Jeffrey Xu CCNP 6-7

12/4/18

Cloud Tacacs and Radius

VMWARE? Azure and AWS

company name

**purpose:**

This lab was an extension on our previous lab, where we would need to set up Tacacs and Radius on a local VM and run it through our network. A VM is an abbreviation for Virtual Machine, it basically allows us to run a server, or a computer virtually on another computer without actually changing our computers own operation systems. While this lab, we would have to set up a VM on the cloud and string it through the public network into our own network. This would be vital for the work force because most of the time, servers on the cloud will be more secure and reliable then running it locally if configured correctly.

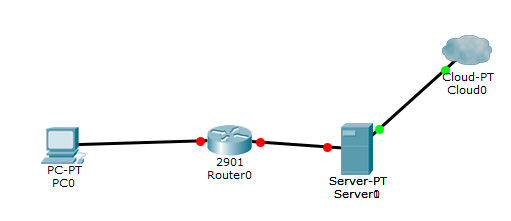
**BACKGROUND:**

As mentioned in the previous paragraph, this lab required us to set up a VM on the cloud through Azure or AWS (Amazon Web Services). Setting us the VMs would allow us to access them anywhere around the world and configure them on the go. So in total, we needed to set up 4 VMs, 2 in Azure and 2 in AWS. Before we started with setting up the VMs, we needed to check the differences between setting up a VM on a local machine and setting it up on the cloud. And to be honest, there isn’t much difference between the 2 with the extra need to Port-forward the correct ports for each type of authentication (Tacacs or Radius), and instead of placing the internal ip address, the external ip address is needed to be used for Radius. Almost everything was going to be the same as setting it up locally.

**LAB SETUP:**

The lab was very simple to setup. The core devices and software needed was going to be a PC, Azure or AWS, and 1 2901 router for testing. I would recommend to have more than one router present if this is your first time setting it up because there is a high chance you will lock yourself out and need to restart. The other differences between the two labs is that now, we need to bridge our networks. The act of bridging a network means to share internet access between two different interfaces. So in our case, we need to share our Wi-Fi connection to our Ethernet port so our routers can acquire a dhcp address.

**Topology:**



Tacacs-Router Configuration:

hostname Tacacs\_router

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

vtp mode transparent

username backup password 0 cisco

interface GigabitEthernet0/0

ip address dhcp

duplex auto

speed auto

ip tacacs source-interface GigabitEthernet0/1

tacacs-server host 18.191.71.165

tacacs-server directed-request

tacacs-server key cisco123

Radius-Router Configuration:

