

LAB 20: DÙNG PYTHON LẤY DANH SÁCH THÔNG TIN CÁC THIẾT BỊ TRONG FABRIC SD-WAN CỦA CISCO

I. Mô tả:

- Học viên thực hiện kết nối đến Sandbox SD-WAN (vManage), viết code để lấy danh sách thông tin các thiết bị bằng ngôn ngữ Python.
- Máy PC phải đáp ứng yêu cầu đã cài đặt trạm làm việc cho developer.

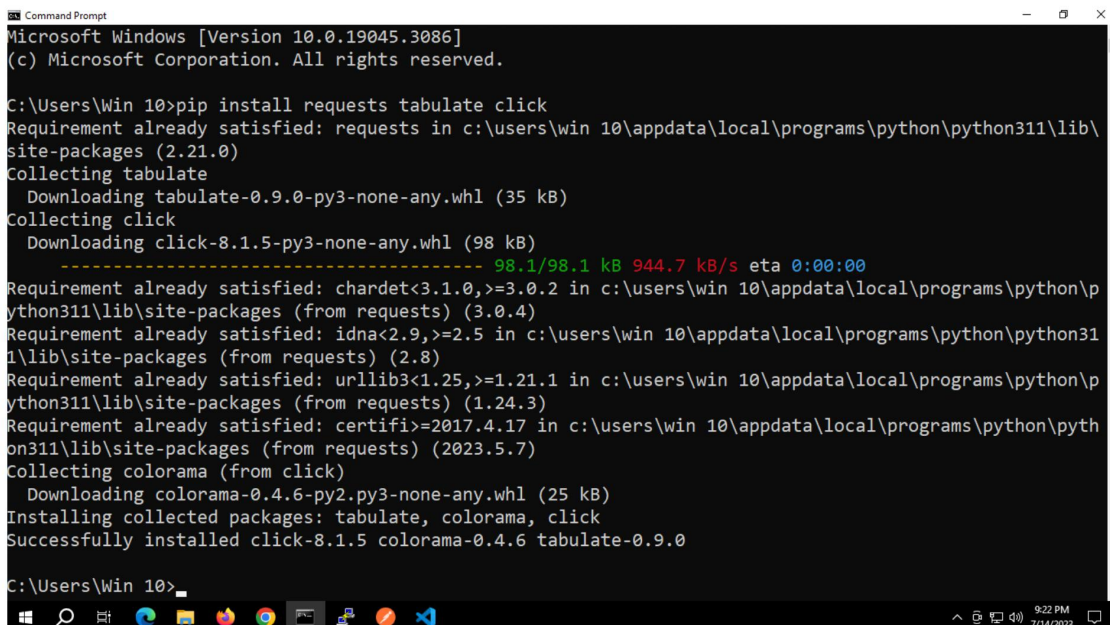
II. Yêu cầu kỹ thuật:

- Cài đặt thư viện requests, tabulate, click trên máy tính.
- Kết nối đến Sandbox SD-WAN (<https://10.215.26.217:8443/>)
- Viết code bằng Python thực hiện yêu cầu:
 - Đăng nhập và xác thực
 - GET requests, POST requests
 - Lấy danh sách thông tin các thiết bị trong Controller

III. Các bước thực hiện:

Bước 1: Cài đặt thư viện

- Bấm tổ hợp phím Win+R để chạy cmd
- Trong màn hình cmd gõ : pip install requests tabulate click



```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Win 10>pip install requests tabulate click
Requirement already satisfied: requests in c:\users\win 10\appdata\local\programs\python\python311\lib\
site-packages (2.21.0)
Collecting tabulate
  Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Collecting click
  Downloading click-8.1.5-py3-none-any.whl (98 kB)
----- 98.1/98.1 kB 944.7 kB/s eta 0:00:00
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\win 10\appdata\local\programs\python\p
ython311\lib\site-packages (from requests) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\win 10\appdata\local\programs\python\python31
1\lib\site-packages (from requests) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in c:\users\win 10\appdata\local\programs\python\p
ython311\lib\site-packages (from requests) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\win 10\appdata\local\programs\python\pyth
on311\lib\site-packages (from requests) (2023.5.7)
Collecting colorama (from click)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: tabulate, colorama, click
Successfully installed click-8.1.5 colorama-0.4.6 tabulate-0.9.0

C:\Users\Win 10>
```

