Radius and Tacacs+ Lab

-Tanishk Singh

Purpose

This lab was performed to carry out the intent to acquire the ability and the experience to conduct *AAA (Authentication, Authorization, and Accounting)* servers running orchestrating *RADIUS (Remote Authentication Dial-In User Service)* and *Tacacs+ (Terminal Access Controller Access-Control System Plus)*, to be further used in another lab, to implement cloud networking. Another skill that this lab would grant us was to understand the working of a Windows Server 2016 and the development of RADIUS on that server, as it would be the server undergoing RADIUS, and understand the operation and installation of TACACS+ on a Linux Server, as it would be the server running TACACS+. During the process of the lab, we would get acquainted with the commands required to insert on the router to establish a connection with the node’s respective AAA server (either running TACACS+ or RADIUS) to grant or deny access to the node when used to gain access to the node.

Background Information

Prior to begin the lab, we researched a bit on the elements that we would work on. We started our research with the two servers that we would implement in the topology. Windows Server 2016 is an Operating System developed by Microsoft, developed concurrently with Windows 10. The server, like some would believe is not a server that can only operate single task at a time or has a fixed role. Instead, Windows Server 2016 can run numerous features at the same time. The list of features that the server can function are: DHCP Server, DNS Server, Active Directory Services (plays an integral part in the completion of this lab) etc. A Linux server is a high-powered variant of the Linux open source operating system that's designed to handle the more demanding needs of business applications such as network and system administration, database management and Web services. Linux servers are frequently selected over other server operating systems for their stability, security and flexibility advantages. The Linux Server Operating system that would be used in this lab would be the Ubuntu OS.

AAA is term for a framework for intelligently controlling access to computer resources, enforcing policies, auditing usage, and providing information necessary to bill for services. These combined processes are considered essential for effective network management and policies. AAA stands for Authentication, Authorization, and Accounting, which is also the procedures for its working.

The first process, *Authentication*, provides a way of identifying a user, typically by having the person attempting to login, enter a valid user name and a password. Then, the AAA Server compares the input of the user with other user credentials stored in a database, where, if the credentials match, then the user is granted access, and if the it varies, then the authentication fails, and the network is denied access.

After the process of Authentication is complete, the permitted user must gain *Authorization* for doing certain tasks. For instance, after logging into a system, the user may try to issue some commands. The Authorization process, however, determines if the user has the authority to issue such commands. Authorization is procedure of enforcing policies: determining what types or qualities of activities, resources, or services a user is permitted.

The final plank in the AAA framework is *Accounting*, which measures the resources the user consumes during access. This can include the amount of system time or the amount of data a user has sent and/or received during a session. Accounting is carried out by logging of session statistics and usage information and is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.

After acquainting ourselves with components that we would be working on, we began researching on the protocols that we would be using. RADIUS is a networking protocol that works on port 1812, that provides centralized AAA management for users who use a network service. It is a client and server protocol that operates in the application layer, and can therefore, user either TCP/UDP as a means of transport. *NAS (Network Access Servers)*, the gateways that control access to a network, contain a RADIUS client component that communicates with server. As mentioned above, the user attempts to gain access to the network through a node. Once the attempt to login is made, the RADIUS Server responds with one of the three responses: -

Access Reject: The user is unconditionally denied access to all requested network resources.

Access Challenge: Server requests additional information from the client, such as, secondary password, a PIN, etc.

Access Accept: The user is granted access into the network.

TACACS+ is a Cisco proprietary, security application that provides centralized validation of users attempting to gain access to a node. Like RADIUS, TACACS+ services are maintained in a database. Almost everything about the theoretical functioning of TACACS+ is similar to that of RADIUS’. The major difference between RADIUS and TACACS+ is that TACACS+ operates on TCP/UDP port number 49, and unlike RADIUS, that encrypts only the users' password as it travels from the RADIUS client to RADIUS server, and rest pf the information including the user name, Authorization, accounting, etc. are transmitted in clear text, which makes it vulnerable to different types of attacks, TACACS+ encrypts the whole information, making it less vulnerable and safer than RADIUS.

Lab Summary

Unlike other times, we did not begin our lab on Cisco Packet Tracer because the level and type of work necessary for the completion of the lab, substantially varies from the level and type of work needed to be done in the actual nodes and server. One of the major difference being that Cisco Packet Tracer does not have the Operating Systems built in the servers that we worked on, hence, not giving us the experience required in this lab. Instead, we initiated the lab on the actual nodes and Virtual Machines of OS of the servers imported in a software called, *VMWare*, that gives us the exact experience that one would gain while working on the Operating systems on an actual machine.

The initial work was done on the Virtual Servers, where on a Windows Server, we had to, at first, configure a domain to connect all the users to, then make a user to login to and add it to a group that would consist all the user names, eligible for logging in to a node. Afterwards, we added the *NPAS (Network Policy and Access Services)* feature to the server, for permitting the client in the topology for logging in to the network and adding the user name and group containing the user name to the database through which, the server could match the credentials and allowing or rejecting an attempt made on the node connected to the server.

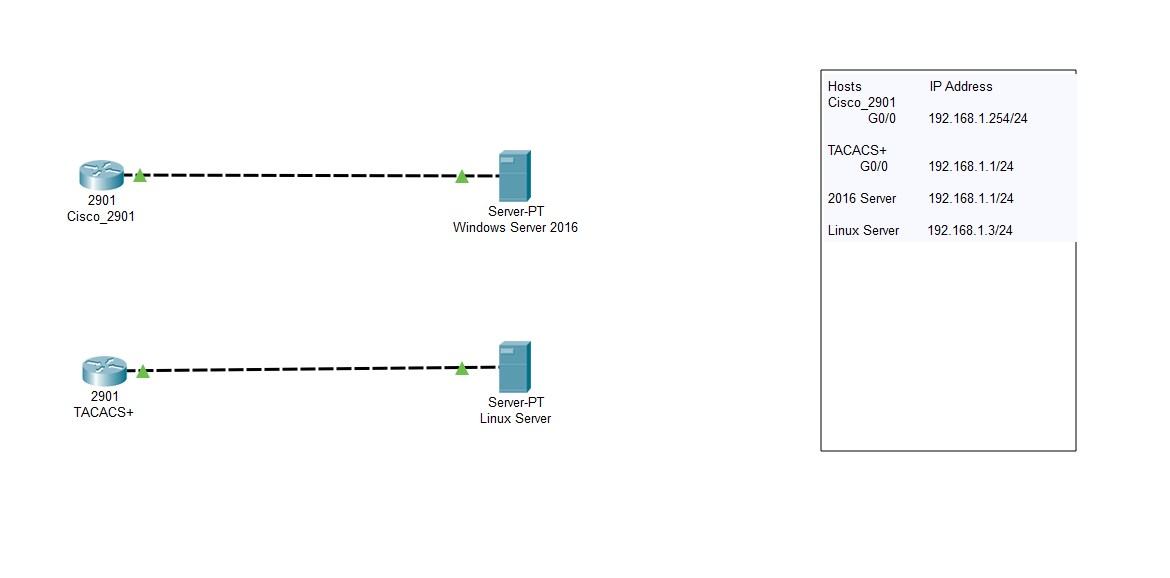
Meanwhile, we connected the Linux machine to the internet to install and updated all the existing features using certain commands. After the process of updating the machine was completed we did some additional commands to install TACACS+ on the server and making a user directory in it.

