



# 基于OpenConnect 构建的 SSL VPN解决方案

SSL VPN solution based on OpenConnect.Virtual  
I private network (VPN) is a way to connect the private network  
of the enterprise through the public network safely,

.....  
成/都/虫/洞/奇/迹/科/技/有/限/公/司

## 目录

---

版权声明.....	3
1. 写在前面.....	4
2. 部署方案.....	4
2.1 服务端部署.....	5
2.2 客户端部署.....	9
2.3 智能路由与限速.....	9
3. 连接测试.....	10
4. 写在最后.....	10

## 版权声明

版权所有 © 虫洞奇迹科技有限公司 2017。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

### 商标声明



和其他成都虫洞奇迹科技有限公司商标均为成都虫洞奇迹科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

### 注意

您购买的产品、服务或特性等应受成都虫洞奇迹科技有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，成都虫洞奇迹科技有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

成都虫洞奇迹科技有限公司  
电话：400-090-2980  
邮箱：contact@lingyuecloud.com  
网址：www.lingyuecloud.com

## 1. 写在前面

---

VPN（Virtual Private Network），虚拟专用网络，是一种通过公用网络安全地对企业内部专用网络进行远程访问的连接方式，可有效保障通信的机密性。如，出差办公人员可通过 VPN 通道安全地访问公司内部 OA 系统。发展至今的 VPN 同时融合了访问控制、路由选择、传输管理等多种功能，在全球的信息安全体系及各行业的信息系统中已发挥着重要作用。

常见的 VPN 主要有：

- PPTP VPN
- L2TP VPN
- IPSec VPN
- SSL VPN

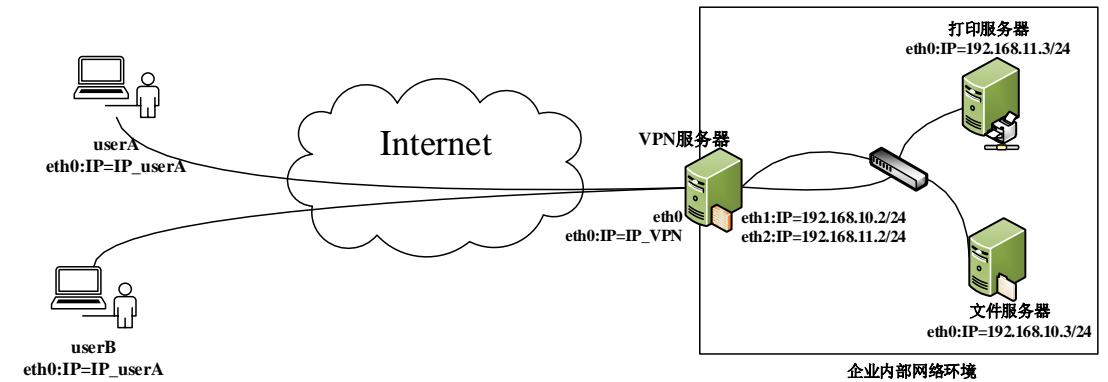
## 2. 部署方案

---

在构建 VPN 方案时，需要考虑服务端、客户端、传输通道三个部分。本次分别选择如下组件进行构建：

- VPN 服务器：ocserv（OpenConnect VPN Server），一款开源的 VPN 服务端软件，可以提供端到端的安全连接服务，可以在思科设备以及众多的 Linux 发行版进行安装和部署；
- VPN 客户端：AnyConnect，由思科推出的 VPN 客户端，目前已有支持 Windows、Android、iOS、OS X、Ubuntu、WebOS 等操作系统的版本；
- 传输协议：SSL。

本次部署场景及达到的效果如图所示。



本次部署测试中相关环境配置如下表所示。

节点	配置信息
userA	IP 地址=IP_userA，其到 VPN 服务器 eth0 的网络可达； VPN 客户端软件：AnyConnect；
userB	IP 地址=IP_userB，其到 VPN 服务器 eth0 的网络可达； VPN 客户端软件：AnyConnect；
VPN 服务器	操作系统：CentOS 7.3 X86_64 位版本 VPN 服务器软件：ocserv 0.11.7 网络配置： (1) eth0: IP=IP_VPN，该地址对外提供服务，外部实体 userA、userB 可访问 (2) eth1: IP=192.168.10.2/24 //该地址与打印服务器地址同一子网、互通 (3) eth2: IP=192.168.11.2/24 //该地址与文件服务器地址同一子网、互通
打印服务器	eth0: IP=192.168.11.3/24
文件服务器	eth0: IP=192.168.10.3/24

