

2.1Lab – Setting Up the Lab Environment

Soy un usuario con dos sistemas operativos los cuales son Windows y Linux, en este caso realice todas mis practicas con Windows por lo cual todas mis operaciones fueron realizadas desde Windows. Hicimos con la conexion por ssh con el programa putty. Primero realice una reservación en cisco con IOS XE on CSR, después de realizar la reservación implementamos la coneccion remota por ssh con el puerto 22 por medio de putty. Mediante nuestra reservación me conecte a un VPN.

2.2Lab – CLI Automation with Python using netmiko

Realizamos la instalacion de netmiko mediante la terminal de Windows por lo cual solamente seguí las instrucciones que se me indicaron en el laboratorio. Despues de su instalación hicimos la conexión para poder configurar los dispositivos tomando en cuenta que Netmiko es una libreria de redes multivendedores basada en Paramiko, que es una libreria estandar para las conexiones ssh Python. Con Netmiko como base, se pueden realizar programas y scripts que faciliten y mejoren la administración de los equipos de redes.

2.3 Lab - Explore YANG models using the pyang tool

Es un validador, convertidor y generador de codigo basado en python. Se puede utilizar para probar los módulos YANG y asegurarse de que funcionan correctamente. correcto, convierta los módulos YANG a otros formatos y guarde. El complemento genera código a partir de módulos. En esta práctica de laboratorio, utilizaremos pyang para analizar el archivo VM XE de IOS, que es la descarga está disponible desde github. con un comando laboratorio, le pedimos que nos mostrara la información en forma de árbol.

2.4 Lab - RESTCONF with Postman.

Este fue uno de los laboratorios más difíciles de realizar tomando en cuenta que son bastantes los pasos y fue algo confuso tomando en cuenta que muchos de los pasos no me salieron a primera cuenta. configuración del dispositivo, pero a diferencia del laboratorio 2.2, usaremos RESTCONF con herramienta de correo. El cual a través de la API lograremos nuestros resultados. En mi opinión, este experimento no es demasiado difícil, solo necesitas un poco conocimiento de Postman más el hecho de que incluso si la API está bien, ya conectado a SDN, había un caso para hacer las solicitudes GET o PUT que usamos en este caso, pero, además, tenemos más características que analizaré a continuación, pero son iguales con las que interactuamos en la API.

¿Que es RESTCONF? Protocolo creado por IETF que nos permite realizar cambios desde una api por medio de http

2.5 Lab - RESTCONF with Python.

A comparación del anterior este fue un poco menos complejo de realizar, tomando en cuenta que este se hizo con Python y no consultas de POSTMAN asi es que importamos json y request y además deshabilitamos las advertencias ssl. Postman, deberá usar la API, los encabezados y el tipo de autenticación para construir su solicitud. Guarda e imprime la información proporcionada en la variable con formato json especificada. Luego use la opción yangConfig. Esta es una opción que proporciona una solicitud para configurar con yang. Luego use un if para enviar el mensaje de respuesta deseado dependiendo del resultado donde se arroje él.

2.6 Lab – raw NETCONF.

Ahora solamente cambiamos de puerto por medio de putty, cambiamos del puerto 22 al 830 ¿Por qué se realiza ese cambio de puerto? La comunicación garantizada por el puerto TCP 830 es la diferencia mayor entre TCP y UDP. El puerto UDP no garantizaría la comunicación como TCP

¿Cuál es la diferencia entre RESTCONF y NETCONF? diferencia de NETCONF y RESTCONF es que nos permite realizar cambios en los dispositivos por medio de una API usando http/tls

2.7 Lab - NETCONF wPython List Capabilities.

Aquí usamos ncclient con el que realizamos peticiones con NETCONF, de ahí los resultados los devolvimos con otro formato.

¿Qué diferencia habría con la biblioteca Netmiko? NCClient es una librería Python escrita para simplificar el scripting y la automatización sobre el protocolo NETCONF. Netmiko es diseñada para actuar como una capa de abstracción sobre la implementación Paramiko Python del protocolo SSHv2.

2.8 Lab - NETCONF wPython Device Configuration..

Empezando con el laboratorio anterior. Se le pedio que copie el resultado y lo pase al navegador CodeBeauty. CodeBeauty Browser se utiliza para ver la salida XML de forma organizada y en cascada. Esto es solo para dar un ejemplo de lo que proporciona el como resultado de agregar la función `toprettyxml()` a su código. Luego puede usar la función `netconf_data` para ejecutar comandos de configuración para el dispositivo el comando está disponible. Se envió uno al dispositivo del proveedor y la estructuración de este pedido debe hacerse en xml. Por ejemplo, estoy ejecutando una acción que crea un nuevo host y una interfaz de loopback, pero ese es un comando arbitrario disponible ahora y Cisco en ese momento. Sin embargo, se requiere conocimiento de NETCONF. Aparte de hablar de Model Drive Programability, una de sus funciones es devolver cambios si la acción en el dispositivo no es incorrecta. Esto es lo que vimos como ejemplo en el ejercicio anterior.

2.9 Lab - NETCONF wPython Get Operational Data

A diferencia del laboratorio anterior que tuvimos que hacer configuración en lo que solo los visualizaremos por el momento Usaremos `xmlodict` para filtrar la información.

¿Qué fin tendría usar los filtros? recupera los datos operativos de estado de interfaces

