基于python的SSH小工具

1. 程序的流程图



1. 程序功能：
2. 实现远程命令的执行，将执行结果返回到当前窗口中（分为执行有返回结果和执行没有返回结果的）
3. 对执行命令进行判断，命令输入是否正确
4. 对执行命令和结果数据进行加密传输(采用对称加密)
5. 服务器采用非阻塞模式，可以连接多个客户端
6. 程序代码
7. 服务器端
8. #!/usr/bin/env python  
     
   #coding: utf8  
     
     
   **import** sys  
   **from** Crypto.Cipher **import** AES  
   **from** binascii **import** b2a\_hex,a2b\_hex  
     
   **class prpcrypt**():  
    **def** \_\_init\_\_(self,key):  
    self.key = key  
    self.mode = AES.MODE\_CBC  
     
    **def encrypt**(self,text):  
    cryptor= AES.new(self.key,self.mode,self.key)  
    length = 16  
    count = len(text)  
    add = length -(count % length)  
    text = text +('\0' \* add)  
    self.ciphertext = cryptor.encrypt(text)  
     
    **return** b2a\_hex(self.ciphertext)  
     
    **def decrypt**(self,text):  
    cryptor = AES.new(self.key,self.mode,self.key)  
    plain\_text = cryptor.decrypt(a2b\_hex(text))  
    **return** plain\_text.rstrip('\0')  
     
   pc = prpcrypt('liangchangyou123')  
     
     
     
   **import** os  
   **import** SocketServer  
   **class myTCPHandler**(SocketServer.BaseRequestHandler):  
    **def handle**(self):  
    **print** "got connection from:",self.client\_address  
    **while** 1:  
    self.data = self.request.recv(4096)  
    e = pc.decrypt(self.data)  
    cmd=os.popen(e)  
    cmd\_result=cmd.read()  
    result = pc.encrypt(cmd\_result)  
    **if** cmd\_result:  
    **print** '\033[31m%s\033[0m' % cmd\_result  
    self.request.sendall(result)  
    **else**:  
    **print** '\033[31msuccess!!\033[0m'  
    a=pc.encrypt('?')  
    self.request.sendall(a)  
   h,p='',9999  
   sever=SocketServer.ThreadingTCPServer((h,p),myTCPHandler)  
     
   sever.serve\_forever()
9. 客户端代码
10. #!/usr/bin/env python  
      
    #coding: utf8  
      
      
      
    **import** sys  
    **from** Crypto.Cipher **import** AES  
    **from** binascii **import** b2a\_hex,a2b\_hex  
      
    **class prpcrypt**():  
     **def** \_\_init\_\_(self,key):  
     self.key = key  
     self.mode = AES.MODE\_CBC  
      
     **def encrypt**(self,text):  
     cryptor= AES.new(self.key,self.mode,self.key)  
     length = 16  
     count = len(text)  
     add = length -(count % length)  
     text = text +('\0' \* add)  
     self.ciphertext = cryptor.encrypt(text)  
      
     **return** b2a\_hex(self.ciphertext)  
      
     **def decrypt**(self,text):  
     cryptor = AES.new(self.key,self.mode,self.key)  
     plain\_text = cryptor.decrypt(a2b\_hex(text))  
     **return** plain\_text.rstrip('\0')  
      
      
    pc = prpcrypt('liangchangyou123')  
      
      
    **import** commands  
    **import** socket  
    **import** time  
      
    h='139.129.47.28'  
    p=9999  
    s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)  
    s.connect((h,p))  
      
    **while** 1:  
     INPUT=raw\_input("INPUT:")  
     **if** INPUT:  
     (status,output)=commands.getstatusoutput(INPUT)  
     **if** status == 0:  
     e=pc.encrypt(INPUT)  
     s.send(e)  
      
     **else**:  
     **print** "\033[31m-bash: %s command not found\033[0m" % INPUT  
     **continue  
     else**:  
     s.close()  
     exit(1)  
     recv\_data=s.recv(4096)  
     data=pc.decrypt(recv\_data)  
     **if** data == '?':  
     **print** 'OUTPUT:\n\033[31m %s Command execution success!\033[0m' % INPUT  
     **continue  
     else**:  
     **print** "OUTPUT:\n\033[31m%s\033[0m" % data

不足：

1. 命令补全
2. 历史命令
3. 持续输出命令 如top