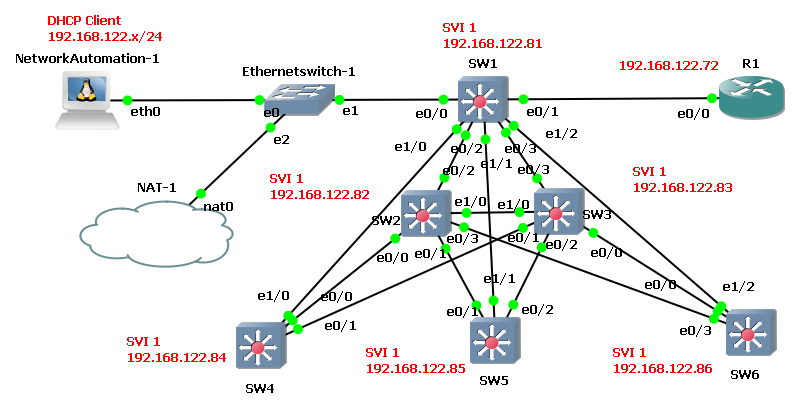
**Basic Automation using Telnet**

**Topology**



**Network Automation-1 pc DHCP config:**

**Step 1**

Change the following:

*root@NetworkAutomation-1:/etc/network#* ***cat interfaces***

*#*

*# This is a sample network config uncomment lines to configure the network*

*#*

*# Static config for eth0*

*#auto eth0*

*#iface eth0 inet static*

*# address 192.168.0.2*

*# netmask 255.255.255.0*

*# gateway 192.168.0.1*

*# up echo nameserver 192.168.0.1 > /etc/resolv.conf*

*# DHCP config for eth0*

*# auto eth0*

*# iface eth0 inet dhcp*

to:

*root@NetworkAutomation-1:/etc/network#* ***cat interfaces***

*#*

*# This is a sample network config uncomment lines to configure the network*

*#*

*# Static config for eth0*

*#auto eth0*

*#iface eth0 inet static*

*# address 192.168.0.2*

*# netmask 255.255.255.0*

*# gateway 192.168.0.1*

*# up echo nameserver 192.168.0.1 > /etc/resolv.conf*

*# DHCP config for eth0*

*auto eth0*

*iface eth0 inet dhcp*

**Step 2**

Reboot the VM (right-click from GNS3)

**Verification**

root@NetworkAutomation-1:~# **ifconfig**

eth0 Link encap:Ethernet HWaddr d6:17:0f:b9:04:21

inet addr:192.168.122.225 Bcast:192.168.122.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:170875 errors:0 dropped:53 overruns:0 frame:0

TX packets:168894 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:7197250 (7.1 MB) TX bytes:7097415 (7.0 MB)

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:65536 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

**Switch and Router configuration**

Here is a sample config that allows Telnet reachability:

SW2(config)#**int range e0/0 - 3 , e1/0**

SW2(config-if-range)#**switchport trunk encapsulation dot1q**

SW2(config-if-range)#**switchport mode trunk**

SW2(config-if-range)#!

SW2(config-if-range)#**int vlan 1**

SW2(config-if)#**no shut**

SW2(config-if)#**ip add 192.168.122.82 255.255.255.0**

SW2(config-if)#!

SW2(config-if)#**line vty 0 4**

SW2(config-line)#**login local**

SW2(config-line)#**transport input all**

SW2(config-line)#**exit**

SW2(config)#**username cisco password cisco**

SW2(config)#**username mikis password cisco**

SW2(config)#**enable password cisco**

**Verification**

root@NetworkAutomation-1:~# **telnet 192.168.122.81**

Trying 192.168.122.81...

Connected to 192.168.122.81.

Escape character is '^]'.