hostname Radius\_router

boot-start-marker

boot-end-marker

aaa new-model

aaa authentication login default group radius local

aaa authorization exec default group radius if-authenticated

vtp domain cisco

vtp mode transparent

username admin privilege 15 secret cisco

interface Embedded-Service-Engine0/0

no ip address

shutdown

interface GigabitEthernet0/0

ip address dhcp

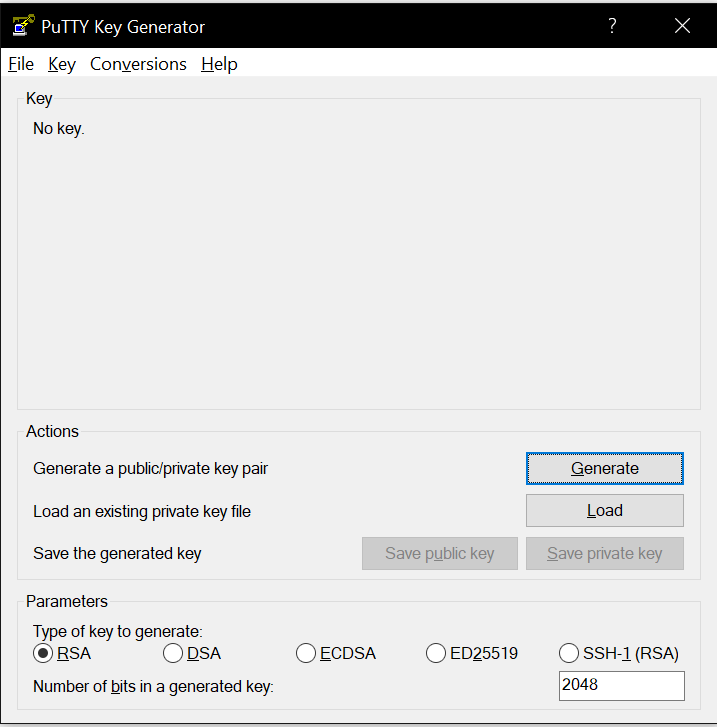
no shutdown

duplex auto

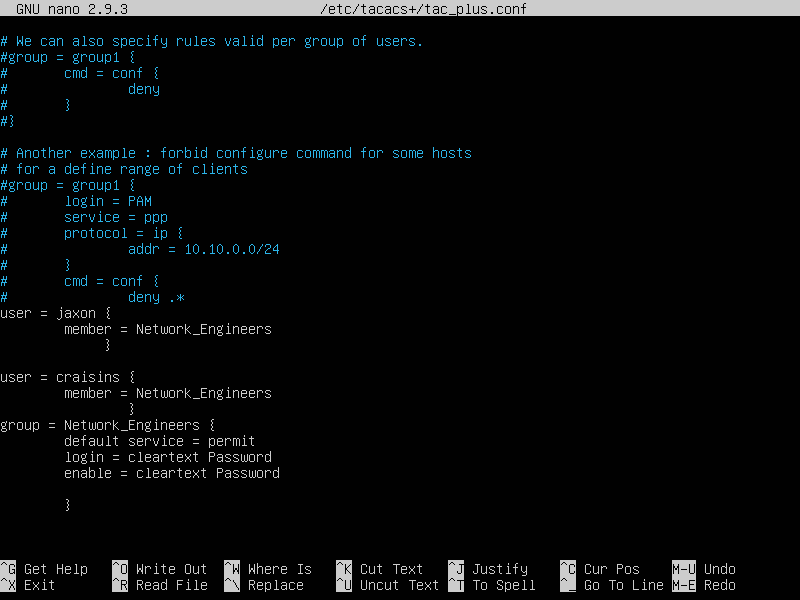
speed auto

Radius-server host "Server Public ip"

Radius-server key 0 cisco123

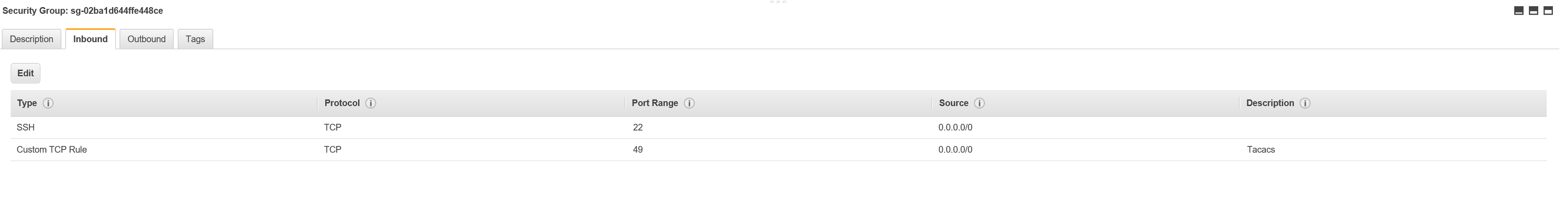
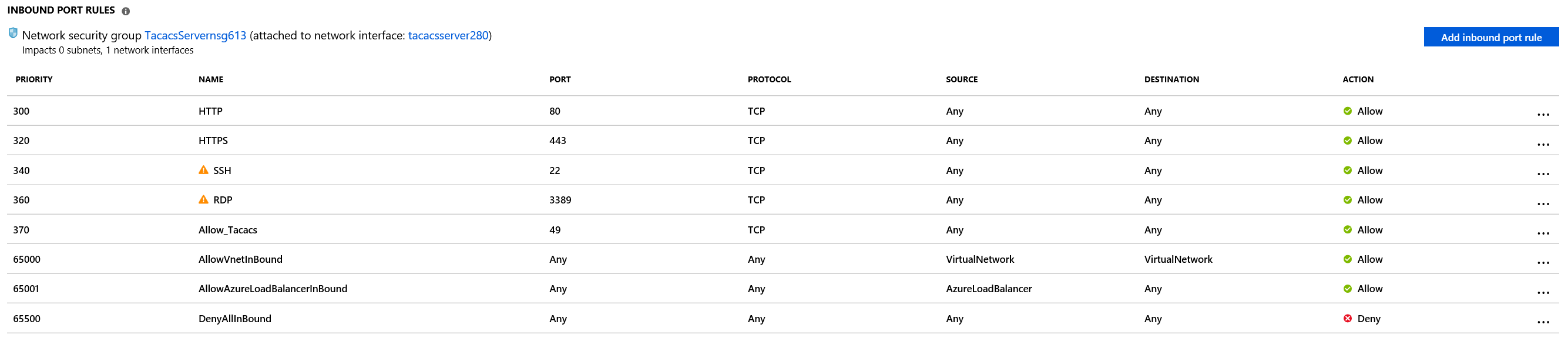


