

OpenVPN + Vagrant应用场景

一半人生 / 2019-09-04 09:10:00 / 浏览数 9443 [安全工具](#) [工具](#) [顶\(0\)](#) [踩\(0\)](#)

这故事得从公司，渗透组因为临时加的小项目，需要搭个渗透综合类的环境，最初就有了这个方案，朋友问怎么整？跨度有点太大了，只因以前吹牛逼，大学学网络，干运维，谷歌还能找到很多OpenVPN的环境部署，国内大多被和谐了。过程中踩了一些坑，分享给圈子，也可以自己搭环境玩一些有意思的东西。

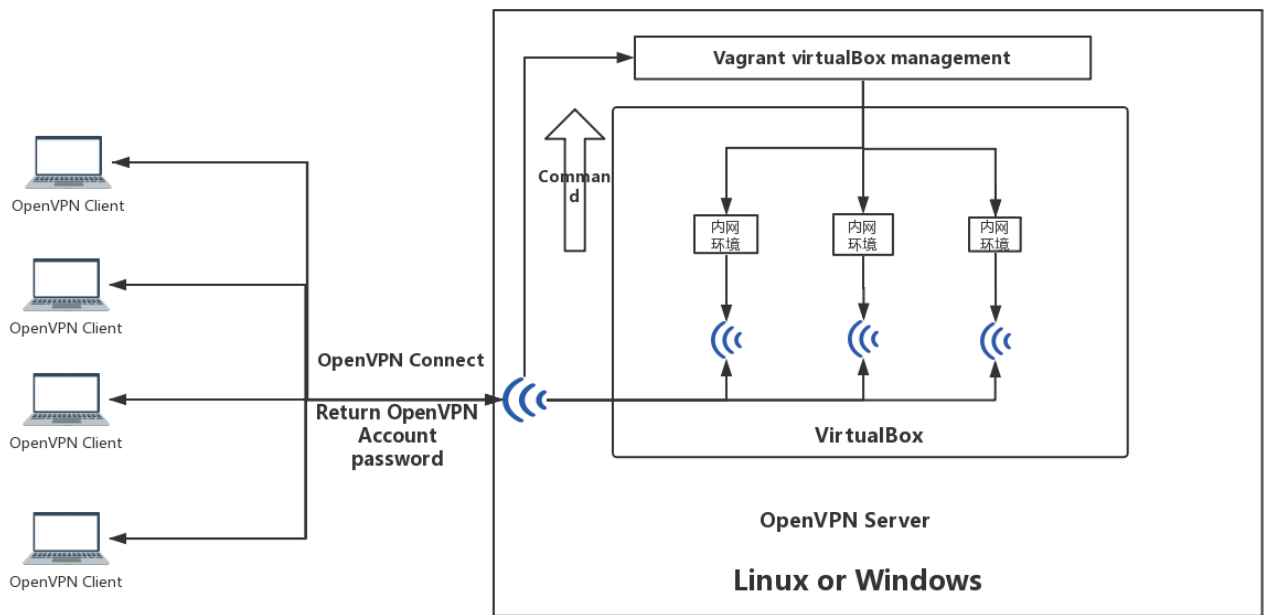
GitHub目前找到最齐全的参考资料：<https://github.com/izombielandgit/OpenVPN-HOWTO/blob/master/1.%20HOWTO.md>

环境如下：

OpenVPNServer: Ubuntu18 Desktop (公网) 假设ip：40.40.40.40 需要装个VirtualBox创建一个虚拟网卡(仅主机模式)：ip：192.168.131.10 gw 192.168.131.1

OpenVPNClient：xxx.xxx.xxx.xxx 注意：不建议在Windows上搭建OpenServer，涉及一些数据包机制转发的会有些麻烦。

为了方便理解这种模式，梳理一副图片，如下所示：



这是部署前梳理流程图，如上图所示，底层是一个Ubuntu or

Windows，客户端利用OpenVPN访问公网，Ubuntu内部开启虚拟容器，开启虚拟机而且仅主机模式。用户想要通过客户端OpenVPN连接内网的虚拟机，可以利用Vagrant