User Access Verification

Username: **cisco**

Password:

SW1>**enable**

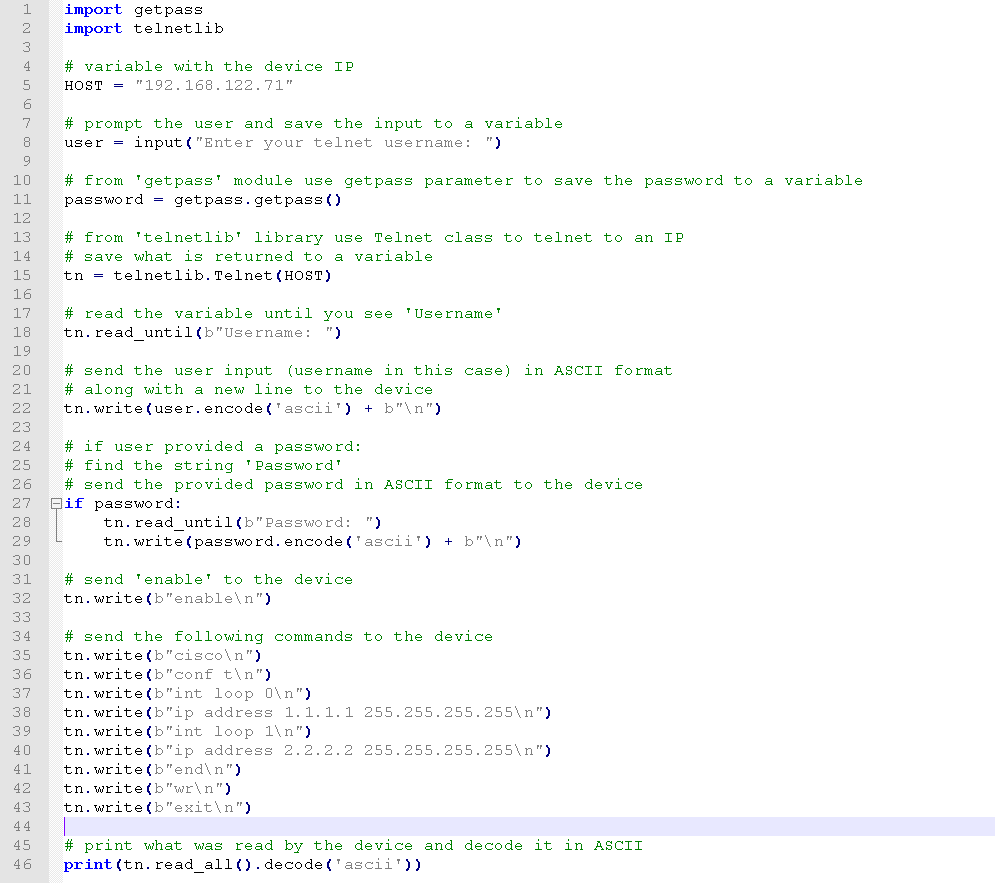
Password:

SW1#

**Section 1 (telnetlib) - Script 1**

The following script:

* Connects via telnet to a device
* The IP of the device is specified manually (the HOST variable)
* The user is prompted for user and password
* The provided credentials are used by the script in the telnet connection
* The script creates 2 loopback interfaces



import getpass

import telnetlib

# variable with the device IP

HOST = "192.168.122.71"

# prompt the user and save the input to a variable

user = input("Enter your telnet username: ")

# from 'getpass' module use getpass parameter to save the password to a variable

password = getpass.getpass()

# from 'telnetlib' library use Telnet class to telnet to an IP

# save what is returned to a variable

tn = telnetlib.Telnet(HOST)

# read the variable until you see 'Username'

tn.read\_until(b"Username: ")

# send the user input (username in this case) in ASCII format

# along with a new line to the device

tn.write(user.encode('ascii') + b"\n")

# if user provided a password:

# find the string 'Password'

# send the provided password in ASCII format to the device

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

# send 'enable' to the device

tn.write(b"enable\n")

