

# Lab8

## 1 Lab8 Network Programming and Automation Bas

---

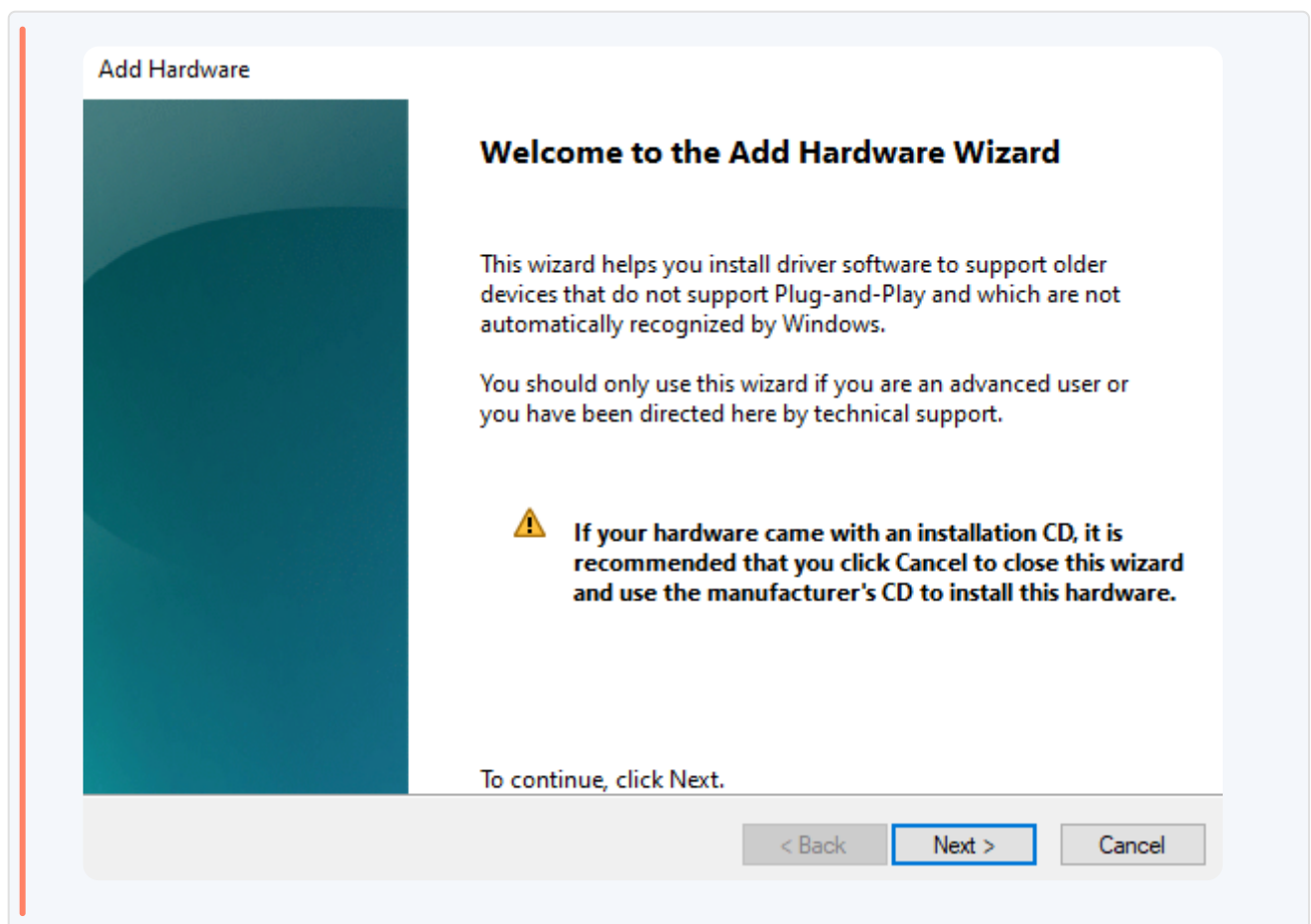
### 1.1 Connecting PC to eNSP

---

