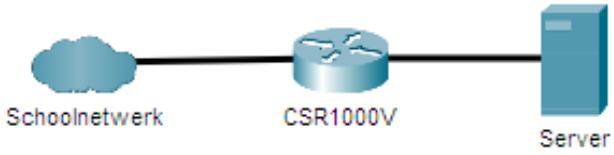
Lab 2 - CLI Automatisering met Python & Netmiko

Topologie



AddresTabel

Device	Interface	IP Address	Subnet Mask	Default Gateway
CSR1000V	G1	DHCP	DHCP	N/A
	G2	172.16.0.254	255.255.255.0	
	Loopback1	172.16.1.1	255.255.255.0	
Server	NIC	172.16.0.2	255.255.255.0	172.16.0.254

Doelstellingen

Deel 1: Vraag vanuit je Python script met de Netmiko bibliotheek informatie op van de router

Deel 2: Pas vanuit je Python script met de Netmiko bibliotheek de configuratie van de router aan

Deel 3: Configureer je opstelling voor gebruik in volgende labo's

Achtergrond / Scenario

Netmiko is een Pythonmodule die je toelaat op een eenvoudige manier je netwerkapparatuur te beheren via een SSH-sessie. Meer info vind je op https://pynet.twb-tech.com/blog/automation/netmiko.html

Opdracht

Deel 1: Informatie opvragen van de router

Stap 1: Test de bereikbaarheid van je router

1. Zoek via onze private cloud het IP-adres dat aan je router toegekend is ('10.129.x.x')
2. Zet indien nodig een VPN-verbinding met het schoolnetwerk
3. Test of je kan pingen naar je router.

Stap 2: 'show ip interface brief' vanuit een Python script

1. Start een Python script in je favoriete Python omgeving (Spyder, PyCharm, Visual Studio Code, Visual Studio 2019…).
   In dat script voer je volgende stappen uit:
2. Importeer de 'ConnectHandler' klasse uit de Netmiko module in je script:

```
from netmiko import ConnectHandler
```

3. Start een SSH-connectie met je server:

```
host = '10.129.x.x'
print("Connecting to {}\n".format(host))
cli = ConnectHandler(
    device_type='cisco_ios',
    host = host,
    port=22,
    username='Automation',
    password='cisco'
    )
```

4. Voer het 'show ip interface brief' commando uit op de router en druk het resultaat af:

```
print("Sending 'sh ip int brief' command...")
output = cli.send_command("show ip int brief")
print("result:\n{}\n".format(output))
lines = output.splitlines()
line = lines[1].split()

print("External IP: {}\n".format(line[1]))
```

5. Test je script (en debug het indien nodig).

Deel 2: Pas vanuit je Python script met de Netmiko bibliotheek de configuratie van de router aan.

Stap 1: Stuur configuratiecommando's naar de router

1. Maak een lijst met configuratiecommando's:

```
config_commands = [
       'int loopback 1',
       'ip address 172.16.1.1 255.255.255.0',

       'description test-interface' ]
```

2. Stuur de configuratiecommando's naar de router en druk het resultaat af:

```
output = cli.send_config_set(config_commands)

print("result:\n{}\n".format(output))
```

3. Schrijf zelf de nodige code om in je script te verifiëren dat het aanmaken van de loopback-interface gelukt is (kijk dat zelf ook na via een interactieve SSH-sessie).

Deel 3: Configureer je opstelling voor gebruik in volgende labo's

Stap 1: Configuratiescript

Schrijf een **Python script** dat je router configureert voor gebruik in de volgende labo-opgaves:

1. Configureer de IP-adressen op de router volgens bovenstaande adressentabel.
2. Schakel de lokale webinterface van de router op poort 80 uit: **'no ip http server'**
3. Configureer Port Forwarding zodat de Webserver van de Server vanop het schoolnetwerk toegankelijk is via het adres van de router (op poort 80).
4. Configureer op de router Port Adres Translation (PAT) via het extern gekregen IP-adres zodat het intern netwerk toegang krijgt tot internet.
5. Voer het script uit (en controleer de werking).

Stap 2: Server

1. Configureer de IP-instellingen van de server volgens bovenstaande adressentabel.
   Stel volgende DNS-servers in: 10.129.28.230 en 10.129.28.232.
2. Test of een ping naar www.odisee.be lukt
3. Zorg ervoor dat er op de server een webserver draait met een homepagina waarin je naam als eenvoudige statische tekst weergegeven wordt:
   - Webserver installeren
   - Service starten
   - Firewall openzetten voor de server
   - Index.html aanpassen
4. Test of je vanaf je thuiscomputer kan surfen naar je webserver (via het extern IP-adres van je router).

Stap 3: Monitoring

1. Door het gebruik van 'Port Forwarding' is de webserver van buitenaf enkel toegankelijk voor HTTP-verkeer naar poort 80. Op zich is dat al een belangrijke beveiliging.
   In een verdere opgave willen we een monitoring opzetten van het aantal webrequests dat er gebeurt naar die webserver.
   Voorzie daarvoor in je Python script op de externe interface een **inbound** ACL met als naam '**monitoring**' en volgende regels (in de vermelde volgorde):
   - Alle 'established' TCP-connections zijn toegelaten.
   - Alle TCP-verkeer naar poort 80 is toegelaten.
   - Alle IP-verkeer is toegelaten.
2. Voer je script uit en test de werking van je access-list: als je van je PC surft naar het IP-adres van de router (via http, niet HTTPS!) en daarna bij de router een '**show access-list monitoring**' ingeeft, krijg je het aantal pakketten dat per regel doorgelaten is. De tweede regel geeft het aantal nieuwe http-connecties dat opgezet is naar de webserver (soms wel met enkele seconden vertraging).
3. Sla je routerconfiguratie op ('copy run start').