**Part 1: Cisco Unified Communication (CUCM) & Voice Over IP (VOIP)**

Juntao Zhong

ccnp (cisco certified network professional)

jan. 2019

NEWPORT HIGH SCHOOL, WA

## Purpose

The purpose of this lab is to configure a Cisco Unified Communication Manager (CUCM) server and connect two phones using Voice Over IP (VOIP).

## Backgroud:

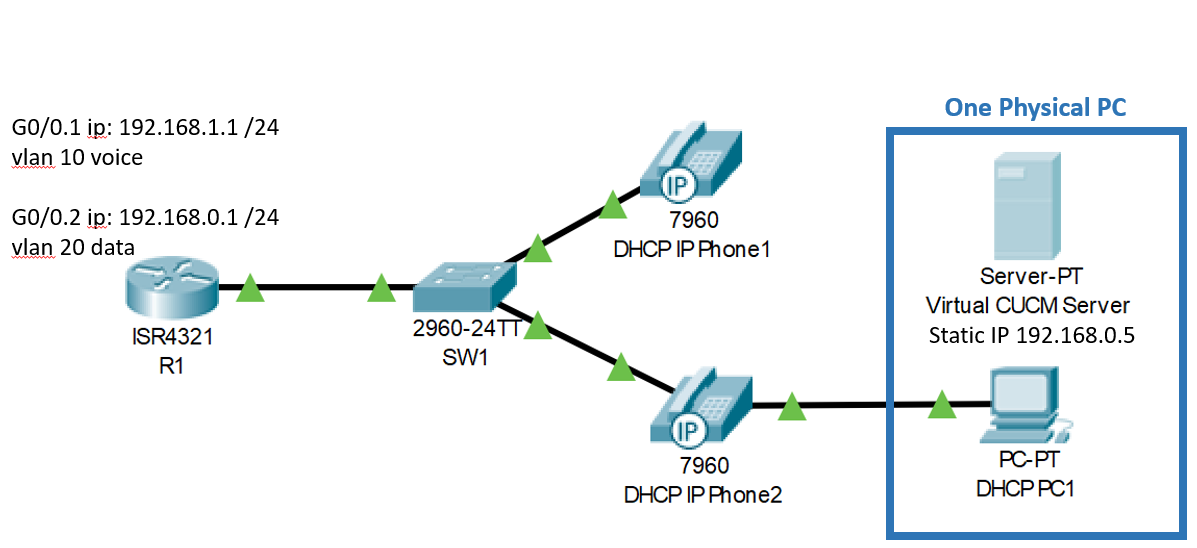
Simple configuration on Cisco router via VoIP Call Manager Express can easily connect two phones. **Cisco Unified Communication Manager** (**CUCM)** is capable of more than simple voice connection. Functions such as voice mails, transfer and conference are available. It stores and backups user information on a separate server, providing extra resilience to network problems. CUCM also supports video conference and can prioritize different traffic, as well as many important functions for video and voice calls of enterprise.

## Lab summary

CUCM has many extensive functions, which I will explore in the future. However, in this lab, I only concern about the setting up a CUCM server and use it to assign phone numbers and connect two phones.

First, I load the CUCM ISO onto a virtual machine using Red Hat Enterprise Operating System. Then, I entered the Graphic User Interface of CUCM through the web browser. After that, I configure the CUCM and assign two numbers to two phones. And two phones can call each other.

## Lab Diagram



## Lab commands

ntp master 1

ntp update-calendar

ntp server 192.168.0.1

option 150 ip 192.168.0.5

telephony-service

max-ephones 2

max-dn 2

ip source-address 192.168.0.5 port 2000

max-conferences 8

## configuration of both servers and routers

### Linux Router configuration:

**Router(config)#** **hostname R1**

aaa new-model

aaa authentication login default group tacacs+ local

aaa authentication enable default group tacacs+ enable

aaa authorization config-commands

aaa authorization commands 0 default group tacacs+ none

aaa authorization commands 15 default group tacacs+ none

aaa accounting send stop-record authentication failure

aaa accounting update newinfo periodic 5

aaa accounting exec default start-stop group tacacs+

aaa accounting network default start-stop group tacacs+

aaa session-id common

ip domain name cisco.com

vtp domain cisco

username backup password 0 cisco

interface GigabitEthernet0/0

ip address 10.10.10.1 255.255.255.0

no shutdown

ip tacacs source-interface GigabitEthernet0/0

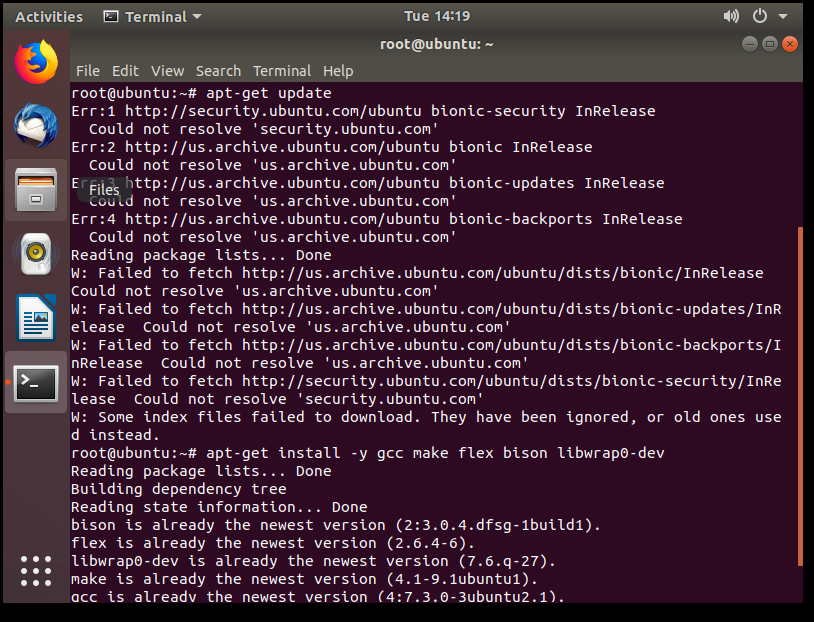
tacacs-server host 10.10.10.3

tacacs-server directed-request

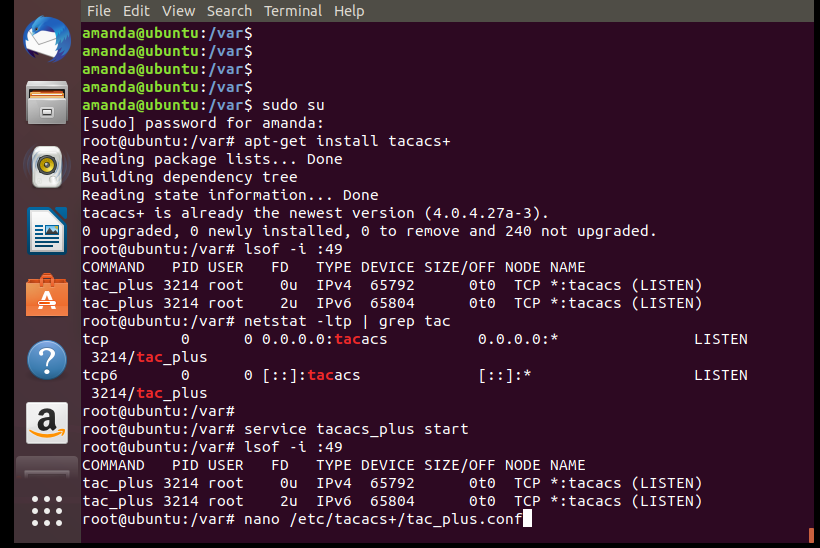
tacacs-server key 123456

### Linux Server Setup:

First, I entered “**sudo -i**” in command prompt to get to Linux root, and issued “**apt-get update**” so that my Linux can download other things from the internet.