Bước 2: Viết code đăng nhập và xác thực

Import các thư viện cần thiết và tắt cảnh báo

```
import requests
import sys
import json
import click
from tabulate import tabulate
import SD_WAN_INFO

requests.packages.urllib3.disable_warnings()
```

Tạo file SD_WAN_INFO.py chứa thông tin kết nối và thông tin đăng nhập, ghi các thông tin như hình dưới, sau đó lưu lại và đóng file

```
C: > Users > Win 10 > Desktop > SD_WAN_INFO.py > ...
1 IP = "10.215.26.217"
2 USERNAME = "admin"
3 PASSWORD = "Admin"
```

Trở về file trước, ta khai báo các thông tin của Sandbox SD-WAN, các biến SDWAN_IP, USERNAME, PASSWORD sẽ lấy giá trị từ file SD_WAN_INFO.py đã tạo ở trên.

```
SDWAN_IP = SD_WAN_INFO.IP
SDWAN_USERNAME = SD_WAN_INFO.USERNAME
SDWAN_PASSWORD = SD_WAN_INFO.PASSWORD
```

Tiếp theo chúng ta sẽ tạo một class rest_api_lib và tạo constructor __init__ của class (lưu ý ngoài các tham số truyền vào chúng ta luôn phải thêm self đại diện cho instance của class và với nó chúng ta có thể kết nối đến các thuộc tính và phương thức của class đó.

```
class rest_api_lib:
    def __init__(self, vmanage_ip, username, password):
        self.vmanage_ip = vmanage_ip
        self.session = {}
        self.login(self.vmanage_ip, username, password)
```

Định nghĩa phương thức login và khai báo login_url

```
def login(self, vmanage_ip, username, password):
    """Login to vmanage"""
    base_url_str = 'https://%s:8443/%s' % (vmanage_ip, vmanage_ip)
    login_action = '/j_security_check'
    login_url = base_url_str + login_action
```

Khai báo login_data chứa username và password để gửi lên xác thực

```
login_data = {'j_username': username, 'j_password': password}
```

Chúng ta sẽ dùng phương thức session từ thư viện request để tạo một phiên làm việc mới, trong phiên làm việc vừa tạo đó gửi yêu cầu post để đưa thông tin đến login_url.

```
sess = requests.session()
login_response = sess.post(url=login_url, data=login_data, verify=False)
```

Để đảm bảo xác thực thành công chúng ta sẽ kiểm tra nội dung trả về và nếu nội dung trả về có tag <html/> thì nghĩa là đăng nhập thất bại. Nếu muốn xem đăng nhập thất bại thì nội dung trả về sẽ như thì sửa lại mật khẩu và bỏ dấu # đầu dòng của đoạn code dưới này.

```
if b'<html>' in login_response.content:
    print ("Login Failed")
#print(login_response.content)
```

Gắn session sess vào làm giá trị của thuộc tính session của class

```
self.session[vmanage_ip] = sess
```

Bước 3: Viết code GET requests

Định nghĩa phương thức get_request, trong đây chúng ta sẽ phải tạo một url mới, tham số api truyền vào tùy mục đích sử dụng nên chúng ta sẽ dùng %s để url có thể thay đổi dễ dàng. Ta sẽ dùng tiếp session vừa được xác thực thành công để gọi các API tiếp theo mà không cần phải gửi kèm theo username, password; get để gửi yêu cầu lên server và nhận lời đáp lại, sau đó gắn thông tin từ lời đáp lại vào biến response. Để lấy dữ liệu, ta sẽ dùng response.content gắn vào biến data và đây cũng là giá trị trả về của phương thức get_request này.

```
def get_request(self, api):
    url = "https://%s:8443/dataservice/%s"%(self.vmanage_ip, api)

    response = self.session[self.vmanage_ip].get(url, verify=False)
    data = response.content
    return data
```