After the configuration of the servers was finalized, we advanced to configure the routers for the proper AAA functioning for their respective servers that they were connected to. The process was concluded with the use of certain commands mentioned below.

Commands

|  |  |
| --- | --- |
| **aaa authentication login**{**group***group-list*[**none**]| **local**| **none**} | Configures login authentication methods.  The *group-list* argument consists of a space-delimited list of group names. The group names are the following:  **radius**—Uses the global pool of RADIUS servers for authentication.  https://www.cisco.com/c/dam/en/us/td/i/templates/blank.gif*named-group*—Uses a named subset of TACACS+ or RADIUS servers for authentication.  The **local** method uses the local database for authentication. The **none** method uses the username only.  The default console login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond. |
| **aaa authentication login default**{**group***group-list*[**none**]| **local**| **none**} | Configures the default authentication methods.  The *group-list*argument consists of a space-delimited list of group names. The group names are the following:  **radius**—Uses the global pool of RADIUS servers for authentication.  *named-group*—Uses a named subset of TACACS+ or RADIUS servers for authentication.  The **local** method uses the local database for authentication. The **none** method uses the username only.  The default login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond. |
| **aaa accounting default**{**group***group-list*| **local**} | Configures the default accounting method.  The *group-list* argument consists of a space-delimited list of group names. The group names are of the following:  **radius**—Uses the global pool of RADIUS servers for accounting.  *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting.  The **local** method uses the local database for accounting.  The default method is **local,**which is used when no server groups are configured or when all the configured server groups fail to respond. |
| **aaa new-model** | Command to enable AAA configuration on the Cisco node |
| **Radius-server host** *{IP Address}* **key** *{shared key}* | Command required to be configured on the mode to establish a connection with the RADIUS server. |
| **Tacacs-server host** *{IP Address}* **key** *{shared key}* | Command required to be configured on the mode to establish a connection with the TACACS+ server. |

Topology and IP Addressing Scheme



Configurations and Result

Cisco\_2901 Configuration

hostname Cisco\_2901

boot-start-marker

boot-end-marker

aaa new-model

aaa authentication login default group radius local

aaa authorization exec default group radius if-authenticated

aaa session-id common

memory-size iomem 5

ip cef

no ipv6 cef

multilink bundle-name authenticated

voice-card 0

license udi pid CISCO2901/K9 sn FTX1520806H

license accept end user agreement

license boot module c2900 technology-package securityk9

license boot module c2900 technology-package uck9

vtp domain cisco

vtp mode transparent

username admin privilege 15 secret 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY