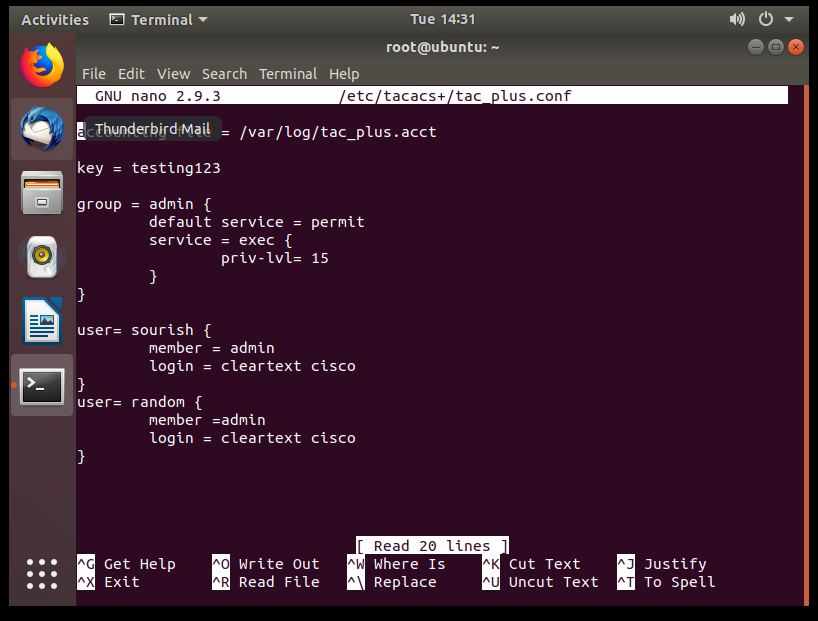
Then, I entered “**apt-get install -y gcc make flex bison libwrap0-dev**” and “**apt-get install tacacs+**” to install TACACS+ service in my Linux.

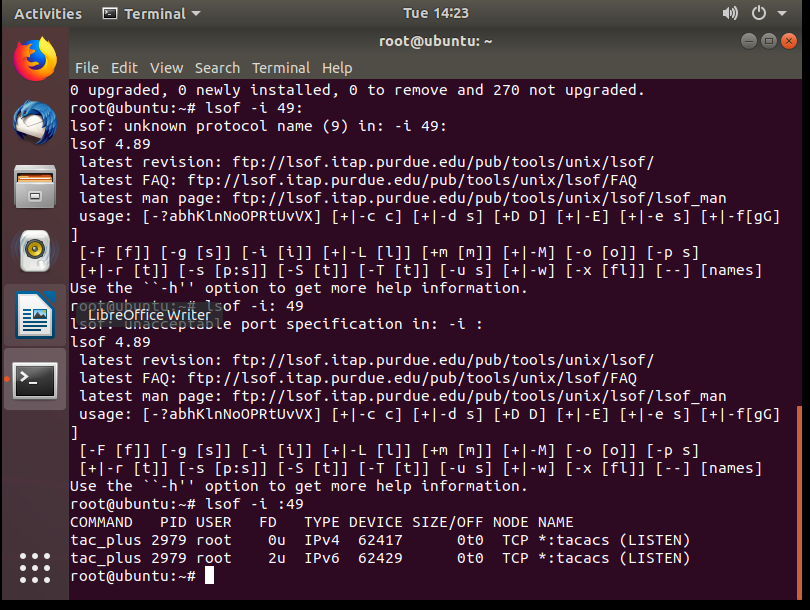


After downloading TACACS+ service, I edited its configuration file section with “**nano …**”to setup TACACS+.

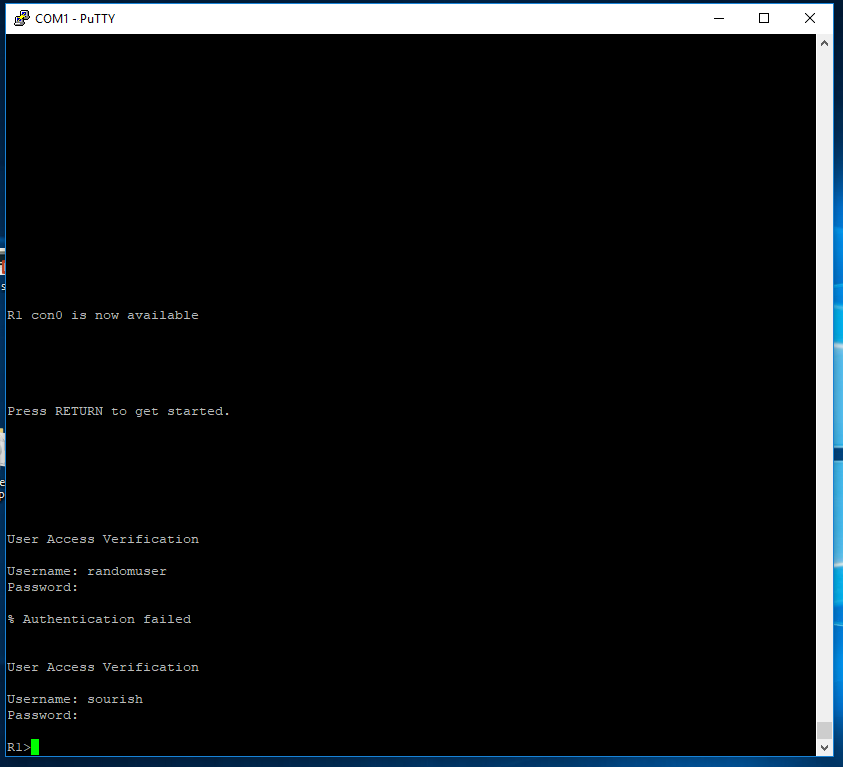


Inside the configuration file, I defined the key (domain) between my server and router, the group with privilege levels and users in such groups.





The last step was to check port 49 with “**lsof -i :49**” to see whether TACACS+ was running. The spelling and space must be exact for it to work. Then, TACACS+ platform was good to go.



That’s what I saw when I console into my router. If ssh or telnet was configured, the User Access Verification should be similar as well.

