LAB 4 – NETWORK INFRASTRUCTURE AND TROUBLESHOOTING

Part 1: Network Infrastructure

Install and configure your lab test environment according to the [network drawing.](https://hogeschoolpxl-my.sharepoint.com/:f:/g/personal/20003844_pxl_be/En7Q-C8mn7VOrETEO9eVLkUBjSpmyaC2kHgt-1Ij3mpJsg?e=zPj8NY) This lab must be executed on hardware.

Scope definition:

* Install, configure and test the network infrastructure based on the network drawing;
* Proactively determine what is needed to ensure the continuity of the system and network infrastructure;
* Apply best practices to configuration and network security;
* Draw up an IP plan and document your solution;
* Make sure you can backup and restore device configuration from a backup environment;

Part 2: Network Troubleshooting

Document your findings and important commands.

Rack 6

voor ospf connect 1 10.199.66.96/27

| Subnet across | 172.16.6.0/28 |
| --- | --- |
| Vlan 10 → managment | 172.16.6.0/28 |
| Vlan 20 → vm hosts | 172.16.6.16/28 |
| Vlann 30 → appliance-servers | 172.16.6.32/28 |
| vlan 40 → data | 172.16.6.48/28 |
| Vlan 50 → Voice-Users | 172.16.6.64/28 |
| Vlan 60 → wireless-users | 172.16.6.80/28 |
| Vlan 80 → Access-Point | 172.16.6.112/28 |
| Native vlan 100 | / |
| RRA06-1 | 172.16.6.1 255.255.255.240  10. |
| SWRA06-1 | 172.16.6.5 255.255.255.240 |
|  |  |

**configuration router: lab-ra06-c-ro3-config:**

!

! Last configuration change at 07:52:18 UTC Tue May 2 2023

!

version 15.4

service timestamps debug datetime msec

service timestamps log datetime msec

service password-encryption

!

hostname RRA06-1

!

boot-start-marker

boot-end-marker

!

!

enable secret 5 $1$5Sq7$UtUZLyNzraPL.3PDize01.

!

no aaa new-model

!

!

!

!

!

!

!

!

!

!

!

!

ip dhcp pool Management

default-router 172.16.6.1

!

!

!

ip domain name www.RA6.com

ip cef

no ipv6 cef

multilink bundle-name authenticated

!

cts logging verbose

!

!

license udi pid CISCO1941/K9 sn FCZ2137B0H3

!

!

username RO1-RA6 password 7 094F471A1A0A → changed to admin

!

redundancy

!

!

ip ssh version 2

!

!

!

!

interface Embedded-Service-Engine0/0

no ip address

shutdown

!

interface GigabitEthernet0/0

no ip address

duplex auto

speed auto

!

interface GigabitEthernet0/0.10

encapsulation dot1Q 10

ip address 172.16.6.1 255.255.255.240

!

interface GigabitEthernet0/0.40

encapsulation dot1Q 40

ip address 172.16.6.49 255.255.255.240

!

interface GigabitEthernet0/1

ip address 10.199.66.106 255.255.255.224

duplex auto

speed auto

!

interface Serial0/1/0

no ip address

shutdown

clock rate 2000000

!

interface Serial0/1/1

no ip address

shutdown

clock rate 2000000

!

router ospf 10

router-id 6.6.6.6

passive-interface GigabitEthernet0/0

network 10.199.66.96 0.0.0.31 area 0

network 172.16.6.0 0.0.0.15 area 0

network 172.16.6.48 0.0.0.15 area 0

!

ip forward-protocol nd

!

no ip http server

no ip http secure-server

!

!

!

!

!

control-plane

!

!

!

line con 0

password 7 1511021F0725

login

line aux 0

line 2

no activation-character

no exec

transport preferred none

transport output pad telnet rlogin lapb-ta mop udptn v120 ssh

stopbits 1

line vty 0 4

login local

transport input ssh

!

scheduler allocate 20000 1000

!

end

**on this router all passwords are “cisco” to avoid confusion.**

**Configuration switch: lab-ra06-a-sw03-config:**

!

! Last configuration change at 00:34:09 UTC Mon Mar 1 1993

!

version 15.0

no service pad

service timestamps debug datetime msec

service timestamps log datetime msec

service password-encryption

!

hostname SWRA06-1

!

boot-start-marker

boot-end-marker

!

enable secret 5 $1$0KCK$p2Vpw3D3M37EkXpGZQlBy.

!

username cisco password 7 060506324F41

no aaa new-model

system mtu routing 1500

!

!

ip domain-name www.SW06.com

!

!

!

!