redundancy

interface Embedded-Service-Engine0/0

no ip address

shutdown

interface GigabitEthernet0/0

ip address 192.168.1.254 255.255.255.0

duplex auto

speed auto

interface GigabitEthernet0/1

no ip address

duplex auto

speed auto

interface GigabitEthernet0/1/0

no ip address

shutdown

duplex auto

speed auto

ip forward-protocol nd

no ip http server

no ip http secure-server

radius-server host 192.168.1.1 key AAARadius

control-plane

mgcp profile default

gatekeeper

shutdown

line con 0

line aux 0

line 2

no activation-character

no exec

transport preferred none

transport input all

transport output pad telnet rlogin lapb-ta mop udptn v120 ssh

stopbits 1

line vty 0 4

transport input all

scheduler allocate 20000 1000

end

TACACS+ Configuration

hostname TACACS+

boot-start-marker

boot-end-marker

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 session-id common

memory-size iomem 10

ip cef

no ipv6 cef

multilink bundle-name authenticated

voice-card 0

license udi pid CISCO2901/K9 sn FTX1520806J

license accept end user agreement

license boot module c2900 technology-package securityk9

license boot module c2900 technology-package uck9

vtp domain cisco

vtp mode transparent

redundancy

interface Embedded-Service-Engine0/0

no ip address

shutdown

interface GigabitEthernet0/0

no ip address

shutdown

duplex auto

speed auto

interface GigabitEthernet0/1

ip address 192.168.1.1 255.255.255.0

duplex auto

speed auto

interface Serial0/0/0

no ip address

shutdown

clock rate 2000000

interface Serial0/0/1

no ip address

shutdown

clock rate 2000000

interface GigabitEthernet0/1/0

no ip address

shutdown

duplex auto

speed auto

ip forward-protocol nd

no ip http server

no ip http secure-server

tacacs-server host 192.168.1.3

tacacs-server directed-request

tacacs-server key IPBALANCE

control-plane

mgcp profile default

gatekeeper

shutdown

line con 0

line aux 0

line 2

no activation-character

no exec

transport preferred none

transport input all

transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh

stopbits 1

line vty 0 4

transport input all

scheduler allocate 20000 1000

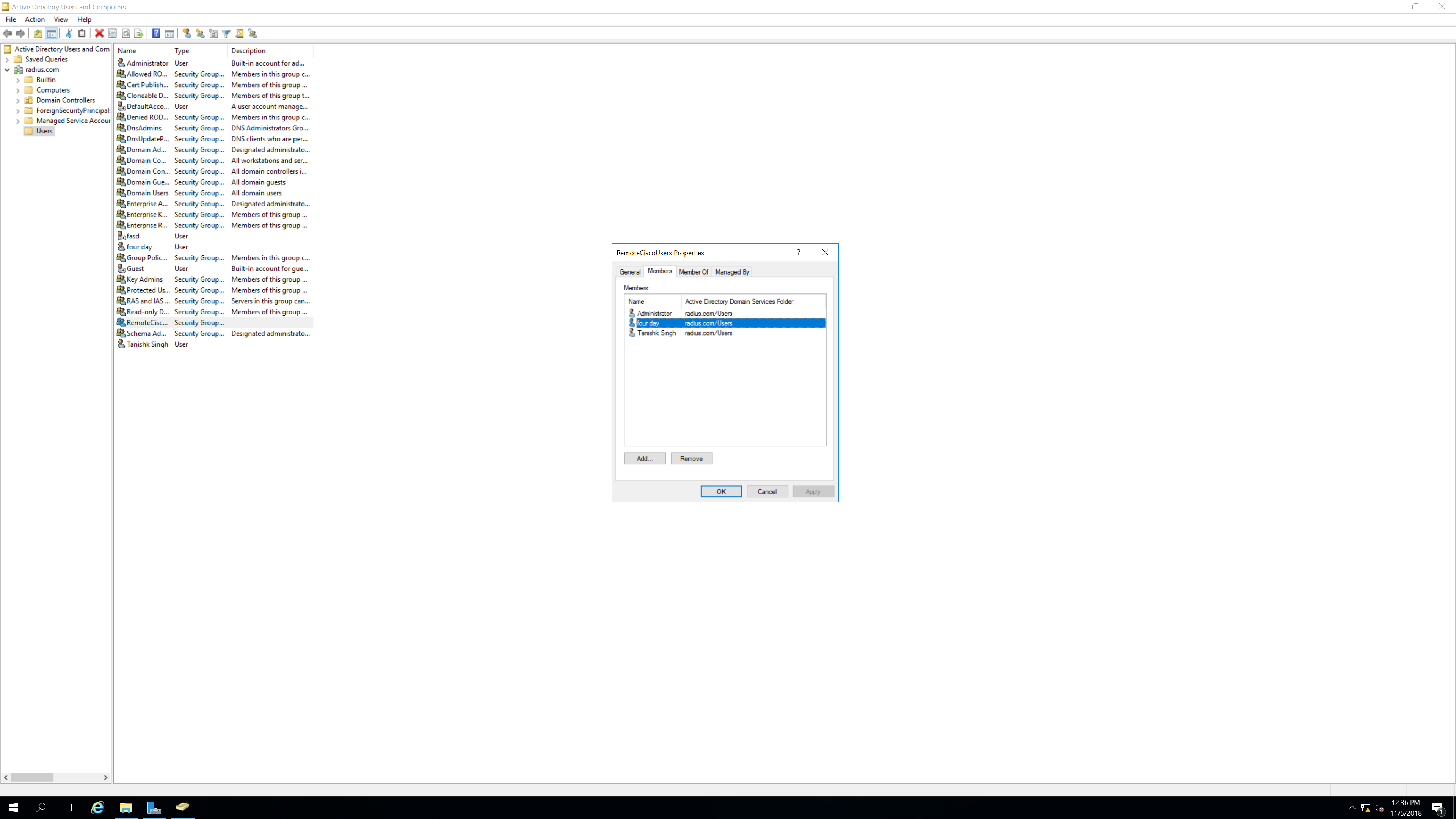
end

Windows Server 2016 Screenshots

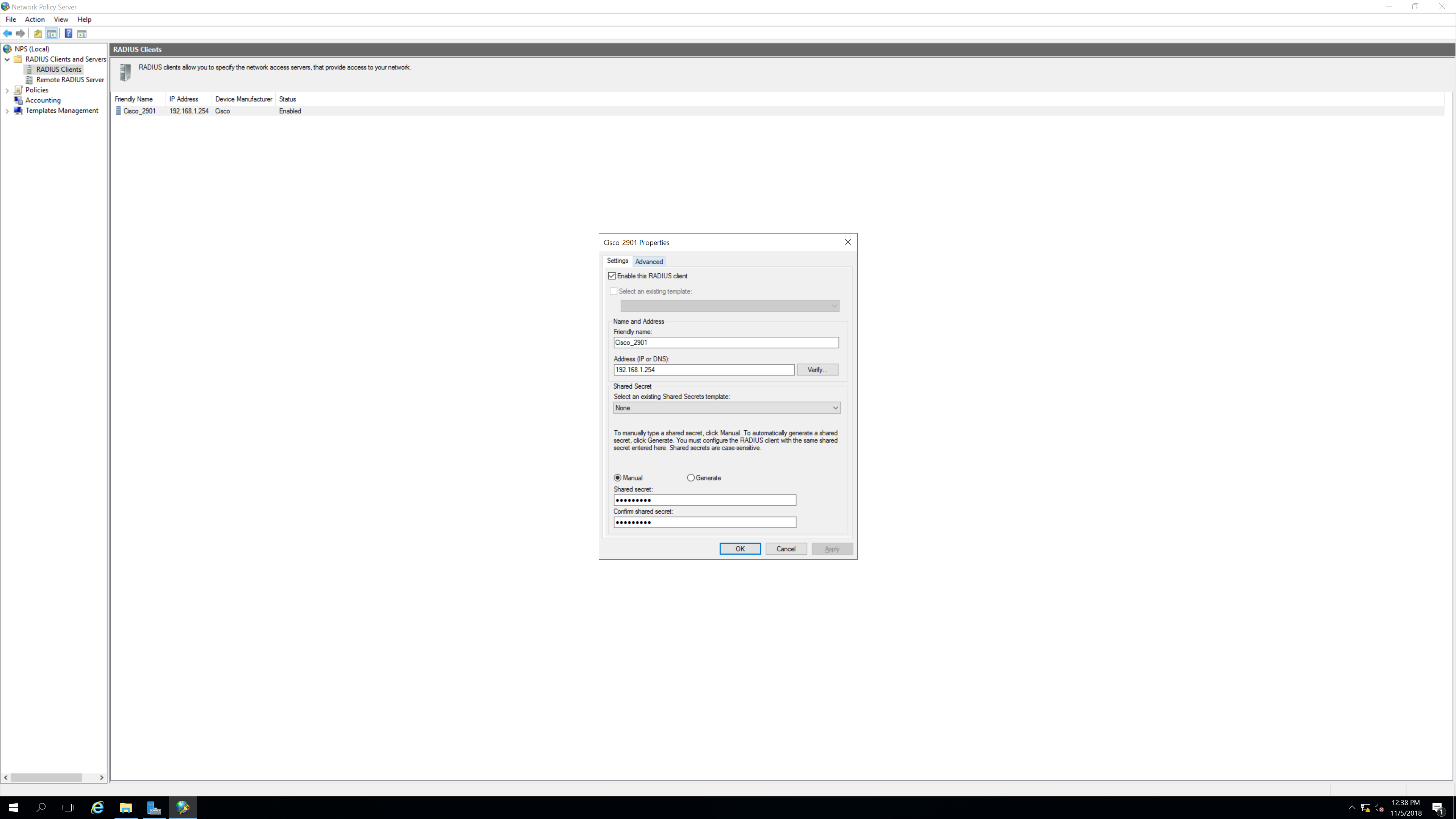
Step 1.