### Windows Router Commands:

**Router(config)#** **hostname R1**

### enable password cisco

### aaa new-model

### aaa authentication login default group radius local

### aaa authorization exec default group radius if-authenticated

### Username backup password cisco

### aaa session-id common

### vtp domain cisco

### interface GigabitEthernet0/0

### ip address 10.10.10.1 255.255.255.0

### no shutdown

### radius-server host 10.10.10.3 key 123456

### line con 0

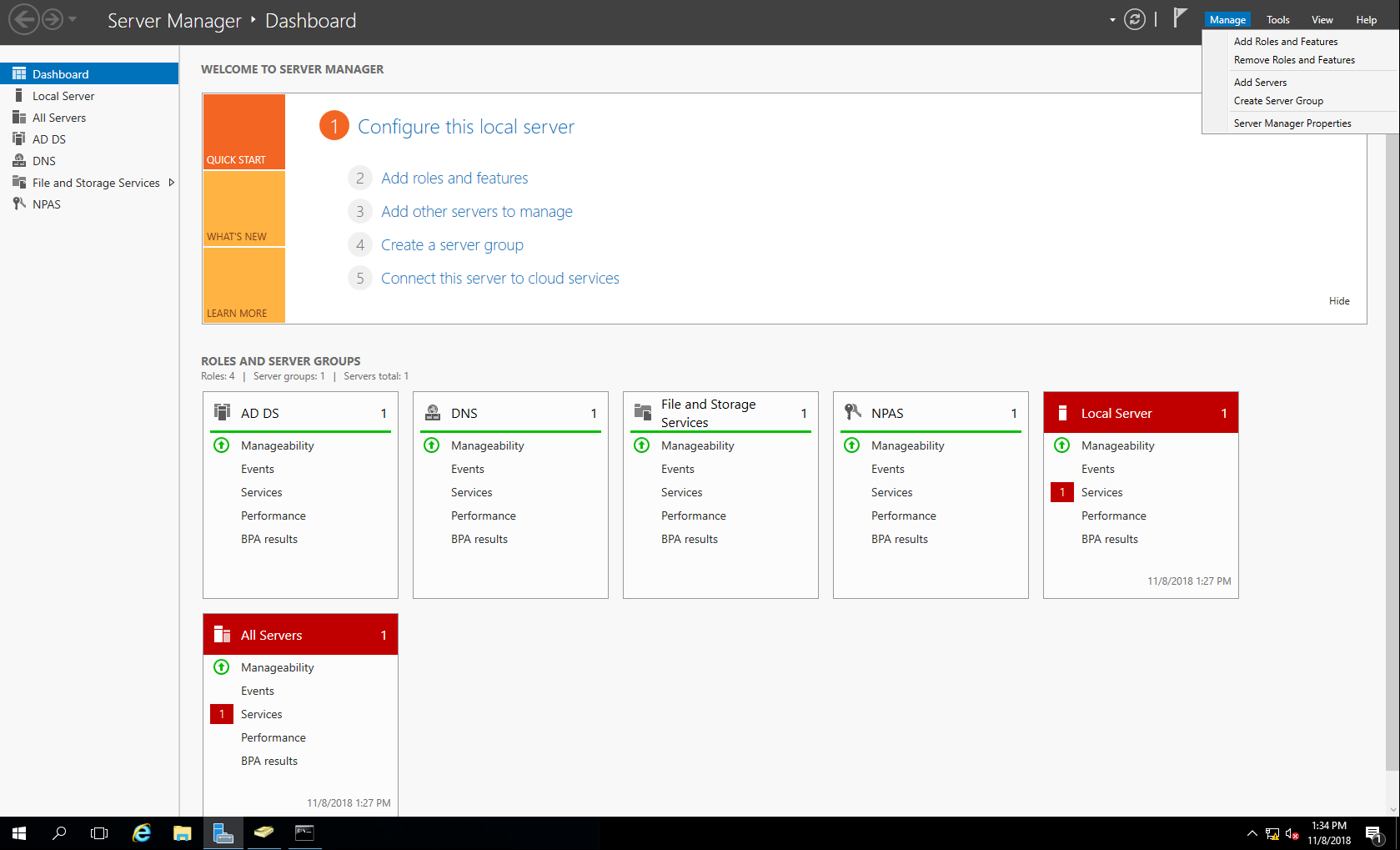
### password 123456

### Windows Server Setup:

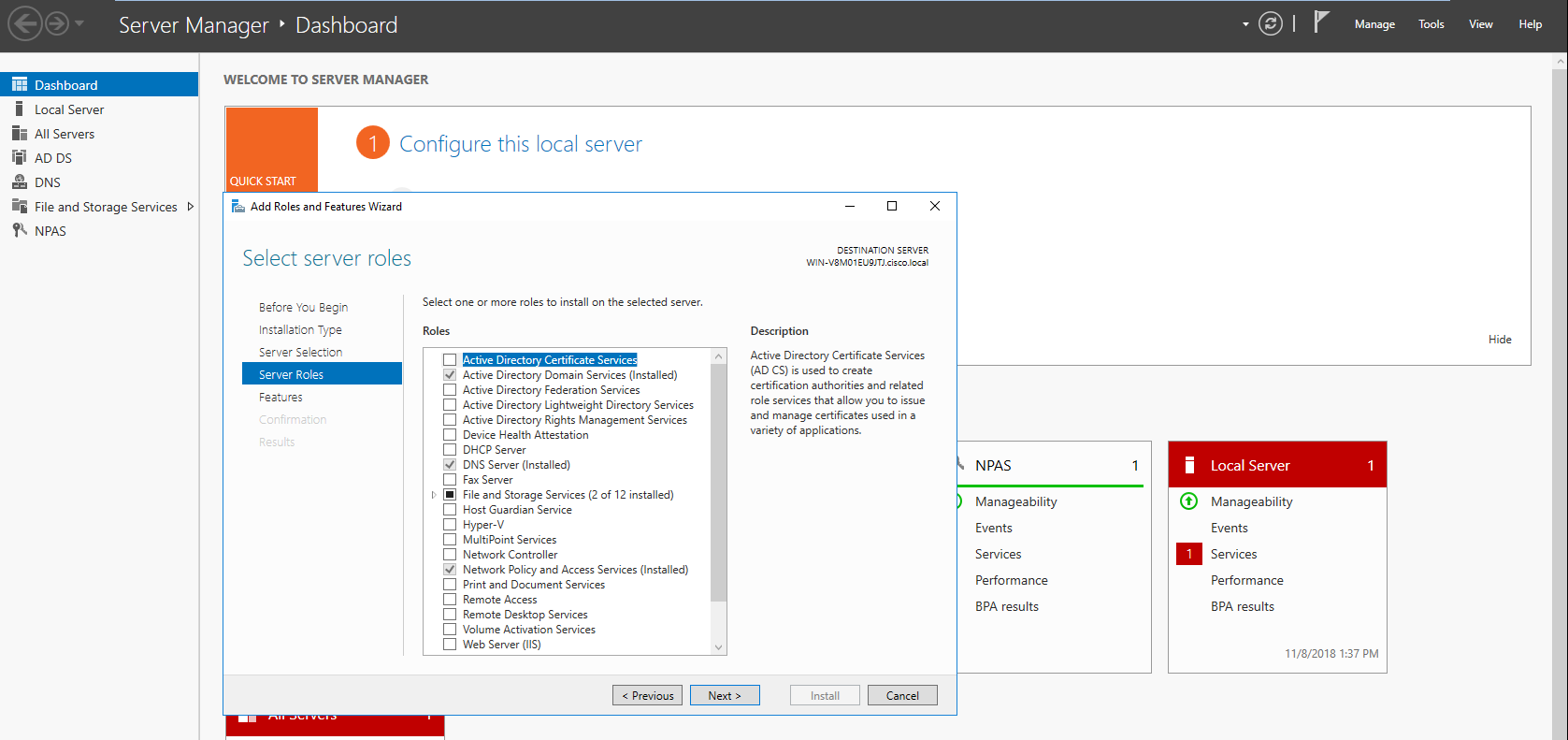
Since Windows has a nice-looking graphic user interface, there are a lot of clicking and screens involved. There are three main components for RADIUS: Active Directory Domain Service (AD DS), Network Policy Server (NPS) and the authentication with routers.

