
# ETWS-1609 雷达系统

# 网络管理设计方案

部　　门：研发部

编　　制：罗敏

版　　本：1.2

使用 LaTeX 撰写于 2018 年 5 月 29 日

# 摘　要

本文主要是关于雷达系统远程控制网络的管理方案设计，目的是实现全国各地雷达系统的远程监测与控制，主要是从两个方面来考虑如何设计实现，第一是从可操作性方面来考虑,因整个控制网连接的设备较多，特别是雷达设备甚至可能是位于郊区等人迹罕至的地方，将所有的接入设备都连接固定 IP 专网接口是不太现实的，而且费用过高，方案必须符合实际具有可操作性。第二个是从安全性的角度，整个系统分布与全国各地，且连接的雷达用户各不相同，需要考虑用户之间的隔离，以及防止非法的网络入侵破坏雷达系统工作。要实现这两个目的，可行的方案是采用 VPN 技术, 将分布于各地的雷达设备、控制主机和后端管理服务器配置成为一个虚拟专网，这样的设计只需要位于后端的 VPN 服务器一个固定 IP，其它设备只需要连接普通的互联网服务便可以实现目的，费用低廉具有可操作性, 且 VPN 的数据传输采用了多重加密技术可以有效防止数据泄露。

**关键字：**　远程监控　VPN　虚拟专网　安全性

# 目录

# 第一章 设计方案

## 1.1 系统结构

要对整个雷达系统进行网络设计，则需要先弄楚整个雷达系统的结构组成，下面主要对雷达系统的特点进行了一些总结：

- 系统庞大设备多，包括雷达、控制服务器、监控摄像头、UPS 设备等等
- 设备分散位于全国各地，后期可能还会不停的在各个地方扩展
- 需要时时监控设备工作情况，随时获取雷达数据

