**Network O&M Technologies**

Student Version



Huawei Technologies Co., Ltd.

|  |
| --- |
| **Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.**  No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.  **Trademarks and Permissions**  HW_POS_RBG_Vertical-150ppi.png and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.  All other trademarks and trade names mentioned in this document are the property of their respective holders.  **Notice**  The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.  The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied. |

|  |  |
| --- | --- |
| Huawei Technologies Co., Ltd. | |
| Address: | Huawei Industrial Base  Bantian, Longgang  Shenzhen 518129  People's Republic of China |
| Website: | <https://e.huawei.com/> |

**Huawei Certification System**

Huawei Certification follows the "platform + ecosystem" development strategy, which is a new collaborative architecture of ICT infrastructure based on "Cloud-Pipe-Terminal". Huawei has set up a complete certification system consisting of three categories: ICT infrastructure certification, platform and service certification, and ICT vertical certification. It is the only certification system that covers all ICT technical fields in the industry. Huawei offers three levels of certification: Huawei Certified ICT Associate (HCIA), Huawei Certified ICT Professional (HCIP), and Huawei Certified ICT Expert (HCIE). Huawei Certification covers all ICT fields and adapts to the industry trend of ICT convergence. With its leading talent development system and certification standards, it is committed to fostering new ICT talent in the digital era, and building a sound ICT talent ecosystem.

Huawei Certified ICT Associate-Datacom (HCIA-Datacom) is designed for Huawei's frontline engineers and anyone who want to understand Huawei's datacom products and technologies. The HCIA-Datacom certification covers routing and switching principles, basic WLAN principles, network security basics, network management and O&M basics, SDN and programmability and automation basics.

The Huawei certification system introduces the industry, fosters innovation, and imparts cutting-edge datacom knowledge.



# Network O&M Technologies

## Background

A company has more than 20 network devices such as Huawei switches and routers. A network administrator needs to inspect these network devices every day (including the card status, CPU usage, CPU temperature, and system alarm logs). In this way, the administrator can learn about the device working status, prevent faults, and detect faults in a timely manner to ensure that network devices work properly. As no automatic monitoring software is available, the network administrator needs to manually log in to each network device through SSH to check their health status, which is time-consuming and inefficient. To improve inspection efficiency, the administrator plans to use the Python programming language to compile simple automated inspection scripts.

## Objectives

Upon completion of this task, you will be able to:

Know how to configure a Python programming environment.

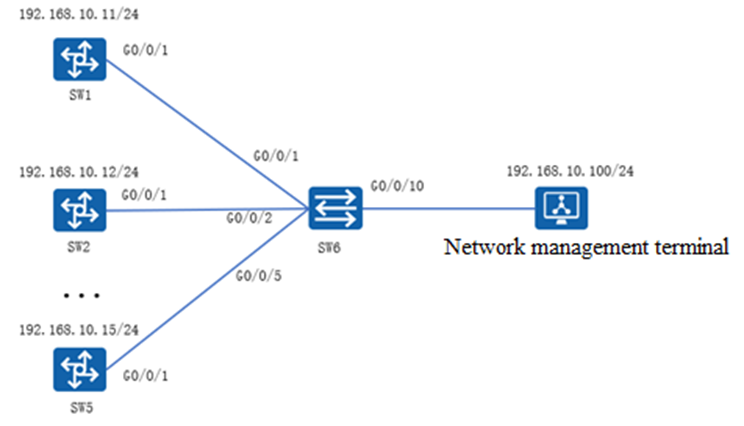
Understand the method to use the Python language.

Understand the method to use the Paramiko module.

Know how to use Python to implement simple automated inspection.

## Topology

Lab topology



In this topology, only some switches are simulated. A network management terminal is used to perform automated inspection for switches SW1 through SW5.

## Implementation

### Roadmap

1. Configure remote management IP addresses on switches SW1 through SW5.
2. Enable SSH on switches SW1 through SW5.
3. Install the Python software to set up the programming environment on the network management terminal.
4. Compile a file containing information about devices to inspect.
5. Compile an inspection command script.
6. Compile an inspection program.

### Procedure

Configure IP addresses.

#Configure remote management IP addresses on switches SW1 through SW5. The following uses SW1 as an example.

<SW1>system

[SW1]interface Vlanif 1

[SW1-Vlanif1]ip address 192.168.10.11 24

[SW1-Vlanif1]quit

[SW1]interface GigabitEthernet 0/0/1

[SW1-GigabitEthernet0/0/1]port link-type trunk

[SW1-GigabitEthernet0/0/1]port trunk allow-pass vlan all

Enable SSH.

[SW1]undo telnet server enable

Warning: The operation will stop the Telnet server. Continue? [Y/N]:y

[SW1]rsa local-key-pair create

The key name will be: SW1\_Host

The range of public key size is (512 ~ 2048).

NOTES: If the key modulus is greater than 512,

it will take a few minutes.