#### **Preparation Step: download the services needed.**

First, I enter the **Dashboard** for Server. Hover over the **Manage** tab and click on “**Add Roles and Features**”. We are going to download AD DS and NPS. Make sure the server connects to Internet.

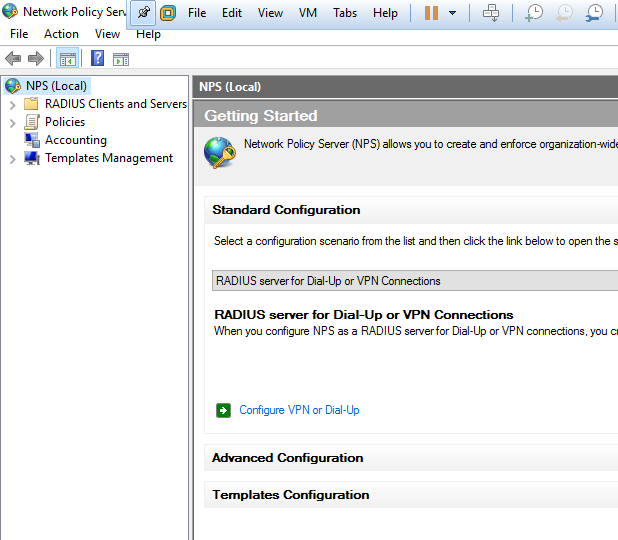


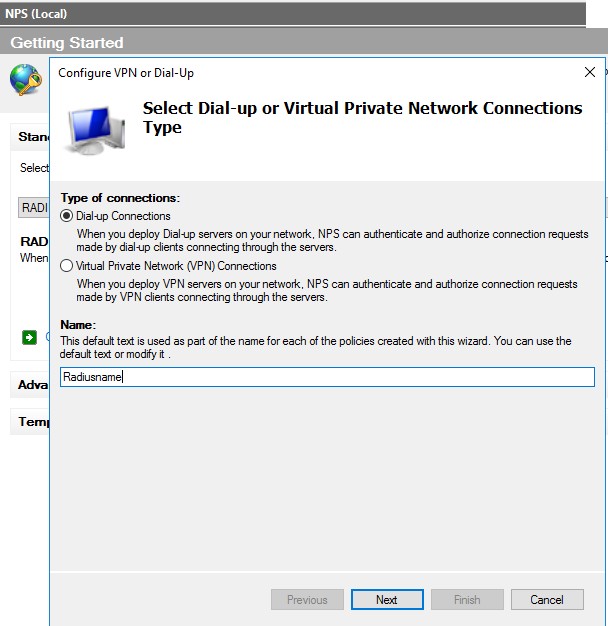
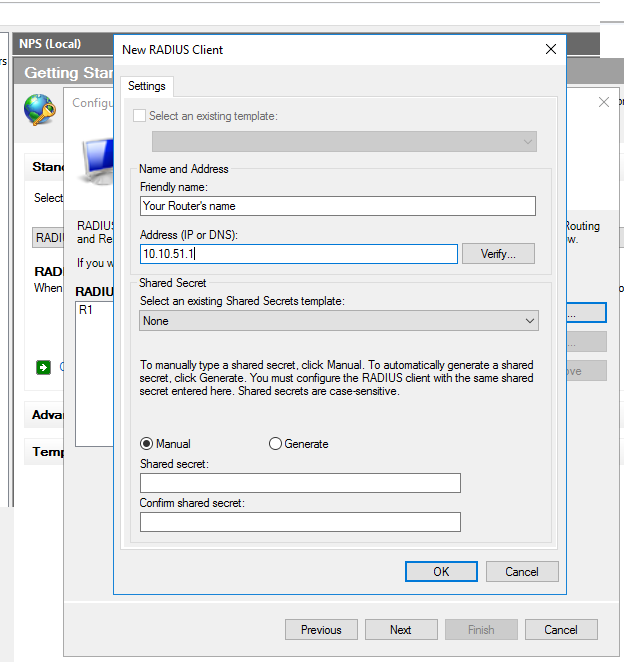
The next step is intuitive: select NPS server and Active Directory Domain to Download. After a successful download, you can find ADDS and NPS from your application list.



#### **Configuring Domain and Authentication in NPS**

First, I created and configured a TACACS+ domain that encompasses all devices under my server’s authentication process.



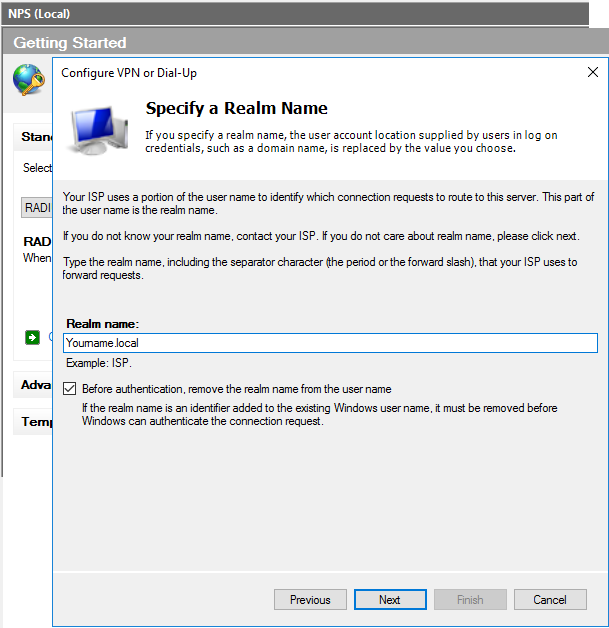


The name should be the name of your RADIUS platform.

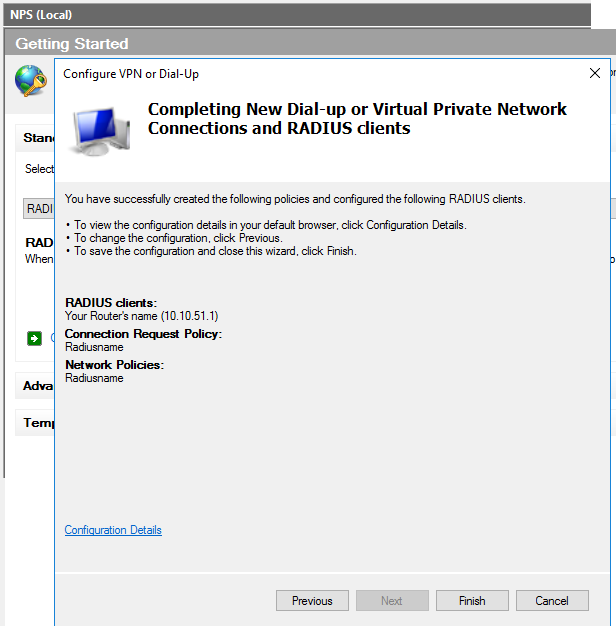
Click “next” after name it. There are a lot of “next” in this GUI, just move forward and follow along with system instruction.