Windows10 OpenVPN部署：

```
1.■■■OpenVpn
2.■■■VPNServer■■■■■
3.Cd Easy-rsa
4.init-config.bat
5.Edit Vars.bat
KEY-COUNTRY = UA
KEY-PROVINCE = Kiev
KEY-CITY = Kiev
KEY- ORG = ServerVPN
6.Cd "c:\Pro OpenVPNPath\easy-rsa"
7.Vars
8.clean-all
9.Build-dh ■■■■■dh4096.pem
10.■■■■■■■■OpenVPN
11.Build-ca CreateCA
12.Build-Key-ServerVPN
13.Build-key ClientVPN
14.Openvpn --genkey -- secret keys/ta.key
15.Cp Server.ovpn --> /$path/OpenVPN/config/Server.ovpn
16.Edit Server.ovpn
17.Edit Client
```

Ubuntu18 OpenVPN部署：

```
wget -P ~/ https://github.com/OpenVPN/easy-r  
mkdir OpenVPNeasy  
cd OpenVPNeasy/  
wget https://github.com/OpenVPN/easy-rsa  
vim easy-rsa // ██████████github  
tar xvf EasyRSA-unix-v3.0.6.tgz  
cd EasyRSA-v3.0.6/  
openvpn --genkey --secret ta.key  
cp ./ta.key /etc/openvpn/  
cp ./pki/ca.crt /etc/openvpn/  
cp ./pki/dh.pem /etc/openvpn/  
cp /usr/share/doc/openvpn/examples/sample-c  
cd /etc/openvpn/  
cd server/  
gzip -d server.conf.gz ███████server.conf
```

```
dev-node "vpn-ada" ██████
mode server
port 12345 ██████
proto tcp4-server ██████
dev tun

tls-server
tls-auth "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\ta.key" 0

tun-mtu 1500
tun-mtu-extra 32
mssfix 1450
```

```
# ██████████xxxxxx██████
push "route 10.10.0.0 255.255.255.0"
push "route 192.168.131.0 255.255.255.0"
```

```
if [ ! -r "${PASSFILE}" ]; then
echo "${TIME_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >> ${LOG_FILE}
exit 1
```

```

fi

CORRECT_PASSWORD=`awk ' !/^;/&&!/^#/&&$1=="'${username}''{print $2;exit}' ${PASSFILE}`

if [ "${CORRECT_PASSWORD}" = "" ]; then
    echo "${TIME_STAMP}: User does not exist: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
    exit 1
fi

if [ "${password}" = "${CORRECT_PASSWORD}" ]; then
    echo "${TIME_STAMP}: Successful authentication: username=\"${username}\"." >> ${LOG_FILE}
    exit 0
fi

echo "${TIME_STAMP}: Incorrect password: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
exit 1

```

```
client

proto tcp
dev tun
# ■■■OpenVPNServer
remote 0.0.0.0 1194
remote-random
resolv-retry infinite

nobind

persist-key
persist-tun

ca ca.crt
# cert gttx-client-vpn.crt
# key gttx-client-vpn.key


auth-user-pass
auth-nocache
remote-cert-tls server
tls-auth ta.key 1
route-method exe

#■■■■■■■■■■■
cipher AES-256-CBC

comp-lzo
status openvpn-status.log
```

当前状态: 连接中

Fri Aug 09 18:52:17 2019 OpenVPN 2.4.7 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [LZ4] [PKCS11] [AEAD]
 Fri Aug 09 18:52:17 2019 Windows version 6.2 (Windows 8 or greater) 64bit
 Fri Aug 09 18:52:17 2019 library versions: OpenSSL 1.1.0i 20 Nov 2018 LZO 2.10


client
×

用户名称:

密码:

☒ 保存密码

OpenVPN GUI 11.13.0.0/2.4.7

断开连接

重新连接

隐藏

C:\Users\Administrator>ping 10.10.0.1

正在 Ping 10.10.0.1 具有 32 字节的数据:
 来自 10.10.0.1 的回复: 字节=32 时间=13ms TTL=64
 来自 10.10.0.1 的回复: 字节=32 时间=13ms TTL=64

