互联网链路利用VPN加密技术传输视频数据的项目研究记录

1. 前言：系统规划

项目分阶段实施，并且把该项目打造成一个网络系统，用于传输视频等数据资源

1. 安装公网服务器，并且安装VPN软件，当做服务器（Windows Server 2008，Ubuntu 14.04 或者是CentOS6.7）

2. 声称定制的证书和密钥，在互联网中任意一台电脑，安装客户端，测试网络连通性。

3. 搭建多个VPN客户端实现连通性能测试

4. 采用NAT转换机制，实现虚拟摄像机的实现原理（Windows客户端实现），可以选择海康NVR，实现网络摄像机的传输。

5. 采用NAT转换机制，将模拟视频接入海康DVN设备，实现对模拟视频的传输

6. 采用NAT原理，在linux客户端上实现虚拟摄像机访问，并且传输图像到VPNServer

7. 采用NAT原理，服务器上安装流媒体服务器，允许多个客户端通过流媒体访问每一台摄像机的视频数据（平板电脑）

8. 利用Android OpenVPN软件（需要翻墙）安装在制定的手机上，实现链路的传输

9. 在Android和服务器上开发制定的配套软件，上传图片、录像数据，和实时观看视频数据

10. 研究OpenWRT系统，学习编译，并且将其刷到制定的硬件上面，实现数据通信，并且在OpenWRT编译参数上，如何与当前的4G网络相结合。（隐藏SSID原理，必须）

11. 关于NAT转换的三种方案：笔记本、mini盒子（安装定制的linux操作系统）、采用OpenWRT结合OpenVPN使用

12. 关于大功率（定向）天线，采用VPN数据通道进行加密，传输视频数据。

1. 关于OpenVPN专用数据通道的安全性介绍

摘抄wiki 论文中的数据 论证VPN网络数据的安全性

结合研究院的项目，接入摄像机的原理，对于其进行分析，原理一样，只允许专用端口的数据通过

我们租用电信的数据链路，实质上也是采用的公网物理链路，通过VPN技术，形成一条虚拟的专用通道。因此，如果以公网接入，可以节省费用？由我们自身搭建VPN链路

1. 关于OpenVPN加密机制在实际业务工作中的使用规范

在业务工作中，每在业务中使用一次，我们便更换一次证书，包括服务器端证书和客户端证书，并且每一个客户端对应唯一的证书。

1. 项目简介

利用互联网传输链路，结合目前网络传输链路最安全的VPN技术，在互联网链路上搭建虚拟专用通道，并且配合目前尚未被破解的加密技术，进行视频的传输，以充分利用互联网链路和加密技术，省去了视频传输中搭建租用电信专线传输的限制和高费用。

1. 实验记录
   1. 实验总目的

实验搭建VPN数据传输链路，实现网络的加密状态下的互联

利用IP，采取开源的VPN软件编译生成安装包，搭建好服务器，实现从任意接入公网的电脑采用VPN技术，实现客户端到服务器的通信技术

公网VPN服务器作为核心，实现接入公网的计算机客户端都通过VPNServer实现互联。

在linux（Ubuntu 14.04LTS 或者是CentOS 7 ）主机上面搭搭建VPNServer，并且实现Linux客户端连接VPNServer。

以上部分实现的是client到Server的连接，下面计划搭建局域网之间的VPN链路，可以采用NAT技术，或者是结合路由器，我们在前端不可能布置一台笔记本，所以引出了专门针对OpenVPN的定制路由器。

* 1. 实施日志记录（实验记录）
     1. VPN服务器搭建 2016.08.01

主要目的：Windows7 上安装OpenVPN，作为VPN服务器，Windows客户端也安装，实现局域网的VPN通信。

实验环境：

服务器：Windows7 64位（台式机） + OpenVPN 2.3.10 （192.168.1.101/24）

客户端：Windows7 64位（笔记本） + OpenVPN 2.3.10 （192.168.1.105/24）

网络环境：两台笔记本连接在同一个交换机上，在原有的网络基础上，可以实现互通。

实验结果：

安装软件之后，配置生成证书、key之类的配置文件，在Windows服务器上启动服务器，之后客户端使用对应的配置文件，启动连接服务器，实现通信。

* + 1. 公网VPN linux服务器 2016.08.02

实验目的：通过固定IP，将对应的端口映射到上面的OpenVPN服务器，将笔记本客户端接入互联网，实现从互联网通过VPN技术接入VPN服务器。

实验结果：在路由器上做一个简单的端口映射即可实现

实验目的：将上述环境全部使用Linux进行搭建

实验环境：

服务器：基于VMWare的Ubuntu14.04 x64

客户端：Ubuntu14.04 x64 客户端

实验过程：

服务器端：编译安装openvpn

sudo apt-get install libssl-dev liblzo2-dev libpam0g-dev

./configure

make & make install

选择配置文件

* + 1. NAT虚拟摄像机访问实现（Windows客户端版本） 2016.08.03

实验目的：客户端实现NAT转化，将网络摄像机接入VPN专用网络

实验环境：同上面一样

客户端Ubuntu14.04 192.168.0.x windows操作系统

摄像机等客户端：192.168.0.x

路由器：接入互联网，客户端和摄像机通过路由器接入互联网

客户端和网络摄像机共同接在一个路由器上面，路由器连接互联网

VPN 网络中的虚拟网络摄像机

由于linux客户端的数据NAT映射知识掌握的很少，暂时没有通过iptables实现NAT映射成功（知识漏洞）

把客户端改成Windows客户端，使用简单的命令便实现了端口的映射转换（记录在电脑里），之后看下linux有没有类似简单的命令，实现简单的NAT映射。

*netsh interface portproxy add v4tov4 listenport=11111 listenaddress=10.10.10.10 connectport=80 connectaddress=10.10.10.11*