This is called Putty-gen, you need to use this application to be able to ssh to your server if you are using AWS by taking the given key and creating a private key so you can ssh from Putty. If you are using Azure, you don’t need to worry about this because they supply you with a terminal on the website.



We are going to be using the same Tacacs server configurations.

You can configure the file by first downloading Tacacs on the server, then doing using the sudo nano command to get to this file.



This is where you are going to port forward any of the ports that you need.

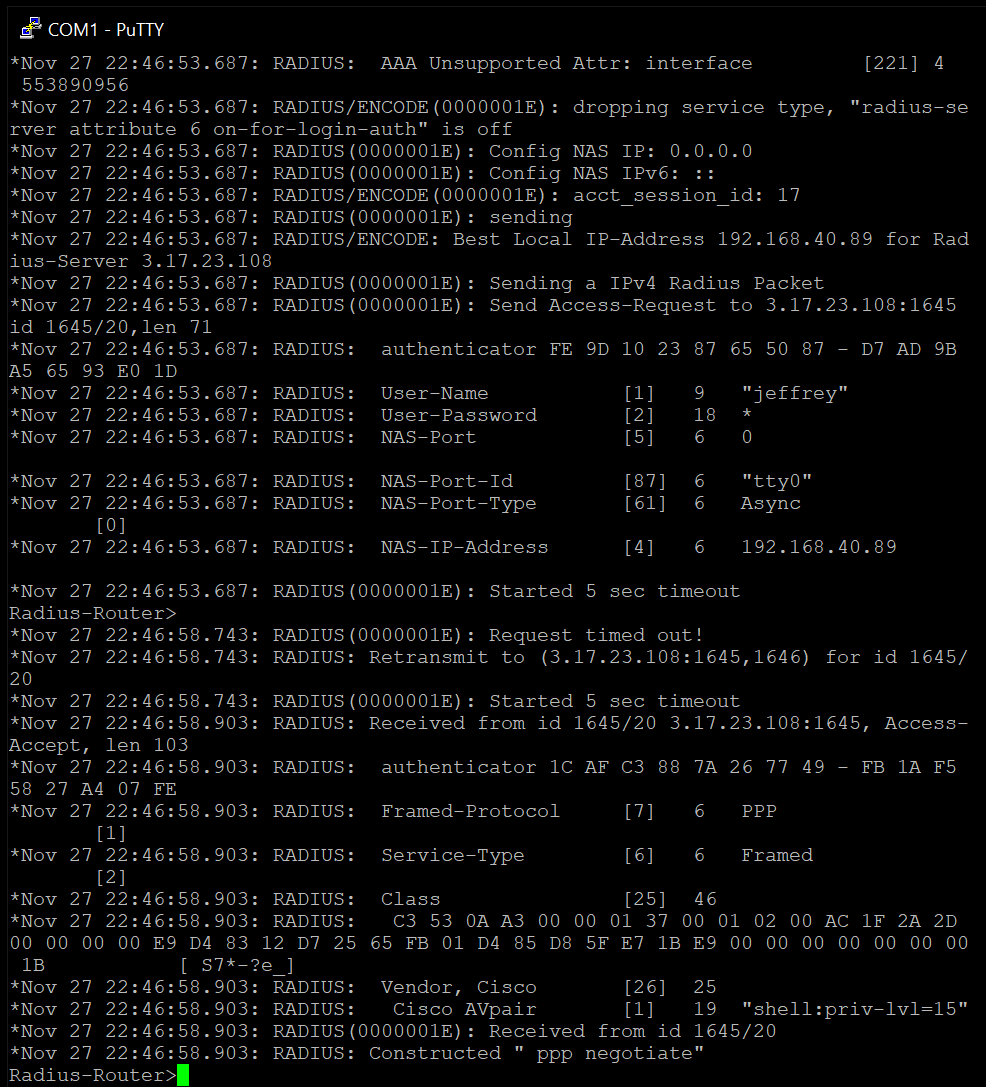
The top one is Azure under the “Networking tab”

