## 实验2 Windows下SSL VPN实验

### 【实验目的】

1. 掌握SSL vpn连接步骤；
2. 掌握SSL vpn的连接原理。

### 【实验内容】

1. 配置SSL VPN 服务端；
2. 配置客户端进行连接。

### 【实验原理】

OpenVPN允许参与建立VPN的单点使用预设的私钥，第三方证书，或者用户名/密码来进行身份验证。它大量使用了OpenSSL加密库，以及SSLv3/TLSv1协议。OpenVPN能在Linux、xBSD、Mac OS X与Windows 2000/XP上运行。它并不是一个基于Web的VPN软件，也不与IPSec及其它VPN软件包兼容。

一、 加密

OpenVPN使用OpenSSL库加密数据与控制信息：它使用了OpesSSL的加密以及验证功能，意味着，它能够使用任何OpenSSL支持的算法。它提供了可选的数据包HMAC功能以提高连接的安全性。此外，OpenSSL的硬件加速也能提高它的性能。

二、 验证

OpenVPN提供了多种身份验证方式，用以确认参与连接双方的身份，包括：预享私钥，第三方证书以及用户名/密码组合。预享密钥最为简单，但同时它只能用于建立点对点的VPN；基于PKI的第三方证书提供了最完善的功能，但是需要额外的精力去维护一个PKI证书体系。OpenVPN2.0后引入了用户名/口令组合的身份验证方式，它可以省略客户端证书，但是仍有一份服务器证书需要被用作加密。

三、 网络

OpenVPN所有的通信都基于一个单一的IP端口，默认且推荐使用UDP协议通讯，同时TCP也被支持。OpenVPN连接能通过大多数的代理服务器，并且能够在NAT的环境中很好地工作。服务端具有向客户端“推送”某些网络配置信息的功能，这些信息包括：IP地址、路由设置等。OpenVPN提供了两种虚拟网络接口：通用Tun/Tap驱动，通过它们，可以建立三层IP隧道，或者虚拟二层以太网，后者可以传送任何类型的二层以太网络数据。传送的数据可通过LZO算法压缩。IANA(Internet Assigned Numbers Authority)指定给OpenVPN的官方端口为1194。OpenVPN 2.0以后版本每个进程可以同时管理数个并发的隧道。OpenVPN使用通用网络协议(TCP与UDP)的特点使它成为IPSec等协议的理想替代，尤其是在ISP(Internet service provider)过滤某些特定VPN协议的情况下。

在选择协议时，需要注意2个加密隧道之间的网络状况，如有高延迟或者丢包较多的情况下，请选择TCP协议作为底层协议，UDP协议由于存在无连接和重传机制，导致要隧道上层的协议进行重传，效率非常低下。

### 【实验步骤】

**步骤一、环境搭建**

Windows下SSL VPN的网络拓扑如图2-1所示，其中：

客户端：本地主机(Windows XP)，IP地址172.22.1.X

服务端：Windows实验台VPN服务器，IP地址：172.22.X.X/16，内网IP为172.20.X.X/16



图2-1 实验网络拓扑图

根据实验环境中的拓扑图，配置服务器(Windows实验台)与客户端(本地主机)的IP地址。

**步骤二、 安装与配置**

这一部分是服务端跟客户端都要做的工作，操作完全相同。具体如下：

双击 openvpn-2.0.9.exe进行安装，点击NEXT、I Agree、NEXT之后开始选择安装路径，手动修改为C:\Program Files\OpenVPN 。点击 Install 开始安装，安装过程如图2-2所示；安装过程中，弹出硬件安装窗口，点击仍然继续，安装虚拟网卡。点击 next、Finish 完成安装。