Bước 4: Viết code POST requests

Phương thức post_request này tương tự với phương thức get_request ở trên nhưng khác ở chỗ post dùng để gửi yêu cầu tạo tài nguyên mới trên server. Và có thêm tham số đầu vào là payload và headers. Payload là nơi sẽ chứa đựng các thông tin gửi kèm theo khi gửi yêu cầu post. Headers khai báo kiểu nội dung là application/json.

```
def post_request(self, api, payload, headers={'Content-Type': 'application/json'}):
    url = "https://%s:8443/dataservice/%s"%(self.vmanage_ip, api)
    payload = json.dumps(payload)
    print(payload)
```

```
response = self.session[self.vmanage_ip].post(url=url, data=payload,
headers=headers, verify=False)
data = response.json()
return data
```

Bước 5: Viết code lấy danh sách thông tin các thiết bị

Tạo instance của class là sdwanp và truyền các tham số SDWAN_IP, SDWAN_USERNAME, SDWAN_PASSWORD

```
sdwanp = rest_api_lib(SDWAN_IP, SDWAN_USERNAME, SDWAN_PASSWORD)
```

Khi muốn cấu hình nhiều decorator @click.command() thì phải sử dụng group() để tạo nhiều decorator trong cùng một đoạn script.

```
@click.group()
def cli():
    pass
```

Tạo decorator @click.command() cấu hình Click để làm việc với hàm Python ngay sau decorator. Click.echo() dùng để in ra màn hình. Json.loads() dùng để giải mã đối tượng json thành đối tượng trong python.

```
@click.command()
def device_list():
    """Retrieve and return network devices list."""
    click.echo("Retrieving the devices.")

    response = json.loads(sdwanp.get_request('device'))
    items = response['data']
```

Khai báo headers để hiển thị trong bảng và khai báo list table. Tạo vòng lặp cứ mỗi item trong biến items chứa dữ liệu này thì sẽ được ghi thêm vào list table. Tiếp theo chúng ta sẽ dùng hàm tabulate để xuất ra màn hình bảng danh sách thông tin các thiết bị trong fabric của SD-WAN.

```
headers = ["Host-Name", "Device Type", "Device ID", "System IP", "Site ID", "Version",
"Device Model"]
table = list()

for item in items:
    tr = [item['host-name'], item['device-type'], item['uuid'], item['system-ip'], item['site-id'], item['version'], item['device-model']]
    table.append(tr)
try:
```



```
click.echo(tabulate(table, headers, tablefmt="fancy_grid"))  
except UnicodeEncodeError:  
    click.echo(tabulate(table, headers, tablefmt="grid"))
```

Cuối cùng là thêm command device_list vào cli() và viết hàm main

```
cli.add_command(device_list)  
  
if __name__ == "__main__":  
    cli()
```

Để chạy chương trình này, chúng ta sẽ vào cmd, đi đến thư mục đặt file python, chạy chương trình python .sdwan.py

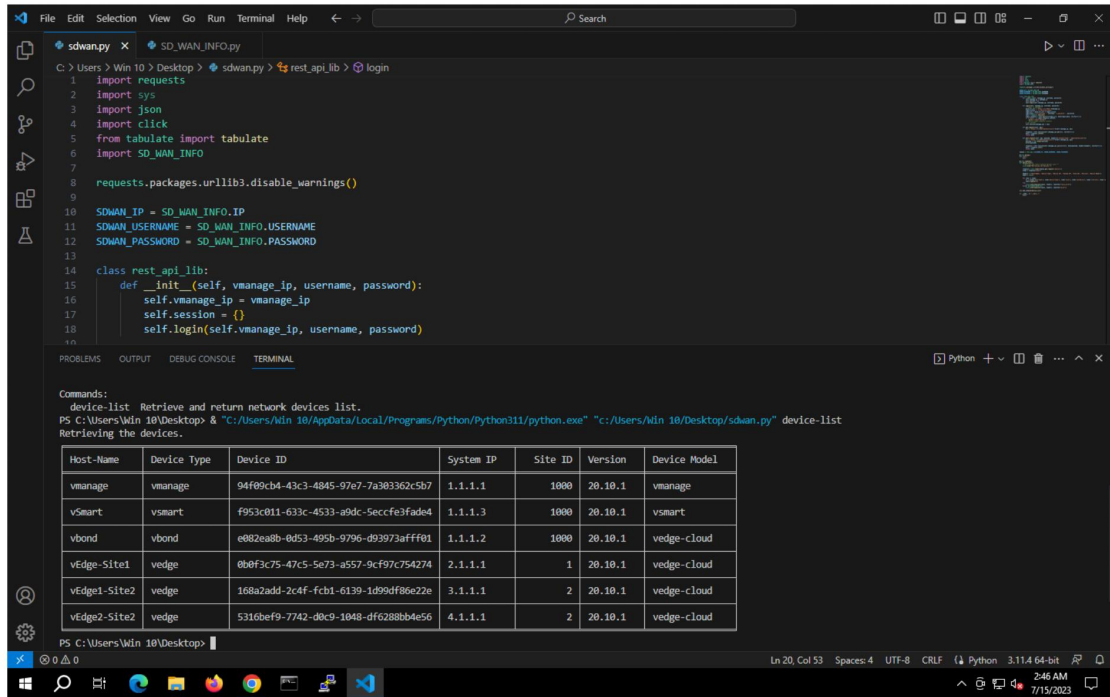
```
PS C:\Users\Win 10\Desktop> python .\sdwan.py  
Usage: sdwan.py [OPTIONS] COMMAND [ARGS]...  
  
Options:  
  --help  Show this message and exit.  
  
Commands:  
  device-list  Retrieve and return network devices list.  
PS C:\Users\Win 10\Desktop>
```