# send the following commands to the device

tn.write(b"cisco\n")

tn.write(b"conf t\n")

tn.write(b"int loop 0\n")

tn.write(b"ip address 1.1.1.1 255.255.255.255\n")

tn.write(b"int loop 1\n")

tn.write(b"ip address 2.2.2.2 255.255.255.255\n")

tn.write(b"end\n")

tn.write(b"wr\n")

tn.write(b"exit\n")

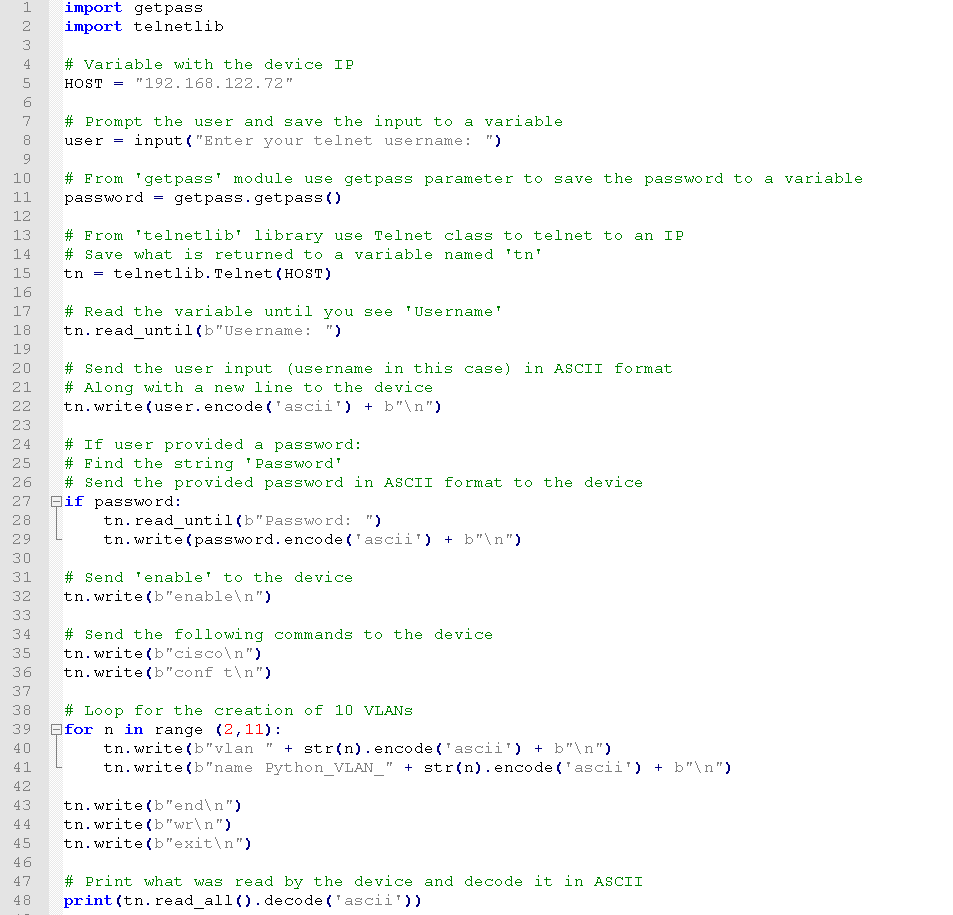
# print what was read by the device and decode it in ASCII

print(tn.read\_all().decode('ascii'))

**Section 1 (telnetlib) - Script 2**

The following script:

* Connects via telnet to a device
* The IP of the device is specified manually (the HOST variable)
* The user is prompted for user and password
* The provided credentials are used by the script in the telnet connection
* The script uses an **if loop** to create 10 VLANs



import getpass

import telnetlib

# Variable with the device IP

HOST = "192.168.122.72"

# Prompt the user and save the input to a variable

user = input("Enter your telnet username: ")

# From 'getpass' module use getpass parameter to save the password to a variable

password = getpass.getpass()

# From 'telnetlib' library use Telnet class to telnet to an IP

# Save what is returned to a variable named 'tn'

tn = telnetlib.Telnet(HOST)

# Read the variable until you see 'Username'

tn.read\_until(b"Username: ")