整个系统的结构如图 1.1 所示：

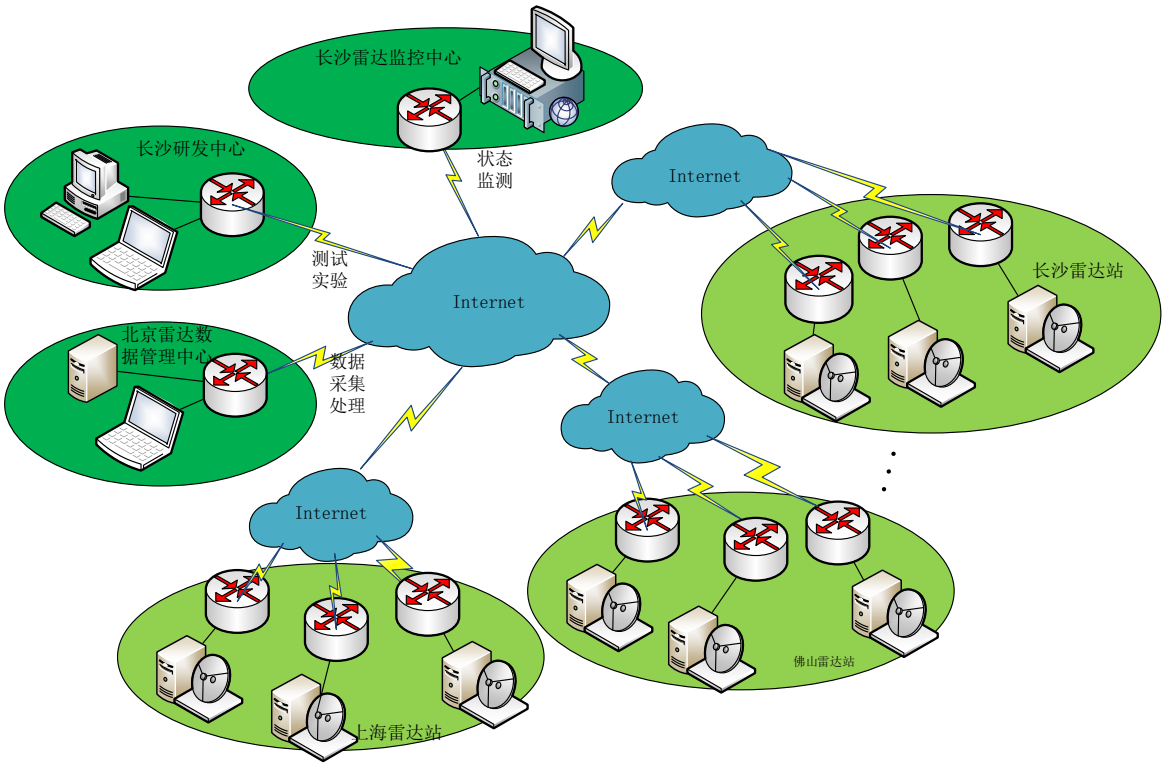


图 1.1: 雷达系统结构图

在图 1.1 可以看到雷达系统组成结构复杂，每个站点有多台雷达，不同站点的雷达也是各自分散的，根据目前的发展情况，后续还会增加更多的雷达站点，除了雷达站点之外雷达系统的监控管理中心在长沙，长沙的研发中心也会有访问雷达站点进行实验测试的需求，北京雷达数据管理中心则是主要需要从各个雷达站点读取数据进行雷达数据的处理。

## 1.2 网络设计

上一节简单的介绍了整个雷达系统的结构，本节根据雷达系统的结构, 进行网络管理方案设计。针对目前的情况和现有的技术，本文决定采用 VPN 技术组建私有专网，具体组网方案如图 1.2 所示：