 10.10.0.1 的 Ping 统计信息:
 数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),
 往返行程的估计时间(以毫秒为单位):
 最短 = 13ms, 最长 = 13ms, 平均 = 13ms

Control-C

C:\Users\Administrator>ping 10.10.1.1

正在 Ping 10.10.1.1 具有 32 字节的数据:
 来自 10.10.1.1 的回复: 字节=32 时间<1ms TTL=64
 来自 10.10.1.1 的回复: 字节=32 时间<1ms TTL=64

 10.10.1.1 的 Ping 统计信息:
 数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),
 往返行程的估计时间(以毫秒为单位):
 最短 = 0ms, 最长 = 0ms, 平均 = 0ms

Control-C

Ping试一试咋样:

```
C:\Users\Administrator>tracert 192.168.131.1
```

通过最多 30 个跃点跟踪到 192.168.131.1 的路由

```
 1      5 ms      1 ms      1 ms      192.168.1.1
 2      4 ms      3 ms      7 ms      10.10.0.1
 3      7 ms      4 ms      4 ms      221.171.224.194
 4      6 ms      *          6 ms      117.177.136.130
 5      *          *          *          请求超时。
 6 ^C
```

```
C:\Users\Administrator>traacert 10.10.0.1
```

'traacert' 不是内部或外部命令，也不是可运行的程序或批处理文件。

```
C:\Users\Administrator>tracert 10.10.0.1
```

通过最多 30 个跃点跟踪到 10.10.0.1 的路由

```
 1      12 ms      13 ms      14 ms      10.10.0.1
```

跟踪完成。

```
C:\Users\Administrator>tracert 10.10.1.1
```

通过最多 30 个跃点跟踪到 SD-20190122QQUB [10.10.1.1] 的路由：

```
 1      <1 毫秒      <1 毫秒      <1 毫秒      SD-20190122QQUB [10.10.1.1]
```

发现问题，起码OpenVPN这条线是没问题，这时候你要考虑如何转发呢？Server端如何做，iptables足够了，所以说利用Iptables做ip转发，简单如下：

```
/etc/sysctl.conf  net.ipv4.ip_forward=1
sysctl -p
```

```
Iptables -F -t nat
```

```
# Completed on Fri Aug 9 00:34:00 2019
```

```
# Generated by iptables-save v1.6.1 on Fri Aug 9 00:50:54 2019
```

```
*nat
```

```
:PREROUTING ACCEPT [121:8996]
```

```
:INPUT ACCEPT [95:7251]
```

```
:OUTPUT ACCEPT [30:2365]
```

```
:POSTROUTING ACCEPT [30:2365]
```

```
-A POSTROUTING -s 10.10.1.0/24 -j SNAT --to-source 10.66.0.1
```

```
-A POSTROUTING -s 10.10.1.0/24 -j SNAT --to-source 192.168.131.0
```

```
COMMIT
```

```
# Completed on Fri Aug 9 00:50:54 2019
```

```
# Generated by iptables-save v1.6.1 on Fri Aug 9 00:50:54 2019
```

```
*filter
```

```
:INPUT ACCEPT [605:70173]
```

```
:FORWARD DROP [24:1456]
```

```
:OUTPUT ACCEPT [371:31256]
```

```
-A INPUT -p tcp -m tcp --dport 12345 -j ACCEPT
```

```
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
-A FORWARD -p tcp -m tcp --dport 12345 -j ACCEPT
```

```
-A FORWARD -s 10.10.1.0/24 -d 192.168.131.0/24 -i tun0 -j ACCEPT
```

```
-A OUTPUT -p tcp -m tcp --sport 12345 -j ACCEPT
```

```
COMMIT
```

```
# Completed on Fri Aug 9 00:50:54 2019
```