*netsh interface portproxy del v4tov4 listenport=11111 listenaddress=10.10.10.10*

* + 1. 手机Android客户端VPN连接服务器

手机客户端连接VPN服务器：将配置文件放在Android客户端，并且下载Android OpenVPN软件实现客户端的连接，这里需要将ta.key关掉，因为在客户端没有这个选项。

此部分作为项目的延伸,可以利用VPN技术的，在客户端和服务器上分别开发应用程序，讲照片、录像、实时视频数据传输至服务器，服务器可以查看图片、视频并且观看实时的视频数据。

已生成了Android-OpenVPN.apk软件

* + 1. 关于linux（Ubuntu）实现NAT功能

把VPN的私有网址放访问NAT到局域网特定主机的某一个服务，实现将局域网后面的服务组员的访问。之前在linux上面使用iptables没有实现，今儿改用windows的命令netsh实现，这里我们linux使用rinetd开源软件实现NAT的功能. 实现了NAT功能,并且将摄像机的资源开放给VPN网络中的其他主机.

关于rinetd的用法比较容易掌握，Ubuntu上可以之间安装软件，也可自行编译源代码，在进行安装。不过自动安装之后，会在开机中自行启动rinetd的程序，因为在操作系统中，我们在开机过程中没有启动VPN服务，因此建议结合附录一的内容，删除关于rinetd的开机自启。在未来最终形成的定制操作系统中，可以开机自启。

* + 1. 搭建VPN 网络中的其他客户端之间的互联

今天实验使用华为的路由器出现了功能性故障，无法通过其访问我们的服务器资源，但是可以访问其他的互联网资源。经过一天的排查找到了问题的原因。（坑死人了，昨晚还可实现VPN服务器资源的访问今天就挂了）

之后采用其他网络资源，生成两个客户端client-a client-b，两客户端之间是可以互相通信的，但是数据流量是否通过VPNServer进行中转，还是不得而知。

关于在Ubuntu中安装了rinetd的开源软件，开机自己启动，但是我们的ＶＰＮ服务没有启动，这样是非常的不合理，因此专门学习了Ubuntu中关于开机自启的问题，详情见[附件1 关于Ubuntu开机自启脚本和系统运行级别](ubuntu自启动服务.docx)

* + 1. 关于linux（Ubuntu）中通过ufw+iptables实现NAT功能

实验目的：利用Ubuntu自带的防火墙ufw和iptables实现NAT的功能

实验过程：

开启OpenVPN Server，

客户端使用Ubuntu14.04（局域网ip 192.168.1.103/24 gw 192.168.1.1）

摄像机IPC：192.168.1.64/24 gw 192.168.1.1 后面改成192.168.1.103

客户端和摄像机IPC通过192.168.1.1 连接到互联网。

实验总结：

在Ubuntu中ufw防火墙是在iptables的基础上的集成的，因此如果开启端口转发，必须要打开防火墙ufw，在此基础上配置iptables规则。

首先要讲FORWARD设置为接受，之后再nat表中进行PREROUTING和POSTROUTING的DNAT和SNAT的转发。此时并没有成功，我们在把摄像机的网关设置成为该客户端的IP地址，这样摄像机的所有数据就会发送到本客户端，经过SNAT转换之后，通过VPN网路转发到请求的VPN主机。

在sysctl.conf中设置ip\_forward为1，开机默认转发数据包.

5.2.8 关于Linux（Ubuntu）仅仅iptable实现NAT的功能

实验环境变量：

VPN服务器：10.8.0.1

VPN客户端：10.8.0.6 局域网ip 192.168.1.103

局域网摄像机IPC：192.168.1.64 GW 192.168.1.103尤为重要

卸载iptables & ufw软件，并且重新安装iptables软件

将IPC摄像机的GW设置为Ubuntu 客户端的IP地址

配置iptables

sudo iptables -P INPUT ACCEPT

sudo iptables -P FPRWARD ACCEPT

sudo iptables –t nat -A PREROUTING -d 10.8.0.6 –p tcp --dport 8080 –j DNAT --to-destination 192.168.1.64:80 #浏览器观看

sudo iptables –t nat –A PREROUTING –d 10.8.0.6 –p tcp --dport 8000 –j DNAT --to-destination 192.168.1.64:8000 # ivs客户端可以使用8000添加设备

sudo iptables –t nat –A POSTROUTING –s 192.168.1.64 –p tcp –j SNAT --to-source 10.8.0.6