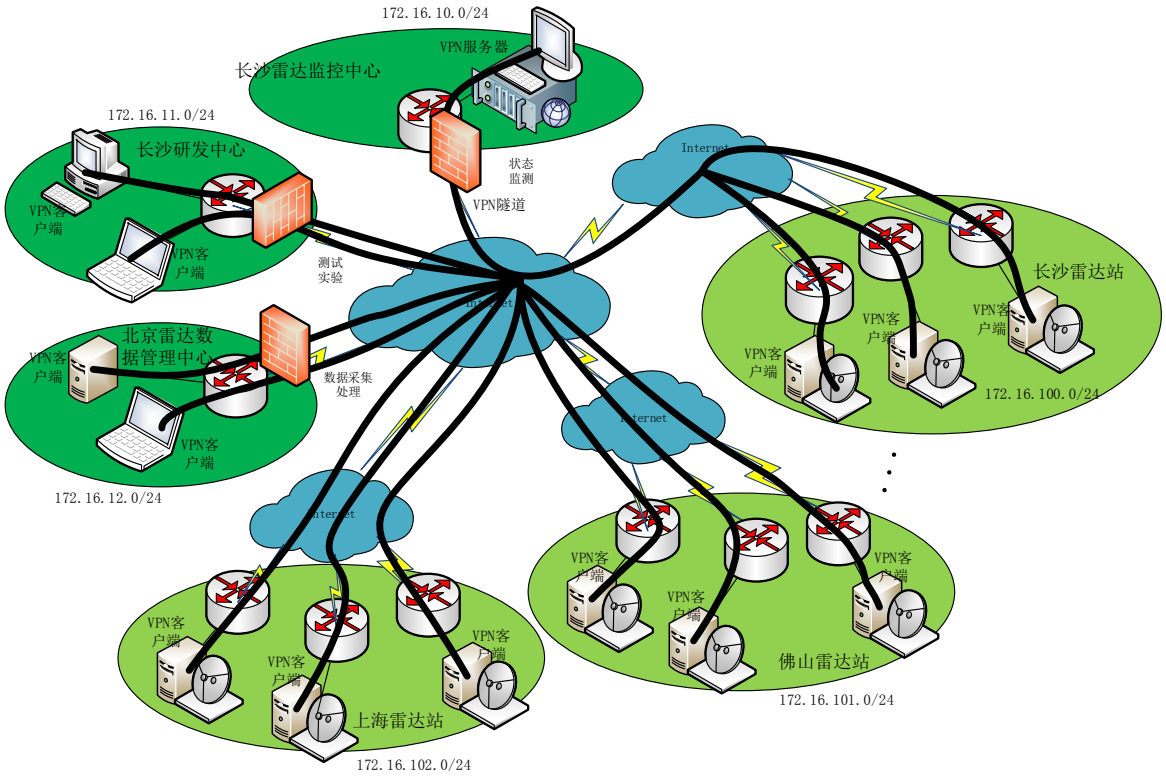


图 1.2: 雷达系统网络设计

在长沙雷达管理中心设置一台 VPN 服务器，来提供 VPN 服务管理，其它的网络节点，则通过 VPN 客户端远程加入 VPN 网络，因此长沙雷达管理中心需要一个专网 IP，以便其它节点可以方便访问。为了系统安全还需要进行网络隔离，各个雷达站之间不能相互访问，同

时需要在各个管理中心添加防火墙，防止雷达站访问到网络中心。

目前考虑没有能够满足要求的集成 VPN 路由器，只能采用在雷达设备的内部服务器上配置 VPN 客户端，后期可以考虑我司自主设计一个多功能网关设备作为雷达站点的控制网关，届时可以将 VPN 客户端配置在多功能集成网关上。

## 1.3 网络划分

在图 1.2 中标注的网络地址是指在接入 VPN 专网时分配的虚拟局域网 IP 地址，具体规划网段如表 1.1。

| 区域 | IP 地址段 | 特殊地址 | 备注 |
|---|---|---|---|
| 长沙监控中心 | 172.16.10.0/24 | 172.16.10.1 | VPN 服务器地址 |
| 长沙研发中心 | 172.16.11.0/24 | - | - |
| 北京数据中心 | 172.16.12.0/24 | - | - |
| 长沙雷达站 | 172.16.100.0/24 | radar01:172.16.100.10<br>radar02:172.16.100.21<br>radar02:172.16.100.30 | 雷达设备地址 |
| 佛山雷达站 | 172.16.101.0/24 | radar01:172.16.101.10<br>radar02:172.16.101.21<br>radar03:172.16.101.30<br>radar01:172.16.101.41<br>radar02:172.16.101.50<br>radar03:172.16.101.61 | 雷达设备地址 |
| 上海雷达站 | 172.16.102.0/24 | radar01:172.16.102.10<br>radar02:172.16.102.21<br>radar02:172.16.102.30 | 雷达设备地址 |
| ... | ... | ... | ... |

表 1.1: VPN 专网 IP 地址划分

表中目前只列出了部分地区的 IP 地址网段，后续根据雷达站的部署可以添加地址网段划分，但是基本的地址划分遵循以下个原则：

- 整个 VPN 虚拟专网使用 172.16.0.0/16 网段
- 长沙监控中心使用 172.16.10.0/24 网段，VPN 服务器使用 172.16.10.1IP 地址
- 长沙研发中心使用 172.16.11.0/24 网段，接入设备自动分配 IP
- 北京数据中心使用 172.16.12.0/24 网段，数据服务器使用用 172.16.12.100 地址，其它接入设备自动分配 IP
- 雷达站使用 172.16.100.0/24-172.16.254.0/24 网段, 具体根据入网先后编址

- 雷达设备使用 172.16.10x.10-172.16.10x.127 的固定地址

清晰的编址可以有效的定位设备、判断问题，是雷达系统网络设备管理的基础, 后续根据讨论之后可以加入新的编址原则以便更好的管理雷达系统网络。

# 第二章 技术方案

## 2.1 VPN 比较

在上一章中提出了采用建立 VPN 虚拟专网的方式来实现远程的设计接入和管理，目前可以采用的 VPN 技术方案有多种, 例如 PPTP、L2TP、OpenVPN 等，下面将几种常见的 VPN 做了个对比，如图 2.1 所示：