Define the Friendly Name with hostname of the router. Also type in the network that your network devices are belong to.

Share Secret should be the same as the shared key in router configuration. In this lab, I set it to a simple “123456”

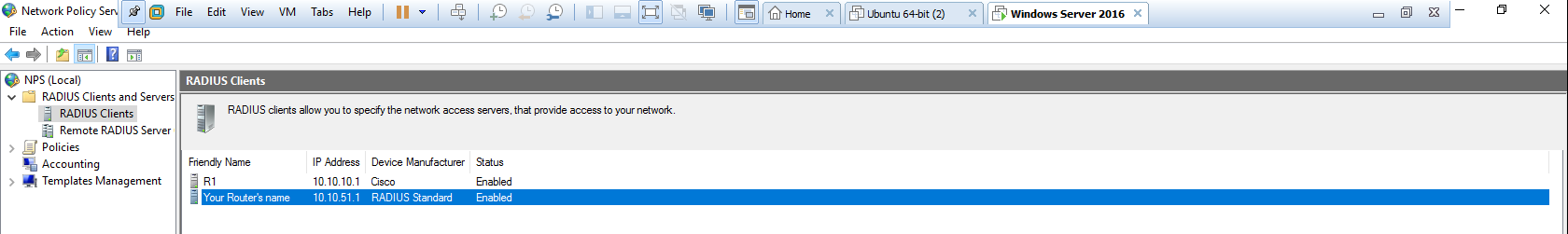


Realm name is important, because I need to use it in future steps. Since I am configuring a local network isolated from the Internet, I used “**local**” as my realm domain. You can choose a name you want before “.**local**”

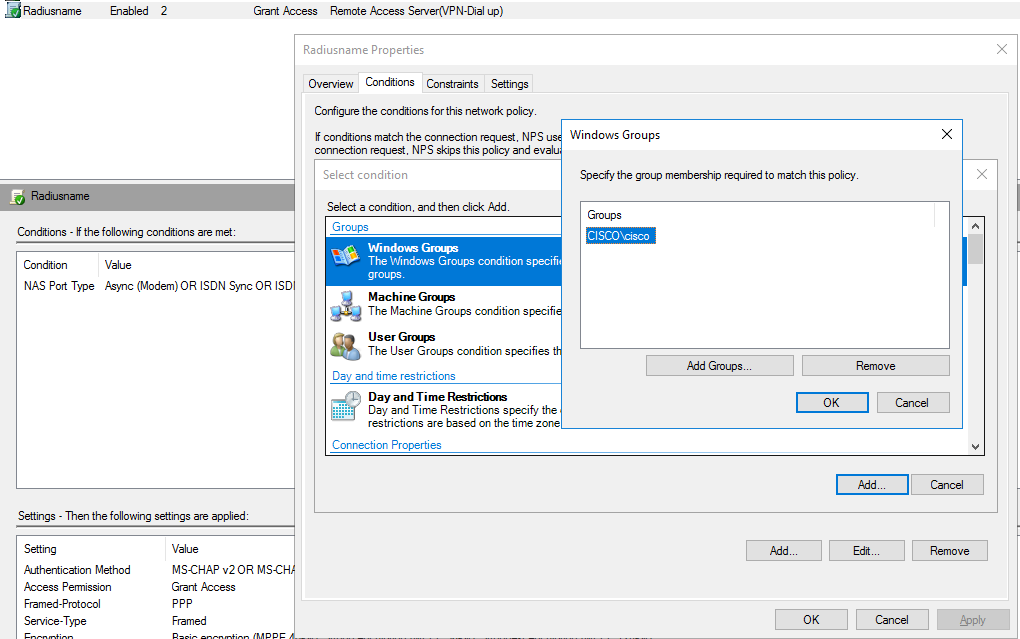
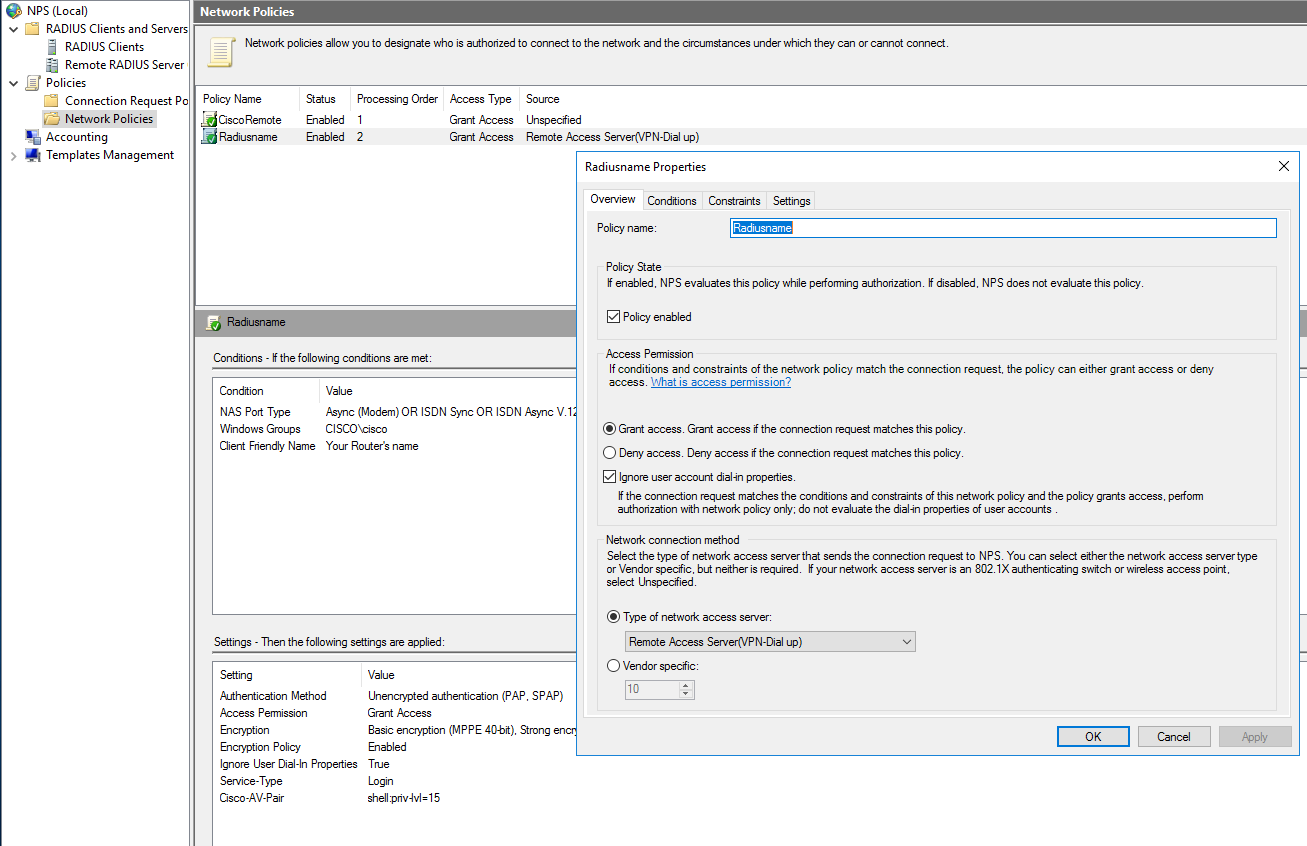


Confirmation step. After you click “Finish”, your RADIUS domain is created.

