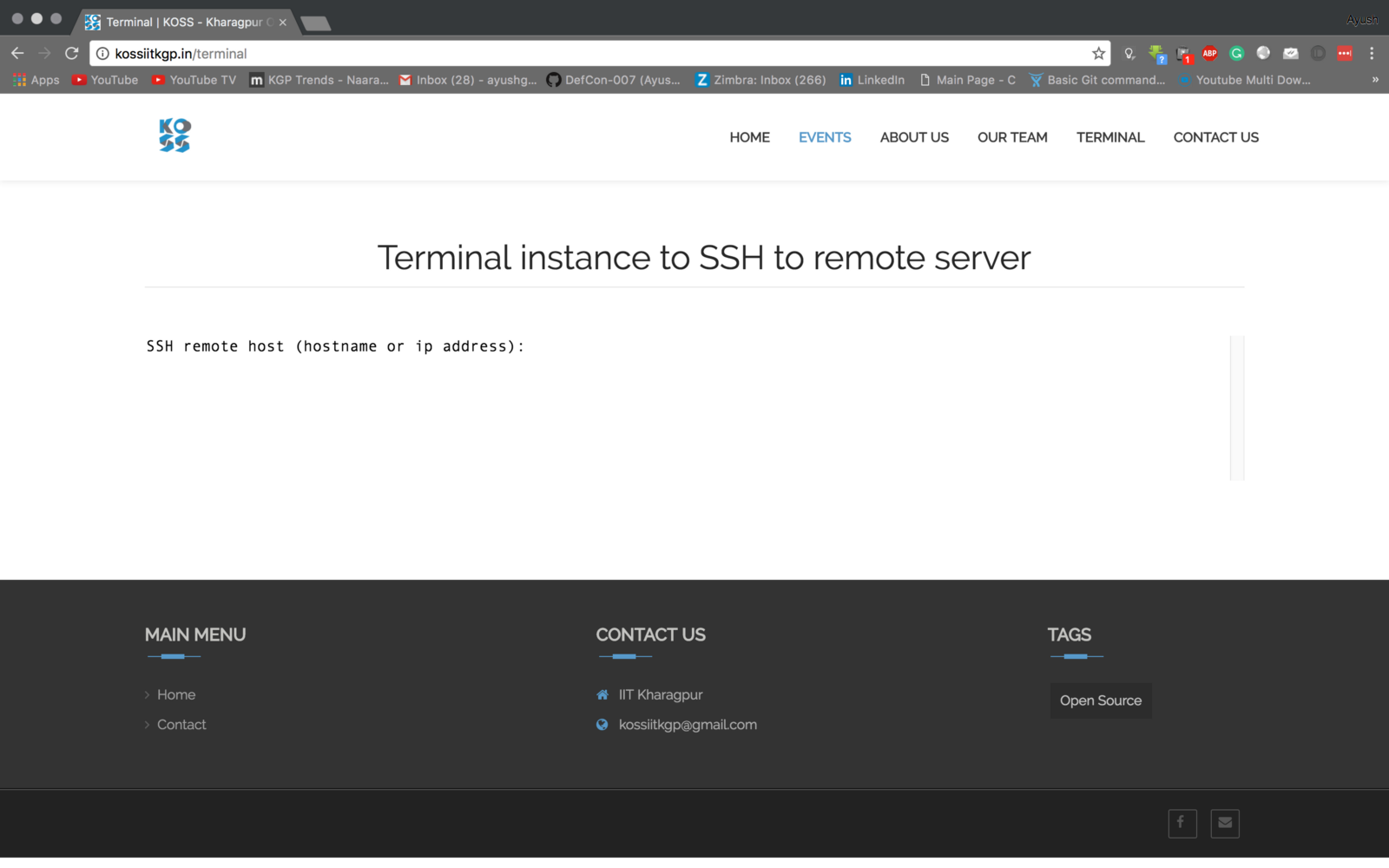
**SSH to a remote server from proxy restricted network**

