Lans behind OpenVPN

here is an example of how to have multiple lans behind OpenVPN from #OpenVPN on freenode.

Our user had a openvpn server with a lan (10.10.2.0/24) behind it, and 2 client with lans behind them:

client1 with lan 10.10.1.0/24
client2 with lan 10.10.3.0/24

He wanted machines on all 3 lans to be able to communicate using a tun (routed) setup.

Every machine with a LAN behind it must have IP forwarding enabled. In this example that means the server, and client1/client2. The user needed the following in his server.conf:

```
route 10.10.1.0 255.255.255.0
route 10.10.3.0 255.255.255.0
push "route 10.10.2.0 255.255.255.0"
push "route 10.10.1.0 255.255.255.0"
push "route 10.10.3.0 255.255.255.0"
client-to-client
```

The route entries adjust the local routing table, telling it to route those networks over the vpn. The push routes are added on the clients connecting, telling them to route those networks over the vpn.

You may realize that client1 should not route 10.10.1.0 traffic over the vpn, and that client2 should not route 10.10.3.0 traffic over the vpn (because those networks are local to each client). Because of the iroute entries you will see below, openvpn knows this too and skips the push for the client.

The route entries are telling his server to add a route for each of 10.10.1.0, and 10.10.3.0 to its kernel's routing table, and both will be routed to the tunnel interface and to openvpn. How will openvpn know what client to send each network to? The answer is iroute!

Iroute does not bypass or alter the kernel's routing table, it allows openvpn to know it should handle the routing when the kernel points to it but the network is not one that openvpn knows about. The iroute entry tells the openvpn server which client is responsible for the network. Without the iroute entry you will find the following in your logfiles:

```
MULTI: bad source address from client [IP ADDRESS], packet dropped
```

IP ADDRESS in that case would be the machine on client LAN which tried to talk through vpn, because openVPN has no clue what that address is. Once you give it the iroute statement, that changes. Iroute is a route internal to openVPN, and has nothing to do with the kernel's routing table. It tells the openvpn server which client owns which network. Note that even if you only have 1 lan behind 1 client, YOU STILL NEED IROUTE. You will need it any time a clients source IP address is different from the IP given to it by the vpn server.

The thing is, we cant just drop the iroute into server.conf because it would then be used for every client, and iroute is only to tell the server at which client it should send traffic destined for a network that the kernel said should go to the openvpn interface. That is why we add the iroute commands to a ccd entry.

You will need client-config-dir /path/to/ccd/ in your server config file to enable ccd entries. ccd entries are basically included into server.conf, but only for the specified client. You put commands in ccd/client-common-name, and they are only included when the client's common-name matches the name of the file in ccd/.

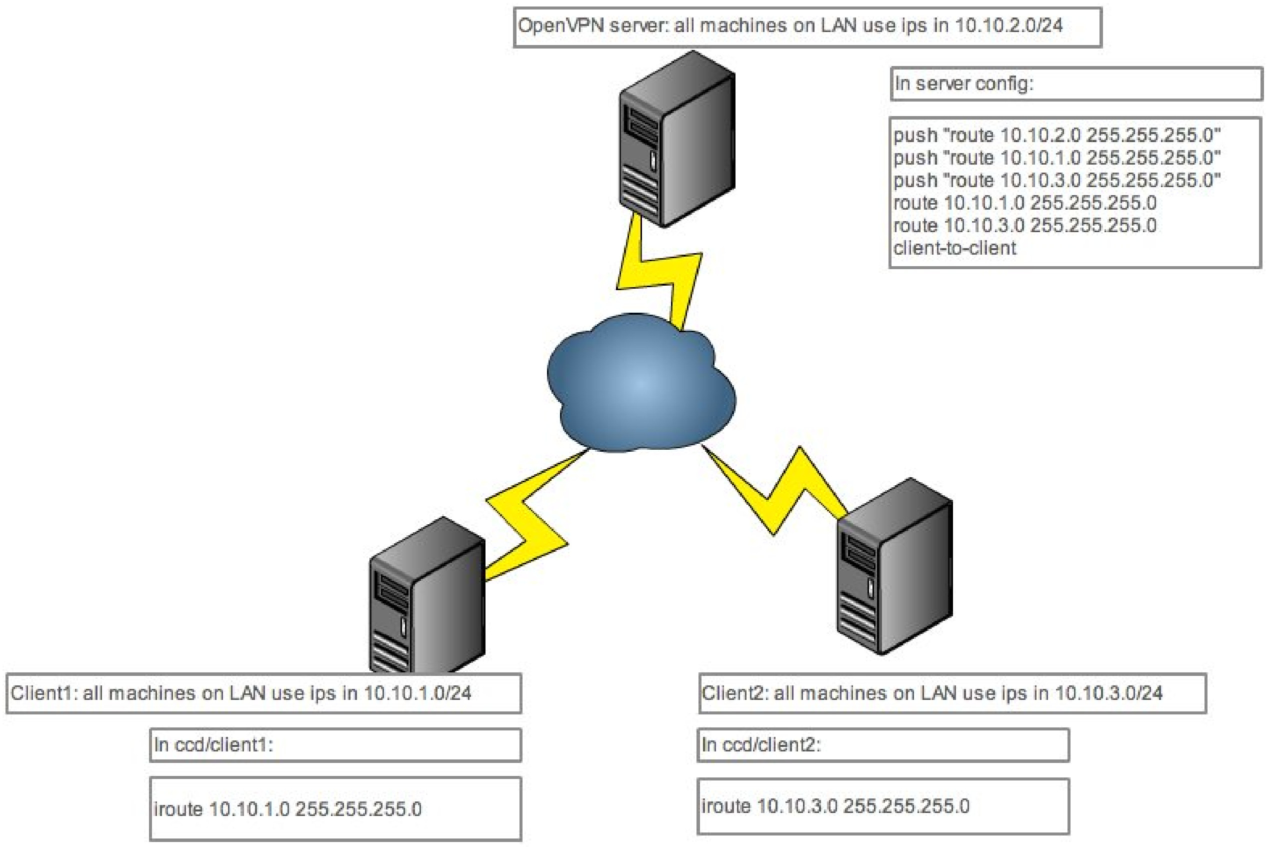
In this example lets assume the client owning the network 10.10.1.0 has a common-name of client1. In ccd/client1 He should have the following:

```
iroute 10.10.1.0 255.255.255.0
```