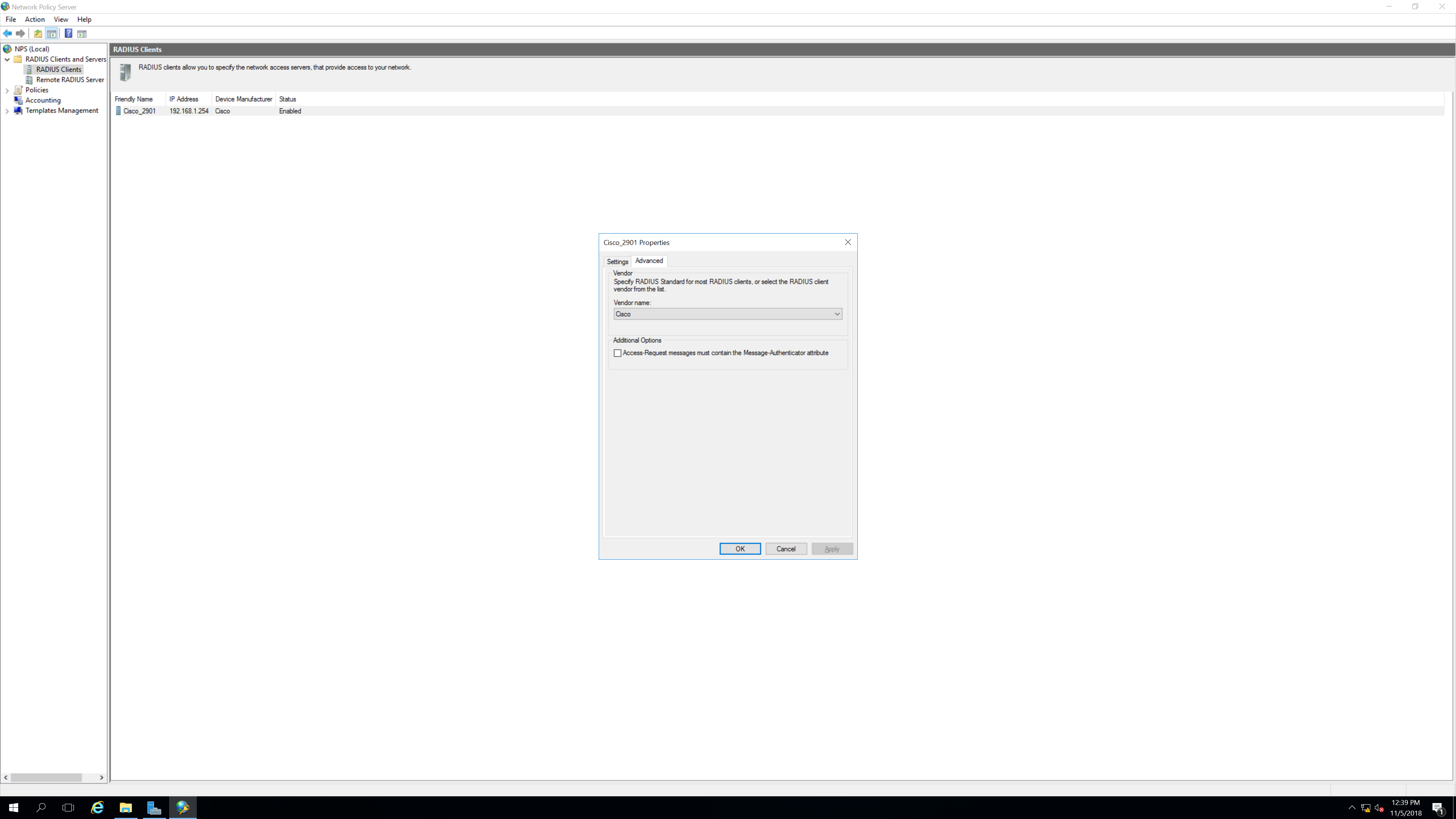
Step 2.



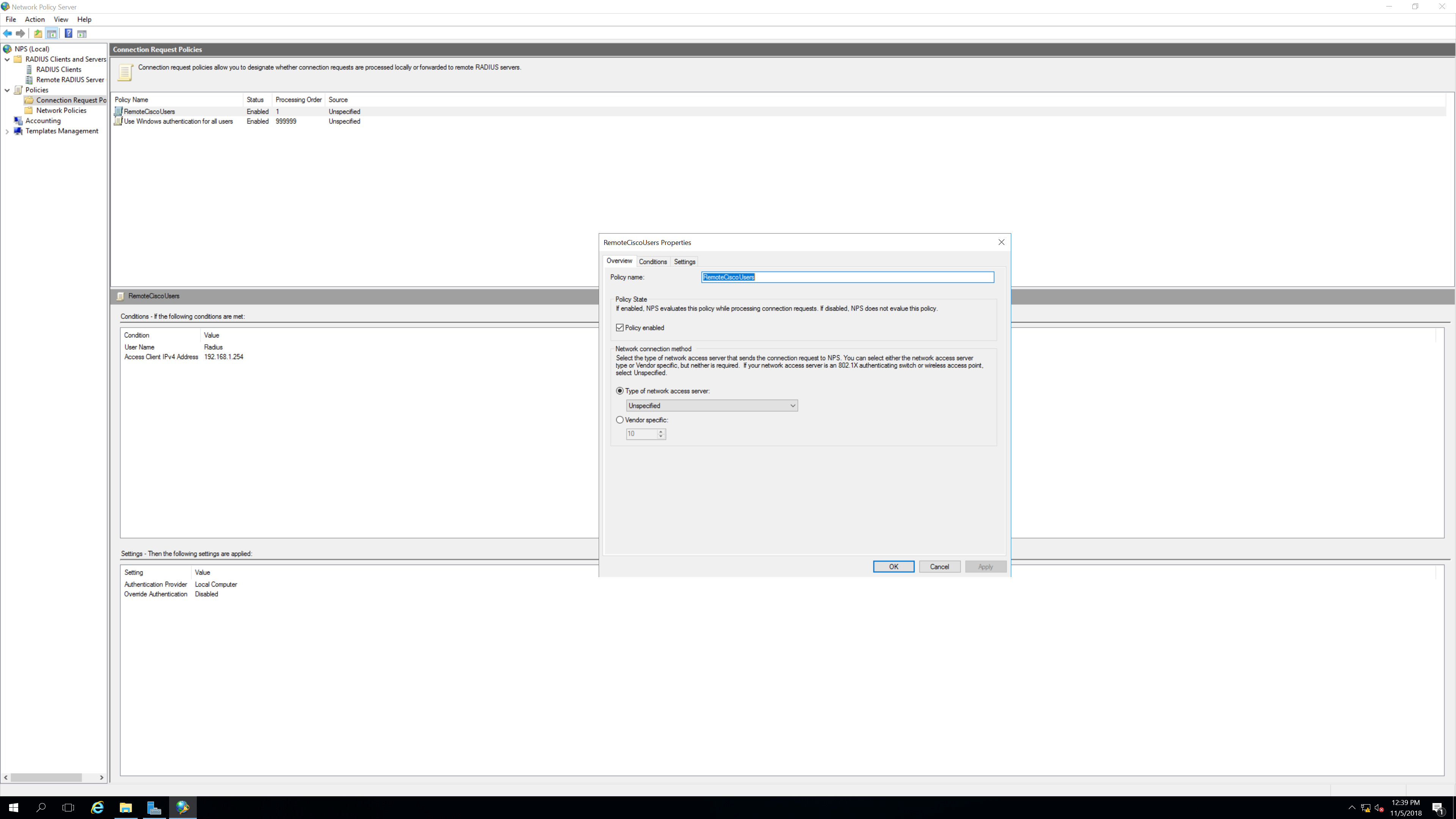
Step 3.