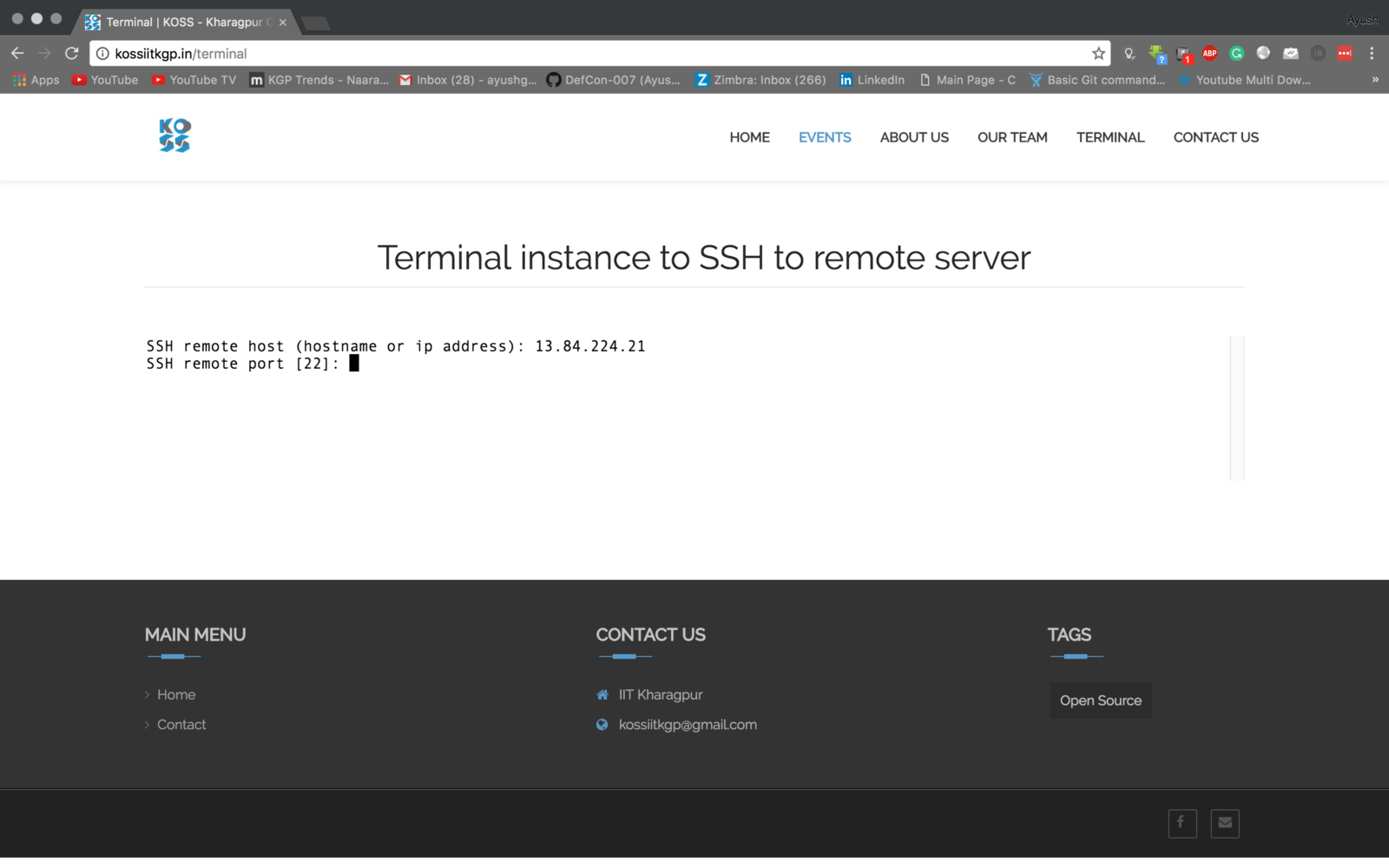
We love to use servers and we all know lots of work can be done efficiently that way. With Digital Ocean and Microsoft Azure giving out free credits to the students we all want to switch one or the other task to the server. However, the main problem with the setup is the port, which is used to SSH to the server. The default port i.e. 22 is blocked by our network administrator. So currently, we have to make some workarounds on our local machine then open port 443 on the server to connect to it. [[1]](https://wiki.metakgp.org/w/How_to_SSH_into_a_server_that_is_outside_the_campus_network)

While I was setting up my first server I faced a lot of problems and I wished to make things easier. I was looking to have a way by which we can SSH to our server without going through all of these steps, which may seem nasty for every first-timer.

