**A note to the programmer:**

You need to create following shell scripts and folders in order to contribute any tactics:

* **register.sh**

This will add the name of the tactic in `index` file and will add tactic folder in the repository. It shall be run only once by the programmer to add the tactic to the repository.

* **install.sh**

This will install the tactic. Both client and server need to run this. It shall be run only once.

* **stop.sh**

This will stop the tactic which is running in the client and server. Both client and server need to run this to stop the ongoing service.

* **tactic\_client**

This folder will contain all the shell scripts need to setup a client (initialize\_client.sh, configuration files etc.)

* **tactic\_server**

This folder will contain all the shell scripts need to setup a server (initialize\_server.sh, configuration files etc.)

**Some guidance to github** (Getting a repository, making changes and uploading in git)

1. Clone a repository

* git clone <https://github.com/signposts/tactics.git>

1. Make changes to the repository (add/delete/modify files/folders)

* cd tactics

1. Add changes

* git add ./

1. Commit changes

* git commit

1. Push back the repository

* git push origin master

**About some tactics:**

1. **openVPN**

**Commands**:

Clone the repository:

* git clone <https://github.com/signposts/tactics.git>
* sudo sh your\_path\_to\_folder\_containg\_tactic/register.sh
* Run this once from any system. It need not be a client or server system.
* This will copy entire folder of the tactic openvpn to repository signposts/tactics and also will add name of the tactic (openvpn) in the signposts/tactics/index file.

**Server**:

* sudo sh ~/tactics/openvpn\_19.3/install.sh

Run this only once as this is needed to install openvpn in the server.

* sudo sh ~/tactics/openvpn\_19.3/openvpn\_server\_me/initialize\_server.sh *country\_name province\_name city\_name origin\_name email*
* This will create a folder `ov\_me` in HOME directory with all the scripts necessary to generate keys and `vars file `containing users’own description for country, city etc.
* NOTE: country\_name and province\_name should have only 2 alphabets.
* cd $HOME/ov\_me
* source vars
* sh ~/tactics/openvpn\_19.3/openvpn\_server\_me/keygen.sh *hostname\_of\_client*

#eg: bishneet(this should match $SUDO\_USER in client system)

This will create all the certificates for server and client. Client certificates need to be copied by them in their HOME directory (It will be done by another shell script)

* sudo sh ~/tactics/openvpn\_19.3/openvpn\_server\_me/server.conf.sh *port\_number*

#here: 5060

This will create server.conf in `ov\_me` and will start openvpn at server on the port specified above. After this, server is up and running and a new interface `tun0` is created with ip: 10.8.0.1. Ping from client using this IP (when the client is up and running).

**Client**:

[Prior to this server should be up and running]

* sudo sh ~/tactics/openvpn\_19.3/install.sh

Run this only once as this is needed to install openvpn in the client.

* sudo sh ~/tactics/openvpn\_19.3/openvpn\_client\_me/initialize\_client.sh *hostname\_of\_server public\_ip\_server*

This will create a folder `ov\_me\_client` and copy all the certificates for client from server machine. Note: this is using scp, so the key to connect to server should be there in ~/.ssh folder in client system.

* sudo sh ~/tactics/openvpn\_19.3/openvpn\_client\_me/client.conf.sh *public\_ip\_server port\_number*

This will create client.conf in `ov\_me\_client`and will start openvpn at client on port specified. After this client is up and running and a new interface `tun0` is created with ip: 10.8.0.x. Ping from server using this IP.

1. **Tor**.

**Commands**:

Clone the repository:

* git clone <https://github.com/signposts/tactics.git>
* sudo sh your\_path\_to\_folder\_containg\_tactic/register.sh
* Run this once from any system. It need not be a client or server system.
* This will copy entire folder of the tactic tor to repository signposts/tactics and also will add name of the tactic (tor) in the signposts/tactics/index file.

**Server**:

* sudo sh ~/tactics/tor\_19.3/install.sh

Run this only once as this is needed to install tor in the server.

* sudo sh ~/tactics/tor\_19.3/tor\_server/initialize\_server.sh *port\_number*
* This will create a folder `tor` in HOME directory. And start tor at server.
* Port number will specify the port at which hidden-service will run. As tor starts, hidden-service will be given a domain name(with .onion extension). This domain name will be stored in a file *$HOME/tor/hidden\_service/hostname*.
* sudo sh ~/tactics/tor\_19.3/tor\_server/filegen.sh

This will change permission of $HOME/tor/hidden\_service which contains file `hostname`. Also, it will copy `hostname` to HOME directory. This `hostname` is copied by the client so that it can connect through this hidden\_service to server.

**Client**:

[Prior to this server should be up and running]

* sudo sh ~/tactics/tor\_19.3/install.sh

Run this only once as this is needed to install tor in the client.

* sudo sh ~/tactics/tor\_19.3/tor\_client/install\_extra.sh

This will install extra softwares like curl, which are needed by client. Run this only once.

* sudo sh ~/tactics/tor\_19.3/tor\_client/initialize\_client.sh *hostname\_server public\_ip\_server port\_number*
* This will create a folder `tor\_me` in HOME directory. `hostname` from server will be pasted in this and contents are stored in a variable. Using these contents (domain name of hidden\_service), connection between client and server is established. Note: this is using scp, so the key to connect to server should be there in ~/.ssh folder in client system.

NOTE: *port\_number* should be the same given by the server where hidden-service is running

* After this client is also up and running. You can see html commands containing elements of HOME directory of server are displayed.

**Special Notes**

* Any running tactic can be stopped by running *stop.sh* present in the respective tactics folder.
* If any changes to repository are made by adding or deleting a tactics, don’t forget to **add** and **commit** those changes and **pushing** the repository back.