



Universidad Tecnológica
del Norte de Guanajuato
Organismo Público Descentralizado del Gobierno del Estado

"Educación y progreso para la vida"

Tecnologías de la Información y Comunicación

Programa educativo: TSU en Infraestructura de Redes Digitales

Área académica: Programación de Redes

Asignatura:

Unidad 3

Grupo: GIR0441

[2.9 Lab - NETCONF wPython Get Operational Data.docx](#)

Alumna Gómez Luna Cinthia Valeria:

Docente: Gabriel Barrón Rodríguez

Dolores Hidalgo, C.I.N., Gto., jueves 24 de noviembre de 2022

Introducción

En el presente documento se podrá observar la documentación del laboratorio 2.9 en donde se realizo se dice que el modulo de xml.dom.minidom este puede ayudar a ejecutar de manera mas sencilla a String los archivos XML que estos los envían a imprimir el documento que nos proporcione nuestro profesor para poder realizar de una manera correcta la ejecución de los comandos fue de gran ayuda por que se logro obtener toda la información requerida para la elaboración a continuación se mostrara el código empleado

```
lab2.9.py - C:/Users/valeria/AppData/Local/Programs/Python/Python311/lab2.9.py (3.11.0)
File Edit Format Run Options Window Help
from ncclient import manager
import xml.dom.minidom

m = manager.connect(
    host="10.10.20.48",
    port=830,
    username="developer",
    password="Cisc012345",
    hostkey_verify=False
)

netconf_filter = """
<filter>
  <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
</filter>
"""

netconf_reply = m.get(filter = netconf_filter)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())

import xmltodict
netconf_reply_dict = xmltodict.parse(netconf_reply.xml)
for interface in netconf_reply_dict["rpc-reply"]["data"]["interfaces-state"]["interface"]:
    print("Name: {} MAC: {} Input: {} Output {}".format(
        interface["name"],
        interface["phys-address"],
        interface["statistics"]["in-octets"],
        interface["statistics"]["out-octets"]
    ))
)
```

```

<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:4c0c2b9e-9b90-43be-b3c6-c89ce21e91ae">
  <data>
    <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet1</name>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:ethernetCsmacd</type>
        <admin-status>up</admin-status>
        <oper-status>up</oper-status>
        <last-change>2022-12-01T02:17:53.000383+00:00</last-change>
        <if-index>1</if-index>
        <phys-address>00:50:56:bf:32:85</phys-address>
        <speed>1024000000</speed>
        <statistics>
          <discontinuity-time>2022-12-01T02:16:39.000345+00:00</discontinuity-time>
          <in-octets>266538</in-octets>
          <in-unicast-pkts>3486</in-unicast-pkts>
          <in-broadcast-pkts>0</in-broadcast-pkts>
          <in-multicast-pkts>0</in-multicast-pkts>
          <in-discards>0</in-discards>
          <in-errors>0</in-errors>
          <in-unknown-protos>0</in-unknown-protos>
          <out-octets>873323</out-octets>
          <out-unicast-pkts>3910</out-unicast-pkts>
          <out-broadcast-pkts>0</out-broadcast-pkts>
          <out-multicast-pkts>0</out-multicast-pkts>
        </statistics>
      </interface>
    </interfaces-state>
  </data>
</rpc-reply>

```

```
File Edit Shell Debug Options Window Help
:iana-if-type">ianaift:other</type>
      <admin-status>up</admin-status>
      <oper-status>up</oper-status>
      <last-change>2022-12-01T02:17:42.000479+00:00</l
ast-change>
      <if-index>0</if-index>
      <phys-address>00:00:00:00:00:00</phys-address>
      <speed>10240000000</speed>
      <statistics>
        <discontinuity-time>2022-12-01T02:16:39.
000411+00:00</discontinuity-time>
        <in-octets>0</in-octets>
        <in-unicast-pkts>0</in-unicast-pkts>
        <in-broadcast-pkts>0</in-broadcast-pkts>
        <in-multicast-pkts>0</in-multicast-pkts>
        <in-discards>0</in-discards>
        <in-errors>0</in-errors>
        <in-unknown-protos>0</in-unknown-protos>
        <out-octets>0</out-octets>
        <out-unicast-pkts>0</out-unicast-pkts>
        <out-broadcast-pkts>0</out-broadcast-pkt
s>
        <out-multicast-pkts>0</out-multicast-pkt
s>
        <out-discards>0</out-discards>
        <out-errors>0</out-errors>
      </statistics>
    </interface>
  </interfaces-state>
</data>
</rpc-reply>

Name: GigabitEthernet1 MAC: 00:50:56:bf:32:85 Input: 316021 Output 1415845
Name: GigabitEthernet2 MAC: 00:50:56:bf:80:ca Input: 0 Output 0
Name: GigabitEthernet3 MAC: 00:50:56:bf:8f:d5 Input: 0 Output 0
Name: Loopback1 MAC: 00:1e:e5:20:a7:00 Input: 0 Output 0
Name: Loopback99 MAC: 00:1e:e5:20:a7:00 Input: 0 Output 0
Name: Loopback100 MAC: 00:1e:e5:20:a7:00 Input: 0 Output 0
Name: Control Plane MAC: 00:00:00:00:00:00 Input: 0 Output 0
>>>
```

Conclusión

Para poder concluir de una manera satisfactoria daremos nuestra opinión de lo que se logro hacer en este laboratorio lo primero que se tenia que hacer es la impresión los envíos de configuración y la filtración de datos esto se realizo con el protocolo puedo lograr decir que se pudo trabajar más fácil con este que con netmiko ya que el otro al momento de ejecutarlo nos marcaba una serie de erros y este no marcaba errores.