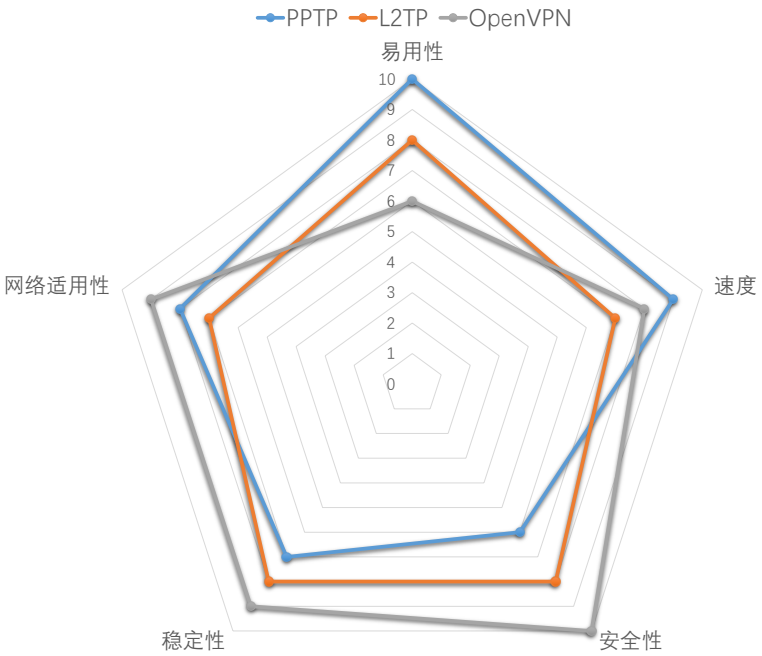


图 2.1: VPN 协议对比 (1)

在图 2.1 中可以看到，PPTP 协议在易用性和速度方面是有优势的，但是 OpenVPN 则在稳定性、安全性和网络适用性上更胜一筹，而 L2TP 协议则是一种比较折中的方案。另外也找了网上的一些关于 VPN 的对比，如图 2.2 所示：

在雷达系统中稳定性和安全性是更应该被着重考虑的因素，综合来看在雷达系统网络管理中采用 OpenVPN 方案是一种更加合适的选择。

| | PPTP | L2TP/IPSEC | OpenVPN |
|---|---|---|---|
| 简介 | 微软推出的第一个VPN协议。占用资源少，应用最为广泛。 | 更高级的VPN协议，支持各种平台。安全性更高，但是不太灵活，容易被封锁。 | 开源的vpn协议，加密性和适应性都比较好，也比较灵活，不容易被封锁。通过udp端口可以获得较好的速度。 |
| 加密 | 支持40位、56位和128位加密 | 256位加密 | 可自定义160位-256位 |
| 平台支持 | ▪ Windows<br>▪ Mac<br>▪ Linux<br>▪ iOS<br>▪ Android<br>▪ DDWRT | ▪ Windows<br>▪ Mac<br>▪ Linux<br>▪ iOS<br>▪ Android | ▪ Windows（第三方软件）<br>▪ Mac（第三方软件）<br>▪ Linux<br>▪ iOS（第三方软件）<br>▪ Android（第三方软件）<br>▪ DDWRT |
| 连接速度 | 很快 | 快 | 一般 |
| 端口 | 1723 TCP | 500 UDP<br>1701 UDP<br>5500 UDP | 可根据需要自定义和更换 |
| 防封锁 | 通过协议和端口很容易被封锁 | 通过协议和端口很容易被封锁 | 比较难封锁 |

图 2.2: VPN 协议对比 (2)

## 2.2 OpenVPN 介绍

VPN 直译就是虚拟专用通道，是提供给企业之间或者个人与公司之间安全数据传输的隧道，OpenVPN 无疑是 Linux 下开源 VPN 的先锋，提供了良好的性能和友好的用户 GUI。使用了 OpenSSL 加密库中的 SSLv3/TLSv1 协议函数库，使得数据安全性更加的有保障。目前 OpenVPN 能在 Solaris、Linux、OpenBSD、FreeBSD、NetBSD、Mac OS X 与 Microsoft Windows 以及 Android 和 iOS 上运行，并包含了许多安全性的功能。