!

!

!

!

spanning-tree mode pvst

spanning-tree extend system-id

!

vlan internal allocation policy ascending

!

ip ssh version 2

!

!

!

!

!

interface FastEthernet0/1

switchport access vlan 10

switchport mode access

switchport port-security violation protect

spanning-tree portfast

!

interface FastEthernet0/2

switchport access vlan 10

switchport mode access

switchport port-security violation protect

spanning-tree portfast

!

interface FastEthernet0/3

switchport access vlan 10

switchport mode access

switchport port-security violation protect

spanning-tree portfast

!

interface FastEthernet0/4

switchport access vlan 10

switchport mode access

switchport port-security violation protect

spanning-tree portfast

!

interface FastEthernet0/5

switchport access vlan 10

switchport mode access

switchport port-security violation protect

spanning-tree portfast

!

interface FastEthernet0/6

switchport access vlan 40

switchport mode access

switchport port-security violation protect

!

interface FastEthernet0/7

switchport access vlan 40

switchport mode access

switchport port-security violation protect

!

interface FastEthernet0/8

switchport access vlan 40

switchport mode access

switchport port-security violation protect

!

interface FastEthernet0/9

switchport access vlan 40

switchport mode access

switchport port-security violation protect

!

interface FastEthernet0/10

switchport access vlan 40

switchport mode access

switchport port-security violation protect

!

interface FastEthernet0/11

!

interface FastEthernet0/12

!

interface FastEthernet0/13

!

interface FastEthernet0/14

!

interface FastEthernet0/15

!

interface FastEthernet0/16

!

interface FastEthernet0/17

!

interface FastEthernet0/18

!

interface FastEthernet0/19

!

interface FastEthernet0/20

!

interface FastEthernet0/21

!

interface FastEthernet0/22

!

interface FastEthernet0/23

!

interface FastEthernet0/24

!

interface GigabitEthernet0/1

switchport trunk native vlan 100

switchport trunk allowed vlan 10,40

switchport mode trunk

!

interface GigabitEthernet0/2

!

interface Vlan1

no ip address

!

interface Vlan10

ip address 172.16.6.5 255.255.255.240

!

interface Vlan40

ip address 172.16.6.50 255.255.255.240

!

interface Vlan70

ip address 10.199.66.98 255.255.255.224

!

ip default-gateway 172.16.6.1

ip http server

ip http secure-server

!

!

line con 0

password 7 13061E010803

login

line vty 0 4

password 7 02050D480809

login local

transport input ssh

line vty 5 15

login local

transport input ssh

!

end

**On this switch all passwords are “cisco” to avoid confusion.**

**restoring config files from tftp server:**

**on router:**

en

conf t

int gig0/1

no shut

ip add 10.199.66.106 255.255.255.224

exit

int gig0/0

no shut

exit

ip route 0.0.0.0 0.0.0.0 gig0/1

copy tftp: running-config

10.199.64.134

lab-ra06-c-ro3-config

**restoring config files from tftp server for switch:**

en

conf t

interface vlan 10

ip address 172.16.6.5 255.255.255.240

no shut

exit

ip default-gateway 172.16.6.1

interface gigabitethernet 0/1

switchport trunk native vlan 100

switchport mode trunk

switchport trunk allowed vlan 10

no shut

exit

copy tftp: running config

10.199.64.134

lab-ra06-a-sw03-config

—--------------------------------------------------------------------------------------------------------------------------------

# Part 1: Software Version Control with Git

Lab netacad: Cisco DEVNET 3.3.11

Document your findings and important commands.

# Part 2: Create a Python Unit Test

Lab netacad: Cisco DEVNET 3.5.7

Document your findings and important commands.

# Part 3: Parse Different Data Types with Python

Lab netacad: cisco DEVNET 3.6.6

Document your findings and important commands.

Search for a script to work and communicate with Cisco IOS network devices using netmiko. Start with a simple script and try to extend your script with additional functions.