While the bottom one is AWS under the “Security Group Tab”

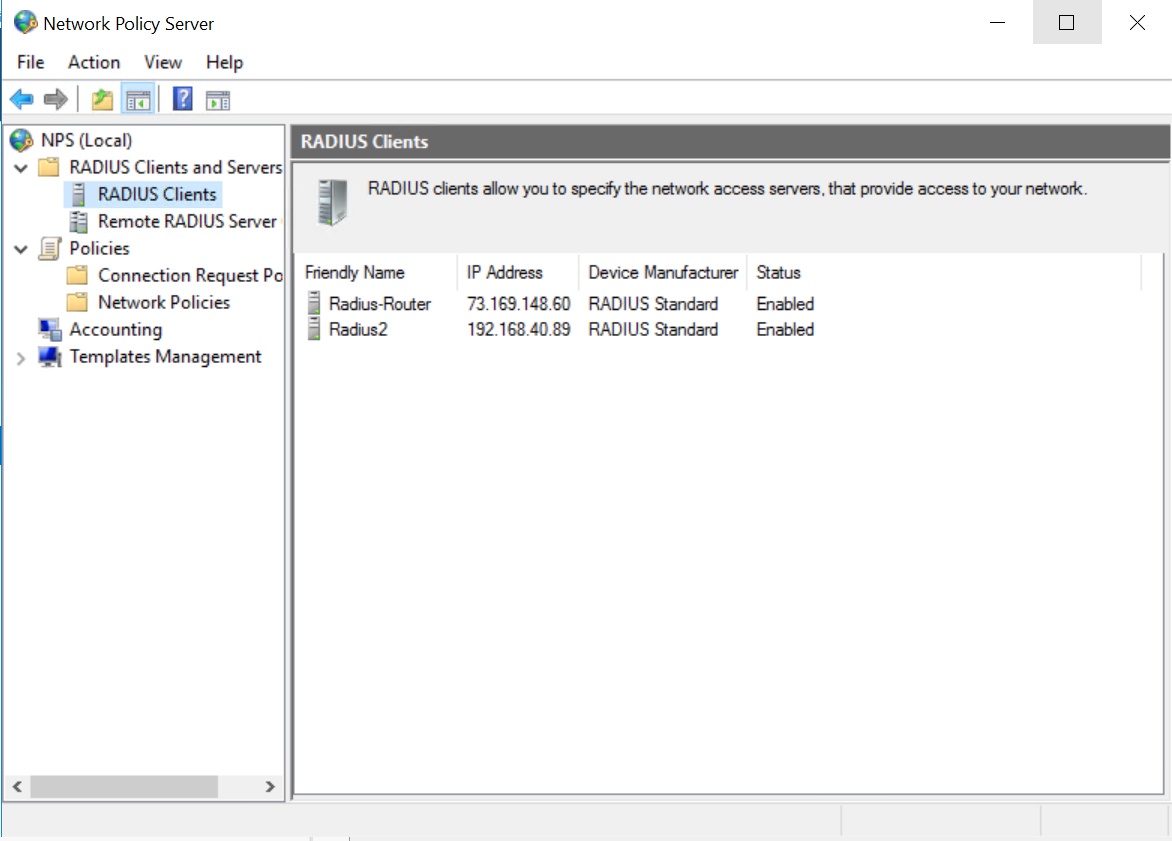
The port for Tacacs is TCP: 49

The ports for Radius is UDP: 1645-1646

SSH is allowed by default on AWS