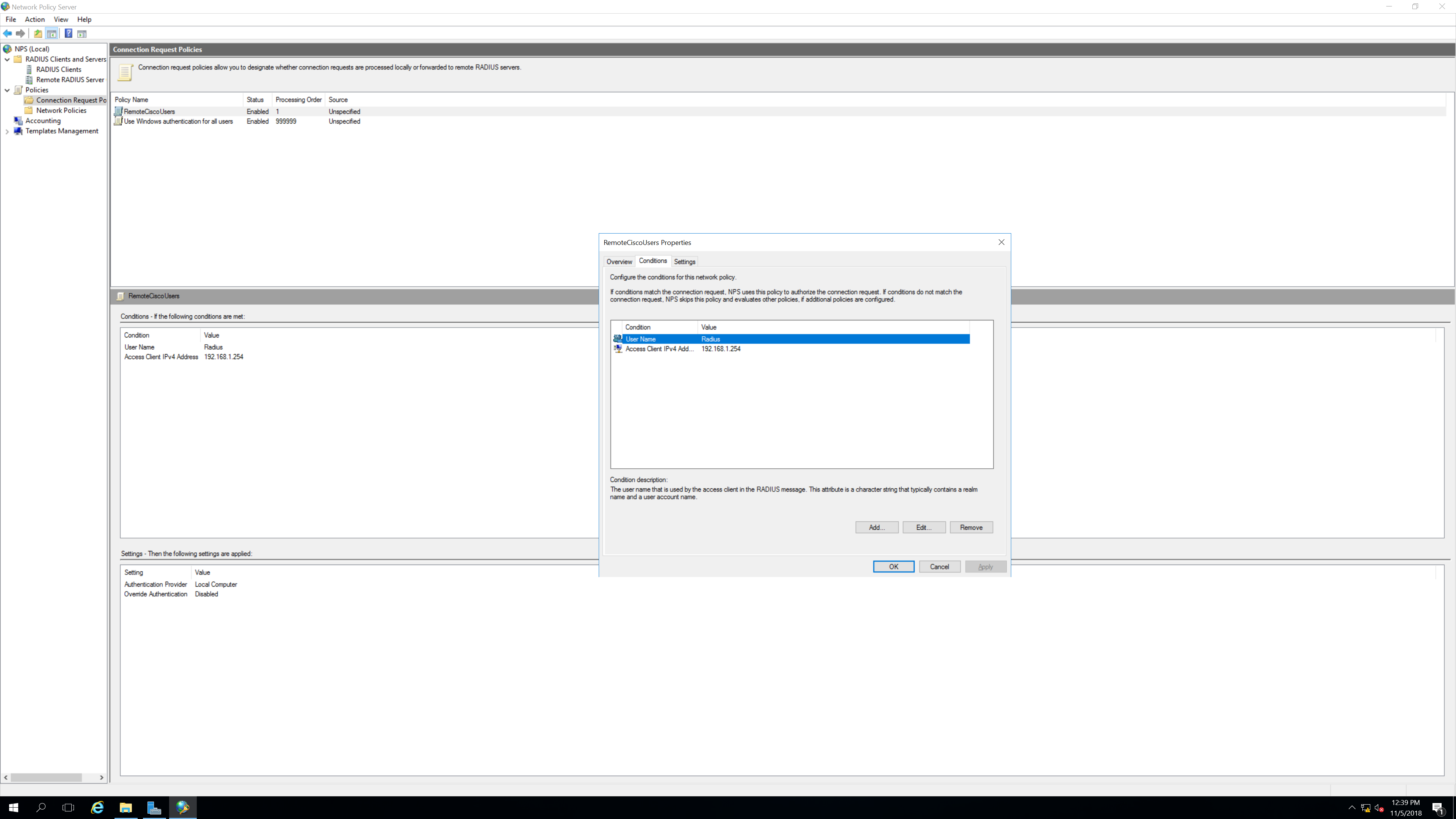
Step 4.



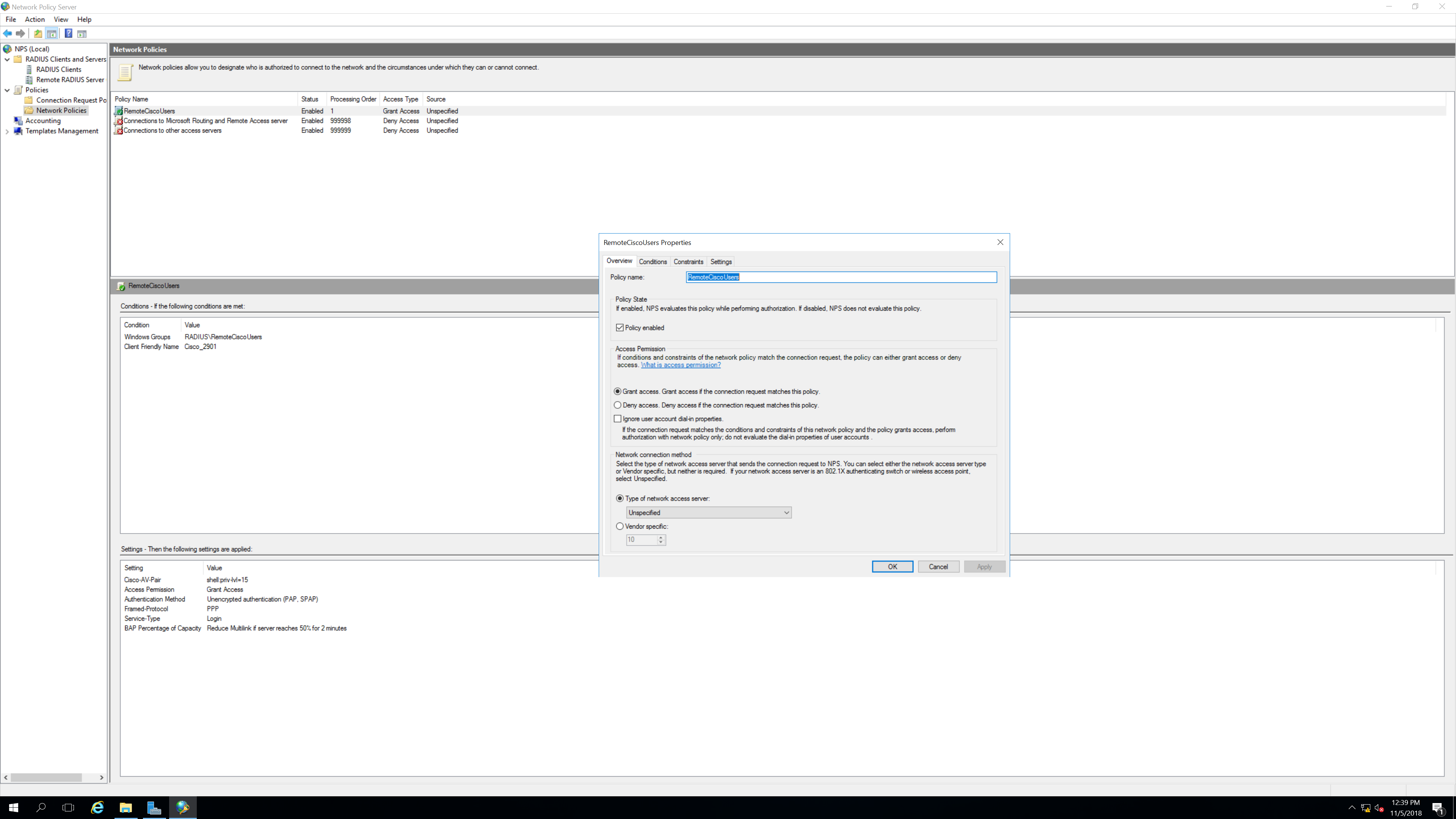
Step 5.



