OpenVPN Deployment and PKI Configuration Laboratory Guide

Contents

1 Laboratory Objective 3

2 Laboratory Architecture 3

2.1 OpenVPN Server 3

2.2 OpenVPN Client 3

2.3 Observer Machine 3

3 Environment Requirements 3

4 Installing OpenVPN and Easy-RSA 3

4.1 Purpose 3

4.2 Command 4

4.3 Explanation 4

5 PKI Initialization 4

5.1 Purpose 4

5.2 Creating PKI Directory 4

5.3 Directory Structure 4

5.4 Explanation 4

6 Creating Certificate Authority 5

6.1 Purpose 5

6.2 Generated Files 5

7 Server Certificate Generation 5

7.1 Purpose 5

7.2 Generated Files 5

8 Client Certificate Generation 5

8.1 Purpose 5

9 OpenVPN Server Configuration 6

9.1 Server Configuration File 6

9.2 Explanation 6

10 Starting OpenVPN Server **6**
10.1 Command 6
10.2 Expected Result 6

11 Client Configuration **7**
11.1 Starting VPN Client 7
11.2 Expected Result 7

12 Observer Machine Role **7**

13 Connectivity Verification **7**
13.1 Ping Test 7
13.2 Expected Result 8

14 Tunnel Interface Verification **8**

15 Conclusion **8**

1 Laboratory Objective

The purpose of this lab is to deploy a secure OpenVPN server using certificate-based authentication through Public Key Infrastructure (PKI). The lab demonstrates:

- Secure server-client communication
- Certificate generation and management
- Tunnel interface creation
- Network traffic verification

2 Laboratory Architecture

The lab consists of three virtual machines:

2.1 OpenVPN Server

Acts as the central authority managing authentication, encryption, and tunnels.

2.2 OpenVPN Client

Connects securely to the server with its own certificate.

2.3 Observer Machine

Monitors VPN traffic and validates tunnel operation.

3 Environment Requirements

- Linux-based system (Ubuntu recommended)
- Administrative privileges
- Internet access

4 Installing OpenVPN and Easy-RSA

4.1 Purpose

OpenVPN provides the VPN service; Easy-RSA handles PKI (certificates and keys).

4.2 Command

```
sudo apt update
sudo apt install openvpn easy-rsa
```

4.3 Explanation

After installation, binaries for OpenVPN and certificate management tools are available.

5 PKI Initialization

5.1 Purpose

PKI allows secure authentication between server and client machines.

5.2 Creating PKI Directory

```
make-cadir ~/openvpn-ca
cd ~/openvpn-ca
```

5.3 Directory Structure

```
openvpn-ca/
├── pki/
│   ├── private/
│   ├── issued/
│   └── reqs/
├── vars
└── openssl-easyrsa.cnf
```

5.4 Explanation

- **private/** stores private keys.
- **issued/** stores signed certificates.
- **reqs/** stores certificate requests.
- **vars** defines certificate parameters.

6 Creating Certificate Authority

6.1 Purpose

The CA signs all server and client certificates to guarantee trust.

```
./easyrsa init-pki
./easyrsa build-ca
```

6.2 Generated Files

- **ca.crt** (public certificate)
- **ca.key** (private key)

7 Server Certificate Generation

7.1 Purpose

Allows clients to verify the server's identity.

```
./easyrsa gen-req server nopass
./easyrsa sign-req server server
```

7.2 Generated Files

- **server.crt**
- **server.key**

8 Client Certificate Generation

8.1 Purpose

Each client needs its own certificate for secure authentication.

```
./easyrsa gen-req client nopass
./easyrsa sign-req client client
```

9 OpenVPN Server Configuration

9.1 Server Configuration File

```
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
server 10.8.0.0 255.255.255.0
keepalive 10 120
persist-key
persist-tun
```

9.2 Explanation

- **port 1194**: VPN communication port
- **proto udp**: UDP transport for speed
- **dev tun**: Creates routed tunnel
- **server**: Defines VPN subnet
- **persist-***: Keeps tunnel and keys active after interruptions

10 Starting OpenVPN Server

10.1 Command

```
sudo systemctl start openvpn-server@server
```

10.2 Expected Result

Check new tunnel interface:

```
ip a
```

11 Client Configuration

```
client
dev tun
proto udp
remote SERVER_IP 1194
ca ca.crt
cert client.crt
key client.key
```

11.1 Starting VPN Client

```
sudo openvpn --config client.ovpn
```

11.2 Expected Result

- Successful TLS handshake
- Tunnel interface created (tun1)
- Assigned VPN IP address

12 Observer Machine Role

Monitors VPN traffic between server and client. Verifies:

- Encrypted packet flow
- Source and destination addresses
- Protocol types

13 Connectivity Verification

13.1 Ping Test

```
ping 10.8.0.1
```

13.2 Expected Result

- ICMP echo replies received
- Stable latency
- No packet loss

14 Tunnel Interface Verification

Each VPN connection generates a virtual interface:

15 Conclusion

This laboratory validates:

- PKI-based authentication
- VPN tunnel creation
- Encrypted communication
- Proper network isolation