Getting started

a quick guide on how to approach tBB

Step 0. Are you allowed to do this?

Before starting at all you should consider that tBB installation and execution require root privileges. If you're not able to supply such privileges for your machine, please contact your system administrator.

Step 1. Setting up the environment.

Before getting the tBB package from whichever source you prefer, you should first check that you're meeting the dependences tBB needs to work. First, you should keep in mind that tBB has been designed to work in a Linux environment. Nevertheless it should be able to run in any other Unix-like OS, but this is not granted.

Furthermore, tBB has dependences other than the OS you're running. Here they are listed:

Name	*How to get it (bash commands)*
`Python 3.4`	`sudo apt-get install python3.4`
`aiohttp 0.22.2 (python library)`	`sudo pip install aiohttp==0.22.2`

Please, note that if you're planning to use tBB_cli for a front-end, it will not require any external libraries, as they come built-into it.

Step 2. Get tBB.

Once you're all set with the dependences you can get tBB from whichever source you prefer.

Step 3. Set up tBB.

Once you got your tBB package ready in your system, navigate to it in a terminal and try see if it is able to run. In the main tBB folder there is a file simply called “tBB”. This is the file tBB is supposed to be executed from. Check that your system is allowing the file to be executed and then start it in your terminal using the command “`$ sudo ./tBB`”. Note that I've used the sudo command. This is due to the fact that tBB is only able to run when launched by the root user. As stated in Step 0, if you don't have such privileges, please contact your system administrator.

Unless you had previously set up one, at this point tBB will complain about the fact that it wasn't able to find a password file and will quit the application prematurely. You'll probably see something like: “`Couldn't find password file!!! Aborting.`”. To set up a password for your tBB simply create a file called “`tBB_access_password`” in your main tBB folder. Edit this file and insert your password in plain-text into it. Once you're good with your password change the permissions of the file to `600`. For instance, use the command “`$ sudo chmod 600 tBB_access_password`”. This way, only the root user is authorized to edit the file. Try executing tBB again to check that everything is working.

Step 4. Configure tBB.

Now, your tBB is fully able to run. At this stage you may want to configure it so as to meet your needs and preferences.

There's a simple way to configure tBB. In the main folder, you'll find a file called "`config_default.json`". This is a JSON-compliant file that will allow you to change the settings of tBB.

Note the default suffix at the end of the file name. This indicates that this is the file tBB will fall-back to when it isn't able to find any specific configuration file.

Specific configuration files are files similar to the one provided along-with tBB. In these files you can define settings to be used only when tBB is monitoring a specific network. Any settings that aren't defined in the specific configuration file will fall-back to the ones tBB finds in the default configuration file. For instance, let's say that for network X I prefer a lighter touch of tBB scans. In this case I'll open the configuration file related to network X and set the field "`time_between_checks`" to a higher value, every field I haven't set in the network X configuration file will fall-back to whatever I have in the default configuration file.

The naming conventions for these specific configuration files are the following: "`config_{Network}.json`". Replace "Network" with any network you like, but use the following syntax: "`{Network IP}\{Network Mask}-{Network Length}`". This allows tBB to use the correct configuration file for every network you may want to monitor.

To learn what settings are there to be configured and how to use them, please refer to the configuration files guide.

Step 5. Using SSL.

If you're planning to use SSL for front-end communications, please keep in mind that the certificates tBB comes with by default are invalid. With these certificates you cannot use host name checking and will therefore be vulnerable to man-in-the-middle attacks. If you wish to use valid HTTPS secure communications, please feed your own certificates at `tBB/certs/cert.pem` and `tBB/certs/key.pem`.

If you're planning to use the invalid certificates tBB comes with, you're going to have to instruct your front-end of choice to not do host name checking either.

Please, keep in mind that non-secure communications are highly discouraged.

You can set SSL usage in the configuration files.

Step 6. You're ready!

Now that you have configured tBB, you're ready to make it work.

There are a couple ways you may want to run tBB. For instance there are the `debug`, `silent` and `developer` modes. Please launch tBB with the `--help` option for further detail.

Happy watching! c: