

LAB - VIẾT CHƯƠNG TRÌNH DÙNG NGÔN NGỮ PYTHON ĐỂ THỰC HIỆN LẤY TICKET TỪ NETWORK CONTROLLER

I. Sơ đồ mạng



Hình 1: Sơ đồ bài Lab

II. Mô tả

Sơ đồ bài thực hành gồm Network Controller và một PC được đấu nối với nhau như hình vẽ. Học viên sẽ viết chương trình Python kết nối đến Network Controller để lấy ticket. Mỗi khi ta cần truy cập tài nguyên hay dịch vụ trên Network Controller, ta cần sử dụng ticket. Mỗi phiên làm việc sẽ có một ticket riêng. Máy PC phải đáp ứng yêu cầu là đã cài đặt các phần mềm cần thiết của developer (Python, Atom, Netmiko).

III. Yêu cầu kỹ thuật

Học viên thực hiện kết nối máy tính đến Network Controller, sau đó cài đặt thư viện requests trên máy tính. Viết code bằng Python thực hiện yêu cầu lấy ticket từ Network Controller. Viết hàm get để lấy thông tin từ Network Controller.

IV. Các bước thực hiện

Bước 1: Cấu hình máy tính ping thành công đến Network Controller

Máy PC để IP động (dùng DHCP) là có thể ping được Network Controller ở địa chỉ 10.215.26.10.

Bước 2: Cài đặt thư viện requests

Bấm tổ hợp phím Windows +R để mở Command Line (cmd).

- Trong màn hình cmd dùng lệnh: `python -m pip install requests --user:`

```
C:\Windows\system32\cmd.exe
C:\Users\pmkdr>python -m pip install requests --user
Collecting requests
  Downloading https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0b2a00aba99ae0f828be89d72b2190f27c11d4b7fb
/requests-2.22.0-py2.py3-none-any.whl (57kB)
    |#####| 61kB 787kB/s
Collecting certifi>=2017.4.17 (from requests)
  Downloading https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744fa380bc73ca47303ccc8b9190fd16a827281eac2
/certifi-2019.9.11-py2.py3-none-any.whl (154kB)
    |#####| 163kB 2.2MB/s
Collecting urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 (from requests)
  Downloading https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a579c09dd7db825bb730ec1fe9c0180fc77bf31448
/urllib3-1.25.6-py2.py3-none-any.whl (125kB)
    |#####| 133kB 3.2MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8
/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    |#####| 143kB 3.3MB/s
Collecting idna<2.9,>=2.5 (from requests)
  Downloading https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9
/idna-2.8-py2.py3-none-any.whl (58kB)
    |#####| 61kB 3.8MB/s
Installing collected packages: certifi, urllib3, chardet, idna, requests
  WARNING: The script chardetect.exe is installed in 'C:\Users\pmkdr\AppData\Roaming\Python\Python38\Scripts' which is not
on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed certifi-2019.9.11 chardet-3.0.4 idna-2.8 requests-2.22.0 urllib3-1.25.6
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\pmkdr>
```

Bước 3: Viết hàm get để lấy ticket

Tạo file getTicket.py trong thư mục C:\python với nội dung như sau:

```
import requests
import json
```

Thư viện requests giúp gửi các yêu cầu loại http đến server và nhận response. Thư viện JSON cho phép mã hóa, giải mã các đối tượng. Khai báo thông tin APIC-EM.

```
APICEM_IP = '10.215.26.10'
VERSION = 'v1'
USERNAME = 'admin'
PASSWORD = 'VnPro@149'
```

Tắt cảnh báo trang web không có xác thực SSL:

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

Tạo hàm get để lấy ticket:

```
def get_ticket(ip= APICEM_IP, ver= VERSION, uname= USERNAME, pword=
PASSWORD, api='', params='')
```

Tạo một jsonObject có hai key username, password có value tương ứng là uname, pword:

```
r_json = {  
    "username":uname,  
        "password":pword  
}
```

Khởi tạo headers với cấu trúc mặc định như trên, headers này để xác nhận chúng ta sử dụng JSON để truyền dữ liệu giữa client và server:

```
headers = {"Content-Type" : "application/json"}
```

Tạo URL (địa chỉ của controller nhận requests). Ví dụ như `https://10.215.26.10/api/v1/ticket` (trong đó ip=địa chỉ ip của server Network Controller, ver= version của server):

```
post_url = "https://" + APICEM_IP + "/api/" + VERSION + "/ticket"
```

Kiểm tra trong phần try “hãy thực hiện tác vụ A” (do A) nếu có lỗi thì sẽ chuyển tới except và thực hiện các hành động B (do B):

```
try:  
    do A  
except:  
    do B
```