And here is the domain you just created. “cisco” is my working RADIUS domain that I configured for this project before.



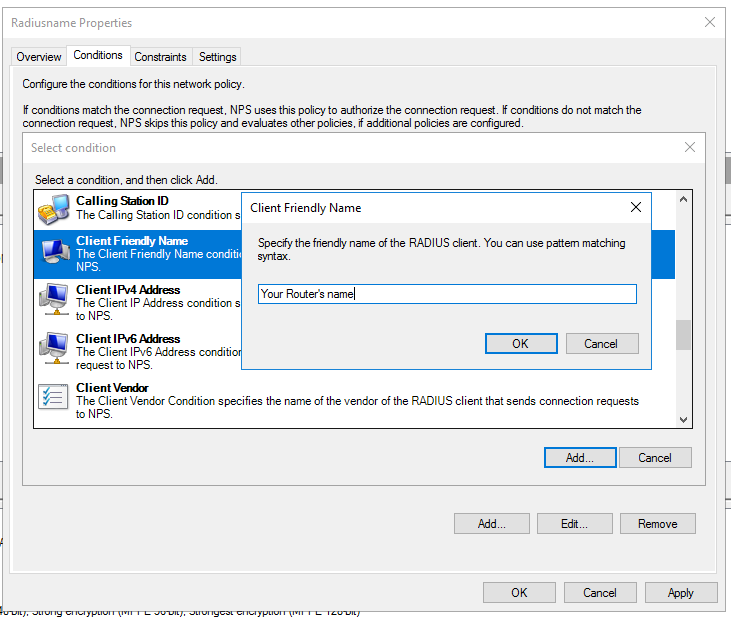
#### **Configuring Privilege and Administrative Network Policy**



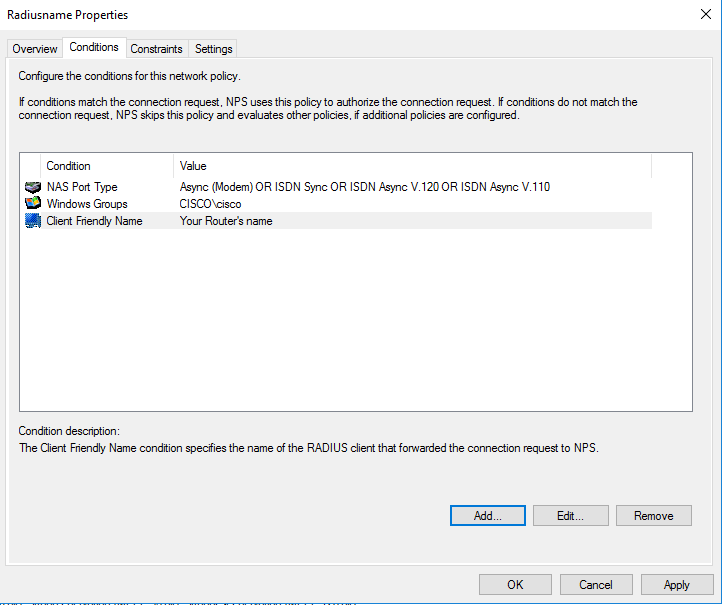
Add your domain into this policy

Next, I needed to configure specific policies in my Radius domain.

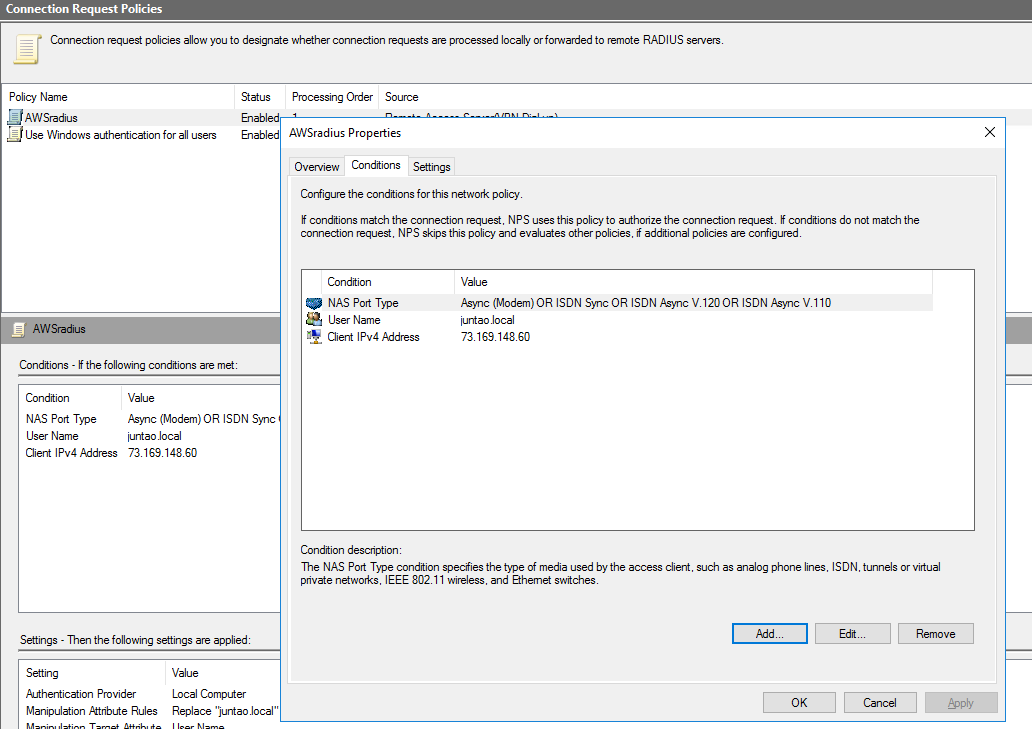
We still operate in NPS but go to “policies” tab and configure.



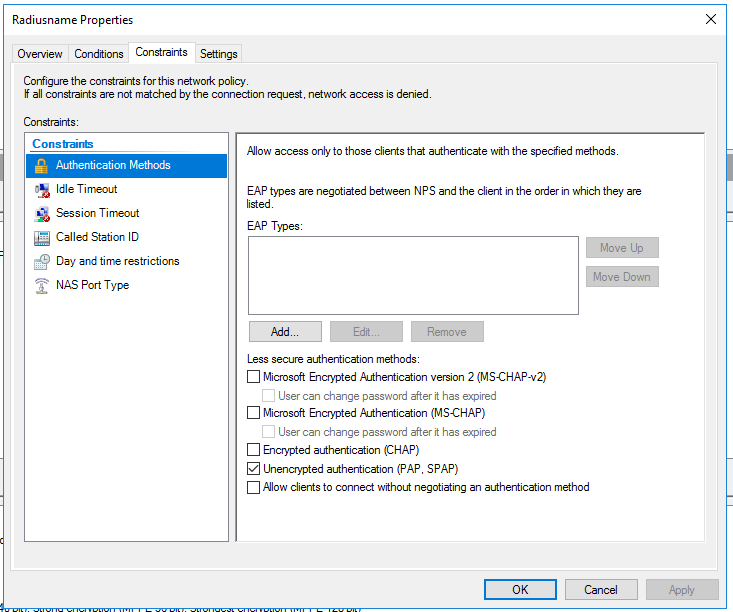
Add your router as part of this policy



Two added. Confirmed.