其中，

- (1) 用户 userA 在利用 VPN 客户端连接 VPN 服务器之后，可访问公司内部打印服务器 192.168.11.3，且带宽被限速为 2MB；
- (2) 用户 userB 在利用 VPN 客户端连接 VPN 服务器之后，可访问公司内部文件服务器 192.168.10.3，无带宽限制。

注：在下述部署中，涉及在服务器 CentOS 7.3 操作系统环境上的所有操作，均以 root 身份登录并执行。

## 2.1 服务端部署

### (一) 安装依赖包

```
#yum -y install wget gcc nettle* gnutls *readline* libev* autogen protobuf*
```

## （二）安装 ocserv 软件

创建文件存储目录/home/centos，并下载 ocserv 源码包：

```
# cd /home
```

```
# mkdir centos && cd centos/
```

下载软件源码包：

```
# wget ftp://ftp.infradead.org/pub/ocserv/ocserv-0.11.7.tar.xz
```

解压源码包：

```
# xz -d ocserv-0.11.7.tar.xz
```

```
# tar -xvf ocserv-0.11.7.tar
```

进入到解压后的 ocserv-0.11.7 文件夹中，执行编译安装：

```
# cd ocserv-0.11.7 && ./configure --prefix=/usr/local/ocserv && make && make install
```

## （三）配置 ocserv

（1）创建目录,用于存储后续生成的证书

```
# mkdir -p /usr/local/ocserv/etc/certificates
```

（2）复制源码目录 ocserv-0.11.7 里面的 sample.config, sample.passwd 文件到 /usr/local/ocserv/etc/目录

```
# cd /usr/local/ocserv/etc && cp /home/centos/ocserv-0.11.7/doc/sample.passwd ./
```

```
# cd /usr/local/ocserv/etc && cp /home/centos/ocserv-0.11.7/doc/sample.config ./
```

之后，通过 **tree** 命令可查看/usr/local/ocserv 目录结构最终应当为：

```
.
├── bin
│   ├── occtl
│   ├── ocpasswd
│   └── ocserv-fw
├── etc
│   ├── certificates
│   │   ├── ca-cert.pem
│   │   ├── ca-key.pem
│   │   ├── ca.tmpl
│   │   ├── server-cert.pem
│   │   ├── server-key.pem
│   │   └── server.tmpl
│   ├── sample.config
│   └── sample.passwd
├── sbin
│   └── ocserv
├── share
│   └── man
│       └── man8
│           ├── occtl.8
│           ├── ocpasswd.8
│           └── ocserv.8
```

7 directories, 15 files

## （3）配置证书

由于采用 SSL 协议作为传输实现，因此还需要在 VPN 服务端进一步配置证书。证书可通过自签名或者购买的方式获取。当前已有众多的证书服务商提供了免费的 DEV 证书可供申请使用。如需要自签名证书，请参考 <https://devcenter.heroku.com/articles/ssl-certificate->

[self](#)。

本次通过下面的脚本 `create_cert.sh` 来创建自签名证书，`create_cert.sh` 脚本的具体内容如下：

```
#!/bin/sh

# create self-signed server certificate:
read -p "Enter your domain [www.lingyuecloud.com]: " DOMAIN
echo "Create server key..."
openssl genrsa -des3 -out $DOMAIN.key 1024
echo "Create server certificate signing request..."
SUBJECT="/C=US/ST=Mars/L=LA/O=iTranswarp/OU=iTranswarp/CN=$DOMAIN"
openssl req -new -subj $SUBJECT -key $DOMAIN.key -out $DOMAIN.csr
echo "Remove password..."
mv $DOMAIN.key $DOMAIN.origin.key
openssl rsa -in $DOMAIN.origin.key -out $DOMAIN.key
echo "Sign SSL certificate..."
openssl x509 -req -sha256 -days 3650 -in $DOMAIN.csr -signkey $DOMAIN.key -out $DOMAIN.crt
```

