Configuração do Servidor

```
# Gerar chaves
cd /etc/wireguard
umask 077
wg genkey | tee server_private.key | wg pubkey > server_public.key
wg genkey | tee client_private.key | wg pubkey > client_public.key

# Criar configuração
sudo nano /etc/wireguard/wg0.conf



[Interface]
Address = 10.200.0.1/24
ListenPort = 51820
PrivateKey = <CONTEUDO_server_private.key>

# IP Forwarding
PostUp = sysctl -w net.ipv4.ip_forward=1
```

```
PostUp = iptables -A FORWARD -i %i -j ACCEPT
PostUp = iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT
PostDown = iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
# Cliente 1
PublicKey = <CONTEUDO_client_public.key>
AllowedIPs = 10.200.0.2/32
```

Configuração do Cliente

```
[Interface]
Address = 10.200.0.2/24
PrivateKey = <CONTEUDO_client_private.key>
DNS = 8.8.8.8

[Peer]
PublicKey = <CONTEUDO_server_public.key>
Endpoint = SEU.IP.SERVIDOR:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25
```

Gerenciar WireGuard

```
# Iniciar VPN
sudo wg-quick up wg0

# Parar VPN
sudo wg-quick down wg0

# Status
sudo wg show

# Habilitar no boot
sudo systemctl enable wg-quick@wg0

# Adicionar cliente sem reiniciar
sudo wg set wg0 peer <PUBLIC_KEY> allowed-ips 10.200.0.3/32
```

---

7. COMANDOS DE ADMINISTRAÇÃO

OpenVPN

```
# Status do servidor
sudo systemctl status openvpn-server@server

# Ver clientes conectados
sudo cat /var/log/openvpn/openvpn-status.log

# Logs em tempo real
sudo tail -f /var/log/openvpn/openvpn.log

# Reiniciar servidor
sudo systemctl restart openvpn-server@server

# Revogar certificado de cliente
cd ~/openvpn-ca
./easyrsa revoke client1
./easyrsa gen-crl
sudo cp pki/crl.pem /etc/openvpn/server/

# Adicionar CRL no server.conf
# crl-verify crl.pem
```

WireGuard

```
# Ver configuração ativa
sudo wg show wg0

# Ver peers conectados
sudo wg show wg0 peers

# Ver estatísticas
sudo wg show wg0 transfer

# Recarregar configuração
sudo wg-quick down wg0 && sudo wg-quick up wg0

# Testar conectividade
ping -c 4 10.200.0.1
```

---

8. TROUBLESHOOTING

OpenVPN

```
# Problema: Servidor não inicia
sudo journalctl -xe -u openvpn-server@server
sudo openvpn --config /etc/openvpn/server/server.conf

# Problema: Cliente não conecta
```

```
# Verificar firewall
sudo ufw status
sudo iptables -L -n -v

# Testar porta
nc -zv SEU.IP 1194

# Problema: Sem internet após conectar
# Verificar IP forwarding
cat /proc/sys/net/ipv4/ip_forward  # Deve ser 1

# Verificar NAT
sudo iptables -t nat -L -n -v
```

WireGuard

```
# Problema: Interface não sobe
sudo dmesg | grep wireguard
sudo wg-quick up wg0

# Problema: Peer não conecta
# Verificar chaves públicas
sudo wg show wg0

# Verificar firewall
sudo ufw allow 51820/udp

# Problema: Sem roteamento
sudo ip route show
```

---

9. MONITORAMENTO

OpenVPN

```
# Script de monitoramento
cat > /usr/local/bin/openvpn-monitor.sh << 'EOF'
#!/bin/bash
echo "=== OpenVPN Status ==="
systemctl status openvpn-server@server | grep Active

echo "=== Connected Clients ==="
cat /var/log/openvpn/openvpn-status.log | grep "^CLIENT_LIST" | awk '{print $2, $3}'

echo "=== Traffic ==="
cat /var/log/openvpn/openvpn-status.log | grep "^CLIENT_LIST" | awk '{print $2, $5, $6}'
EOF
```

```
chmod +x /usr/local/bin/openvpn-monitor.sh
```

WireGuard

```
# Status completo
watch -n 2 'sudo wg show all'

# Script personalizado
cat > /usr/local/bin/wg-monitor.sh << 'EOF'
#!/bin/bash
echo "=== WireGuard Status ==="
sudo wg show wg0 | grep -E "interface|peer|transfer"
EOF

chmod +x /usr/local/bin/wg-monitor.sh
```

---

10. SEGURANÇA

Boas Práticas

```
# OpenVPN - Hardening
tls-version-min 1.3
cipher AES-256-GCM
auth SHA512
tls-cipher TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384

# Limitar tentativas
max-clients 10
client-cert-not-required off
```

Firewall Adicional

```
# Limitar conexões por IP
sudo iptables -A INPUT -p udp --dport 1194 -m state --state NEW -m recent --set
sudo iptables -A INPUT -p udp --dport 1194 -m state --state NEW -m recent --update --secon
ds 60 --hitcount 4 -j DROP
```

---

11. EXERCÍCIOS

OpenVPN Básico: Configure servidor e conecte 1 cliente

Multi-cliente: Crie 3 clientes diferentes

WireGuard: Configure WireGuard e compare performance

Site-to-Site: Configure VPN entre duas redes

Monitoramento: Implemente script de monitoramento

RESUMO

Protocolos:

• OpenVPN: Robusto, multi-plataforma

• WireGuard: Rápido, moderno, simples

Comandos OpenVPN:

• systemctl start openvpn-server@server

• easyrsa build-ca/sign-req

Comandos WireGuard:

• wg-quick up/down wg0

• wg show

Arquivos:

• /etc/openvpn/server/server.conf

• /etc/wireguard/wg0.conf

Próximo: Módulo 4 - Servidores Web (Apache/Nginx)