Instructions for joining the final contest via OpenVPN

You have been sent file `<your-teamname-X>.tar` where `X` is `A`, `B`, or `C`. Below you see a tar file for me (without the `X`) in directory `CDX` on my laptop. I recommend the tar file you were sent should be placed in a clean directory because when it is untarred a lot of files will show up. Change directory to where you put the `tar` file and run the `ls` command like this:

```
[franco@franco ~]$ cd ~/CDX
[franco@franco CDX]$ ls
franco.tar
[franco@franco CDX]$
```

The `tar` file contains contest `VPN` credentials in directory `keys`, the configuration file `client.conf` used to join the `VPN`, a contest configuration file `Parms`, and scripts to join the `VPN` and to start and stop the Client. Untar the file and run `ls` to get a new directory, in this case named `franco`:

```
[franco@franco CDX]$ tar xf franco.tar
[franco@franco CDX]$ ls
franco/  franco.tar
[franco@franco CDX]$
```

Now change directory to the new directory and look at its contents with `ls`:

```
[franco@franco CDX]$ cd franco
[franco@franco franco]$ ls
keys/  Parms  sbin/  run.client*  stop.client*  client.conf  run.vpn*
[franco@franco franco]$
```

You may also see files `email.txt` and `instructions.pdf` (you do not need to worry about `email.txt` and `instructions.pdf` is what you are reading so you already have it). Use command '`ls keys`' to see the contents of directory `keys` which contains OpenVPN credentials:

```
[franco@franco franco]$ ls
keys/  Parms  sbin/  run.client*  stop.client*  client.conf  run.vpn*
[franco@franco franco]$ ls keys
ca.crt  client100.crt  client100.key
[franco@franco franco]$
```

File ca.crt is the OpenVPN server's certificate. File `client100.crt` is my OpenVPN certificate and file `client100.key` is my `OpenVPN` key. Now take a look at the last lines of the configuration file `client.conf` with tail:

```
[franco@franco franco]$ tail -n 18 -f client.conf
# Socks proxy at localhost, port 8080
# Comment this line if the proxy is not needed and modify run.client
# accordingly by taking out the lines that create a socks proxy.
# Comment the line by placing a ';' as the first character in the line.
;socks-proxy 127.0.0.1 8080
#
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca keys/ca.crt
cert keys/client100.crt
key keys/client100.key
#
# The hostname/IP and port of the vpn server
remote 10.52.10.254 1194
```

Note the following:

1. The line at the bottom of the file is remote 10.52.10.254 1194. The 10.52.10.254 IP address is the address of the machine that hosts the OpenVPN server. The address is the one *inside* UC's perimeter. The 1194 is the port on which the server is listening. If you are joining from inside UC's perimeter there is no problem because UCIT allows that port to be used from inside the perimeter. From outside the perimeter, say from India or Turkey, or even Mason, Ohio, you will need to tunnel into UC's network to get to the OpenVPN server. You do this with a socks proxy. See below about how to do this.
2. The server certificate, your certificate, and your key are stated in this file with the same names as in directory `keys`. The middle arrow in the above figure shows where in `client.conf` to find this information. The location of those files is also stated as directory `keys`.
3. There is no specification in the configuration file for the IP address that your VM will get when it joins. The IP address is determined by the OpenVPN server that maps your certificate and key to a `VPN` address.
4. There is a socks-proxy line that begins with a semi-colon, and is identified in the figure above by the topmost red arrow – the semi-colon means this line is commented out. If you need to use a socks proxy because you are joining the contest from outside of UC then this line must be uncommented by removing the semi-colon and you must start a socks proxy like this:

```
[franco@franco franco]$ ssh -N -f -T -D 8080 visitor@helios.ececs.uc.edu
visitor@helios.ececs.uc.edu's password:
[franco@franco franco]$ pstree -paul | grep visitor
  |   |   |   |   |-grep,408558 visitor
  |   |-ssh,408527,franco -A -N -f -T -D 8080 visitor@helios.ececs.uc.edu
[franco@franco franco]$
```

   The password is `iwarsdemo`. Use `pstree` to make sure the socks proxy is running, as shown above - the socks proxy is enabled with process `ssh` running with process number 408527.

