STI
MEI
2024/2025

Practical class #3

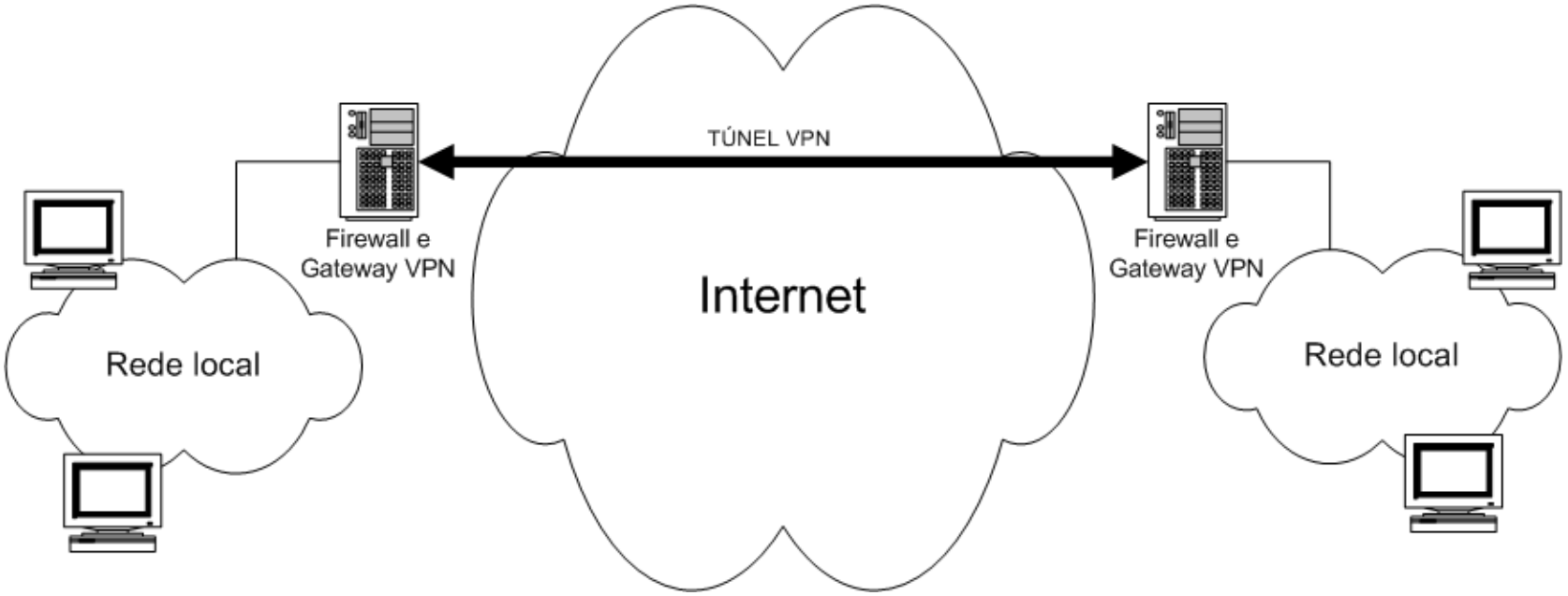
- VPN using OpenVPN

OpenVPN

- VPN using SSL/TLS
- UDP or TCP
- TAP (Network Tap) or TUN (Network Tunnel) virtual network devices
- Supports common VPN usage scenarios
- Various client authentication methods (e.g. passwords, private keys, smart cards, certificates)

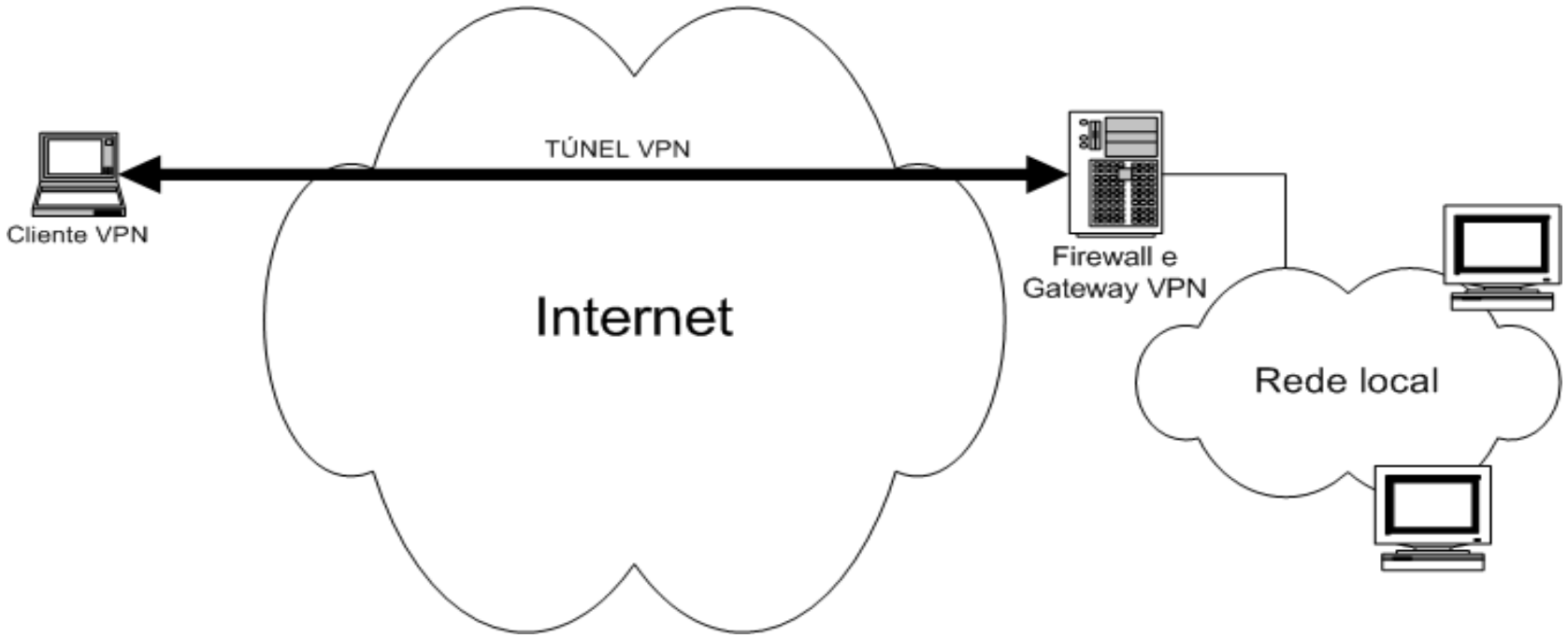
OpenVPN (Usage Scenarios)