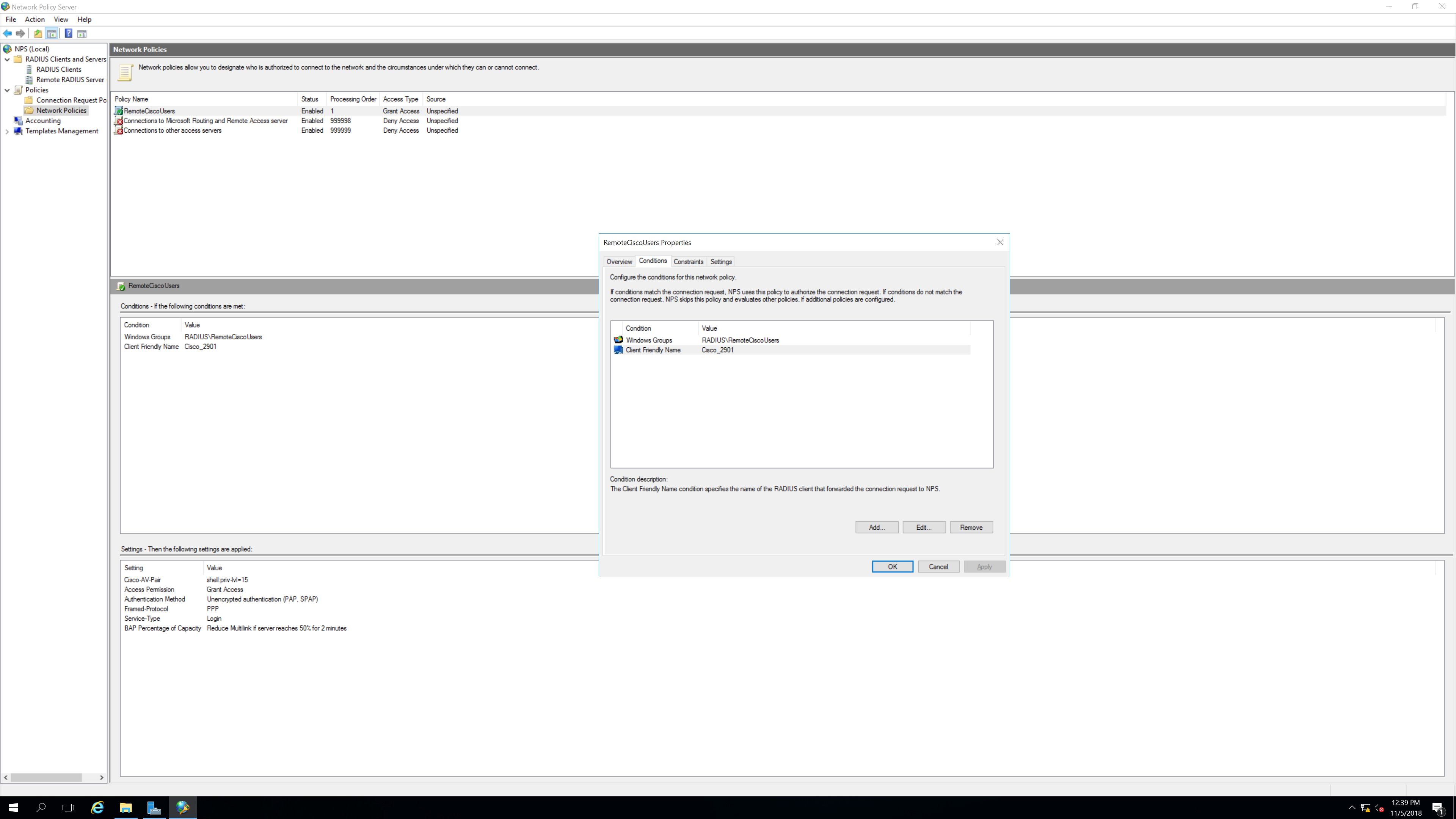
Step 6.



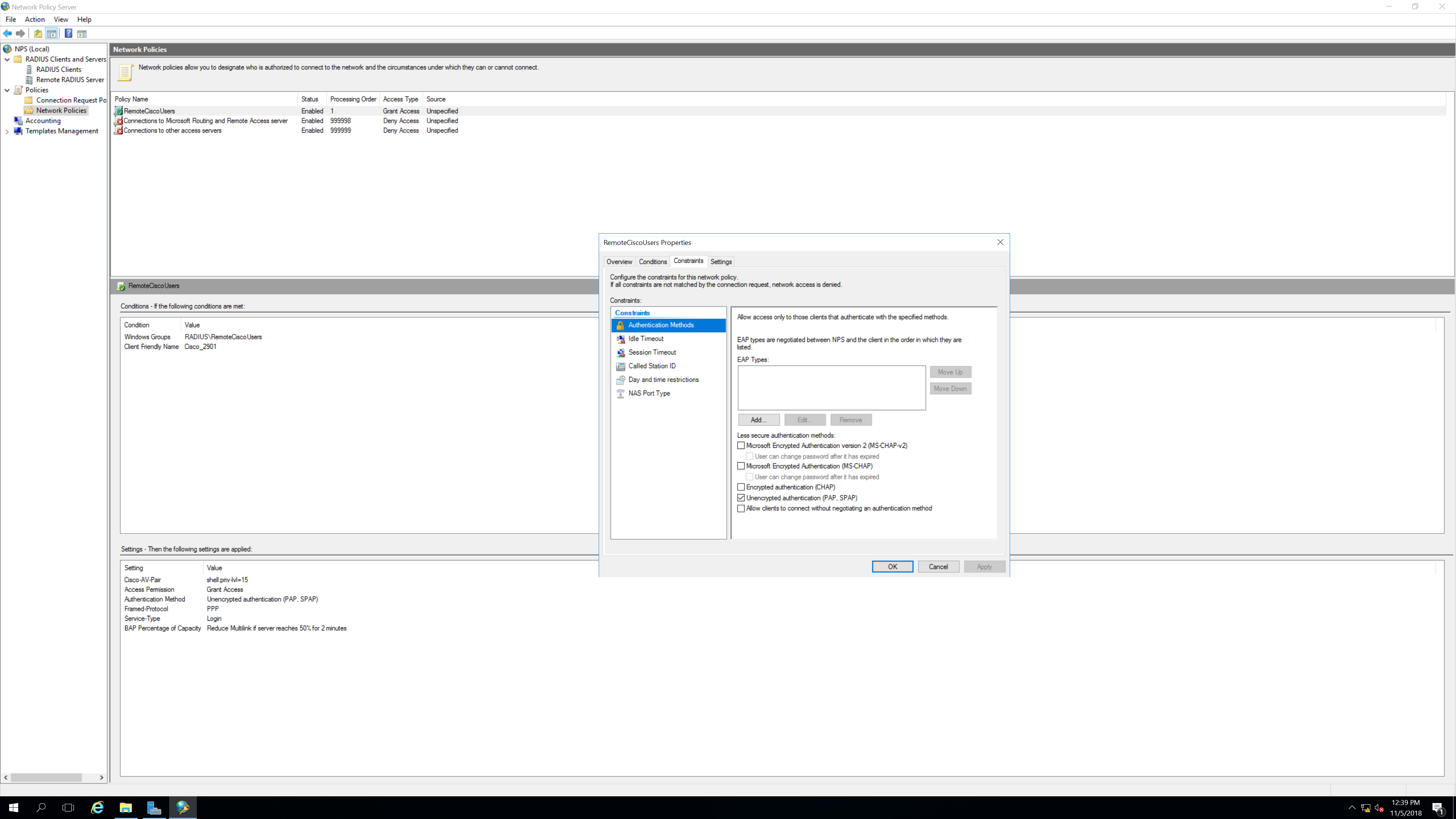
Step 7.