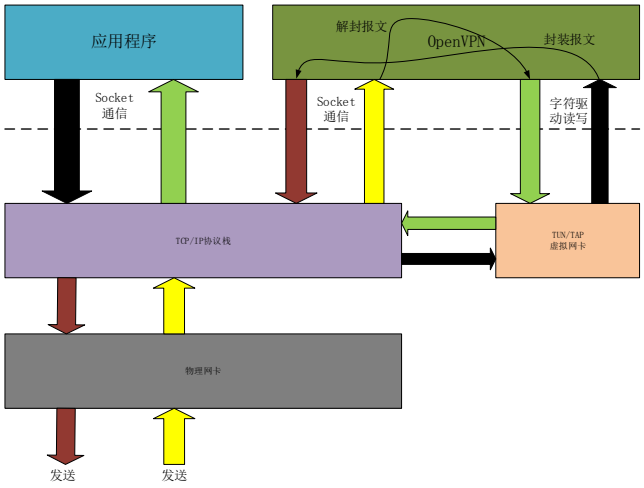


图 2.3: OpenVPN 原理

OpenVPN 基本工作原理，如图 2.3 所示，应用层的外出数据，经过系统调用接口传入核心 TCP/IP 层做处理，在 TCP/IP 经过路由到虚拟网卡，虚拟网卡的网卡驱动发送处理程序 hard_start_xmit() 将数据包加入 skb 表并完成数据包从核心区到用户区的复制，OpenVPN 调用虚拟网卡的字符处理程序 tun_read()，读取到设备上的数据包，对读取的数据包使用 SSL 协议做封装处理后，通过 socket 系统调用发送出去。物理网卡接收数据包，经过核心 TCP/IP 上传到 OpenVPN，OpenVPN 通过 link_socket_read() 接收数据包，使用 SSL 协议进行解包处理，经过处理的数据包 OpenVPN 调用虚拟网卡的字符处理程序 tun_write() 写入虚拟网卡的字符设备，设备驱动程序完成数据从用户区到核心区的复制，并将数据写入 skb 链表，然后调用网卡 netif_rx() 接收程序，数据包再次进入系统 TCP/IP 协议栈，传到上层应用程序。如果想要了解更详细可以参考官方网站 [1]。

# 第三章 网络安全

网络的安全性是在部署远程控制网络时另外一个需要重点考虑的问题，本文提出可以从 OpenVPN 自身安全策略和防火墙 (Firewall) 安全策略两方面来保障网络的安全性。

## 3.1 OpenVPN 安全策略

OpenVPN 提供的安全策略主要有以下几个方面：

- 认证证书机制，用户需要通过 VPN 服务器分发的认证证书才能接入服务端
- 用户名和密码机制，每个客户端可以设置不同的用户名和密码
- 传输数据加密机制，保障所传输的数据是安全可靠的

可以看到 OpenVPN 提供了多重的网络和数据安全保障，在此基础上还需要配合防火墙策略来进行各个区域的权限控制。

## 3.2 Firewall 策略

根据雷达系统网络控制的需求，可以确定以下几点的防火墙策略原则：

- 每个接入虚拟专网的区域均需要添加防火墙策略
- 权限区分，管理中心、研发中心和数据中心需要更高权限，雷达站则是低权限
- 设置管理员权限，可以访问网络上所有设备

根据确定的防火墙策略制定原则，可以制定雷达系统网络管理的一些基本防火墙策略，如表 3.1 所示，由表可以看到管理员用户是可以访问所有网段的，主要用于维护整个网络和进行相关的配置，属于 admin 权限，长沙研发中心用户和北京数据中心用户可以访问所有雷达站点设备和 VPN 服务器，属于 user 权限，各个雷达站的则是 connetor 权限，只能访问 VPN 服务器。

| 类别 | IP 地址 | 权限级别 | 备注 |
|---|---|---|---|
| 管理员 | 172.16.10.0/24 | admin | 可以访问所有网段 |
| 长沙研发中心用户 | 172.16.11.0/24 | user | 只能访问 VPN 服务器和雷达站网段 |
| 北京数据中心用户 | 172.16.12.0/24 | user | 只能访问 VPN 服务器和雷达站网段 |
| 长沙雷达站设备 | 172.16.100.0/24 | connector | 只能访问 VPN 服务器 |
| 佛山雷达站设备 | 172.16.101.0/24 | connector | 只能访问 VPN 服务器 |
| 上海雷达站设备 | 172.16.102.0/24 | connector | 只能访问 VPN 服务器 |
| ... | ... | ... | ... |