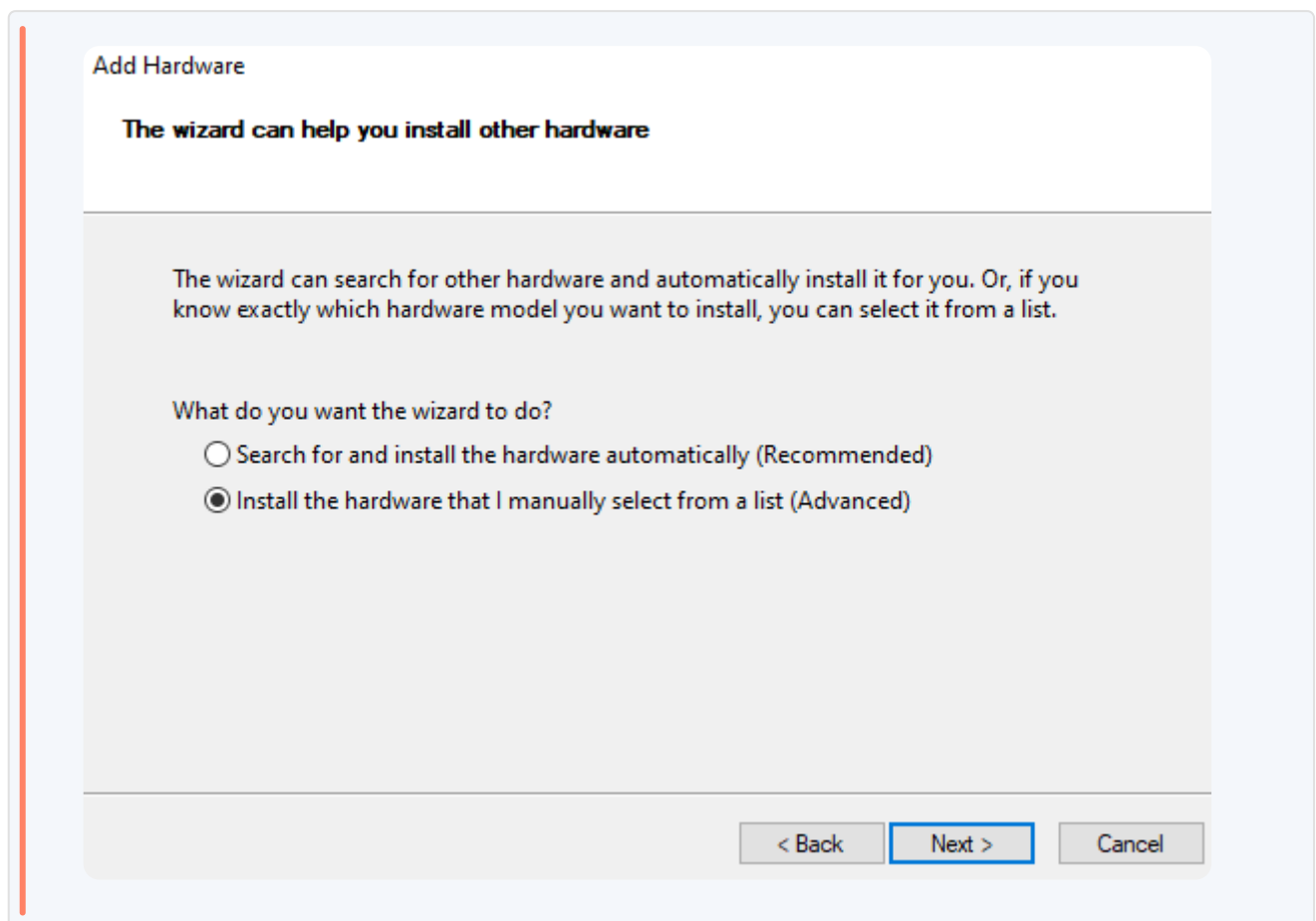
This section is about establishing a connection between your physical computer and the eNSP simulation environment.