Input the bits in the modulus[default = 512]:2048

Generating keys...

................................................................................

..+++

.............................................+++

.++++++++

.........................++++++++

[SW1]user-interface vty 0 4

[SW1-ui-vty0-4]authentication-mode aaa

[SW1-ui-vty0-4]protocol inbound ssh

[SW1-ui-vty0-4]quit

[SW1]aaa

[SW1-aaa]local-user admin password cipher Huawei@123

[SW1-aaa]local-user admin service-type ssh

[SW1-aaa]quit

[SW1]ssh user admin service-type stelnet

Info: Succeeded in adding a new SSH user.

[SW1]ssh user admin authentication-type password

[SW1]stelnet server enable

Info: Succeeded in starting the Stelnet server.

[SW1]quit

<SW1>save

The current configuration will be written to the device.

Are you sure to continue?[Y/N]y

Info: Please input the file name ( \*.cfg, \*.zip ) [vrpcfg.zip]:

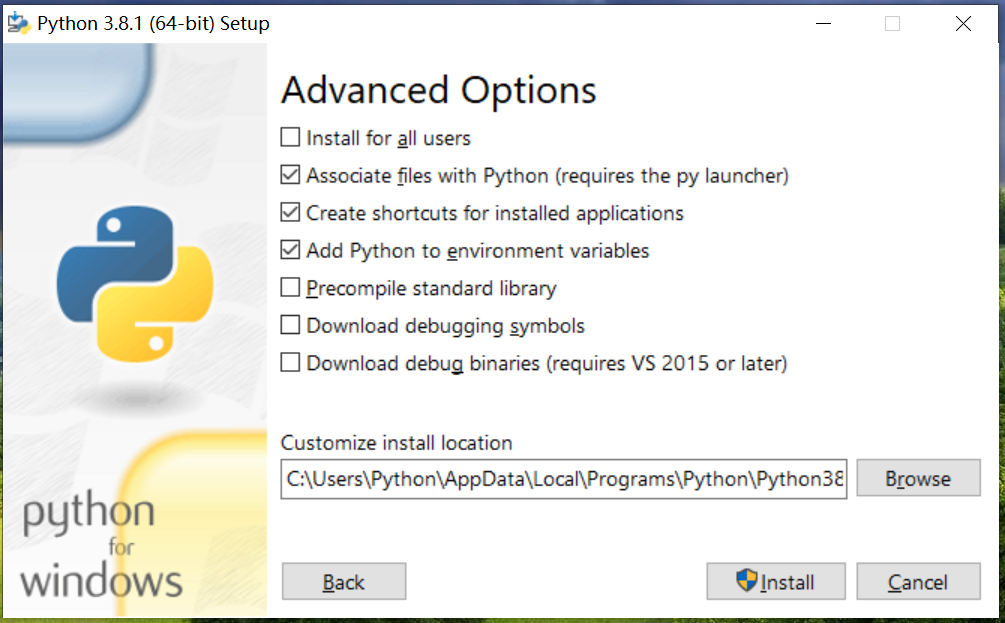
Jan 30 2020 15:25:18-08:00 SW1 %%01CFM/4/SAVE(l)[6]:The user chose Y when decidi

ng whether to save the configuration to the device.

Install the software, for example, in the Windows operating system.

#Download the [Python installation package for Windows](https://www.python.org/downloads/) from the Python official website, and double-click the installation package to install the Python software. Select **Customize installation**. Note that you need to select **Add Python to environment variables** in the **Advanced Options** step and retain the default settings in other steps. After the installation is complete, run the **Python** command in the command prompt window. The Python CLI is displayed.

Setting environment variables



Compile a file containing information about devices to inspect.

#Write device information (including the IP address, user name, and password) into the **device\_info.txt** file. When the inspection program is executed, the program reads device information from this file to connect to the device. In the **device\_info.txt** file, the first field is the device management address, the second field is the SSH login user name, and the third field is the login password. Use commas (,) to separate the fields. The following lists only part of the device information. To add more devices to be inspected, add their information to the file.

192.168.10.11,admin,Huawei@123

192.168.10.12,admin,Huawei@123

192.168.10.13,admin,Huawei@123

192.168.10.14,admin,Huawei@123

192.168.10.15,admin,Huawei@123

Compile an inspection command script.

#Write the inspection commands into the **cmd.txt** script. To add inspection items, add the corresponding commands to the script. After the inspection program is executed and connected to the device, the program reads the inspection commands in the **cmd.txt** script and collects device information.

display device

display environment

display alarm urgen

display memory-usage

display cpu-usage

display logbuffer level 0

display logbuffer level 1

display logbuffer level 2

display logbuffer level 3

display logbuffer level 4

Compile an inspection program.