表 3.1: Firewall 策略

如果使用图来表示将会更加清晰，在图 3.1 中，红色区域表示 admin 区域，绿色区域表示 user 区域，黄色区域表示 connector 区域。

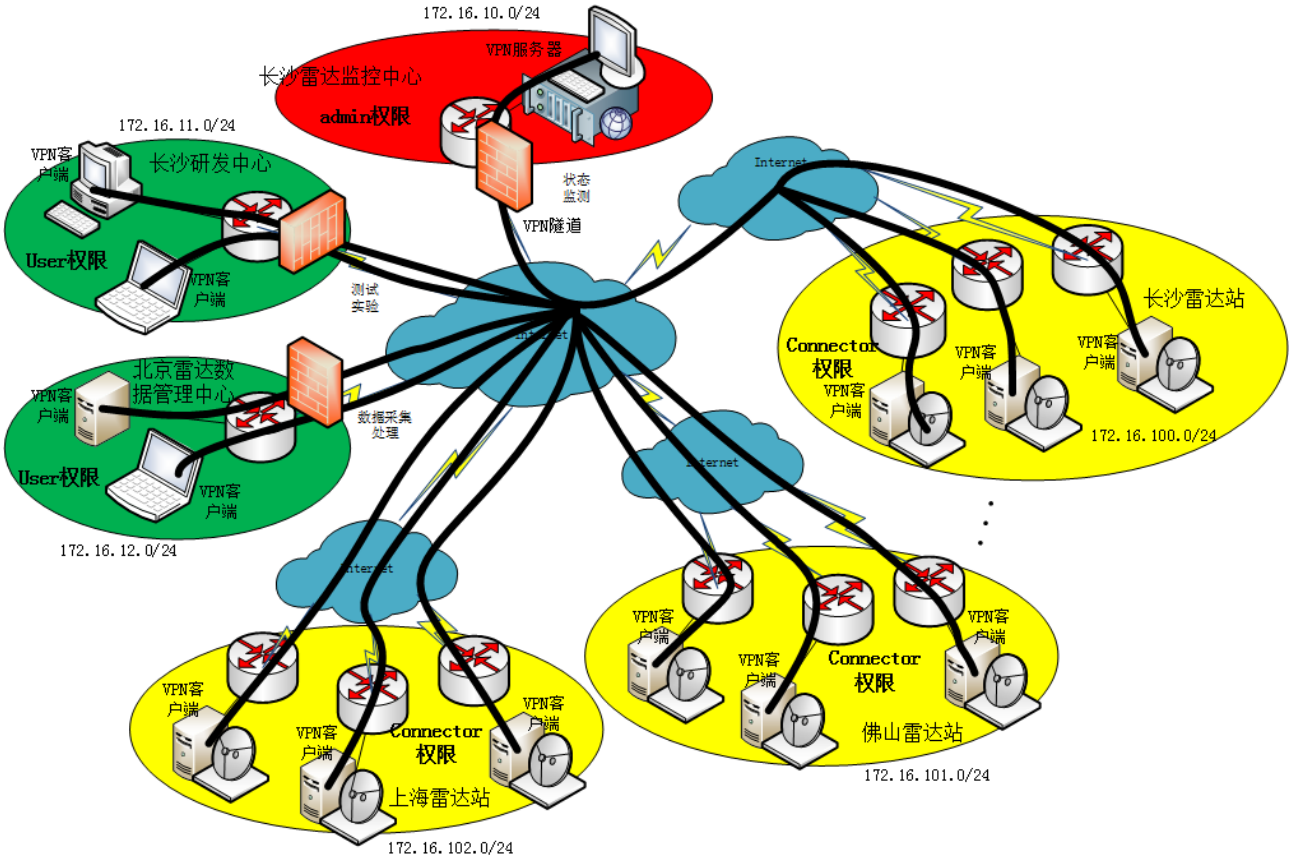


图 3.1: 防火墙策略区域

# 第四章 配置步骤

## 4.1 OpenVPN 安装

OpenVPN 官方基本提供了多个平台的 OpenVPN 包，在 Windows 平台可以直接安装 Openvpn 安装包，本节主要介绍在 Linux 平台编译安装 OpenVPN 包。在编译 OpenVPN 软件前最好先编译安装 openssl 软件，因为 OpenVPN 调用了 Openssl 函数库，OpenVPN 的客户端和服务端建立 SSL 链接的过程是通过调用 Openssl 来实现的。