As you can see our user will make a ccd/ entry for each client with a lan behind it. The ccd entry will have an iroute command for the network behind the client.

That means that client2 on the 10.10.3.0 LAN would have the following entry for its ccd/client2 file:

```
iroute 10.10.3.0 255.255.255.0
```


OpenVPN server: all machines on LAN use ips in 10.10.2.0/24
In server config:
push "route 10.10.2.0 255.255.255.0"
push "route 10.10.1.0 255.255.255.0"
push "route 10.10.3.0 255.255.255.0"
route 10.10.1.0 255.255.255.0
route 10.10.3.0 255.255.255.0
client-to-client
Client1: all machines on LAN use ips in 10.10.1.0/24
In ccd/client1:
iroute 10.10.1.0 255.255.255.0
Client2: all machines on LAN use ips in 10.10.3.0/24
In ccd/client2:
iroute 10.10.3.0 255.255.255.0

ROUTES TO ADD OUTSIDE OF OPENVPN

If you are not running openvpn on the router for each lan, you have some more routes to add. This diagram explains it pretty well.

Lets say our server is 10.10.2.10 on its lan, and uses 10.10.2.1 as its default route, and you want the 2.x lan to be accessible or able to access over the vpn. 2.1 would need a route for every network that 2.x will access or be accessed by. That means in our example: 10.10.2.1 must know that for 10.10.1.x 10.10.3.x and the vpn internal network (for example, 10.8.0.x), it sends the traffic to 10.10.2.10 This is true for any number of lans you want to connect, whether server or client.

If you fail to add this route, here is what would happen if a VPN client (for example, 10.8.0.6) wanted to send traffic to 10.10.2.20:

1. The vpn client sends traffic to 10.10.2.20, with a source address of 10.8.0.6
2. The vpn server (10.8.0.1 and 10.10.2.10) receives the traffic, has IP forwarding enabled, and passes the traffic to 10.10.2.20
3. 10.10.2.20 gets it and tries to respond to 10.8.0.6 but has no entry in its routing table
4. Because 10.10.2.20 has no route for 10.8.0.6, it sends the traffic to its default gateway which is 10.10.2.1
5. 10.10.2.1 checks its routing table, has no route for 10.8.0.6, and sends the traffic to its default gateway which is likely its ISP
6. The ISP ignores it, because it is a RFC 1918 ip (aka lan only)

the annoying work-around would be to add the route to every box on the LAN, in which case step 3 above would work.

If this needs clarification ask me about it and I will update this page after discovering how to make it clearer.

On Jan26, 2010 I changed this article to no longer use 192.168.1.0 192.168.2.0 and 192.168.3.0 for my subnets. I did this because it is important for people to not use common subnets such as 192.168.1/0.x when pushing routes to clients. It does not matter if you know where every client connects from, but once you add a single road warrior to the VPN you will run in to a problem. If the road warrior is connecting from a LAN where he has 192.168.0.X and he gets pushed a route to 192.168.0.0/24 to flow over the vpn, he will lose all connectivity to the internet until he kills the vpn. This is because the client loses his route to his gateway... he tries to contact the gateway over the VPN, but he has no route to the VPN because he needs to access his gateway to reach it. In short, if your lan that you want to access using openvpn uses a common subnet such as 192.168.0.x or 192.168.1.x, CHANGE IT.

Caveats

If you have a problem adding routes in Windows, check the following. If you are using OpenVPN-GUI, do not run it as administrator but make sure the service `OpenVPNServiceInteractive` is running. If you are not using the OpenVPN-GUI make sure you have openvpn starting as administrator.

In some rare cases you may also need to use one of these options:

```
route-method exe
route-delay
```

The first option changes how windows adds a route The second option waits to add the route (helpful for making sure you get a DHCP lease before messing with routes).
If those don't help, try turning off routing and remote access in administrative tools - routing and remote access

Written by krzee @ #OpenVPN @ freenode IRC, mirrored from http://www.secure-computing.net/wiki/index.php/OpenVPN/Routing

Feel free to discuss this document on the unofficial OpenVPN forum at:
https://forums.openvpn.net/topic98.html

Last modified on 03/25/18 16:57:07

Attachments (1)