#### 1.1.1 Add a microsoft loopback interface

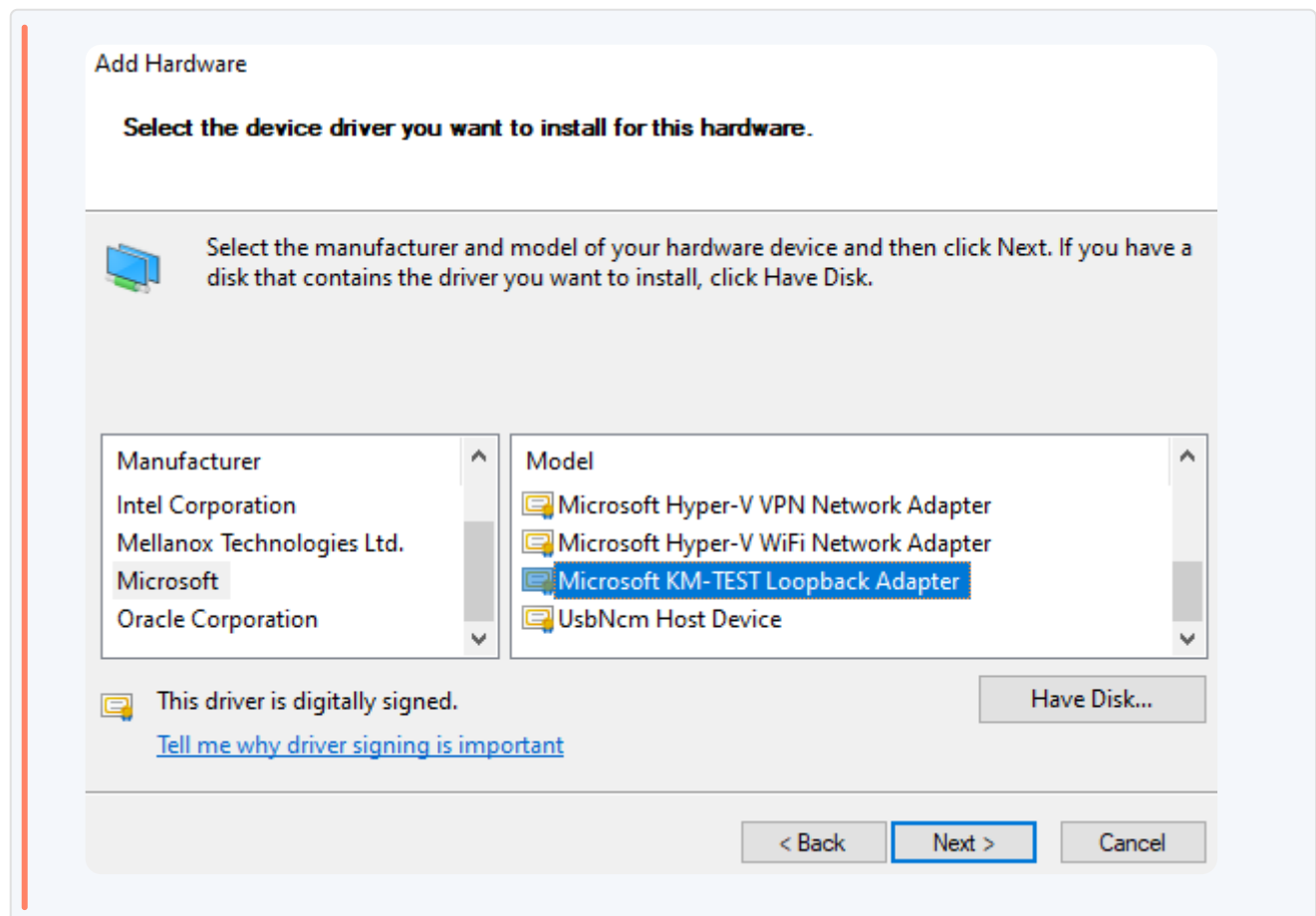
---



From device manger then select your device then action then add legacy hardware



Then choice Network adapter



After initiating the addition of legacy hardware, you would select "Network adapter" from the list of hardware types. This will allow you to manually install a network adapter that can be used within the eNSP simulation.