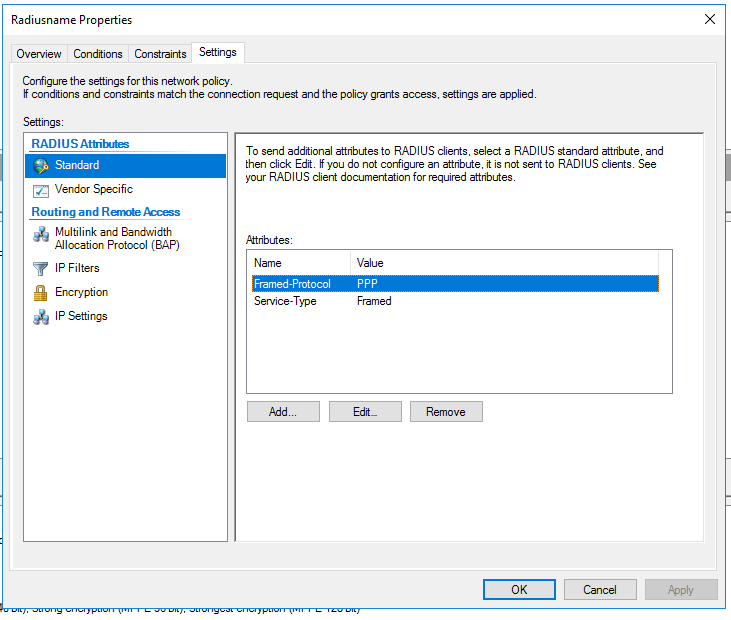


This is right. The pictures before was wrong!!!!!



Next, I needed to define what privilege/ policies I have in my domain.

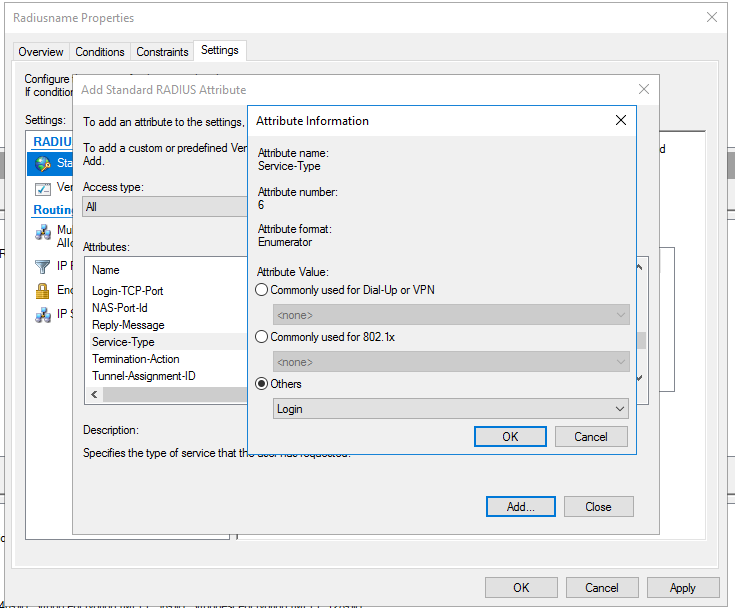
In “Constraints” tab, it showed me the Authentication Method between routers and server. Since it was a practice lab, I used unencrypted authentication.



Then, in “Settings => Standard”, I set up what service type did I want.

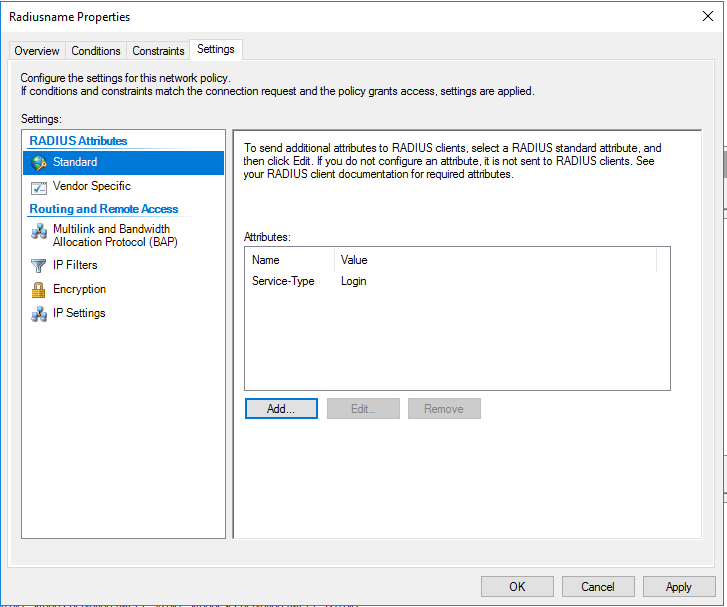
Since I want to login into my RADIUS devices, I wanted only “login” service.

So, I removed two existing Attributes.



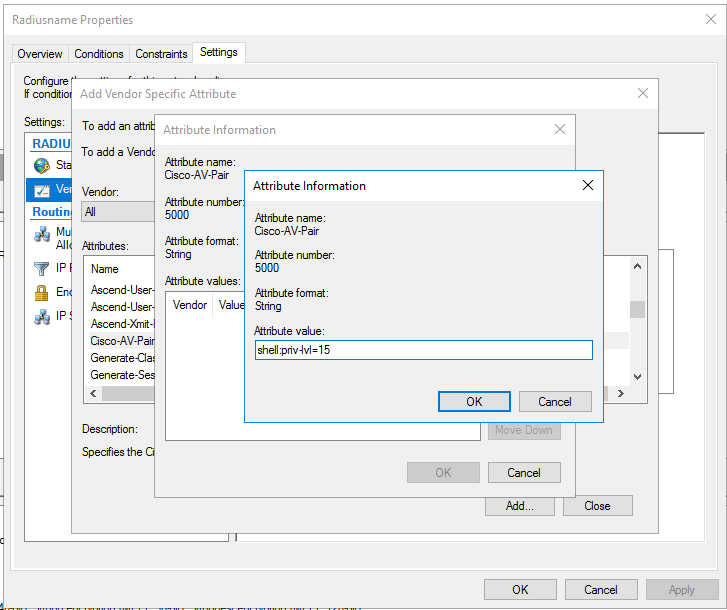
I clicked “Add” to load “login” service.

I clicked “Service-Type” and select “login” from “others”



Confirmed.

Two other original attributes are removed and “login” should be the only attribute.

