Azure and AWS 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 used on the cloud services: *Microsoft Azure* and *Amazon Web Services (AWS)*, to implement cloud networking. Another skill that this lab would grant us, would be to use our machine that we would work on, which in this lab was a PC running Windows 10, to route the data from the node that would be connected to it on the local network, out to the internet. During the process of the lab, we would get acquainted with the operation of the AAA servers on the internet.

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 web services that we would utilize. Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. In other words, one can effectuate certain applications and services on the Internet. In this lab, we used a service that Microsoft terms as *Infrastructure as a Service (IaaS)*, which allows users to launch general-purpose Microsoft Windows and Linux virtual machines, as well as preconfigured machine images for popular software packages. We deployed two Virtual Machines on our Microsoft Azure account. We ran a Windows Server 2016 machine that would act as a RADIUS Server, and a Ubuntu Lts 18.04 Server, which would function as a TACACS+ Server. For information on the two servers, kindly refer to the “RADIUS and TACACS+ Lab.”

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's version of virtual computers emulate most of the attributes of a real computer including hardware (CPU(s) & GPU(s) for processing, local/RAM memory, hard-disk/SSD storage); a choice of operating systems; networking; etc. The browser acts as a window into the virtual computer, letting subscribers log-in, configure and use their virtual systems just as they would a real physical computer. They can choose to deploy their AWS systems to provide internet-based services for themselves and/or their customers. We deployed two Virtual Machines on our AWS account. We ran a Windows Server 2016 machine that would act as a RADIUS Server, and a Ubuntu Lts 18.04 Server, which would function as a TACACS+ Server.

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. 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. TACACS+ is a Cisco proprietary, security application that provides centralized validation of users attempting to gain access to a node. Unlike RADIUS, TACACS+ uses TCP/UDP port 49. TACACS+ is more secure than RADIUS. For more information on the two protocols, kindly head back to the “RADIUS and TACACS+ Lab.”

Lab Summary

Unlike usual labs, 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 Azure and AWS.

The initial work was done on our local network, which consisted of a connection of a router to a PC, which was further connected to the Internet. We bridged the wi-fi and the ethernet connections so that the router on the local network gains access to the cyberspace. We then configured the router to request and obtain an IP address from a DHCP Server on the Comcast network. After having linked our local network to the Internet, we checked the public IP Address on different computers on the internet, as we had a doubt that the Comcast was using *Port Address Translation (PAT)*. It permits multiple devices on a local network to be mapped to single or a set of public IP address. After checking the public IP address on several machines, we confirmed that PAT was using just one public IP address.

Later, we created the Virtual Machines on Microsoft Azure and AWS and configured them according to their respective protocols.

After the configuration of the servers was finalized, we advanced to configure the routers for the proper AAA functioning. The process was concluded with the use of certain commands mentioned below. We used the public IP address for the configuration of the whole process as those would be the addresses that the machines and nodes would use to connect.