A ở đây cụ thể là:

```
r=requests.post(post_url,data = json.dumps(r_json), headers =  
headers,verify=False)  
r.raise_for_status()      #Lấy mã trạng thái, 200 là thành công,  
                          404 là not found  
token = r.json()["response"]["serviceTicket"]  
    return {  
        "token" : token}
```

Hàm trên ta thực hiện gửi request đến server và sẽ gán response của server vào biến r. Trong đó:

- post là tạo dữ liệu.
- post_url là địa chỉ ta muốn gửi đến.
- json.dumps dùng để mã hóa username và password.
- verify=False để không xác thực SSL.

Sau khi nhận được response, ta tạo biến token để lấy ticket từ r và return để trả về giá trị ticket

B ở đây cụ thể là:

```
print("Status: %s" %r.status_code)
print("Response: %s" %r.text)
```

Để in giá trị ticket kết quả ra màn hình ta thêm vào đoạn code:

```
ticket = get_ticket()
print(ticket)
```

Tạo thêm hàm get để lấy thông tin, truyền các thông số của Network Controller vào hàm get:

```
# Tạo hàm get để lấy thông tin
def get(ip=APICEM_IP, ver=VERSION, uname=USERNAME, pword=PASSWORD,
        api='', params='')
```

Lấy ticket bằng cách gọi hàm get_auth_token đã giới thiệu ở bài lab trước:

```
ticket = get_auth_token(ip, ver, uname, pword)
```

Gắn ticket vừa lấy được vào headers, headers này sẽ được truyền đi kèm theo requests gửi lên server để server kiểm tra ticket có đúng không:

```
headers = {"X-Auth-Token": ticket['token']}
```

Tạo URL (địa chỉ của server nhận requests). Ví dụ như <https://10.215.26.10/api/v1/network-device> (trong đó ip=10.215.26.10, ver= v1, api= network-device).

```
url = "https://" + ip + "/api/" + ver + "/" + api
```

Kiểm tra trong phần try (do A) nếu có lỗi thì sẽ chuyển tới mục except và thực hiện các hành động trong đó (do B):

```
try:
    do A
except:
    do B
```

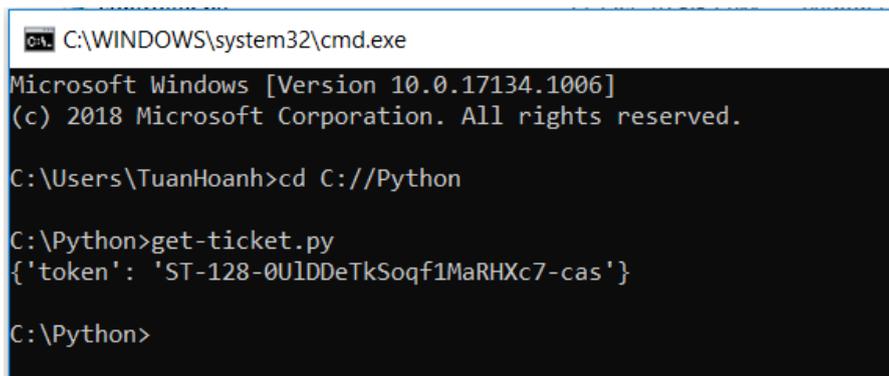
Truyền các tham số vào để gửi đến server bao gồm url, headers, params, verify. Params có thể để trống, verify phải bằng False để bỏ qua phần xác thực SSL:

```
resp=requests.get(url,headers=headers,params=params,verify=False)
print("GET '%s' status" %api,resp.status_code,'\n')
return(resp)
```

Câu lệnh `return(resp)` dùng để gán giá trị trả về của hàm `get`. Ví dụ như `response = get("network-device")` thì kết quả trả về của hàm `get` này sẽ được gán vào biến `response`. Để chạy đoạn chương trình trên, ta vào dấu nhắc `cmd` nhập lệnh như bên dưới:

```
Cd C://pythonGet-ticket.py
```

Kết quả:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1006]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\TuanHoanh>cd C://Python

C:\Python>get-ticket.py
{'token': 'ST-128-0U1DDeTkSoqf1MaRHxc7-cas'}

C:\Python>
```

Network Controller đã trả về ticket thành công. Còn hàm `get` thông tin sẽ được sử dụng trong bài tiếp theo. Vì một ticket chỉ dùng trong 1 phiên làm việc nên không thể tách rời hàm lấy ticket và hàm `get` thông tin thiết bị ra 2 file khác nhau.