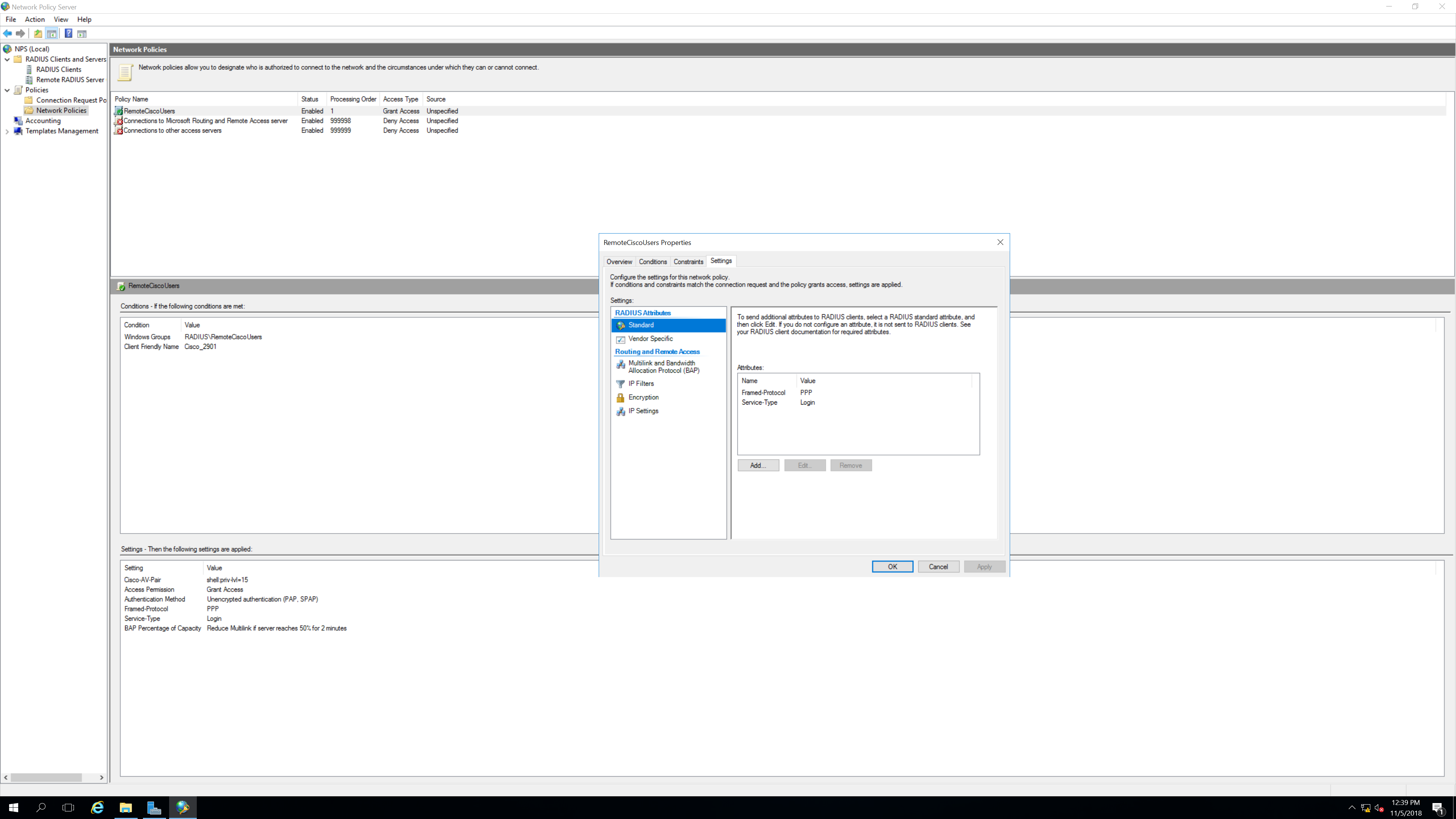
Step 8.