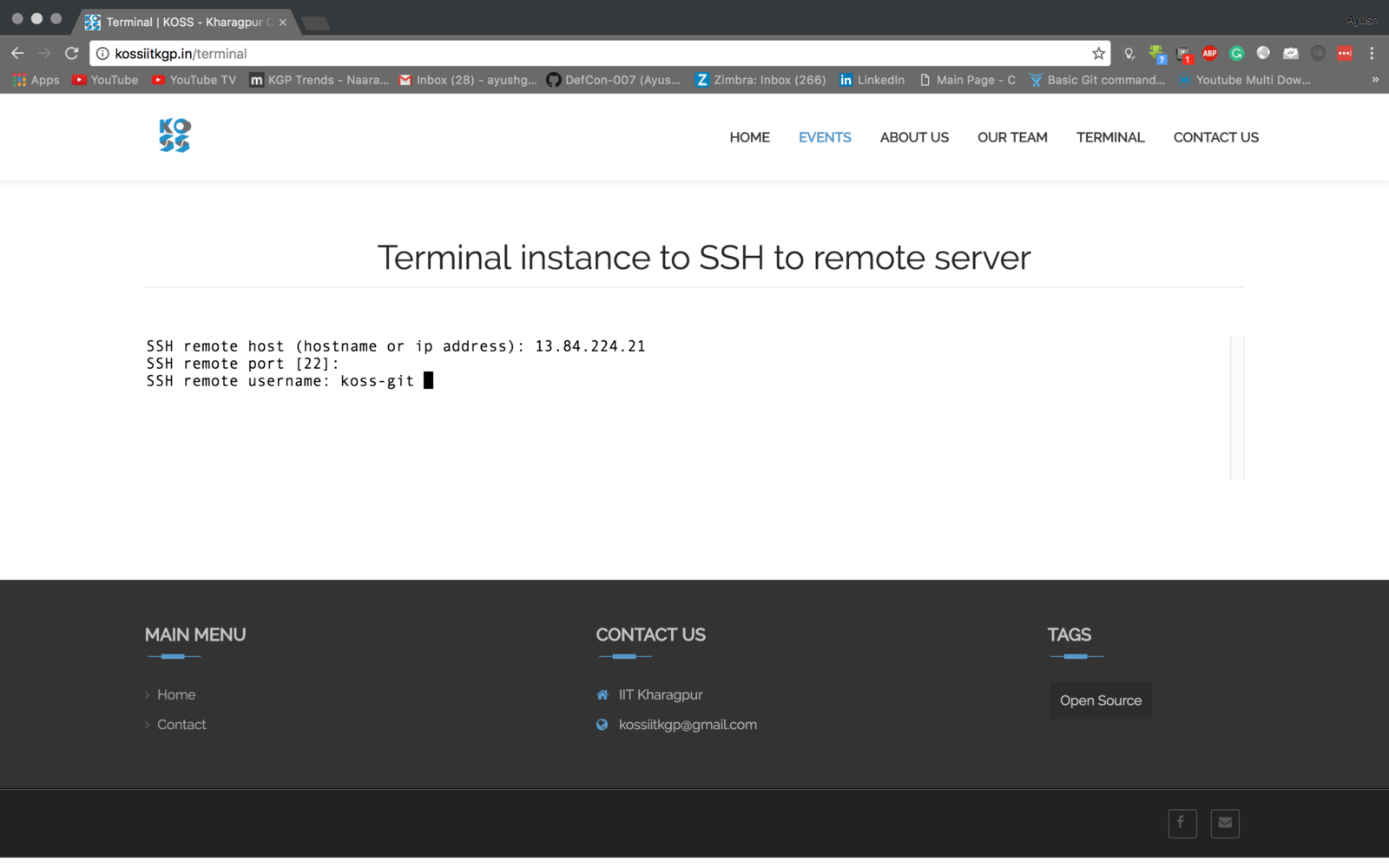
So, here it goes. Visit <http://kossiitkgp.in/terminal> .