but if you want to enable it on Azure you are going to need to enable it manually during setup or in the Networking tab.

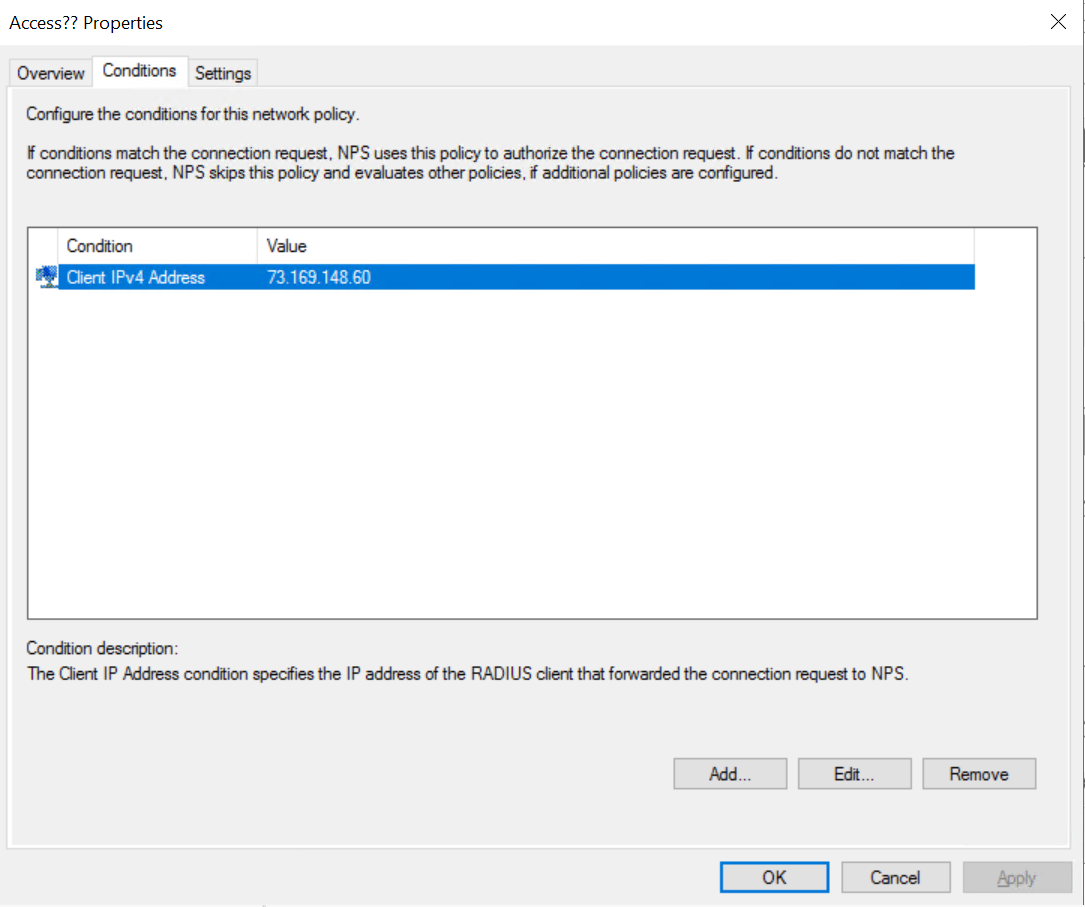


Proof of Radius is Authenticating.