To connect to the competition network run `run.client` like this:

```
[franco@franco franco]$ ls
keys/  Parms  sbin/  run.client*  stop.client*  client.conf  run.vpn*
[franco@franco franco]$ ./run.client
[franco@franco franco]$ Sun Nov  1 17:25:04 2020 us=970128 Current Parameter Set
tings:
Sun Nov  1 17:25:04 2020 us=970164   config = 'client.conf'
Sun Nov  1 17:25:04 2020 us=970185   mode = 0
Sun Nov  1 17:25:04 2020 us=970197   persist_config = DISABLED
Sun Nov  1 17:25:04 2020 us=970208   persist_mode = 1
Sun Nov  1 17:25:04 2020 us=970214   show_ciphers = DISABLED
```

What you see above is the beginning of a long output that ends like this if you're actually connected:

```
Sun Nov  1 17:25:07 2020 us=895395 OPTIONS IMPORT: route-related options modifie
d
Sun Nov  1 17:25:07 2020 us=896802 TUN/TAP device tap0 opened
Sun Nov  1 17:25:07 2020 us=897008 TUN/TAP TX queue length set to 100
Sun Nov  1 17:25:07 2020 us=897090 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv
6_setup=0
Sun Nov  1 17:25:07 2020 us=897174 /sbin/ifconfig tap0 10.8.0.200 netmask 255.25
5.255.0 mtu 1500 broadcast 10.8.0.255
Sun Nov  1 17:25:07 2020 us=905332 GID set to nogroup
Sun Nov  1 17:25:07 2020 us=905452 UID set to nobody
Sun Nov  1 17:25:07 2020 us=905522 Initialization Sequence Completed
```

If you do not see "`Initialization Sequence Completed`" then something is wrong. In that case there are two things to try to correct the problem. First, make sure there are no `openvpn` processes running before running `run.client`. Use `pstree | grep openvpn` for this check. To kill `openvpn` processes use `sudo killall openvpn`. Another possibility is you are trying to use a socks proxy but you forget to uncomment the socks-proxy line in `client.conf`.

To make sure you are connected use `/sbin/ifconfig` or something similar like this:

```
[franco@franco franco]$ /sbin/ifconfig
enp0s25: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether c4:34:6b:28:9b:c6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xd0700000-d0720000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 113012  bytes 1419113619 (1.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 113012  bytes 1419113619 (1.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.8.0.200  netmask 255.255.255.0  broadcast 10.8.0.255
        inet6 fe80::88f5:88ff:fe33:4761  prefixlen 64  scopeid 0x20<link>
        ether 8a:f5:88:33:47:61  txqueuelen 100  (Ethernet)
        RX packets 1  bytes 70 (70.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 60  bytes 8374 (8.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The interface `tap0` has been created by the OpenVPN server. The IP address given by the server is `10.8.0.200`. If `tap0` or `tap1` is missing from the output of `/sbin/ifconfig` then something is wrong but this is unlikely if `run.client` produced a successful output.

To quit the `VPN` use `stop.client`.

The main purpose of the file `Parms` is to supply the start and end dates and times of the competition.

Summary

Your VM

Connecting to the contest using OpenVPN

The following assumes a directory called `keys` with files `ca.crt`, `client`**`X`**`.key`, `client`**`X`**`.crt`, an OpenVPN configuration file client.conf with a section like this:

```
ca keys/ca.crt
cert keys/client X.crt
key keys/client X.key
```

where **`X`** is a number from 0 to 150, and a line indicating the internal address of the OpenVPN server like this:

```
remote 10.52.10.254 1194
```

plus scripts `run.client` and `run.player` for starting the OpenVPN connection and Client

Using OpenVPN with a socks proxy

1. edit `client.conf` → remove semi-colon from `;socks-proxy 127.0.0.1 8080`
2. check whether there is a connection to socks proxy → `pstree -paul | grep ssh`
   example result:
   `-ssh,6687,user -N -f -T -D 8080 visitor@helios.ececs.uc.edu`
3. if there is no result as above (no connection) then do this:
   a. connect to socks proxy → `./run.proxy` (password is `iwarsdemo`) where run.proxy should be created by you with execute permissions and the following content:

```
#!/bin/sh
# This is just an example - replace visitor@... with the user and
# location of your proxy server
ssh -N -f -T -D 8080 visitor@helios.ececs.uc.edu
```

4. check whether openvpn is running → `pstree -paul | grep openvpn`
   example result:
   `-openvpn,6984,nobody client.conf`
5. if there is a result as above (openvpn is running) then do this:
   a. kill the runnung process → `sudo killall openvpn`
6. start the openvpn client → `./run.client`

Using OpenVPN without a socks proxy

1. complete steps 4, 5, 6 above