

Forum:[Hack Forums](#)

Thread: [Basic Python IRC Scripts](#)

Dawnfang - April 16, 2020, 12:21 pm

Hello guys, I created a basic python IRC script and wanted to see what you guys thought about it. I'm interested in any thoughts or improvements about the scripts.

Client.py (The client used to send messages)

```
Code import socket
import os
import sys
import pickle
import threading
from threading import Thread
```

```
class Connection:
```

```
    def __init__(self, port, host, name):
        self.name = name
        self.host = host
        self.port = port
        self.messages = []
        self.clients = []
```

```
def send():
```

```
    while True:
        os.system('clear')
        for message in Conn.messages:
            print message
        print "\ntype '--exit' to quit or '--r' to refresh"
        print "Type message here: "
        message = raw_input(" ")
        if message == "--exit":
            sys.exit()
        if message == "--r":
            pass
        else:
            send = "%s: %s" % (Conn.name, message)
            Conn.s.sendto(pickle.dumps(send), (Conn.host, Conn.port))
```

```
def main():
```

```
    print "Username: "
    option = raw_input('>>> ')
    global Conn
    Conn = Connection(7890, '10.183.1.16', option)
    Conn.s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    send()
```

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

Server.py (Receives and directs messages to clients)

```
Code import socket  
from client import Connection  
import pickle  
import errno  
import threading  
from threading import Thread
```

```
def clisten():  
    i = 0  
    while True:  
        s1.listen(1)  
        exec("c%i, addr%i = s1.accept()" % (i, i))  
        exec("Conn.clients.append(c%i)" % i)  
        i+=1
```

```
def mlisten():  
    while True:  
        message = s2.recvfrom(1024)  
        send(message[0])
```

```
def send(message):  
    for client in Conn.clients:  
        #Checks to see if client exists  
        #If not reader is removed from client list  
        try:  
            client.send(message)  
        except IOError as e:  
            if e.errno == errno.EPIPE:  
                print "%r is offline" % client
```

```
def main():  
    global Conn  
    Conn = Connection(7890, "10.183.1.16", "main_server")  
    #Socket S1 is socket for client prompt (tcp)  
    #Message recieving socket  
    global s1  
    s1 = socket.socket()  
    s1.bind((Conn.host, Conn.port))  
    print s1  
    #Socket S2 is socket for client reader (udp)  
    #Message Sending socket  
    global s2  
    s2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
s2.bind((Conn.host, Conn.port))
Thread(target = mlisten).start()
Thread(target = clisten).start()
```

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

Reader.py (Receives messages and updates them to terminal to be viewed)

Code import socket

import os

from client import Connection

import time

import threading

from threading import Thread

import pickle

```
def listen():
```

```
    while True:
```

```
        messg = Conn.s.recv(1024)
```

```
        messg = pickle.loads(messg)
```

```
        Conn.messages.append(messg)
```

```
def printMessage():
```

```
    while True:
```

```
        os.system('clear')
```

```
        for message in Conn.messages:
```

```
            print message
```

```
        time.sleep(1)
```

```
def main():
```

```
    global Conn
```

```
    Conn = Connection(7890, '10.183.1.16', "reader")
```

```
    Conn.s = socket.socket()
```

```
    Conn.s.connect((Conn.host, Conn.port))
```

```
    Thread(target = listen).start()
```

```
    Thread(target = printMessage).start()
```

```
if __name__ == "__main__":
```

```
    main()
```

The reason I have the Reader and Client separated into 2 separate scripts was because, I couldn't figure out a way to be able to send messages without the terminal being cleared every time the client receives a new message.

D34r_Us3rn4m3 - May 4, 2020, 1:47 am

Cool :) A suggestion would be to add an encryption, something like RSA to exchange keys and then AES 256 bit. This is really easy to do having the script that you have (I know because I've done it), if you need some help hit me up.

I've also tried XOR encryption for "perfect secrecy" but that's way harder to implement having the two-time-pad vulnerability and trying to avoid it.

Dawnfang - May 5, 2020, 12:33 pm

(May 4, 2020 01:47 AM)D34r_Us3rn4m3 Wrote: [\[CITE\]](#) Cool :) A suggestion would be to add an encryption, something like RSA to exchange keys and then AES 256 bit. This is really easy to do having the script that you have (I know because I've done it), if you need some help hit me up. I've also tried XOR encryption for "perfect secrecy" but that's way harder to implement having the two-time-pad vulnerability and trying to avoid it.

Appreciate the feedback ? Adding some sort of encryption definitely would be a good idea. I was also planning on having chat rooms that could be viewed from a menu or something, so you could host your chat server and have it be available to everyone using this client.

D34r_Us3rn4m3 - May 14, 2020, 1:27 am

(May 5, 2020 12:33 PM)Dawnfang Wrote: [\[CITE\]](#) Appreciate the feedback ? Adding some sort of encryption definitely would be a good idea. I was also planning on having chat rooms that could be viewed from a menu or something, so you could host your chat server and have it be available to everyone using this client.ah, I thought you were creating a RAT :D But nevertheless, good idea, keep going.