5 min. read

Python程序设计作业#2

2021-07-16 | 2020-11-30 | 3 | 0

Python程序设计#2作业

截止时间：2020年11月02日23:59:59

作业题目

实现localProxy双协议（SOCKS5和HTTP tunnel）本地代理。

支持（SOCKS5代理）基于#1作业的成果。

支持HTTP tunnel（即HTTP CONNECT method）可用于HTTPS代理。

关于HTTP tunnel可以参见：https://www.zhihu.com/question/21955083

作业内容

程序源代码嵌入下方的code block中。

```
import asyncio
import struct
import socket
import logging
logging.basicConfig(level=logging.INFO)
import nest_asyncio
nest_asyncio.apply()
VERSION = 5
async def socks5(first, reader, writer):
```

```
    addr_from = writer.get_extra_info('peername')
    logging.info(f'connect from{addr_from}')
    header = await reader.read(1)
    header = first + header
    ver, num_method = struct.unpack("!BB", header)
    logging.info(f'ver == VERSION:{ver == VERSION}')
    logging.info('num_method = %d' % num_method)
    methods = []
    for i in range(num_method):
        methods.append(ord(await reader.read(1)))
    if 0 not in methods:#无需认证
        writer.close()
        writer.wait_closed()
        return
    #回应一个数据包，包括协议版本号，指定认证方法
    writer.write(struct.pack("!BB", VERSION, 0))
    await writer.drain()
    request = await reader.read(4)
    ver, cmd, rsv, atype = struct.unpack("!BBBB", request)
    assert ver == VERSION
    #ipv4
    if atype == 1:
        address = socket.inet_ntoa(await reader.read(4))
    #域名
    elif atype == 3:
        domain_length = await reader.read(1)
        address = await reader.read(domain_length[0])
    #ipv6
    elif atype == 4:
        address = socket.inet_ntop(socket.AF_INET6, await reader.read(16))
    else:
        writer.close()
        writer.wait_closed()
        return
    port = struct.unpack('!H', await reader.read(2))
    try:
        if cmd == 1:
            reader_remote,writer_remote = await asyncio.open_connection(address,p
        else:
            writer.close()
            writer.wait_closed()
    except Exception as error:
        logging.error(error)
    reply = struct.pack("!BBBBIH", VERSION, 0, 0, 1, 0, 0)
    writer.write(reply)
    await writer.drain()

```

```
57      #第一个字节为0表示成功代理
58      if cmd == 1 and reply[1] == 0:
59          tasks = [read_trans(reader, writer_remote), write_trans(reader_remote, wr
60          await asyncio.wait(tasks)
61
62  async def read_trans(reader, writer_remote):
63      while True:
64          data = await reader.read(4096)
65          if not data:
66              logging.info('disconnect')
67              break
68          writer_remote.write(data)
69          await writer_remote.drain()
70
71  async def write_trans(reader_remote, writer):
72      while True:
73          data = await reader_remote.read(4096)
74          if not data:
75              logging.info('disconnect')
76              break
77          writer.write(data)
78          await writer.drain()
79
80  async def httptunnel(first, reader, writer):
81      http_connect = (await reader.read(1024))
82      http_connect = (first + http_connect).decode()
83
84      logging.info(http_connect)
85      i = 0
86      while(http_connect[i] != ':'):
87          i += 1
88      domain_name = http_connect[8 : i]
89      j = i
90      while(http_connect[j] != ' '):
91          j += 1
92      port = http_connect[i + 1 : j]
93
94      logging.info('domain_name:%s ' % domain_name)
95      logging.info('port:%s' % port)
96      reply = 'HTTP/1.1 200 OK\r\n\r\n'
97      writer.write(reply.encode())
98      await writer.drain()
99      reader_remote,writer_remote = await asyncio.open_connection(domain_name,port)
100     tasks = [read_trans(reader, writer_remote), write_trans(reader_remote, writer
101     await asyncio.wait(tasks)
102
103 async def test(reader, writer):
```

```
104         first = await reader.read(1)
105         if(first == b'\x05'):
106             await socks5(first, reader, writer)
107         elif(first == b'C'):
108             await httptunnel(first, reader, writer)
109
110     async def main():
111         server = await asyncio.start_server(test, '0.0.0.0', 10086)
112         async with server:
113             await server.serve_forever()
114
115     asyncio.run(main())
116     import asyncio
117
118     if __name__ == '__main__':
119         pass
```

代码说明（可选）

源代码中不要出现大段的说明注释，如果需要可以可以在本节中加上说明。

Python程序设计作业#4 剑指 Offer 52. 两个链表的第一个公共节点