- Gateway-to-gateway (VPN) tunnel



OpenVPN (Usage Scenarios)

- Road warrior (user remote access)



OpenVPN (Installation & basics)

Install OpenVPN

yum install epel-release

yum install openvpn

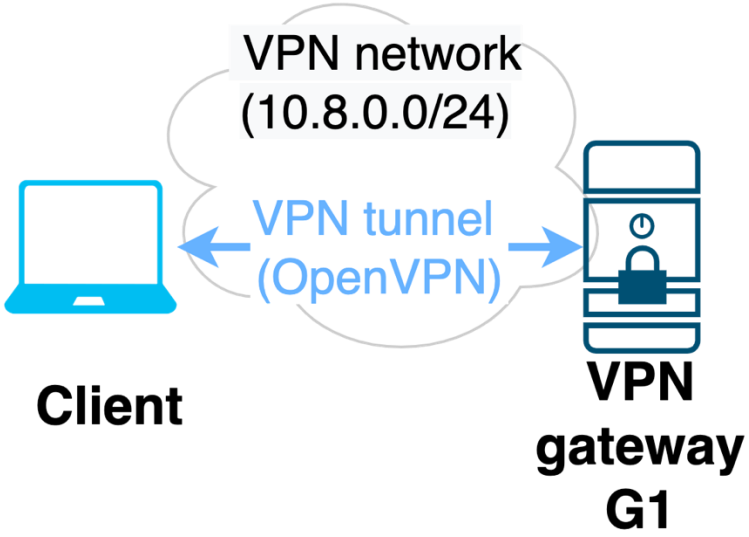
Start service

systemctl start openvpn

Example documented configuration files can be found at:

/usr/share/doc/openvpn/sample/sample-config-files/

OpenVPN – exercise scenario



OpenVPN – Exercise in 3 phases

1. **OpenVPN authentication with X.509 certificates**
2. **OpenVPN authentication with passwords**
3. **Check OpenVPN functioning and security hardening**

Phase 1 - OpenVPN Configuration

 # Example server configuration

```
local 192.168.1.1   # Modify according your settings.
port 1194
proto udp
dev tun
ca ca.crt           # Consider the settings of the previous class (private CA)
cert gw-vpn.crt     # Certificate generated for server
key gw-vpn.key      # Private key of server
dh dh2048.pem       # Diffie Hellman parameters
server 10.8.0.0 255.255.255.0
```

 # Start server (for testing)

```
# openvpn --config /etc/openvpn/server.conf
```

 # Check output of server

Phase 1- OpenVPN Configuration (client)

 # Example client configuration
client
dev tun
proto udp
remote <IP/hostname> 1194 #Modify accordingly
persist-tun
persist-key
ca ca.crt
cert cliente_vpn.crt
key cliente_vpn.key

 # Start client with (for testing, this blocks terminal)
openvpn --config /etc/openvpn/client.conf

 # Check output of client

Phase 1- Checking connectivity

 # on the server (look for tun0 interface)

ifconfig

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.1  netmask 255.255.255.255  destination 10.8.0.2
```

 # on the client (look for tun0 interface)

ifconfig

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.6  netmask 255.255.255.255  destination 10.8.0.5
```

 # client should be able to ping server IP address used in tunnel

```
sti@node2:~$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.567 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.639 ms
```

 # server should be able to ping client IP address used in tunnel

```
sti@debian:~$ ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
64 bytes from 10.8.0.6: icmp_seq=1 ttl=64 time=0.576 ms
64 bytes from 10.8.0.6: icmp_seq=2 ttl=64 time=0.892 ms
```

Phase 2 – Modify OpenVPN Configurations

Relevant information on:

/usr/share/doc/openvpn/README.auth-pam

man openvpn

 # Server configuration (to add)

; Need to use plugin **openvpn-plugin-auth-pam.so** with login service type

 # Client configuration (to add)

; Need to activate option authenticate with server using username/password

; Password should be requested to the user

```
2022-02-25 20:22:21 OpenVPN 2.5.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO]
2022-02-25 20:22:21 library versions: OpenSSL 1.1.1k  25 Mar 2021, LZO 2.10
Enter Auth Username: sti
Enter Auth Password: *******
```

Phase 3 – Hardening OpenVPN configuration (tls-auth)

Relevant information on:

https://openvpn.net/community-resources/hardening-openvpn-security/

man openvpn

Allows to add HMAC signature to SSL/TLS handshake packets for integrity verification

 # Server configuration (to add)

; Need to to generate key and secret

; Need to transfer information to clientes

; Need to activate configuration tls-auth key 0

 # Client configuration (to add)

; Need to activate configuration tls-auth ta.key 1

Phase 3 – Hardening OpenVPN configuration (tls-auth)