## 1.1.2 Rename the network adapter to eNSP

---

#### Related settings

[Change adapter options](#)

[Change advanced sharing options](#)

[Network and Sharing Center](#)

[Windows Firewall](#)



[Give feedback](#)



eNSP

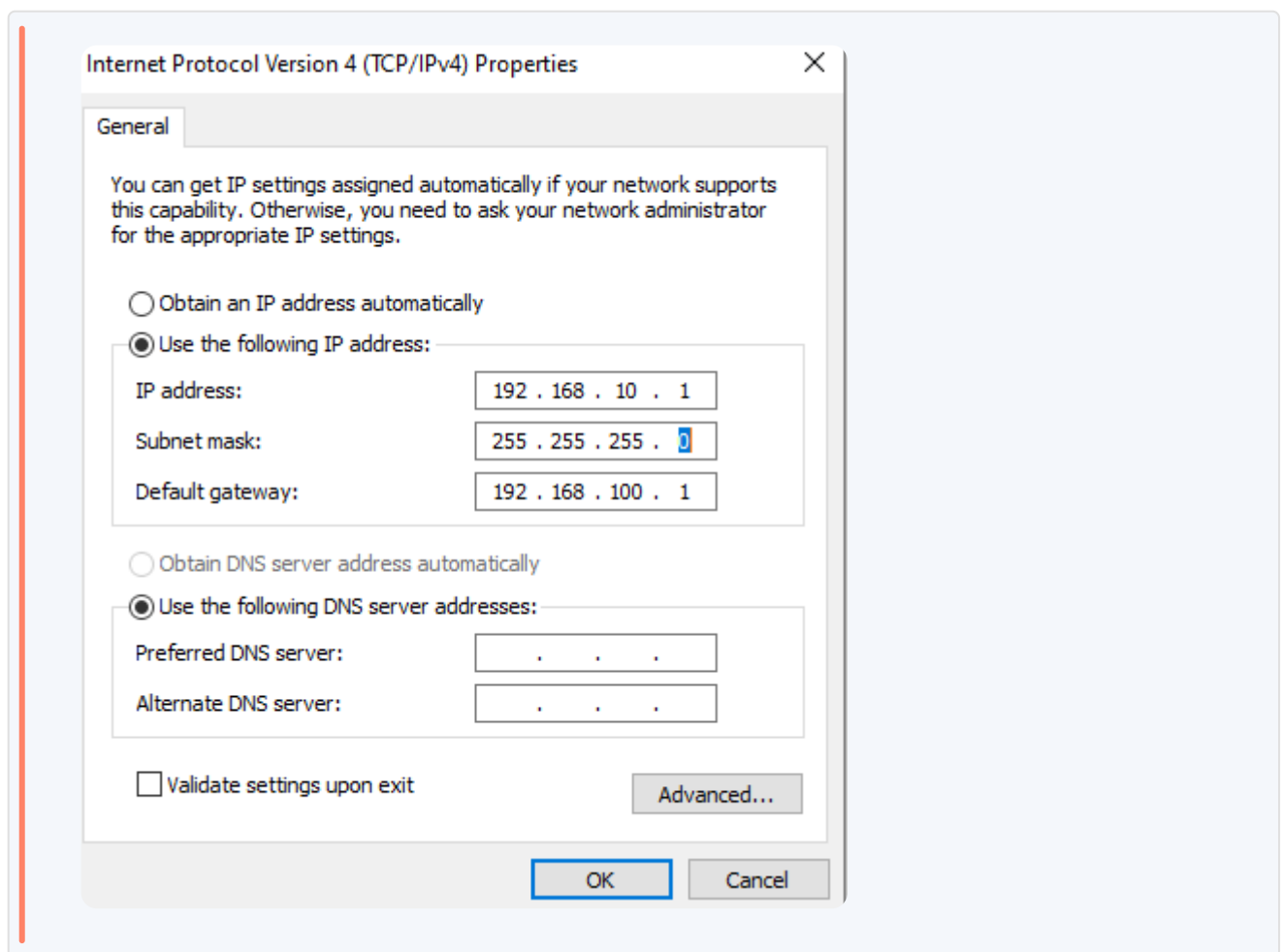
Unidentified network

Microsoft KM-TEST Loopback Ad...

Once the network adapter is installed, you are instructed to rename it to "eNSP." This step is probably done for easier identification when configuring network simulations in eNSP.

### 1.1.3 Put at IP address 192.168.10.2/24

---



## 1.1.4 Open eNSP and connect the cloud to the router

---

IO Config

Port Building

BindingInfo:

UDP

Warning: Please don't bind the public network,otherwise maybe breakdown.

