MENU

Debian下搭建Ocserv(openconnect server),并启用证书验证

月杪 | 11 OCT 2015 on Ocserv

安装编译依赖:

```
apt-get install build-essential autogen pkg-config
apt-get install libtalloc-dev libreadline-dev libpam0g-dev libhttp-parser-dev libpcl1-dev
apt-get install libgnutls28-dev libev-dev
apt-get install libprotobuf-c-dev libhttp-parser-dev gnutls-bin
# 0.11.8版本后如果系统也为Debian8可能需要
apt-get install -t jessie-backports libgeoip-dev
# 如果为Debian9则直接
apt-get install libgeoip-dev
# ocserv 0.12.0开始需要添加一个新的依赖(该依赖在0.12.0以前为可选),不然预编译的时候会出警告告诉你worker进程无法
apt-get install libseccomp-dev
```

ocserv编译安装(目前最新版):

```
wget ftp://ftp.infradead.org/pub/ocserv/ocserv-0.12.3.tar.xz
tar Jxvf ocserv-0.12.3.tar.xz
```

```
cd ocserv-0.12.3
./configure --prefix=/usr --sysconfdir=/etc
make && make install
```

在预编译前如果需要ocserv支持更多的路由表需要编辑src/vpn.h(新版本以后已经不需要了):

```
#define DEFAULT_CONFIG_ENTRIES 200 // 默认96,iOS Anyconnect客户端最多支持到200条路由表
```

配置证书:

CA模板，创建ca.tmpl，按需填写，这里的cn和organization可以随便填。

```
cn = "Your CA name"
organization = "Your fancy name"
serial = 1
expiration_days = 3650
ca
signing_key
cert_signing_key
crl_signing_key
```

CA密钥

```
certtool --generate-privkey --outfile ca-key.pem
```

CA证书

```
certtool --generate-self-signed --load-privkey ca-key.pem --template ca.tmpl --outfile ca-cert.pem
```

同理，我们用CA签名，生成服务器证书。先创建server.tmpl模板。这里的cn项必须对应你最终提供服务的hostname或IP，否则AnyConnect客户端将无法正确导入证书。

```
cn = "Your hostname or IP"
organization = "Your fancy name"
expiration_days = 3650
signing_key
encryption_key
tls_www_server
```

Server密钥

```
certtool --generate-privkey --outfile server-key.pem
```

Server证书

```
certtool --generate-certificate --load-privkey server-key.pem --load-ca-certificate ca-cert.pem --load
```

将CA，Server证书与密钥复制到以下文件夹

```
cp ca-cert.pem /etc/ssl/certs/my-ca-cert.pem
cp server-cert.pem /etc/ssl/certs/my-server-cert.pem
cp server-key.pem /etc/ssl/private/my-server-key.pem
```

生成证书认证需要的客户端证书

######创建user.tmpl

```
cn = "some random name"
unit = "some random unit"
expiration_days = 365
signing_key
tls_www_client
```

User密钥

```
certtool --generate-privkey --outfile user-key.pem
```

User证书

```
certtool --generate-certificate --load-privkey user-key.pem --load-ca-certificate ca-cert.pem --load-c
```

```
openssl pkcs12 -export -inkey user-key.pem -in user-cert.pem -certfile ca-cert.pem -out user.p12
```

将生成的客户端证书拷贝到可以在线下载的地址或者其他你可以导入到客户端的地方,比如我直接放到nginx的默认目录下以方便直接下载

```
cp user.p12 /var/www/html
```

ocserv配置文件

配置文件放置在 `/etc/ocserv/ocserv.conf` 以下只保留重要内容

```
auth = "certificate"

tcp-port = 443 #端口可自定义,如果ISP对udp限制较高可尝试注释掉udp端口
udp-port = 443

# Keepalive in seconds
keepalive = 32400

# MTU discovery (DPD must be enabled)
try-mtu-discovery = true

cookie-timeout = 86400

mobile-dpd = 1800

server-cert = /etc/ssl/certs/my-server-cert.pem
server-key = /etc/ssl/private/my-server-key.pem
ca-cert = /etc/ssl/certs/my-ca-cert.pem

cert-user-oid = 2.5.4.3

dns = 8.8.8.8
```

网络

```
iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE # 如果是ovz的服务器网卡应该为venet0
iptables -I INPUT -p tcp --dport 443 -j ACCEPT # 端口应与ocserv配置中配置的端口对应
iptables -I INPUT -p udp --dport 443 -j ACCEPT
```

现在直接使用

```
ocserv
```

就可以启用服务,需要查看日志的话可以使用

```
ocserv -f -d 1
```

在iOS端安装Anyconnect导入证书就可以了

月杪

在大学教过书,做过游戏,iOS应用,虚幻4.只想安静的当一条咸鱼.

Chongqing, China *moonagic.com*

Share this post

READ THIS NEXT

Debian下搭建Shadowvpn服务端

Shadowvpn衍生自libsodium,主要是为低端硬件编写的,比如一些路由器. 但是也能当做vps之间的传输工具(比如国内跳板?) 而Github上的项目更新到2.0后安装说明没有得到及时更新...前几天按照旧的说明始终不行 目前的安装流程是这样的: ##### 安装编译依赖 ```bash apt-get install build-essential automake libtool git ``` ##### 从github得到源码并安装 ```bash git clone https...

YOU MIGHT ENJOY

解决Chrome系统菜单字体无法禁用DirectWrite的问题

Chrome从37开始支持DirectWrite,但是在低分屏下显示效果反而更差,并且与mactype冲突. 当时的解决办法为在flags中禁用DirectWrite,也确实解决了问题,但是似乎从

Chrome42以后这个方法不再适用于Chrome的系统菜单部分. 在忍受了很久之后终于得知解决办法,添加Chrome启动参数 --disable-directwrite-for-ui