在下载好的源码包中，有个 INSTALL 文件，里面有具体的安装步骤和注意事项，可按需进行查阅。进入 openvpn 目录，需要手动生成编译脚本 configure。

```
autoreconf -i -v -f //BUILD COMMANDS FROM SRC REPOSITORY CHECKOUT
```

生成 configure 脚本之后就可以执行 configure 脚本了：

```
./configure –prefix=/usr/local/OpenVPN –disable-lzo //如需禁用 lzo, 加入此参数
make
make install
```

编译完成之后默认是安装在/usr/local 目录下的，如果需要改变安装目录，在./configure 时添加 –prefix=PREFIX 即可，或者直接建立软连接。

```
ln -s /usr/local/OpenVPN/sbin/openvpn /usr/sbin/openvpn
```

## 4.2 OpenVPN 配置

OpenVPN 的配置是分成几个阶段的，第一阶段先需要进行密钥的生成, 第二阶段才是具体的 OpenVPN 功能的配置，下面分成两个小节来叙述 OpenVPN 配置。

### 4.2.1 密钥配置

要完成 OpenVPN 的密钥配置需要先安装配置 easy-rsa，且基于 easy-rsa3 来进行配置。easy-rsa 源文件需要从 github 进行下载[2]，其和 OpenVPN 是在同一个软件库。完成下载之

后按照下面的步骤进行配置。

```
mkdir /etc/openvpn
mkdir /usr/local/OpenVPN/etc
ln -s /usr/local/OpenVPN/etc /etc/openvpn
cd easyrsa3/
cp vars.example vars
```

主要需要对 vars 中的一些选项进行配置, 具体修改如图 4.1，

```
set_var EASYRSA_REQ_COUNTRY     "CN"
set_var EASYRSA_REQ_PROVINCE    "SH"
set_var EASYRSA_REQ_CITY        "ChangSha"
set_var EASYRSA_REQ_ORG         "Copyleft Certificate Co"
set_var EASYRSA_REQ_EMAIL       "luomin@estonetech.com"
set_var EASYRSA_REQ_OU          "My Organizational Unit"
```

图 4.1: vars 文件修改

接下来，开始初始化 pki:

```
./easyrsa init-pki
```

创建根证书 CA:

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa build-ca

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/usr/local/OpenVPN/easy-rsa/easyrsa3/pki/private/ca.key.EUNv9gSqxP'
Enter PEM pass phrase:                          #输入密码，用来证书签名
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:myhost  #输入一个common name

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/usr/local/OpenVPN/easy-rsa/easyrsa3/pki/ca.crt
```

图 4.2: 创建根证书

创建服务器端证书:

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa gen-req server nopass

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.....+++
.+++
writing new private key to '/usr/local/OpenVPN/easy-rsa/easyrsa3/pki/private/server.key.WCjfTjvd25'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]:host-server        #此处的common name不要同于CA的
common name

Keypair and certificate request completed. Your files are:
req: /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/reqs/server.req
key: /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/private/server.key
```

图 4.3: 创建服务器端证书

签约服务器端证书:

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa sign server server

Note: using Easy-RSA configuration from: ./vars


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 3650 days:

subject=
    commonName                = host-server


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes   #在此输入yes
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/private/ca.key:    #输入CA的数字签名密码
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :PRINTABLE:'host-server'
Certificate is to be certified until Jul 12 02:07:19 2027 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/issued/server.crt
```

图 4.4: 签约服务器端证书

创建 Diffie-Hellman，确保 key 穿越不安全网络：

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa gen-dh

Note: using Easy-RSA configuration from: ./vars
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....................................................................................................
.....................................................................................................
.....................................+.....................++*++*

DH parameters of size 2048 created at /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/dh.pem
```