Port Type:

GE

☐ Public UDP Port

Listening Port:

30000

Suggestion: (30000-35000)

Peer IP:

0 . 0 . 0 . 0

Peer Port:

0

Modify

Add

Delete

No.	Port Type	Port Num	UDP Port	Port Open Status	Binding Info
1	GE	1	None	Public	eNSP -- IP: 192.168.10.1
2	GE	2	5380	Internal	UDP

Port Map Setting

Port Type:

GE

Local Port Num:

2

Remote Port Num:

1

☒ Two-way Channel

The mapping has been in list

Add

Port Mapping

No.	Local Port Num	Remote Port Num	Port Type
1	1	2	GE
2	2	1	GE

Delete

## 1.1.5 Put an IP address on the router 192.168.10.2/24

Make sure loopback for physical device in same domain as in ensp environment

## 1.1.6 ping from your pc to router

```
C:\Users\Bakaito>ping 192.168.10.101

Pinging 192.168.10.101 with 32 bytes of data:
Reply from 192.168.10.101: bytes=32 time=2ms TTL=255
Reply from 192.168.10.101: bytes=32 time=9ms TTL=255
Reply from 192.168.10.101: bytes=32 time=5ms TTL=255
```

## 1.2 Introduction

## 1.2.1 About This Lab

---

After completing this lab activity, you will learn how to use Python `telnetlib` for network automation.

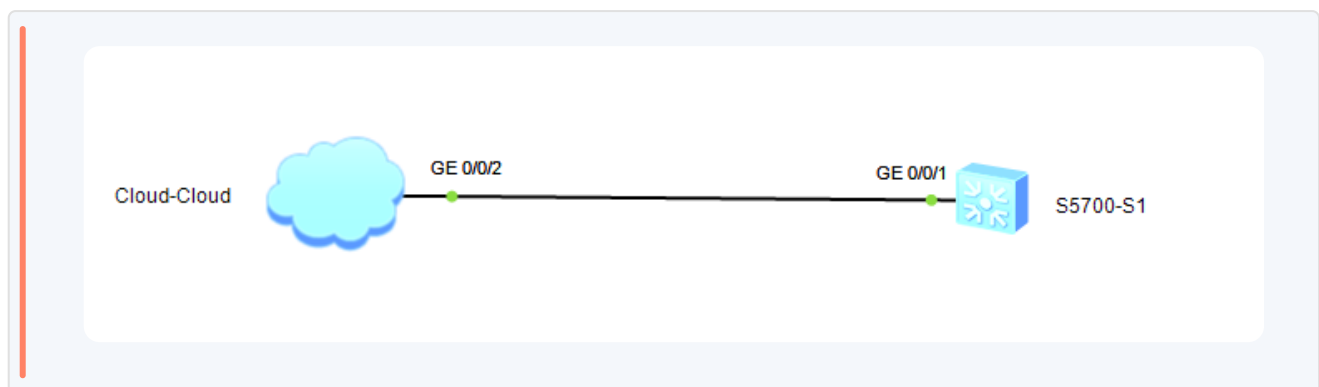
## 1.2.2 Objectives

---

- Understand basic Python syntax.
- Learn to use `telnetlib` for network tasks.

## 1.2.3 Networking Topology

---



A company's switch has a management IP address of `192.168.56.101/24`. The task is to automate viewing the current configuration file of the device.

## 1.3 Lab Configuration

---

### 1.3.1 Configuration Roadmap

---

1. **Configure Telnet:** Set up Telnet access with a password.

2. **Compile a Python script:** Use `telnetlib` to log into the device and retrieve its configuration.

## 1.3.2 Configuration Procedure

---

### 1.3.2.1 Step 1: Configure ip address on interface

---

```
1 [S1]interface vlanif1
2 [S1-vlanif1]ip address 192.168.10.101 24
```

### 1.3.2.2 Step 2: Configure Telnet on the Switch

---

```
1 [S1]user-interface vty 0 4
2 [S1-ui-vty0-4]authentication-mode password
3 [S1-ui-vty0-4]set authentication password simple Huawei@123
4 [S1-ui-vty0-4]protocol inbound telnet
5 [S1-ui-vty0-4]user privilege level 15
6 [S1]q
7 [S1]telnet server enable
```

Set up a Telnet login password.

Enable Telnet service for access.