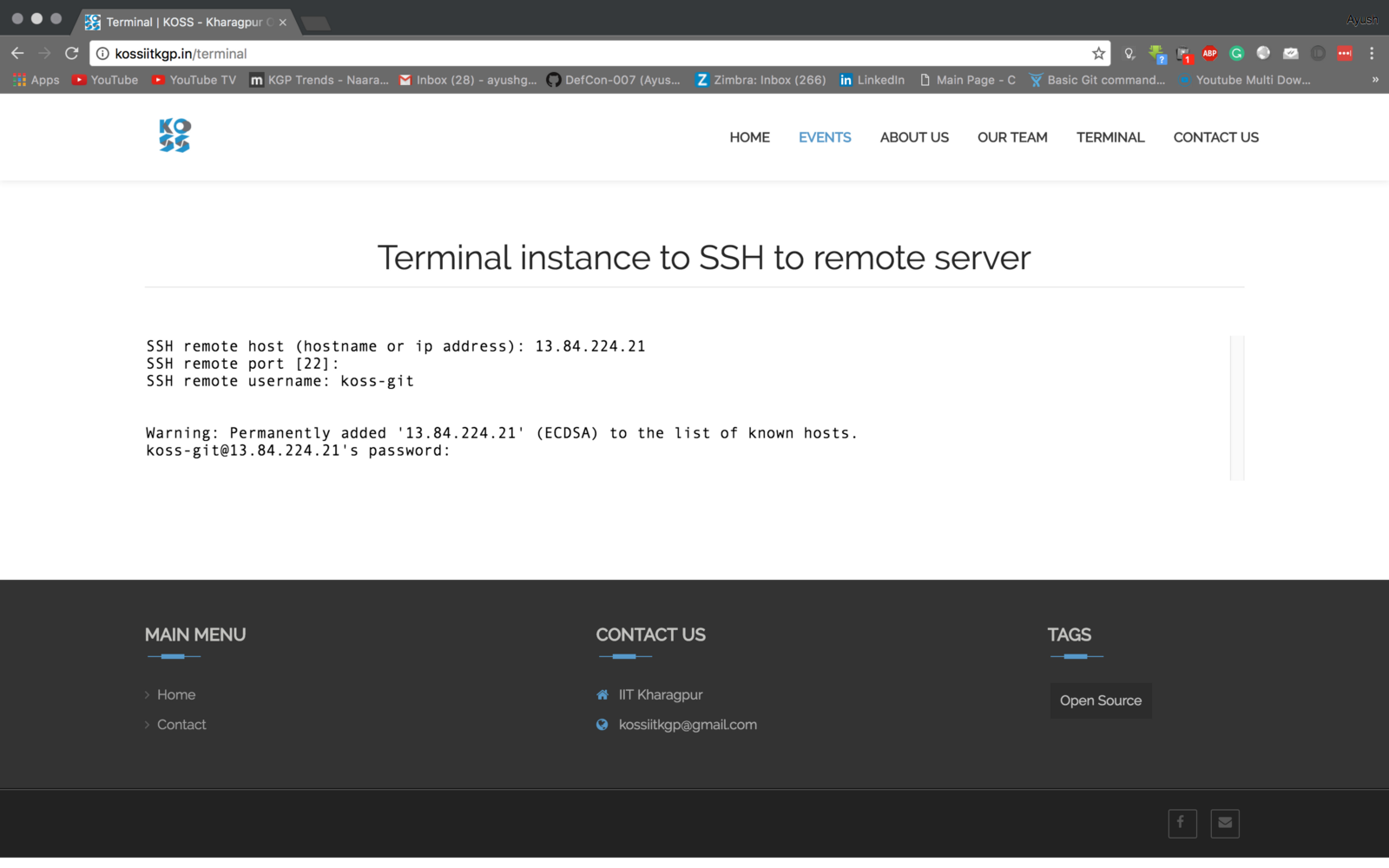
Now type the IP address of your remote server and press enter.



You can leave the port field blank unless your server listens on any other port. Press enter and type your username.



Now, just type your password and press enter.



Voìla … You have successfully SSHed to your remote server.

The above steps are good for a beginner but security risks are very high in them.   
To overcome those there are actually two ways :

1. Shellinabox
2. Butterfly

They both run on your own server so there are no security risks.

**Shell in a box :**

Shell In A Box implements a web server that can export arbitrary command line tools to a web based terminal emulator. This emulator is accessible to any JavaScript and CSS enabled web browser and does not require any additional browser plugins.