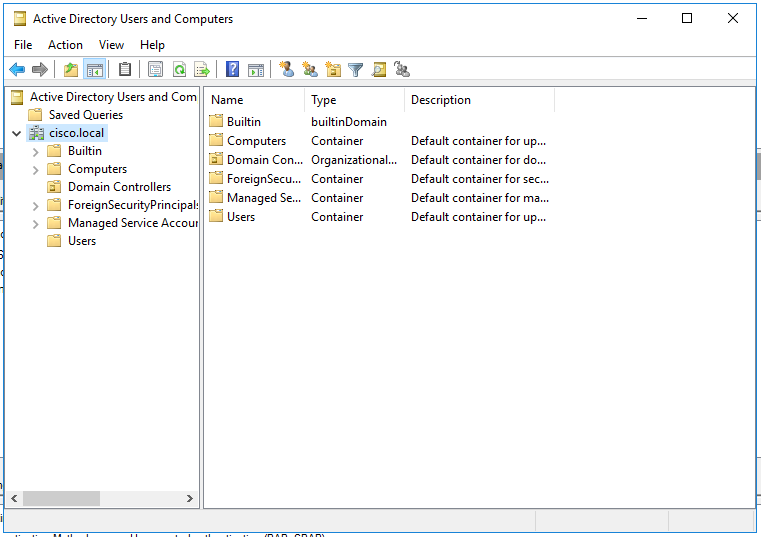


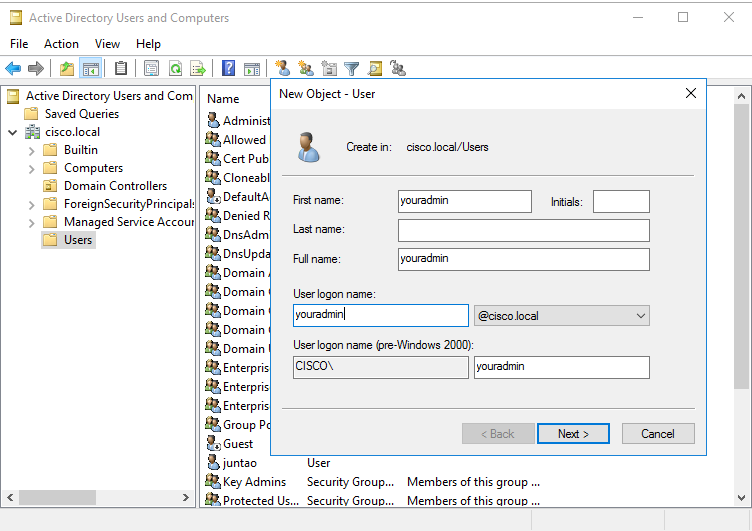
Last step in NPS: configure the privilege level for my domain.

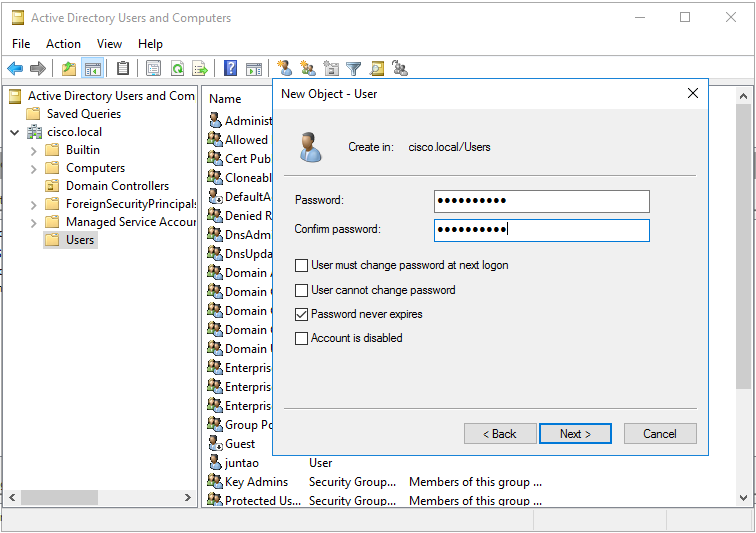
Privilege 15 means that I gave anyone in this domain unreserved privilege to view and write anything.

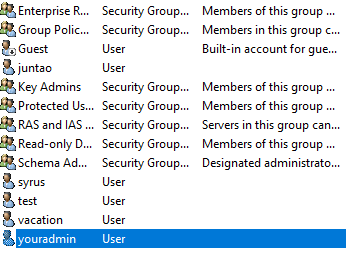
#### **Configuring Active Directory Domain Service (AD DS).**

The next step is to add various users/ administrators into my domain. Open Active Directory Domain Service. Go to your domain (cisco.local) => users and click add users. The password is configured in this process, just follow the system instruction.









User Confirmed.

Last step, add your user into the Your routers’ group.

After that, your router is ready to go. The console authentication is almost the same as TACACS+, so I didn’t show it here.

You can login into your RADIUS authenticated router now.

## Problems encountered

**Linux TACACS+ Administrator directory:**

when I first configure the file for TACACS+ user directory, I thought that I only need to add a user into the directory with username and password. My username and password didn’t work, and I couldn’t log in to my router. However, I later found I must configure a group with defined privilege, and then assign users/admins under that group for TACACS+ to work. Now it makes sense to me because a privilege level must be defined in order for TACACS+ to operate.

**Log Out of Routers, Can’t Log In:**

When I configure first AAA accounting on routers, I didn’t set up my servers correctly, which leads to the failure in authentication. Since authentication failed, I couldn’t log in or do anything with my routers. At first, I had to power cycle my routers to erase my mistake, but that took a lot of time.

To solve this question, I need to configure a local user with:

**R1(config)**# username ***backup*** secret ***#####***

Now, I can log in to and configure my routers even though TACACS+ or RADIUS don’t work. Note that this local user will only be valid when the server is unavailable (for example, unplug the ethernet cable between server and routers).

**Not Sure About the Direction to Go:**

Before, I only configured routers in cisco CLI, or computer in GUI. I was clueless when two distinct interfaces intercepted, so I spent a lot of time following websites that don’t contribute to my project. After extensive research and discussions with my classmates, I learned that basic components of AAA authentication are an active directory, a shared key between routers and server and a defined network policy, any other things are security/ management preferences. After these three essential parts, I’ve built the skeleton of my AAA accounting platform.

**VMware connection issue. Two different modes**

The last problem is not really a problem in actual servers. Because my servers are virtual servers based on the ISO images in VMware, there are two ways that a virtual computer can connect to the physical network: either uses the same ip address as the physical PC, or a unique ip address different from the physical PC. In this lab, my virtual and actual computer each needs an ip address, so I need two ip address for one physical computer.

## Summary

I used configured TACACS+ server on Linux and RADIUS server on Windows 2016. TACACS+ and RADIUS are platforms that centralizes the router login authentication into one server. In this way, I increase the security of my system because I can monitor one server more closely than a lot of individual network devices. There are three essential part of for this platform to run: group policies, user directory and authentication configuration on both routers and servers.

In this project, Linux is much easier for the people who set up the server because all essential parts of the AAA platform are concentrated in a script of codes. Though it’s not easy for a person new to Windows Server to find the right buttons to click, it has a lot of extra features and explicated options; and after all, it has a nice graphic user interface close to PCs that we use every day, which makes it more user friendly in some degree.