Kết quả:

```
PS C:\Users\Win 10\Desktop> & "C:/Users/Win 10/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Win 10/Desktop/sdwan.py"  
Usage: sdwan.py [OPTIONS] COMMAND [ARGS]...  
  
Options:  
  --help  Show this message and exit.  
  
Commands:  
  device-list  Retrieve and return network devices list.  
PS C:\Users\Win 10\Desktop>
```

Để ý phần Usage có hướng dẫn cách sử dụng click, hiện tại commands hiện có là device-list nên trong cmd chúng ta sẽ gõ: python sdwan.py device-list

Kết quả là chúng ta lấy được danh sách các thiết bị hiện có trong mạng fabric SD-WAN.



```
File Edit Selection View Go Run Terminal Help
sdwan.py SD_WAN_INFO.py
C:\Users\Win 10\Desktop> sdwan.py rest_api_lib login
1 import requests
2 import sys
3 import json
4 import click
5 from tabulate import tabulate
6 import SD_WAN_INFO
7
8 requests.packages.urllib3.disable_warnings()
9
10 SDWAN_IP = SD_WAN_INFO.IP
11 SDWAN_USERNAME = SD_WAN_INFO.USERNAME
12 SDWAN_PASSWORD = SD_WAN_INFO.PASSWORD
13
14 class rest_api_lib:
15     def __init__(self, vmanage_ip, username, password):
16         self.vmanage_ip = vmanage_ip
17         self.session = {}
18         self.login(self.vmanage_ip, username, password)
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Commands:

device-list Retrieve and return network devices list.

PS C:\Users\Win 10\Desktop> & "C:\Users\Win 10\AppData\Local\Programs\Python\Python311\python.exe" "c:\Users\Win 10\Desktop\sdwan.py" device-list

Retrieving the devices.

Host-Name	Device Type	Device ID	System IP	Site ID	Version	Device Model
vmanage	vmanage	94f09cb4-43c3-4845-97e7-7a303362c5b7	1.1.1.1	1000	20.10.1	vmanage
vSmart	vsmart	f953c011-633c-4533-a9dc-5eccfe3fade4	1.1.1.3	1000	20.10.1	vsmart
vbond	vbond	e082ea8b-0d53-495b-9796-d93973afff01	1.1.1.2	1000	20.10.1	vedge-cloud
vEdge-Site1	vedge	0b0f3c75-47c5-5e73-a557-9cf97c754274	2.1.1.1	1	20.10.1	vedge-cloud
vEdge1-Site2	vedge	168a2add-2c4f-fcb1-6139-1d99df86e22e	3.1.1.1	2	20.10.1	vedge-cloud
vEdge2-Site2	vedge	5310bef9-7742-d0c9-1048-df6288bb4e56	4.1.1.1	2	20.10.1	vedge-cloud

PS C:\Users\Win 10\Desktop>