如何开发一套 VPN
0. 写在前面
1. VPN 的原理
1.1 Linux 服务端
1.2 安卓客户端
1.3 Windows 客户端
2. 服务端开发
2.1 Linux
3. 客户端开发
3.1 安卓
3.2 Windows
4. 使用 VPN 访问内网

如何开发一套 VPN

0. 写在前面

1. 这篇文章讲的是如何**开发**一套VPN, 不是如何使用某个 VPN.
2. 一套, 包括Linux服务端, 安卓客户端, Windows客户端.
3. VPN 一般用于实现访问局域网, 当然也可以有其他用途.
4. VPN 协议特征过于明显, 很容易被识别并封禁, 本文主要是介绍原理, 有兴趣的同学可以基于此优化.
5. 咦? 为什么会有人要识别和封禁 VPN 呢? 我也布吉岛...

1. VPN 的原理

以下会描述得比较通俗, 不太专业, 请专业人士见谅, 也欢迎您指正.

1.1 Linux 服务端

1. 如果Linux有两张网卡, K1 和 K2, 其中 K1 可以上网, K2 不可以.
2. 我们可以设置 Linux 使它将 K2 收到的流量转向 K1, 让 K1 帮把 K2 的流量转发出去, 等到数据回来到 K1 时, K1 再转给 K2.
3. 这就是路由器的工作原理.
4. 我们购买到的 VPS 一般只有一张上网的网卡, 这时, 我们需要创建一张虚拟网卡.
5. VPN 服务程序只需要将从客户端收到的数据写入 K2, 再从 K2 读取回包, 再发回给客户端就行.

1.2 安卓客户端

1. 安卓提供了一个方法, 可以新建一个虚拟网卡, 叫做 tun, 然后让所有 app 的网络连接都发到 tun.
2. 然后让某个 app 可以对发到这张虚拟网卡的流量进行管理.
3. 这个可以管理虚拟网卡流量的 app 就是 VPN 客户端 app.
4. VPN 客户端 app 只需要将从 tun 读取到的数据发向服务端, 等服务端回包时, 再将数据写入 tun 就行.

1.3 Windows 客户端

1. Windows 可以安装第三方虚拟网卡.
2. 我们可以让程序修改路由表, 使得流量转向虚拟.
3. 客户端程序对虚拟网卡进行监听, 读取住转向虚拟网卡的流量.
4. 客户端程序只需要将从虚拟网卡读取到的数据发向服务端, 等服务端回包时, 再将数据写入虚拟网卡就行.

2. 服务端开发

2.1 Linux

以下命令均需要root权限执行.

一. 首先我们要设置 Linux 的 ip_forward. 使它能在不同网卡间转发流量.

```
# 编辑这个文件:
vim /etc/sysctl.conf
# 将文件里的这行取消注释
# net.ipv4.ip_forward = 1

# 然后执行这条命令:
sysctl -p
# 如果有看到 net.ipv4.ip_forward = 1 说明成功了.
```

二. 需要新建一个虚拟网卡, 并设置它的ip, 以及设置将其流量转发到真实网卡. 一般命名为 tun0, tun1, tun2...

```
ip tuntap add tun0 mode tun
ip link set dev tun0 up
ifconfig tun0 192.168.194.224 netmask 255.255.255.0 promisc
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# 第一步有可能遇到如下错误:
# Object "tuntap" is unknown, try "ip help".
# 解决方法: 使用tunctl添加tun
# Unbuntu:
apt-get install uml-utilities bridge-utils
# CentOS:
yum install tunctl brctl
# 执行下面这条命令后继续第二步
tunctl -n -t tun0 -u root
```