假设 VPN 服务端域名为 `www.lingyuecloud.com`，运行 `create_cert.sh` 脚本生成服务端的自签名证书和 key。

```
# sh create_cert.sh
```

根据提示填入相应的信息：

```
Enter your domain [www.lingyuecloud.com]: www.lingyuecloud.com //红色为手动输入的 VPN 服务器的域名，此处仅为示例，请按照实际的 VPN 服务器域名填写
Create server key...
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for www.lingyuecloud.com.key: //请输入密码
Verifying - Enter pass phrase for www.lingyuecloud.com.key: //请输入密码
Create server certificate signing request...
Enter pass phrase for www.lingyuecloud.com.key: //请输入密码
Remove password...
Enter pass phrase for www.lingyuecloud.com.origin.key: //请输入密码
writing RSA key
Sign SSL certificate...
Signature ok
subject=/C=CN/ST=SC/L=CD/O=Longyuan/OU=Longyuan/CN=www.lingyuecloud.com
Getting Private key
```

如上，首先输入 VPN 服务端的域名，例如：`www.lingyuecloud.com`，在实际部署中请按照实际的 VPN 域名填写。之后，设置和确认解析密码，四个待输入的密码请保持一致。

`create_cert.sh` 脚本执行完成后，将在当前目录生成如下 4 个文件：

```
www.lingyuecloud.com.crt
www.lingyuecloud.com.csr
www.lingyuecloud.com.key
www.lingyuecloud.com.origin.key
```

其中，文件名为 `www.lingyuecloud.com.crt` 和 `www.lingyuecloud.com.key` 的文件，即 VPN

服务端需要的证书和 key，将上述两个文件拷贝到 `/usr/local/ocserv/etc/certificates/` 目录下：

```
# cp -r www.lingyuecloud.com.crt www.lingyuecloud.com.key /usr/local/ocserv/etc/certificates/
```

#### (4) 配置 **ocserv**

在 `ocserv` 的配置文件 `/usr/local/ocserv/etc/sample.config` 中，参考下述的配置进行相应参数的修改，相关配置项如下：

```
auth = "plain[passwd=./sample.passwd]" # 认证方式及密钥路径
tcp-port = 443 # 监听端口
udp-port = 443
run-as-user = root # 启动用户
run-as-group = root
socket-file = /var/run/ocserv-socket
server-cert = /usr/local/ocserv/etc/certificates/www.lingyuecloud.com.crt # 证书，来源于第
```

#### (3) 步生成的自签名证书

```
server-key = /usr/local/ocserv/etc/certificates/www.lingyuecloud.com.key # 证书的 key，来源于
```

#### 第 (3) 步生成的自签名证书的 key

```
isolate-workers = false
max-clients = 16 # 最大连接数
max-same-clients = 2 # 相同用户最大连接设备
keepalive = 32400
dpd = 90
mobile-dpd = 1800
switch-to-tcp-timeout = 25
try-mtu-discovery = false
cert-user-oid = 0.9.2342.19200300.100.1.1
tls-priorities = "NORMAL:%SERVER_PRECEDENCE:%COMPAT:-VERS-SSL3.0"
auth-timeout = 240
min-reauth-time = 300
max-ban-score = 50
ban-reset-time = 300
cookie-timeout = 300
deny-roaming = false
rekey-time = 172800
rekey-method = ssl
use-occtl = true # 是否可以使用 occtl 进行管理
pid-file = /var/run/ocserv.pid
device = vpns # 建立隧道的设备名
predictable-ips = true
ipv4-network = 172.16.16.0 # 隧道设备的 IP 段
ipv4-netmask = 255.255.255.0
ping-leases = false
route = 192.168.10.0/255.255.255.0 # 全局路由，需要路由的 IP 或者 IP 段
# no-route = 192.168.0.0/255.255.0.0
cisco-client-compat = true
dtls-legacy = true
```