# Send the user input (username in this case) in ASCII format

# Along with a new line to the device

tn.write(user.encode('ascii') + b"\n")

# If user provided a password:

# Find the string 'Password'

# Send the provided password in ASCII format to the device

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

# Send 'enable' to the device

tn.write(b"enable\n")

# Send the following commands to the device

tn.write(b"cisco\n")

tn.write(b"conf t\n")

# Loop for the creation of 10 VLANs

for n in range (2,11):

tn.write(b"vlan " + str(n).encode('ascii') + b"\n")

tn.write(b"name Python\_VLAN\_" + str(n).encode('ascii') + b"\n")

tn.write(b"end\n")

tn.write(b"wr\n")

tn.write(b"exit\n")

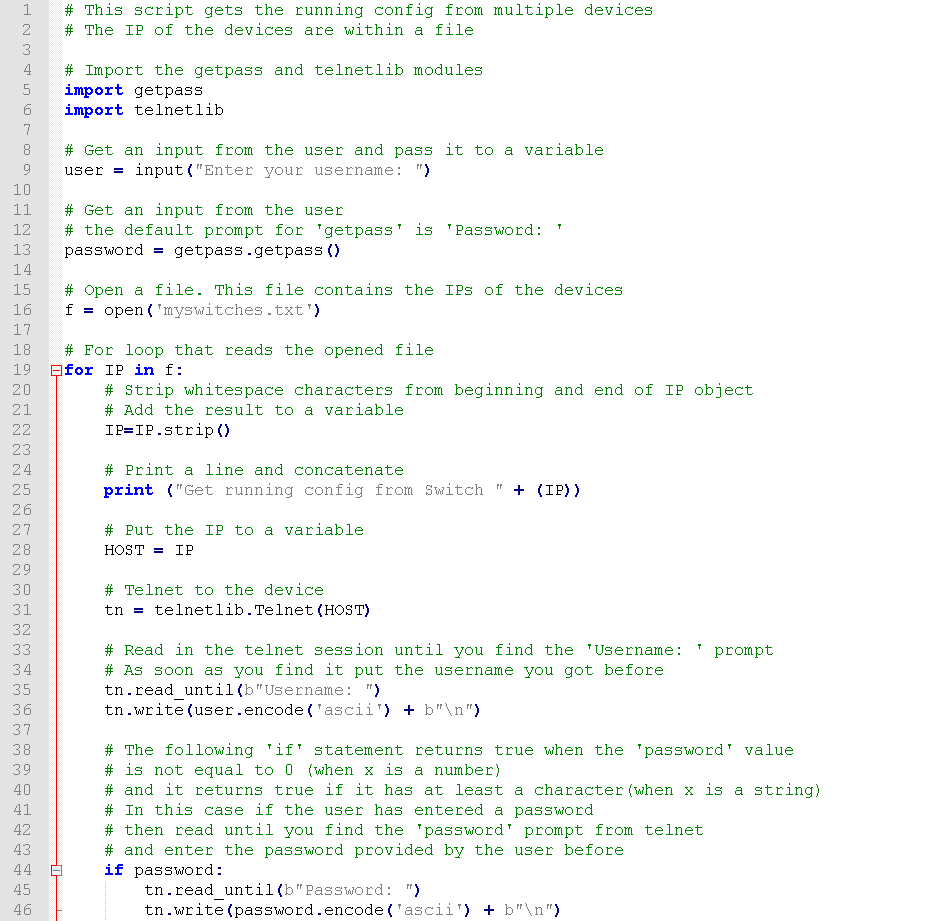
# Print what was read by the device and decode it in ASCII