图 4.5: 创建 Diffie-Hellman

同样的在客户端也是需要证书的，需要在另外一个路径 clone 一份 easyrsa 源码。然后进入源码执行初始化操作。

```
./easyrsa init-pki
```

生成客户端证书生成请求和 key：

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa gen-req Ops

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.+++
..............................+++
writing new private key to '/root/client/easy-rsa/easyrsa3/pki/private/Ops.key.8uUfDoiUPb'
Enter PEM pass phrase:       #输入密码
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [client_one]:Ops        #输入common name

Keypair and certificate request completed. Your files are:
req: /root/client/easy-rsa/easyrsa3/pki/reqs/Ops.req
key: /root/client/easy-rsa/easyrsa3/pki/private/Ops.key
```

图 4.6: 创建客户端证书

执行下面的命令导入客户端证书：

```
cd /usr/local/OpenVPN/easy-rsa/easyrsa3
./easyrsa import-req /root/client/easy-rsa/easyrsa3/pki/reqs/Ops.req Ops
```

签约客户端证书：

```
[root@vm172-26-0-5 easyrsa3]# ./easyrsa sign client Ops

Note: using Easy-RSA configuration from: ./vars


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 3650 days:

subject=
    commonName                = Ops


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes      #输入yes
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/private/ca.key:     #输入CA的数字签名密码
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :PRINTABLE:'Ops'
Certificate is to be certified until Jul 12 03:15:23 2027 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /usr/local/OpenVPN/easy-rsa/easyrsa3/pki/issued/Ops.crt      #客户端证书生成
```

图 4.7: 签约客户端证书

至此，客户端和服务端的证书已经配置完毕，接下来将相关文件拷贝至 openvpn 安装目录下:

```
cp ca.crt /etc/OpenVPN/
cp private/server.key /etc/Openvpn/
cp issued/server.crt /etc/Openvpn/
cp dh.pem /etc/Openvpn/
```

拷贝客户端证书和秘钥至客户端 client 目录，在后面客户机连接服务器的时候是会要用到的：

```
cp ca.crt /root/client/
cp issued/Ops.crt /root/client/
cp /root/client/easy-rsa/easyrsa3/pki/private/Ops.key /root/client/
```

本小节主要介绍了 OpenVPN 的密钥生成和配对，整个过程稍显复杂，不过 OpenVPN 的安全性便是由密钥和证书来保证，所以这是必不可少的一个环节。

### 4.2.2 网络配置

本节主要介绍 OpenVPN 的一些基本网络配置，如下所示，列出了服务端 server.conf 文件的一些基本配置，可以看到端口被配置成了 5094，使用的通行模式是 udp，配置的 OpenVPN 服务器网段为 172.16.10.0/24，相应的还配置了之前生成的密钥文件。

```
port 5094
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key #This file should be kept secret
dh /etc/openvpn/dh.pem
server 172.16.10.0 255.255.255.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
tls-auth /etc/openvpn/ta.key 0 # This file is secret
cipher AES-256-CBC
persist-key
persist-tun
status /tmp/openvpn-status.log
verb 3
explicit-exit-notify 1
```

可以再看一下客户端的配置文件 client.conf，如下所示，主要配置了程序为 client 模式，设置了服务器的地址及端口，以及上一节中介绍的相关客户端的证书和密钥。

```
client
dev tun
proto udp
remote 218.76.8.148 5094
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/Ops.crt
key /etc/openvpn/Ops.key
remote-cert-tls server
tls-auth /etc/openvpn/ta.key 1
cipher AES-256-CBC
verb 3
```

# 第五章 总结

整个雷达系统的网络管理纷繁复杂，主要是因为设备和用户较多，并且需要各种不一样的权限划分，数据需要远程传输，设备需要远程管理，各种安全性的要求也一一呈现。本文主要提出了采用 OpenVPN+Firewall 的管理方案，在 OpenVPN 的基础上添加相关的防火墙策略，划分各个区域，确定每个接入设备的 IP 地址和接入用户的使用权限。在相关配置都论述清楚的情况下，针对雷达系统的管理也就变得清晰明了。

# 参 考 文 献

[1] https://www.openvpn.net/

[2] https://github.com/OpenVPN/easy-rsa.git