#### (5) 启动 **ocserv** 服务

```
# /usr/local/ocserv/sbin/ocserv -f -c /usr/local/ocserv/etc/sample.config -d 1
```

建议使用 `supervisor` 进行启动。关于 `supervisor` 的安装和配置，请参考 <http://supervisord.org/installing.html>



### （6）配置 **Linux** 路由功能

增加地址伪装：

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

```
# iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

开启 Linux 路由转发：

有两种方式，一种是永久打开路由转发，另一种是在 Linux 本次运行过程中打开路由转发功能，重启后此功能将无效。配置方式分别如下：

永久添加：`# echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf`

临时添加：`# echo 1 > /proc/sys/net/ipv4/ip_forward`

### （7）创建用户账户 userA、userB

```
# /usr/local/ocserv/bin/ocpasswd -c /usr/local/ocserv/etc/sample.passwd userA
```

将提示输入 userA 的密码

```
# /usr/local/ocserv/bin/ocpasswd -c /usr/local/ocserv/etc/sample.passwd userB
```

将提示输入 userB 的密码

至此，服务端的配置已完成。此处 userA、userB 的账号和密码即为后续用户使用 AnyConnect 登录时使用的账户。

## 2.2 客户端部署

依据 userA、userB 使用的操作系统环境，访问 <https://www.ed.ac.uk/information-services/computing/desktop-personal/vpn/vpn-cisco-client>，下载适用于对应操作系统的 VPN 客户端软件即可。

## 2.3 智能路由与限速

通过下述配置，使得 userA 将走专门的路由，而 userB 会走全局路由；且 userA 的上传、下载带宽将被限速为 2M。

### （1）为 userA 新建配置文件

```
# vi /usr/local/ocserv/etc/config-per-user/userA
```

在上述文件添加下述信息：

```
route = 192.168.11.0/24 #局部路由，userA 需要路由的 IP 或者 IP 段
```

```
rx-data-per-sec = 2000000 #实现接收限速配置
```

```
tx-data-per-sec = 2000000 #实现发送限速配置
```

### （2）修改 ocserv 的配置文件 sample.conf

```
# vi /usr/local/ocserv/etc/sample.conf
```

在 sample.conf 配置文件中添加以下配置：

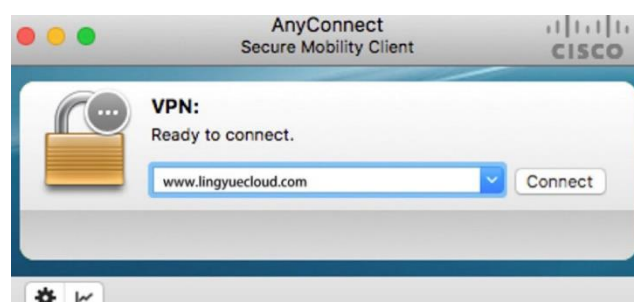
```
config-per-user = /usr/local/ocserv/etc/config-per-user/
```

(3) 重启服务端

```
# pkill ocserv && sleep 2 && /usr/local/ocserv/sbin/ocserv -f -c /usr/local/ocserv/etc/sample.config -d 1
```

### 3. 连接测试

在 userA、userB 客户端打开 AnyConnect 软件，输入域名（注：请填写实际配置的 VPN 服务器的域名，且该域名 userA、userB 可正常访问。若测试环境中 userA eth0、userB eth0、VPN 服务器 eth0 位于同一子网内，则可通过在 userA、userB 系统对应的 hosts 文件中添加如“IP VPN 地址 www.lingyuecloud.com”信息，即可实现基于域名的访问。）之后，点击 Connect 即可连接 VPN 服务器，如下图所示。



之后，点击 connect 按钮，将弹出如图所示的用户登录窗口、密码输入窗口，分别输入在第 (2.1) 节中创建的用户名、密码即可。



### 4. 写在最后

至此，基于 OpenConnect 构建的 SSL VPN 解决方案构建完成，并具有智能路由和带宽限速功能。在实际的应用场景中，还可进一步结合 FreeIPA 来实现企业级的统一用户身份认证与授

权管理，关于 ocserv 与 FreeIPA 的结合应用将在后续更新。请持续关注公众号更新。