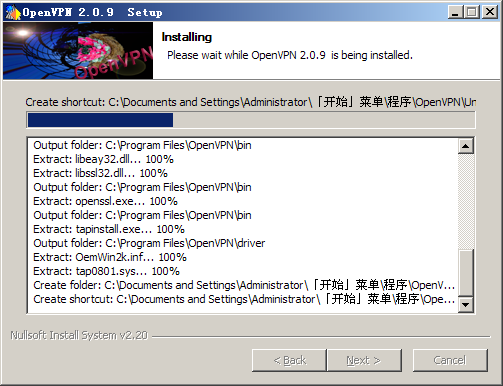


图2-2 OpenVPN安装界面

**步骤三、VPN服务器初始化配置**

在进行操作之前，首先进行初始化工作：

打开命令提示符：“开始|运行”，键入cmd，回车，进入命令提示符；或者“开始|程序|附件|命令提示符”；

进入C:\Program Files\openvpn\easy-rsa目录下，开始初始化，具体命令如下：

cd C:\Program Files\openvpn\easy-rsa

init-config

vars

clean-all

如图2-3所示。

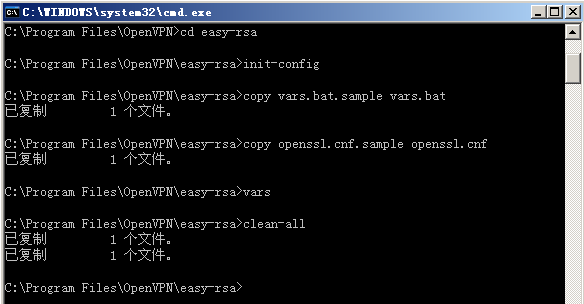


图2-3 初始化界面

上面是初始化工作，以后，在进行证书制作工作时，仍旧需要进行初始化，但只需要进入openvpn\easy-rsa目录，运行vars就可以了，不需要上面那些步骤了。

**步骤四、服务器证书的制作**

1）生成根证书

输入build-ca.bat，如图2-4所示。

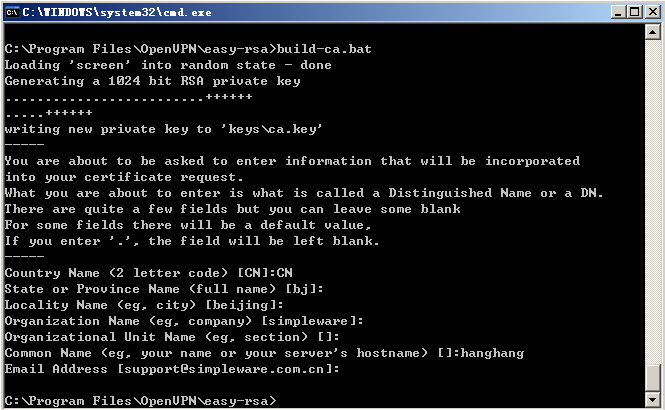


图2-4 生成根证书

2）输入build-dh.bat，如图2-5所示。

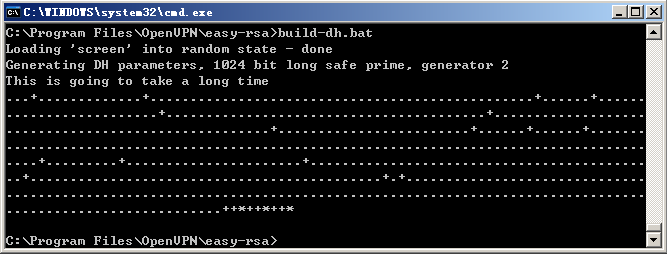


图2-5 生成DH参数

3）生成服务端密钥

输入build-key-server server，生成服务端密钥；生成服务端密钥的过程中，所填写的common name需要与build-ca中所输入的common name名称一致，其余的摁空格选择默认或手动输入皆可；具体如图2-6所示。

