Direct and Reverse Connections

Direct Connections->
For example you in a country and your friend is in our country and now your friend wants to you to fix some problem in his laptop.so what we can do is we can remotely connect to his computer using command shell or terminal. This type of connection is achieved by something known as direct connection
In direct connection first we create a socket in our laptop which basically means to open the line of communication between computers then we will bind our port and host to the socket and send the request to our friend on his id address. If he accept our request then we will be able to remotely access his laptop using our command prompt and fix his problem
So for direct connections we require the ip address of our laptop as well as ip of our friend laptop

Problem With Direct Connections

1) It is difficult to get IP address
2) Even if we get his ip address the ip address is dynamic so its always changing
3) Even we could get regular updates to the dynamic IP address the computer has bunch of firewalls which makes it impossible to get to his computer

Reverse Connections
To understand the reverse Connections i thing the best is we take the example of a hacker trying to access his victim without his knowledge.in reverse connection instead of trying to initiate or start the connection from his computer , the connection is initiated from the victim's computer . so what hacker do is that they create a python file called reverse shell and in that file the ip address and port number of hacker is stored and they send this file to victim via mail and when the victim open up this file it creates a reverse connection to the hacker computer . Now because the victim is started this connection the hacker has not to worry about the ip address of the victim computer. So even if the ip address is dynamic it does not really matter because every time the ip address changes the file installed on the victim laptop calibrates accordingly but there is still one more problem to solve and that is the hacker laptop still has a dynamic ip address so the address store in the file will be useless after some time . to mitigate this problem hackers create the server and they store the ip address of the server in the reverse shell file because the ip address of the server remains static .

Server

1) Remote computer or a laptop which is never turn off.
2) where you can store files or host websites
3) It has a static address
4) You need internet to access this remote computer

REVERSE SHELL

It is a way to connect anyone's personal computer anywhere in the world. It uses the Concept of reverse connection

Now we are going to implement the same with the help of python

For doing it we need to create a server .py and client.py.
server.py is to be installed on cloud server that we can host on digital ocean or aws

And client .py is intallled on the computer whom you want to connect

MULTIPLE CLIENT SUPPORT

How to handle multiple client with one server file?

So every time we accept a connection we get 2 output.
One is conn object and another is address list

For mutli there are two tasks of server.py file

1) Sending commands to an already connected client
2) Listen and accept connections from other clients

But you must be thinking how can one file do two things in same time?

This is done with the help of threading

Threads

-> Multitasking Support System

->Speaking - Thread 1
Thinking->Thread 2

-> Listen and accept connection from other clients - Thread1
Sending commands to an already connected client- Thread 2

Thread Flow in multi threading

1) Create worker thread
   a)Use a ‘for ’loop
   b)Create threads using t=threading.Thread()
   c) Assign t.daemon=True-> this means we are telling the thread to release the memory after the program ends
   d) Start the thread using t.start()
2) Store jobs in Queue because thread look for jobs ina queue and not in lists
3) Create a work function and get the queue
   a) If the job number in queue is 1 then handle connection
   b) If the job number in queue is 2 then send commands