上述很快就布置完成，那就在客户端ping，192.168.131.xx，发现不通.....，因为一开始Server是Windows，排查错误没有想着客户端排查，折腾许就在服务器端抓包发现问题，所以调整思路，客户端抓包，果真路由的问题，因为客户端的网段也在192.168.XX.XX，下一跳的过程无法识别到底走那个网卡，按照本机网卡去跑的，客户端主机网卡去转

```
C:\Users\Administrator>tracert 192.168.131.1
```

通过最多 30 个跃点跟踪到 192.168.131.1 的路由

跃点	源 IP	估计源 RTT	估计中间 RTT	估计目标 RTT	目标 IP
1		4 ms	1 ms	1 ms	192.168.1.1
2		42 ms	3 ms	3 ms	192.168.1.1
3		45 ms	38 ms	41 ms	192.168.1.1
4	*	*	*	*	请求超时。
5	*	*	*	*	请求超时。
6	*	*	*	*	请求超时。

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Administrator>tracert 10.10.1.1
```

通过最多 30 个跃点跟踪到 SD-20190122QQUB [10.10.1.1] 的路由:

跃点	源 IP	估计源 RTT	估计中间 RTT	估计目标 RTT	目标 IP
1		<1 毫秒	<1 毫秒	<1 毫秒	SD-20190122QQUB [10.10.1.1]

跟踪完成。



这明显就有问题，那么发现了问题，看一下OpenVPNClient网卡：

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : TAP-Windows Adapter V9
物理地址. . . . . : 00-FF-20-AC-72-EB
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
本地链接 IPv6 地址. . . . . : fe80::dd09:ae2e:ae6e:cf58%4(首选)
IPv4 地址 . . . . . : 10.10.1.1(首选)
子网掩码 . . . . . : 255.255.255.252
获得租约的时间 . . . . . : 2019年8月31日 星期六 7:07:08
租约过期的时间 . . . . . : 2020年8月30日 星期日 7:07:07
默认网关. . . . . :
DHCP 服务器 . . . . . : 10.10.1.2
DHCPv6 IAID . . . . . : 67174176
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-23-E2-C5-95-50-7B-9D-1C-80-A9
DNS 服务器 . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
TCPIP 上的 NetBIOS . . . . . : 已启用
```



客户端局域网Ip如下，所以ping

192.168.131.1的时候，走的是本机这跳网络，没有走OpenVPN，我们需要添加一条路由由很明确的告诉他，如何在二层、三层走：

无线局域网适配器 WLAN:

```
连接特定的 DNS 后缀 . . . . . :  
描述. . . . . : Intel(R) Dual Band Wireless-AC 3160  
物理地址. . . . . : -83-87-30-70  
DHCP 已启用 . . . . . : 是  
自动配置已启用. . . . . : 是  
本地链接 IPv6 地址. . . . . : fe80::2896:eecf:14fd:932b%8(首选)  
IPv4 地址 . . . . . : 192.168.1.83(首选)  
子网掩码 . . . . . : 255.255.255.0  
获得租约的时间 . . . . . : 2018/5/8 11:25:14
```

先知社区

```
route add 192.168.131.0 mask 255.255.255.0 10.10.1.2
```

```
C:\WINDOWS\system32>route add 192.168.131.0 mask 255.255.255.0 10.10.1.2  
操作完成!
```

```
C:\WINDOWS\system32>ping 192.168.131.1
```

```
正在 Ping 192.168.131.1 具有 32 字节的数据:  
来自 192.168.131.1 的回复: 字节=32 时间=14ms TTL=64  
来自 192.168.131.1 的回复: 字节=32 时间=12ms TTL=64  
来自 192.168.131.1 的回复: 字节=32 时间=12ms TTL=64  
来自 192.168.131.1 的回复: 字节=32 时间=18ms TTL=64  
  
192.168.131.1 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 12ms, 最长 = 18ms, 平均 = 14ms
```

```
C:\WINDOWS\system32>tracert 192.168.131.1  
  
通过最多 30 个跃点跟踪到 192.168.131.1 的路由  
  
 1      11 ms      11 ms      10 ms  192.168.131.1
```