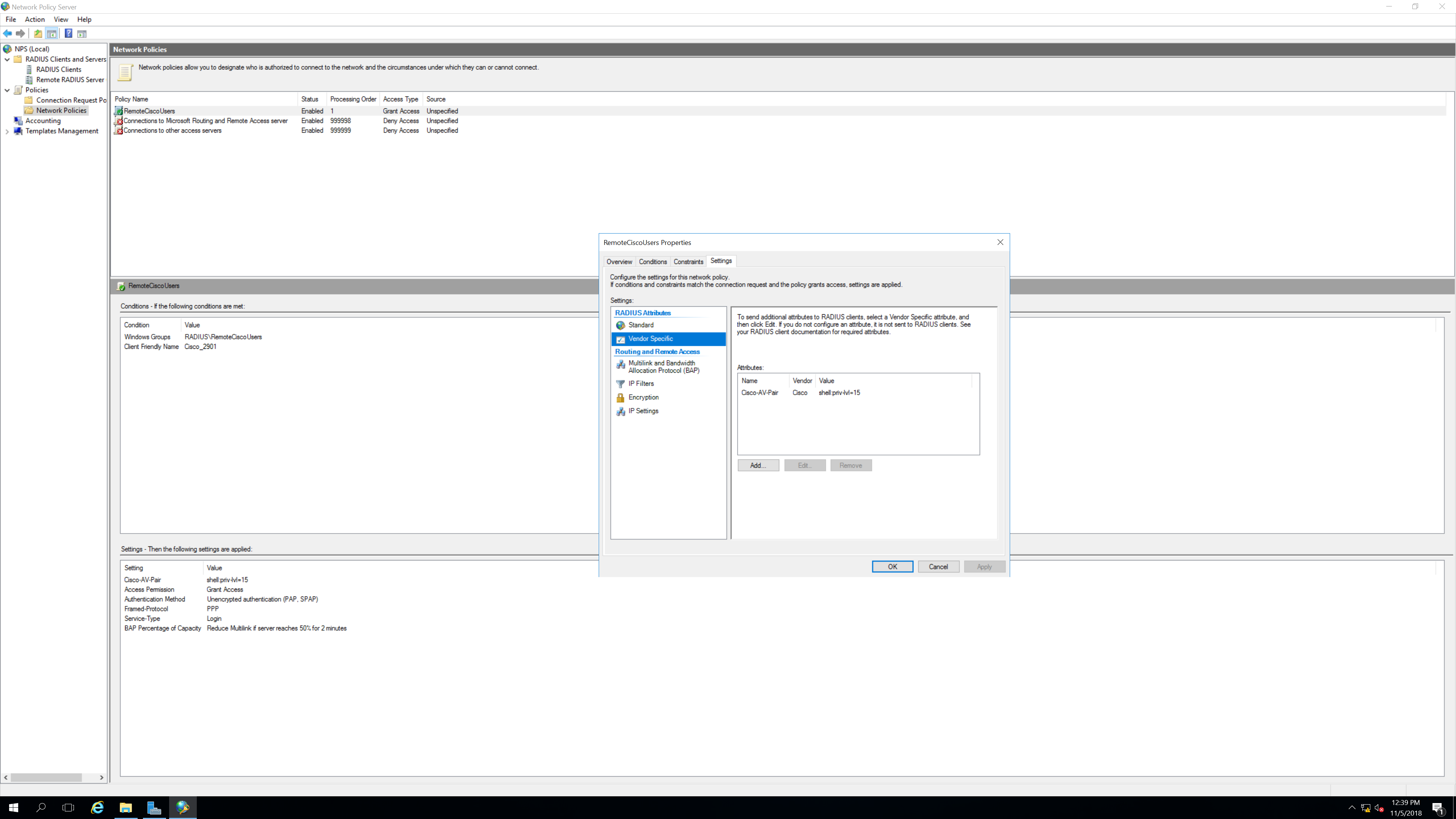
Step 9.



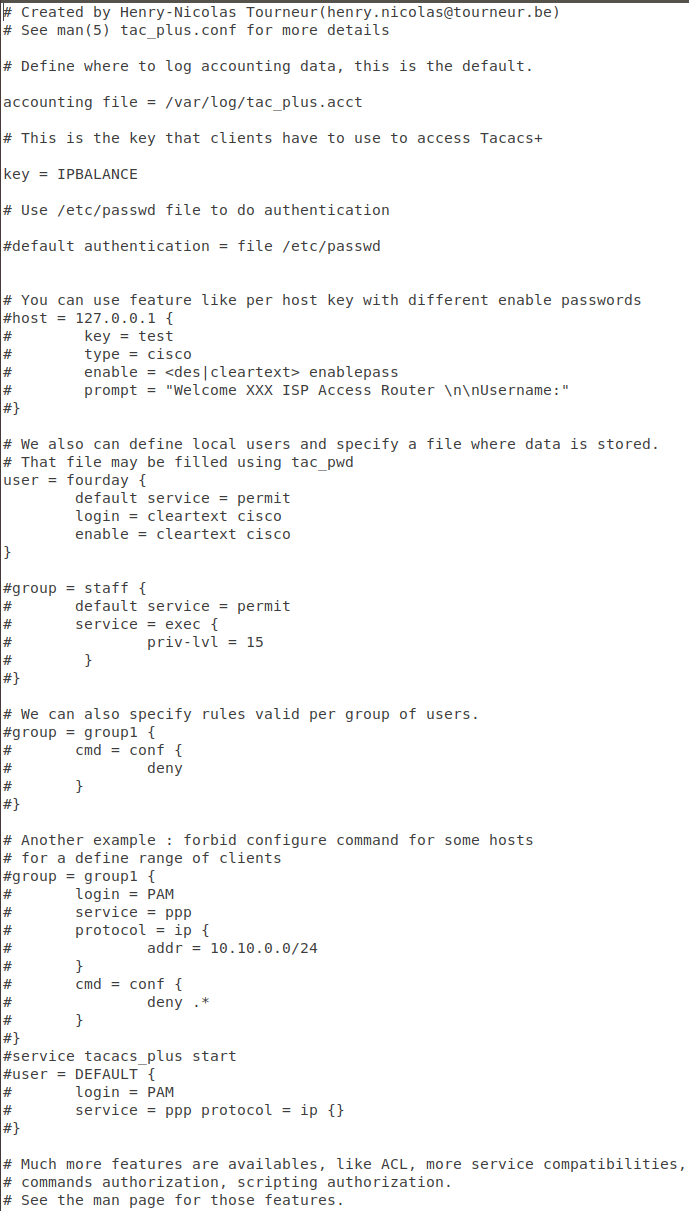
Step 10.

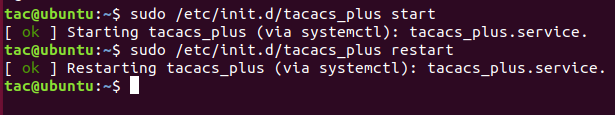


Step 11.



Linux Server Configuration





Problems: -

We faced most of our problems on the Windows Server 2016. One of the first steps into getting the lab working is setting up users and groups in the Active Directory, which serves as a database to carry the user credentials, on the server. However, after reaching the *Active Directory Users and Computers* folder, we could not locate the *user* folder, in which, one must set up users and groups. After a little research on the internet, we found out that prior to configuring the users and groups, one must configure a domain for the whole server. After we connected the server to a domain and went back to the AD folder, we were able to spot the user folder, and were able to set up users and groups that would be saved in the database and used in the authentication process.

Towards the end of the lab, we could not successfully login to the node, even though the user name and passwords were matching the credentials in the database. We assumed that we had wrong configuration on the router, and so we added numerous, unneeded commands, which complicated the lab. After a week and a half, we thought of starting the anew, and reloaded the router and re-installed the Server. To our surprise, everything worked fine afterwards, and the authentication was successful.

Conclusions: -

This was one of the most challenging labs that we have done so far, with us not being able to find a possible outcome on numerous occasions. It was a very new experience, and we had to reconfigure our nodes and servers, many times. So much, so that we have become familiar with the concept, and after the completion of the lab, I could confidently claim that I can conduct the whole lab again, without any help needed from anything, get both, RADIUS and TACACS+ working.