三. 关键代码 (使用udp进行服务端和客户端间的数据传输)

```
// 打开虚拟网卡 tun0, 获取文件描述符 tun_fd

int tun_fd;
char *clonedev = "/dev/net/tun";
if ((tun_fd = open(clonedev, O_RDWR)) < 0) {
    printf("error1");
    return -1;
}
struct ifreq ifr;
memset(&ifr, 0, sizeof(ifr));
ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
strncpy(ifr.ifr_name, "tun0", IFNAMSIZ);
int err;
if ((err = ioctl(tun_fd, TUNSETIFF, (void *) &ifr)) < 0) {
    close(tun_fd);
    return err;
}
```

```
uint8_t buf[MAX_PKG_LEN] = { 0 };
struct sockaddr_in* clent_addr;

// 将收到的数据写入 tun0
int n = recvfrom(udp_fd, buf, MAX_PKG_LEN, 0, (struct sockaddr*)clent_ad
write(tun_fd, buf, n);

// 读取 tun0 收到的回包并发回客户端
int n = read(tun_fd, buf, MAX_PKG_LEN);
sendto(udp_fd, buf, n, 0, clent_addr, sizeof(struct sockaddr_in));
```

四. 完整代码

vpn-server-demo

代码完全使用 C 语言编程.

使用 epoll 进行文件描述符管理.

为了方便演示没有对数据进行加密, 在实际使用中必须加密!

3. 客户端开发

3.1 安卓

一. 关键代码:

```
//请求打开vpn
Intent intent = VpnService.prepare(MainActivity.this);
if (intent != null) {
    startActivityForResult(intent, 0);
} else {
    onActivityResult(0, RESULT_OK, null);
}

//用户同意后,开启vpn服务
@SuppressLint("ResourceAsColor")
protected void onActivityResult(int request, int result, Intent data) {
    super.onActivityResult(request, result, data);
    if (result == RESULT_OK) {
        Intent intent = new Intent(this, MyVpnService.class);
        startService(intent);
    }
}
```

```
import android.net.VpnService;
public class MyVpnService extends VpnService { ... }

static String SERVER_ADDR = "192.168.1.169";
static int SERVER_PORT = 7194;
static int MAX_BKG_LEN = 65535;

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Builder builder = new Builder();
    builder.setSession("MyVPNService");
    builder.addAddress("192.168.194.1", 24); //这个ip要和服务器的虚拟网卡ip在
    builder.addDnsServer("8.8.8.8");
    builder.addRoute("0.0.0.0", 0);

    static ParcelFileDescriptor mInterface = builder.establish();
    FileInputStream in = new FileInputStream(mInterface.getFileDescripto
    FileOutputStream out = new FileOutputStream(mInterface.getFileDescri

    InetAddress serverAddr = InetAddress.getByName(SERVER_ADDR);
    DatagramSocket sock = new DatagramSocket();
    sock.setSoTimeout(0); //超时为无穷大
    protect(sock); //保护这个连接的数据不会进入虚拟网卡

    //启动两个线程一收一发

    return START_STICKY
}

//发线程
@Override
public void run() {
    int length;
    byte[] ip_pkg = new byte[MAX_BKG_LEN];
    while ((length = in.read(ip_pkg)) >= 0) {
        if (length == 0) {
            continue;
        }
        DatagramPacket msg = new DatagramPacket(
            ip_pkg, length, serverAddr, SERVER_PORT);
        sock.send(msg);
    }
    in.close();
}

//收线程
@Override
public void run() {
    byte[] ip_buf = new byte[MAX_BKG_LEN];
    while (true) {
        DatagramPacket msg_r = new DatagramPacket(
            ip_buf, MAX_BKG_LEN, serverAddr, SERVER_PORT);
        sock.receive(msg_r);
        int pkg_len = msg_r.getLength();
        if (pkg_len == 0) {
            continue;
        } else if (pkg_len < 0) {
            break;
        }
        out.write(ip_buf, 0, pkg_len);
    }
    out.close();
}

// 注意: 代码里省略了很多 try catch.
```

二. 完整代码

vpn-client-demo

这是一个完整的安卓项目

3.2 Windows

敬请期待

4. 使用 VPN 访问内网

敬请期待