# Part 1: Connecting to a single iOS device

- Sending single show command

- -Sending multiple show commands

- Send multiple configuration commands to a single device

script voor een show commando

from netmiko import ConnectHandler

cisco\_881 = {

'device\_type': 'cisco\_ios',

'host': '172.16.6.5',

'username': 'cisco',

'password': 'cisco',

'port' : 22, # optional, defaults to 22

'secret': 'csco', # optional, defaults to ''

}

net\_connect = ConnectHandler(\*\*cisco\_881)

output = net\_connect.send\_command('show ip int brief')

print(output)

~

script voor multiple show commands

from netmiko import ConnectHandler

cisco\_01 = {

"device\_type": "cisco\_ios",

"host": "172.16.6.5",

"username": "cisco",

"password": "cisco",

"secret": "cisco" # Enable password

}

connection = ConnectHandler(\*\*cisco\_01)

show\_commands = ['show vlan brief', 'show interface', 'show clock'] # List of commands to send

for command in show\_commands: # for loop

print(f'\n \*\*\* Sending { command} \*\*\* \n')

output = connection.send\_command(command)

print(output)

connection.disconnect()

script voor multiple conf

from netmiko import ConnectHandler

device = {

"device\_type": "cisco\_ios",

"ip": "172.16.6.5",

"username": "cisco",

"password": "cisco",

"secret": "cisco",

}

with ConnectHandler(\*\*device) as conn:

if not conn.check\_enable\_mode():

conn.enable()

cfg = conn.send\_config\_set(["int range f0/15-16", "switchport mode access","switchport access vlan 60"])

print(cfg)

# Part 2: Connect to multiple IOS devices

- Send show commands to multiple devices

from netmiko import ConnectHandler

switch = {

'device\_type': 'cisco\_ios',

'host': '172.16.6.5',

'username': 'cisco',

'password': 'cisco',

'port' : 22, # optional, defaults to 22

'secret': 'csco', # optional, defaults to ''

}

router = {

'device\_type': 'cisco\_ios',

'host': '172.16.6.1',

'username': 'admin',

'password': 'cisco',

'port' : 22, # optional, defaults to 22

'secret': 'csco', # optional, defaults to ''

}

net\_connect = ConnectHandler(\*\*switch)

ShowInt = net\_connect.send\_command('show ip int brief')

print(ShowInt)

net\_connect = ConnectHandler(\*\*router)

Showroute = net\_connect.send\_command('show ip route')

print(ShowInt, Showroute)

- Send configuration commands to multiple devices

from netmiko import ConnectHandler

switch = {

'device\_type': 'cisco\_ios',

'host': '172.16.6.5',

'username': 'cisco',

'password': 'cisco',

'port' : 22, # optional, defaults to 22

'secret': 'cisco', # optional, defaults to ''

}

router = {

'device\_type': 'cisco\_ios',

'host': '172.16.6.1',

'username': 'admin',

'password': 'cisco',

'port' : 22, # optional, defaults to 22

'secret': 'cisco', # optional, defaults to ''

}

with ConnectHandler(\*\*switch) as conn:

if not conn.check\_enable\_mode():

conn.enable()

cfg = conn.send\_config\_set(["int range f0/15-16", "switchport mode access","switchport access vlan 60"])

print(cfg)

with ConnectHandler(\*\*rouer) as conn:

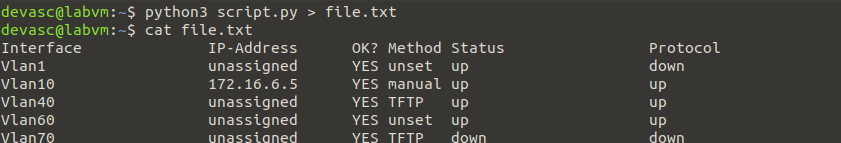
if not conn.check\_enable\_mode():

conn.enable()

conf\_router = conn.send\_config\_set(["int g0/0","no shutdown", "exit"])

print(conf\_router)

- Run show commands and save the output



- Backup the device configurations

switch = {

"device\_type": "cisco\_ios",

"host": "172.16.6.5",

"username": "cisco",

"password": "cisco",

"secret": "cisco" # Enable password

}

connection = ConnectHandler(\*\*switch)

connection.enable() # Enable method

connection.config\_mode() # Global config mode

connection.send\_command('copy running-config tftp:')

connection.send\_command('10.199.64.134')

connection.send\_command('lab-ra06-a-sw03-config')

print("bye bye")

connection.disconnect()

- Configure a subset of Interfaces

- Send device configuration using an external file

- Connect using a Python Dictionary

- Execute a script with a Function or classes

- Execute a script with a statements (if, ifelse, else)

# Part 3: Connect to IOS-XE devices

- Send show commands to multiple devices

- Send configuration commands to multiple devices

- Run show commands and save the output

- Backup the device configurations

- Configure a subset of Interfaces

- Send device configuration using an external file

- Connect using a Python Dictionary

- Execute a script with a Function or classes

- Execute a script with a statements (if, ifelse, else)

# Part 4: Create an challenging excited script as a network

- Create an exciting and challenging script that a network engineer in a programmable era would use every day. Surprise your lecturer!

Document your findings and important commands.