跟踪完成。

先知社区

Windows下Vagrant安装:

```
Vagrant■■■■https://www.vagrantup.com/downloads.html
```

```
■■■■■  
box■■■  
■■■■vagrant box  
Usage: vagrant box <subcommand> [<args>]
```

```
Available subcommands:  
  Add          vagrant box add [options] <name, url, or path>  
  list  
  outdated  
  prune  
  remove  
  Repackage  
Update  
pull■■■■■■■■■■https://app.vagrantup.com/boxes/search
```

```
■■■pull■■■■Vagrant box add https://app.vagrantup.com/generic/boxes/ubuntu1604
```

```
vagrant up  
■■■■■■■■BOX■■■■■■■■init  
Vagrant init name
```

```
■■■■■  
vagrant status
```



```

#####
vagrant halt

#####
Vagrant destroy [name|id]

Vagrant#####api#####curl#####Ruby#####Python#####curl#####.
★ #####
    vagrant plugin install vagrant-scp
    Vagrant global-status
# #####
    vagrant scp /home/vincent/backend/go-dev/proxy-v default:~

★ #####
#####
    vagrant snapshot save your_snapshot_name
#####
    vagrant snapshot list
#####
    vagrant snapshot restore your_snapshot_name
#####
    vagrant snapshot delete your_snapshot_name

```

Vagrant只支持BOX镜像，也就是说Vagrant镜像，封装好的，如何去打包自己的镜像的？

```

1.#####
#####VboxManage list vms

```

```

root@ubuntu:~# vboxmanage list vms
"zus" {65cbaa43-9555-4b27-a759-398ffc7eff55}
"zeus2019_default_1565399808459_52019" {ec883686-1327-47fd-9d31-a23e1701184d}
"node1" {cee0faaf-6162-49df-8649-5f0fb82597d2}
"total_default_1565754382541_99721" {a22f95fd-fe75-4645-b9e3-9a406313201c}
"centos" {ba1884cb-c0cd-44a3-9931-c596f62e2513}
"ubuntu-1" {21c9079d-d103-4c94-b774-de31a26337de}

```

先知社区

```

2.#####
#####vagrant package --base "#####" --output #####os#####
3.#####Compressing package to : //#####
4.vagrant box add zus2019 D:\virtualbox-1\boxtest\zus2019.box #####.box

```

收集了一些常用的Vagrant指令，在这也分享一下：

```

1.vagrant box list #box#####
2.#####box
    vagrant box add (box_name) (file_path)
    ######box box_name #####box##### file_path #####
    vagrant box add (vagrant_box)
    ######box vagrant box#####vagrant box#####box#####
    #vagrant box add laravel/homestead
    #vagrant box add laravel/homestead --box-version=0.4.3
    ######
3.vagrant init (box_name)
    ###### box_name #####box#####
4.vagrant up ######
5.vagrant ssh #ssh#####
6.vagrant halt ######
7.vagrant reload ######
8.vagrant destroy ######
9.vagrant suspend ######
10.vagrant status ######
11.vagrant box remove (boxname) ######box#####
12.vagrant package ######
13.vagrant resume ######

```

上述内容虽然看起来部署挺快的，对于没有部署过的人来说还是非常耗时间与精力.....，环境搭建不容易，且搭且珍惜。

点击收藏 | 1 关注 | 1

[上一篇：【实战3】记一次内网中反弹shell...](#) [下一篇：传统XSS攻击引发持久型ATO漏洞...](#)

1. 2 条回复



[zzzhhh](#) 2019-09-08 02:55:04

感谢分享，1.%20HOWTO.md好全。最近搭建OpenV皮恩，疑惑为什么本地客户端连接了OpenV皮恩，就无法浏览网页了。

0 回复Ta



[一半人生](#) 2019-09-12 18:19:09

[@zzzhhh](#) 这个得根据实际情况，按照上述思路可疑追踪一下，看看是不是连接vpn后本地ping www.baidu.com 跳得节点有问题，可以根据本地路由加route更正就好。

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)