### 1.3.2.3 Step 3: Write the Python Code





Python



```
1  import telnetlib
2  import time
3
4  # Define the host and password for your device
5  host = '192.168.56.101'
6  password = 'Huawei@123'
7
8  # Establish a connection to the host using Telnet on
  default port 23
9  tn = telnetlib.Telnet(host)
10
11 # Read until the password prompt appears
12 tn.read_until(b"Password:")
13
14 # Send the password followed by a newline character to
  simulate pressing Enter
15 tn.write(password.encode('ascii') + b"\n")
16
17 # Send command to display current configuration on the
  device
18 tn.write(b'display cu \n')
19
20 # Wait for one second to ensure command execution is
  complete before proceeding
21 time.sleep(1)
22
23 # Read any data available from output buffer, decode it
  from ASCII, and print it out
24 print(tn.read_very_eager().decode('ascii'))
25
26 # Close Telnet session after completing tasks
27 tn.close()
```



Code Interpretation

`telnetlib` for Telnet communication and `time` for pausing the script execution when necessary.

define the IP address of the host (network device) and the corresponding password. We then create a Telnet object ( `tn` ) that connects to this host. The script waits until it encounters the "Password:" prompt before sending over the encoded password with an appended newline character ( `\n` ) to log in.

After successfully logging into the network device, we use `write()` method of our Telnet object ( `tn` ) to issue commands to it. In this case, we send over `"display cu \n"` which is a shorthand command for displaying current configuration settings on Huawei devices. A brief pause is introduced with `time.sleep(1)` to allow time for command execution and output generation. Finally, we read eagerly any available data from output buffer, decode it from ASCII encoding, and print it on console.

The last step involves closing our Telnet session by calling `close()` method on our Telnet object ( `tn` ). This is important because network devices typically have limited VTY (Virtual Teletype) connections available; closing sessions ensures these resources are freed up for other users or processes.

### 1.3.2.4 Step 4: Execute the Compiler

---

Use Jupyter Notebook or any other preferred compiler to run the script.

### 1.3.2.5 Step 5: Output Example

---

```

D:\Shared\Huawei\HCIA\Labs\Lab13>py Script.py
D:\Shared\Huawei\HCIA\Labs\Lab13\Script.py:1: DeprecationWarning: 'telnetlib' is deprecated and slated for removal in Python 3.13
import telnetlib

Info: The max number of VTY users is 5, and the number
      of current VTY users on line is 2.
      The current login time is 2024-04-22 00:02:23.
<S1>display cu
#
sysname S1
#
cluster enable
ntdp enable
ndp enable
#
drop illegal-mac alarm
#
diffserv domain default
#
drop-profile default
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme default
domain default
domain default_admin
local-user admin password simple admin
local-user admin service-type http
#
interface Vlanif1
ip address 192.168.56.101 255.255.0.0
---- More ----

D:\Shared\Huawei\HCIA\Labs\Lab13>
D:\Shared\Huawei\HCIA\Labs\Lab13>

```

## 1.4 Quiz Questions for Revision

### ? Question1&2

**Q1:** How do you use `telnetlib` to configure a device, such as setting the IP address of its management interface?

**Q2:** How do you save the configuration file to a local directory?



Python



```

1  import telnetlib
2  import time
3
4  # Set variables for the host IP address and host
5  host = '192.168.56.101'
6  password = 'huawei'
7
8  # Create a Telnet session to the host.
9  tn = telnetlib.Telnet(host)
10 tn.read_until(b"Password:")

```

```
11
12 # Send the password followed by a newline character to
    log in.
13 tn.write(password.encode('ascii') + b"\n")
14
15 # Enter system view mode on the device by sending
    "system-view".
16 tn.write(b'system-view \n')
17
18 # Select interface `gig0/0/1` by sending "interface
    gig0/0/1".
19 tn.write(b'interface gig0/0/1 \n')
20
21 # Assign IP address `192.168.56.101` to that interface
    with "ip address 192.168.56.101".
22 tn.write(b'ip address 192.168.56.101 \n')
23
24 # Exit from interface configuration mode with "q"
    (quit).
25 tn.write(b'quit \n')
26
27 # Save the configuration changes with "save".
28 tn.write(b'save \n')
29 time.sleep(1)
30
31 # Close the Telnet connection.
32 tn.close()
```