

## Lab 2.9– NETCONF w/Python: Get Operational Data

```
lab29_sharonolvera.py > ...
1  #Autor: Sharon Michelle Olvera Ibarra
2  #Descripción: Configuración NETCONF con Python: Configuración del dispositivo
3  #Fecha: 05/12/2023
4
5  from ncclient import manager
6  import xml.dom.minidom
7
8  m = manager.connect(
9      host = "10.10.20.48",
10     port = 830,
11     username = "developer",
12     password = "Cisco12345",
13     hostkey_verify = False
14 )
15
16 netconf_filter = """
17 <filter>
18 | <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
19 </filter>
20 """
21 netconf_reply = m.get(filter = netconf_filter)
22 print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
23
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... ^ X

```
return cls(self._session,
          ~~~~~~
File "C:\Users\VIP\AppData\Local\Programs\Python\Python312\Lib\site-packages\ncclient\operations\retrieve.py", line 91, in request
    return self._request(node)
          ~~~~~~
File "C:\Users\VIP\AppData\Local\Programs\Python\Python312\Lib\site-packages\ncclient\operations\rpc.py", line 375, in _request
    raise self._reply.error
ncclient.operations.rpc.RPCError: {'type': 'protocol', 'tag': 'unknown-element', 'app_tag': None, 'severity': 'error', 'info': '<?xml version="1.0" encoding="
UTF-8"?><error-info xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"><bad-element>filter</bad-element>\n</er
ror-info>\n', 'path': '\n /rpc/get\n ', 'message': None}
© PS C:\Users\VIP\Desktop\unidad3>
```

```

from ncclient import manager
import xml.dom.minidom
import xmltodict

m = manager.connect(
    host = "10.10.20.48",
    port = 830,
    username = "developer",
    password = "C1sco12345",
    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())

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"]
    ))

```

```

PS C:\Users\HP\Desktop\unidad3> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe c:/Users/HP/Desktop/unidad3/lab29_sharonolvera.py
Traceback (most recent call last):
  File "c:/Users/HP/Desktop/unidad3/lab29_sharonolvera.py", line 22, in <module>
    netconf_reply = m.get(filter = netconf_filter)
    ~~~~~^~~~~~
  File "C:/Users/HP/AppData/Local/Programs/Python/Python312/Lib/site-packages/ncclient/manager.py", line 257, in execute
    return cls(self._session,
    ~~~~~^~~~~~
  File "C:/Users/HP/AppData/Local/Programs/Python/Python312/Lib/site-packages/ncclient/operations/retrieve.py", line 91, in request
    return self._request(node)
    ~~~~~^~~~~~
  File "C:/Users/HP/AppData/Local/Programs/Python/Python312/Lib/site-packages/ncclient/operations/rpc.py", line 375, in _request
    raise self._reply.error
ncclient.operations.rpc.RPCError: {'type': 'protocol', 'tag': 'unknown-element', 'app_tag': None, 'severity': 'error', 'info': '<?xml version="1.0" encoding="UTF-8"?><error-info xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"><bad-element>filter</bad-element>\n</error-info>\n', 'path': '\n /rpc/get\n ', 'message': None}
PS C:\Users\HP\Desktop\unidad3>

```

En este código hubo algunos errores, lo cual creo que pudieron ser por que el servidor NETCONF podría no admitir el módulo YANG. o porque el servidor NETCONF no reconoce el elemento <filter> en la solicitud, o también puede ser por las credenciales.