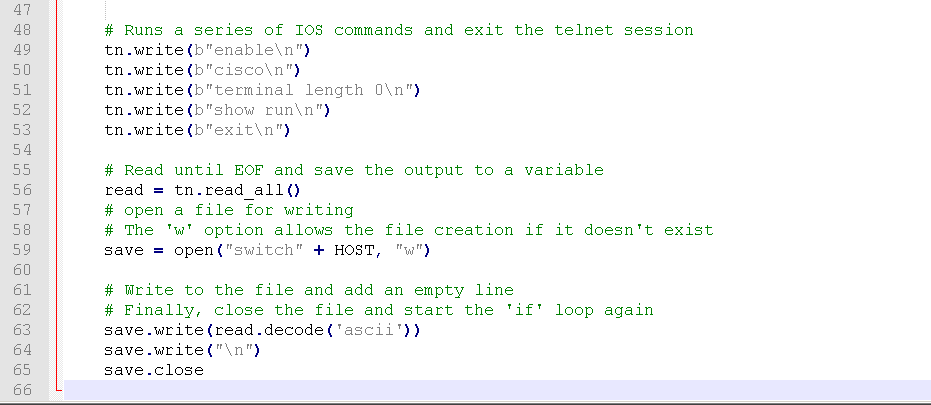
print(tn.read\_all().decode('ascii'))

**Section 1 (telnetlib) - Script 3**

The following script:

* Connects via telnet to multiple devices
* The IP of the device are read by the script from a file
* The user is prompted for user and password
* The provided credentials are used by the script in the telnet connection
* The script saves the running config of the remote devices to separate files





# This script gets the running config from multiple devices

# The IP of the devices are within a file

# Import the getpass and telnetlib modules

import getpass

import telnetlib

# Get an input from the user and pass it to a variable

user = input("Enter your username: ")

# Get an input from the user

# the default prompt for 'getpass' is 'Password: '

password = getpass.getpass()

# Open a file. This file contains the IPs of the devices

f = open('myswitches.txt')

# For loop that reads the opened file

for IP in f:

# Strip whitespace characters from beginning and end of IP object

# Add the result to a variable

IP=IP.strip()

# Print a line and concatenate

print ("Get running config from Switch " + (IP))

# Put the IP to a variable

HOST = IP

# Telnet to the device

tn = telnetlib.Telnet(HOST)

# Read in the telnet session until you find the 'Username: ' prompt

# As soon as you find it put the username you got before

tn.read\_until(b"Username: ")

tn.write(user.encode('ascii') + b"\n")

# The following 'if' statement returns true when the 'password' value

# is not equal to 0 (when x is a number)

# and it returns true if it has at least a character(when x is a string)

# In this case if the user has entered a password

# then read until you find the 'password' prompt from telnet

# and enter the password provided by the user before

if password:

tn.read\_until(b"Password: ")

tn.write(password.encode('ascii') + b"\n")

# Runs a series of IOS commands and exit the telnet session

tn.write(b"enable\n")

tn.write(b"cisco\n")

tn.write(b"terminal length 0\n")

tn.write(b"show run\n")

tn.write(b"exit\n")

# Read until EOF and save the output to a variable

read = tn.read\_all()

# open a file for writing

# The 'w' option allows the file creation if it doesn't exist

save = open("switch" + HOST, "w")

# Write to the file and add an empty line

# Finally, close the file and start the 'if' loop again

save.write(read.decode('ascii'))

save.write("\n")

save.close

**Basic Automation using SSH**

**Section 2 (netmico) - Script 1**

**Step 1**

Prepare the devices for SSH access

SW1(config)#**hostname SW1**

SW1(config)#**ip domain-name mikis.lab**

SW1(config)#**crypto key generate rsa**

The name for the keys will be: SW1.mikis.lab

Choose the size of the key modulus in the range of 360 to 4096 for your

General Purpose Keys. Choosing a key modulus greater than 512 may take

a few minutes.

How many bits in the modulus [512]: **1024**

% Generating 1024 bit RSA keys, keys will be non-exportable...

[OK] (elapsed time was 0 seconds)

SW1(config)#**line vty 0 4**

SW1(config-line)#**transport input ssh**

SW1(config-line)#**login local**

SW1(config)#**username cisco password cisco**

SW1(config)#**enable secret ssh**