**Installation**

Install the OpenSSL and Shell in a box package.

sudo apt-get install openssl shellinabox

**Configuring port**

By default, **shellinaboxd** listens on **TCP** port **4200** on **localhost**. For security reason, you can change this default port to a random (say 6175) to make it difficult for anyone to reach your SSH box. Also, during installation a new self-signed **SSL** certificate automatically created under “**/var/lib/shellinabox**” to use **HTTPS** protocol.

sudo vim /etc/default/shellinabox

Now update the file according to the following snippet.

# TCP port that shellinboxd's webserver listens on  
SHELLINABOX\_PORT=6175  
# specify the IP address of a destination SSH server  
SHELLINABOX\_ARGS="--o-beep -s /:SSH:172.16.25.125"  
# if you want to restrict access to shellinaboxd from localhost only  
SHELLINABOX\_ARGS="--o-beep -s /:SSH:172.16.25.125 --localhost-only"

**Starting Shell in a box**

Once the configuration is complete you can start the shell in a box with the following command.

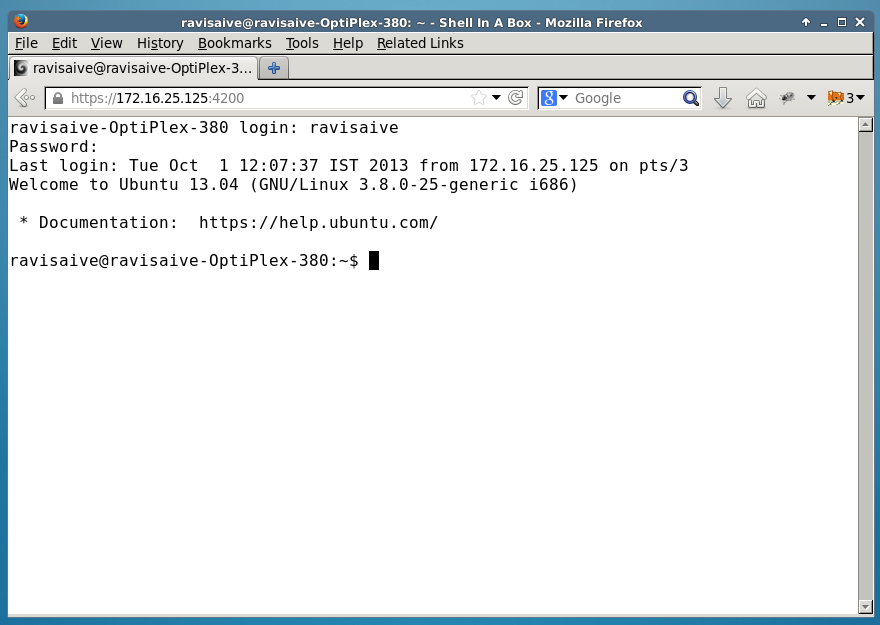
sudo service shellinaboxd start

**Verifying**

You can verify whether **Shellinabox** is running on port **6175** using “**netstat**” command.

sudo netstat -nap | grep shellinabox

Now open up your web browser, and navigate to [https://Your-IP-Adress:6175](https://your-ip-adress:6175/). You should be able to see a web-based SSH terminal. Login using your username and password and you should be presented with your shell prompt.



**Butterfly**

This one is quite easier to setup than Shell in a box

**Installing**

sudo pip install butterfly

**Configuring**

It relies on a server running in background.   
Systemd provides a way to automatically activate daemons when needed (socket activation). Use the following to do it.

$ cd /etc/systemd/system  
$ curl -O https://raw.githubusercontent.com/paradoxxxzero/butterfly/master/butterfly.service  
$ curl -O https://raw.githubusercontent.com/paradoxxxzero/butterfly/master/butterfly.socket  
$ systemctl enable butterfly.socket  
$ systemctl start butterfly.socket

Don’t forget to update the /etc/butterfly/butterfly.conf file with your server options (host, port, shell, …)

You can set the bind host with butterfly.server.py --host="0.0.0.0"(Replace 0.0.0.0 with your server IP) which will allow other users to connect to your terminal. A password will be asked but **IT IS NOT SECURE!** So it’s recommended as of now to run this only on a local network for testing purposes.

**References :**

1. <http://paradoxxxzero.github.io/2014/02/28/butterfly.html>
2. <https://github.com/paradoxxxzero/butterfly>
3. <http://www.tecmint.com/shell-in-a-box-a-web-based-ssh-terminal-to-access-remote-linux-servers/>
4. <https://github.com/shellinabox/shellinabox>