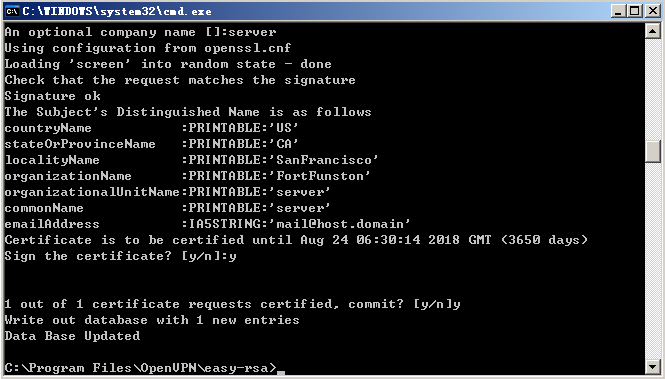


图2-6 生产服务器端密钥

4）生成客户端密钥

输入build-key client1生成第一个VPN客户端密钥，如图2-7所示；

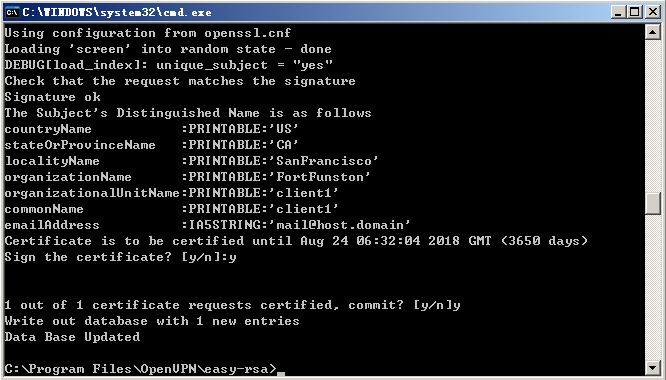


图2-7 生成客户端密钥

build-key client2 //可以继续配置第二个VPN客户端密钥；

生成的密钥存放于C:\Program Files\openvpn\easy\rsa\keys目录下。

如果client密钥生成不成功（生成的client密钥文件为0kb）C:\Program Files\OpenVPN\easy-rsa\openssl.cnf文件中的[policy] = policy\_match修改成[policy] = policy\_anything

**步骤五、配置服务器**

在C:\Program Files\OpenVPN\easy-rsa\keys目录下，将生成的“ca.crt”、“dh1024.pem”、“server.crt”、“server.key”复制到C:\Program Files\OPENVPN\KEY目录下（如果没有可以自己创建）,这四个文件是VPN服务端运行所需要的文件。

|  |
| --- |
| 注：“ca.crt”“dh1024.pem”“server.crt”“server.key”这四个文件是VPN服务端运行所需要的文件。“ca.crt”“client.crt”“client.key”是VPN客户端所需要的文件 |

在C:\Program Files\OpenVPN\config目录下创建server.ovpn，服务器端文件(server.ovpn)示例：

|  |
| --- |
| local 172.22.1.X #建立VPN的IP  port 443 #端口号，根据需要，自行修改，如果是用http代理连接，请不要修改  proto tcp-server #通过TCP协议连接  dev tap #win下必须设为tap  server 172.20.0.0 255.255.0.0 # 虚拟局域网网段设置，请根据需要自行修改，不支持和拔号网卡位于同一网段  push "route 0.0.0.0 0.0.0.0" #表示client通过VPN SERVER上网  keepalive 20 180  ca "C:\\Program Files\\OPENVPN\\KEY\\ca.crt" #CA证书存放位置，请根据实际情况自行修改  cert "C:\\Program Files\\OPENVPN\\KEY\\server.crt" #[服务](http://www.3800hk.com/)器证书存放位置，请根据实际情况自行修改  key "C:\\Program Files\\OPENVPN\\KEY\\server.key" #[服务](http://www.3800hk.com/)器密钥存放位置，请根据实际情况自行修改  dh "C:\\Program Files\\OPENVPN\\KEY\\dh1024.pem" #dh1024.pem存放位置，请根据实际情况自行修改  push "redirect-gateway def1"  push "dhcp-option DNS 219.141.140.10" #DNS，请根据实际情况自行修改  mode server  tls-server  status "C:\\Program Files\\OPENVPN\\log\\openvpn-status.log" #LOG记录文件存放位置，请根据实际情况自行修改  comp-lzo  verb 4 |

**步骤六、配置客户端**

“ca.crt”“client.crt”“client.key”是VPN客户端所需要的文件，复制到客户端C:\Program Files\OPENVPN\KEY目录下（如果没有可以自己创建）。

在客户端安装完成之后，需要将 ca.crt client1.crt client1.key 这三个文件拷贝到C:\Program Files\openvpn\key目录下，这三个文件由[服务](http://www.3800hk.com/)端生成，所以，连接谁的[服务](http://www.3800hk.com/)器，就需要跟谁索取这三个文件。

然后，编辑一个 client.ovpn的配置文件存放到C:\Program Files\openvpn\config目录下，客户端就可以进行连接了；客户端文件(client.ovpn)示例：

|  |
| --- |
| client  dev tap #windows下面用tap,LINUX下用tun  proto tcp-client  remote 172.22.1.X 443 #VPN服务器的域名或IP 端口  resolv-retry infinite  nobind  mute-replay-warnings  ca "C:\\Program Files\\OPENVPN\\KEY\\ca.crt"  cert "C:\\Program Files\\OPENVPN\\KEY\\client1.crt" #这里改成每个客户端相应的证书  key "C:\\Program Files\\OPENVPN\\KEY\\client1.key" #这里改成每个客户端相应的密钥  comp-lzo  verb 4  status openvpn-status.log |

**步骤七、VPN服务端命令行启动**

Openvpn.exe "C:\Program Files\OpenVPN\config\server.ovpn" //启动VPN到443端口

**步骤八、VPN客户端命令行连接**

Openvpn.exe "C:\Program Files\OpenVPN\config\client.ovpn"