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. |

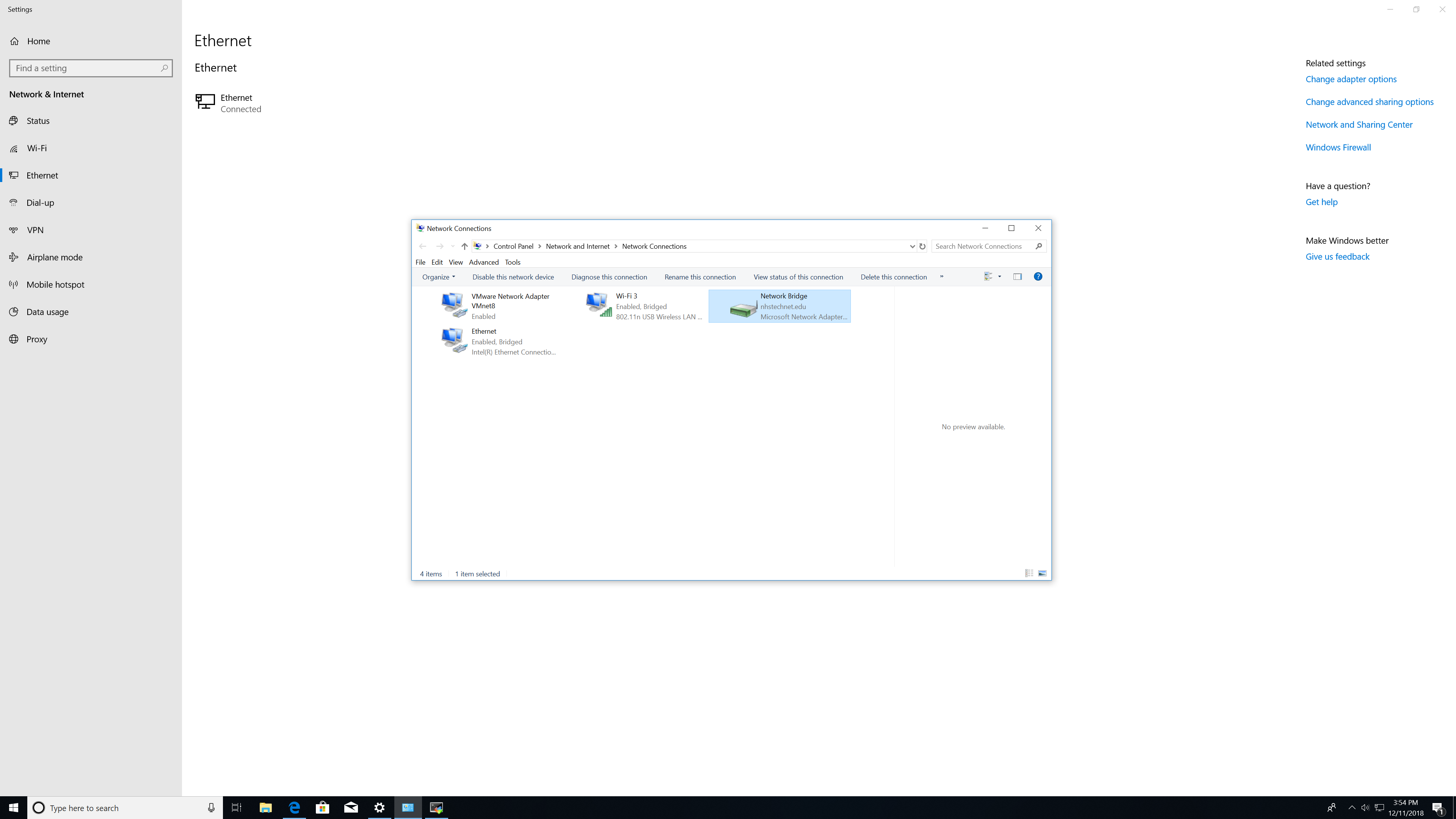
Topology and IP Addressing Scheme

A screenshot of a map

Description generated with very high confidence

Configurations

Step 1.



Connect a router to the PC in the local network, and link to the PC to the internet, in this scenario, we utilized a wireless connection to a home router which was further connected to the Internet. After confirming that the two connections (in this case the ethernet and the wi-fi connections) are recognized by the local machine, bridge both the connections in order for all the devices in the local network to attain an internet connection.

A screenshot of a computer

Description generated with very high confidence

Step 2.

Proceed to the Router in the local network and open the interface that the PC is connected to and type the following command : -

Ip address dhcp

Make certain to execute a no shut command on the interface and successfully obtain an IP Address from a DHCP server established in the ISP’s network.

Step 3.

A screenshot of a cell phone

Description generated with very high confidence

Advance to Microsoft Azure and after a successful login attempt to your Azure account, create one of the two highlighted virtual servers on the cloud with your desired preferences.

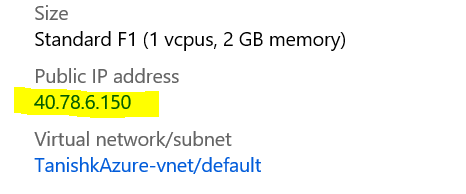
Step 4.

A screenshot of a cell phone

Description generated with very high confidence

Prior to heading to the next step, you should check the public IP address from a trusted source on the internet. One should also ensure that the ISP is using PAT with just a single public IP Address, by checking the public IP address on other devices connected to the same wi-fi.

Step 5.