#Import the Paramiko module.

import paramiko

import time

#Read information about inspected devices.

dev\_filepath = r"d:\Python\device\_info.txt"

dev\_file = open(dev\_filepath,"r")

while 1:

##Read one line of information in the file each time.

dev\_info = dev\_file.readline()

if not dev\_info :

break

else :

##Read the IP address, user name, and password of the device and assign values to the variables.

devs = dev\_info.split(',')

ip = devs[0]

username = devs[1]

password = devs[2].strip()

password = password.strip('\n')

#Read the inspection commands.

cmd\_filepath = r"d:\Python\cmd.txt"

cmd\_file = open(cmd\_filepath,"r")

cmds = cmd\_file.readlines()

#Remotely connect to the device.

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname=ip,username=username,password=password)

print("Connected",ip)

#Send an inspection command.

command = ssh.invoke\_shell()

for line in cmd:

command.send(cmd+'\n')

#Save inspection results.

output=command.recv(65535)

str\_output=str(output)

outprint=str\_output.strip().split('\\r\\n')

log=open("d:\Python\py\xunjian\\"+ip+".txt",'a')

log.write(start\_info+'\n\n'+str(outprint)+'\n\n'+end\_info)

log.close()

#Improve and generate a complete inspection program.

#-\*- coding:UTF-8 -\*-

import paramiko

import time

starttime = time.strftime('%Y-%m-%d %T')

start\_info = "Inspection start time:"+str(starttime)

cmd\_filepath = r"d:\Python\py\xunjian\cmd.txt"

cmd\_file = open(cmd\_filepath,"r")

cmds = cmd\_file.readlines()

dev\_filepath = r"d:\Python\py\xunjian\device\_info.txt"

dev\_file = open(dev\_filepath,"r")

while 1:

dev\_info = dev\_file.readline()

if not dev\_info :

break

else :

devs = dev\_info.split(',')

ip = devs[0]

username = devs[1]

password = devs[2].strip()

password = password.strip('\n')

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname = ip,username = username,password = password)

print("Connected",ip)

command = ssh.invoke\_shell()

time.sleep(3)

command.send('N\n') #This line is optional.

command.send('screen-length 0 temporary\n') #Cancel the split-screen display.

for cmd in cmds:

command.send(cmd+'\n')

time.sleep(5)

output = command.recv(65535).decode()

log = open(r"d:\Python\py\xunjian\\"+ip+".txt",'a')

endtime = time.strftime('%Y-%m-%d %T')

end\_info = "Inspection end time:"+str(endtime)

log.write(start\_info+'\n\n'+output+'\n\n'+end\_info)

log.close()

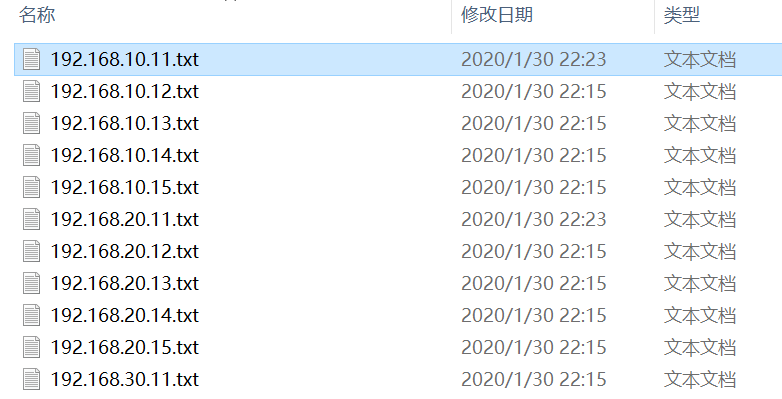
dev\_file.close()

* 1. **Verification**

#### Checking Generated Inspection Files

Run the inspection program. After the inspection is complete, inspection files named after the device IP addresses are generated, as shown in Figure 1-3.

Files generated during the inspection



#### Checking the Contents of Inspection Files

The inspection file of SW1 (IP address: 192.168.10.11) contains the following contents:

Inspection start time: 2020-01-30 22:23:31

' ---------------------------------------------',

'User last login information: ',

'-----------------------------------------------',

'Access Type: SSH',

'IP-Address : 192.168.10.11 ssh',

'Time: 2020-01-30 22:23:55-08:00',

' ---------------------------------------------',

'<SW1>display device',

"S3700-26C-HI's Device status:",

'Slot Sub Type Online Power Register Status Role ',

'-----------------------------------------------------------------------',

'0 - 3726C Present PowerOn Registered Normal Master',

'<SW1>display environment',

'Environment information:',

'Temperature information:',

'SlotID CurrentTemperature LowLimit HighLimit',

' (deg c ) (deg c) (deg c )',

'0 0 0 70',

'<SW1>display alarm urgen',

'Alarm: ',

'Alarm Slot Date Time Location',

'-----------------------------------------------------',

'Temp low 0 2020/01/30 22:23:59 Slot 0',

'Fan abnormal 0 2020/01/30 22:23:59 Slot 0',

'<SW1>display memory-usage',

......

Inspection end time: 2020-01-30 22:23:37