**步骤九、VPN安全性验证**

上面的配置拔号成功后，VPN SERVER的IP为172.20.1.Y，VPN client的IP为172.20.1.Y。

1）在VPN client上ping VPN SERVER；

2）分别抓取真实网卡与虚拟网卡的数据包作对比。

具体结果如下：

1）真实网卡：抓取的为加密ssl数据包(图2-8为实际环境举例)。

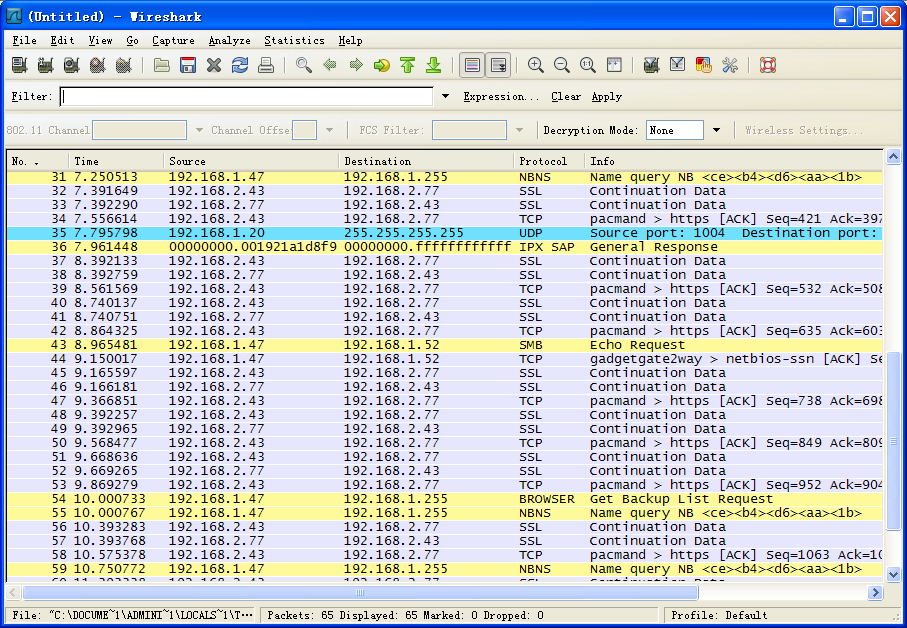


图2-8 真实网卡抓包

2）虚拟网卡：虚拟网卡显示为的ping命令所使用的协议，如图2-9所示。

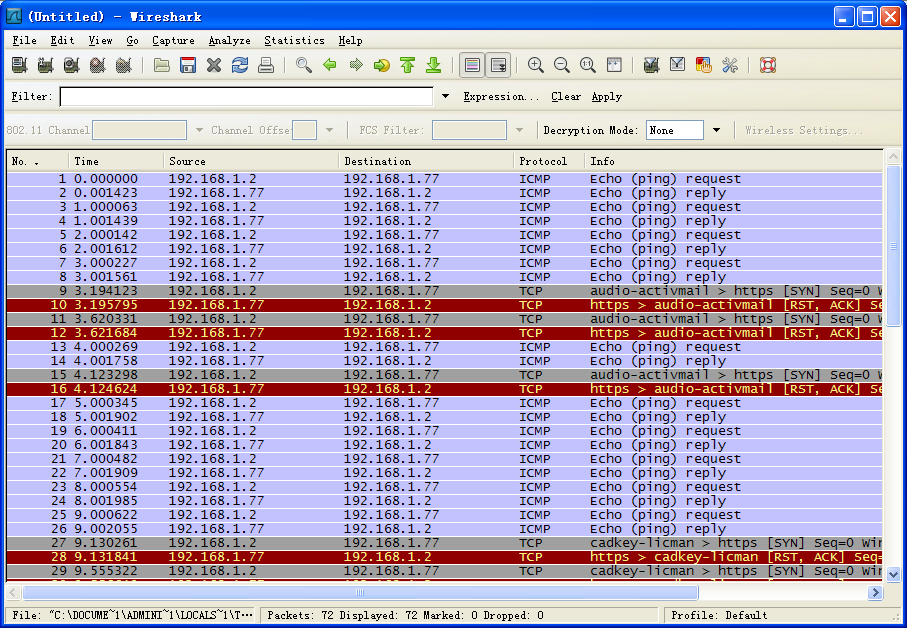


图2-9 虚拟网卡抓包

### 【实验报告】

1. 说明实验过程。
2. 进行结果分析。