Before starting the lab on the Windows Server 2016, note the public IP address of the server shown on Azure and utilize the in-built application called *Remote Desktop Management* on the Windows PC in the local machine to remotely manage the Windows Server.

A screenshot of a cell phone

Description generated with very high confidence

Step 6.

Perform the similar processes from the “RADIUS and TACACS+ Lab” to complete the RADIUS configurations on the server except for the following :-

A screenshot of a cell phone

Description generated with very high confidence

Under the “Clients” section, instead of the private IP address of the router on the local network, which will act as the RADIUS client in this lab, fill the public IP address that was learnt by checking on the internet, and :-

A screenshot of a cell phone

Description generated with very high confidence

In the Network Policies, under the NAS Port Type Section, which happens to be under the constraints section, check the boxes which show: “Async (Modem)”

Step 7.

At this point, the RADIUS configuration on the server is complete. Progress to the router in the LAN and type the following commands :-

aaa new-model

aaa authorization login default group radius local

aaa authorization exec default group radius if-authenticated

radius-server host 40.78.6.150 key AzureRadius

If all the procedures are finely followed, the user should have a successful login attempt.

Step 8.

For completing the TACACS+ process, create the Ubuntu server, and instead of using RDP to remotely manage the server, head to the Serial Console section in the cloud serve that is being used and type in the following commands :-

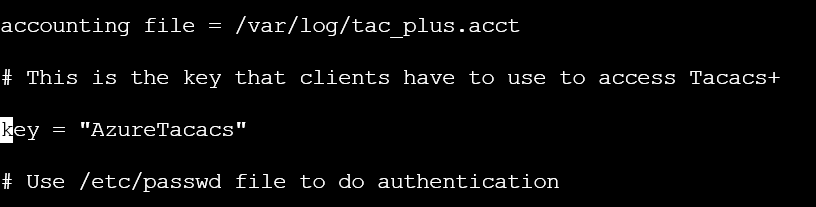
sudo apt-get install tacacs+

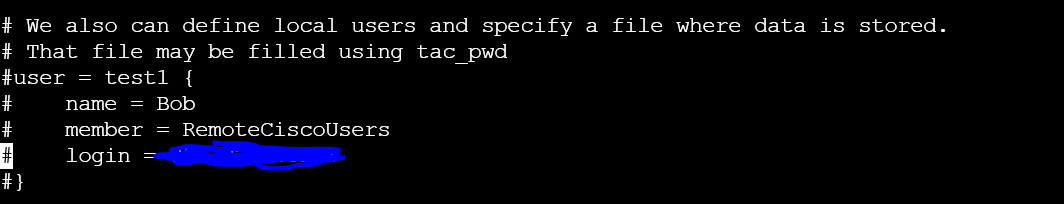
service tacacs\_plus start

nano /etc/tacacs+/tac\_plus.conf

Step 9.

After executing the third command from the given set of commands above creat a user, group and a TACACS+ key that can be used by the router in the local network.





A close up of a black background

Description generated with high confidence

Ensure that everything except for the name of the group remains similar, and username must be a part of the new group created.

Step 10.

We are left with a minor step before moving to the router for the completion of the lab. There are two commands that are essential for the completion that must be deployed on the server :-

chmod o-r /etc/tacacs+/tac\_plus.conf

service tacacs\_plus reload

Step 11.

Finally, proceed to the router. If you are using the same router as used in the configuration of RADIUS, make certain to reload te router and erase the startup-config file if anything was saved there. Type the set of commands to ensure the proper operation of the TACACS+ on the LAN :-

Aaa new-model

aaa authentication login default group tacacs+ local none

aaa authorization exec default group tacacs+ local none

aaa authorization commands 0 default group tacacs+ local none

tacacs-server host 40.78.42.250

tacacs-server key AzureTacacs

Step 12.

Repeat the same steps on AWS for concluding the lab.

Problems: -

We faced most of our problems on the Windows Server 2016 on Azure. After finishing the steps on the Windows Server, we could not successfully login on the router. We later inspected the NPAS Events log on the Server and could not locate any detail that showed evidence of the connection made. We kept attempting until after a few days the login attempt was successful. We researched about the topic and came to conclusion that there was some problem with the internet connection, as it was probably slower to deliver the request packets to the server.

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 anyone.