This is where I put the Radius clients. The public ip address is the one being used.

Note: On the Cisco Wi-Fi, we are using PAT, so everyone on the network has the same External IP address.

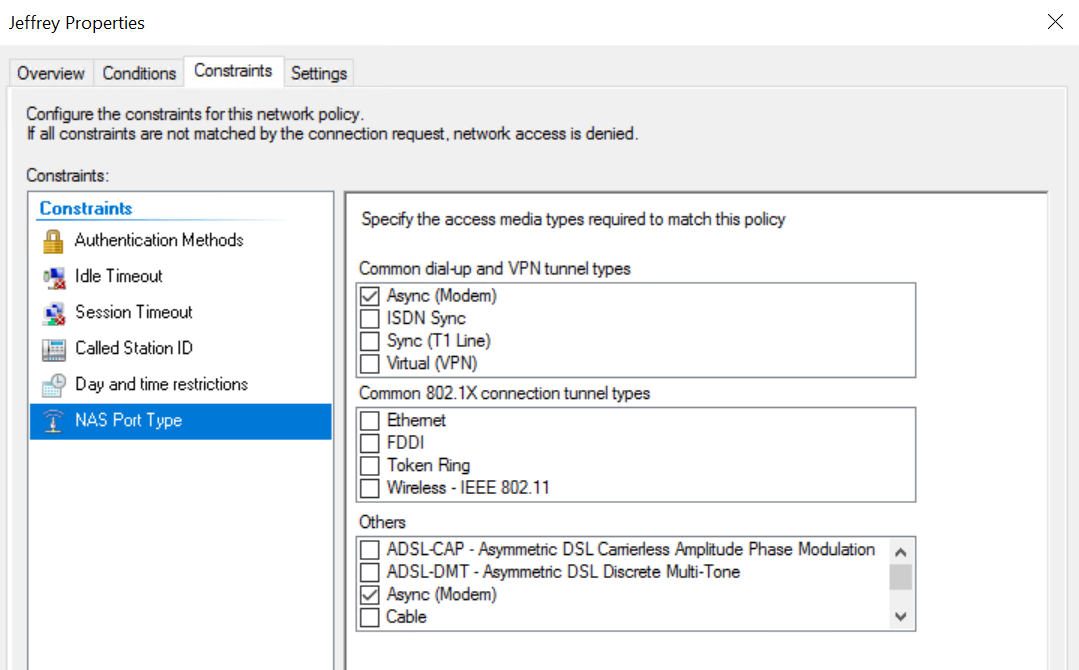


This is my connection policy where I only let the IP address of my computer allowed to be authenticated on this server.

D:\Radius-Tacacs_-_ Cloud\Radius\Config Pictures\Domain Admins.PNGD:\Radius-Tacacs_-_ Cloud\Radius\Config Pictures\User.PNG

These are the group and user I made. The User is placed in the group while the group is placed into the network policy and connection policy to have all the members in the group to be able to be authenticated by the server.

Problem Pictures:



Problems:

There were no problems with setting up Tacacs, but with radius, there were a few problems with setting it up. Such as selecting Async as the NAS port type. Although it doesn’t affect how it authenticates, if it is not matching with your router, it will deny authentication. And if you do a debug Tacacs or Radius and you can’t connect to the server, you need to check your Port Forwarding for the correct protocol and port range. The last problem I had was when I had authenticated successfully only because I had specifically allowed my user without going through the NPAS. But I figured out that it was because I had a windows group in my network policy and had to remove it from the settings and that allowed NPAS to authenticate correctly.

Conclusion: If the internet didn’t drop every few seconds and the bridging was more reliable, I would have finished it on an earlier time because it would prevent me from testing my configurations. But I’m glad that im finished with it.