AAA

*RADIUS and TACACS+*

CCNP Lab 9

Jason Liu

CCNP – Mr. Mason & Mr. Hansen

Period 6, 7, 8

*Lab 9: AAA - RADIUS and TACACS+*

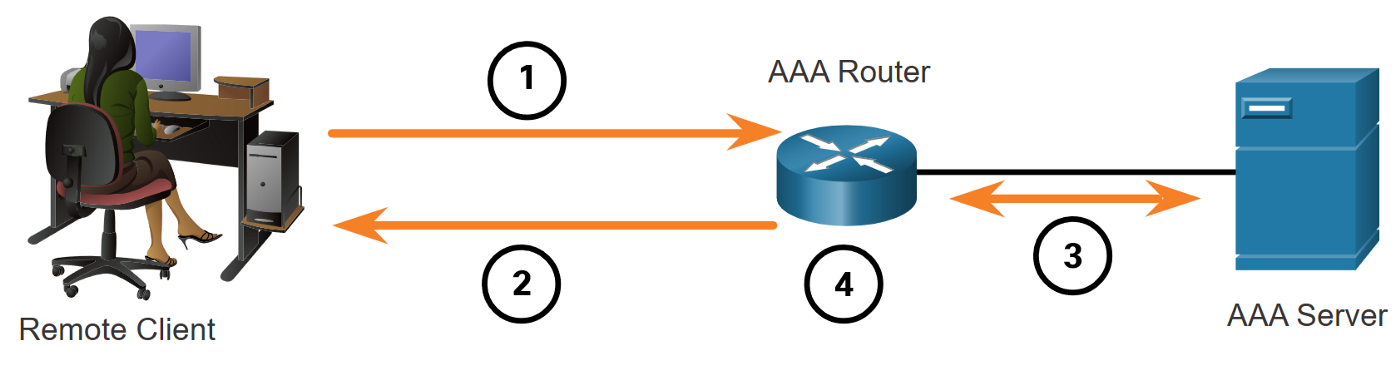
**Purpose**

The objective of the lab was to understand and configure the Authentication, Authorization, and Accounting (AAA) framework and set an authentication protocol in two forms: The open-source RADIUS protocol and the Cisco-proprietary TACACS+. The goal was to establish the remote authentication protocol to validate registered clients for login. In other words, requesting and accepting created user credentials.

**Background Information**

AAA, short for Authentication, Authorization, and Accounting, is a security framework to facilitate user login. It ensures only authorized users may access the device, especially when remote access is the main method of configuration. Each user can also then be assigned a level of privilege, determining what they are authorized to do. Lastly, user activity can be monitored, captured, and saved to a database server. Accounting is negligible for the purpose of this lab. Different authentication protocols serve this AAA framework, and the two most widely used are RADIUS and TACACS.

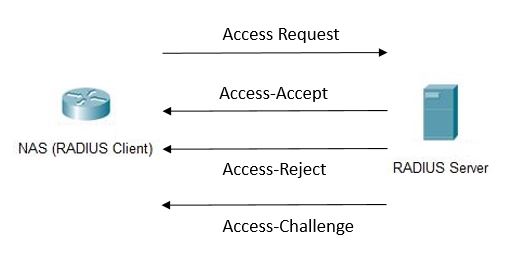
A router can be the AAA client, but a Network Access Server (NAS) is the general term for devices that serve the same role. It is the element that users first contact to get access to the network. As it is the single point of access to the remote resource, it also serves as the gateway to guard it. The NAS should be located within the accessible network. Then, the NAS is connected to the centralized AAA server. The protocol for communication between this NAS and the central server is called the authentication protocol and is what RADIUS and TACACS are used for. Below, the process of AAA communication is shown.



1. The remote client tries to access the router. Information is sent via the link-layer, such as through the Point-to-Point Protocol (PPP).
2. The AAA router/NAS prompts user credentials, such as Username and Password.
3. The AAA router/NAS checks with the authentication server through a protocol to determine validity and authorized privileges.
4. The AAA router/NAS replies to the client again with either a accept or a deny message.

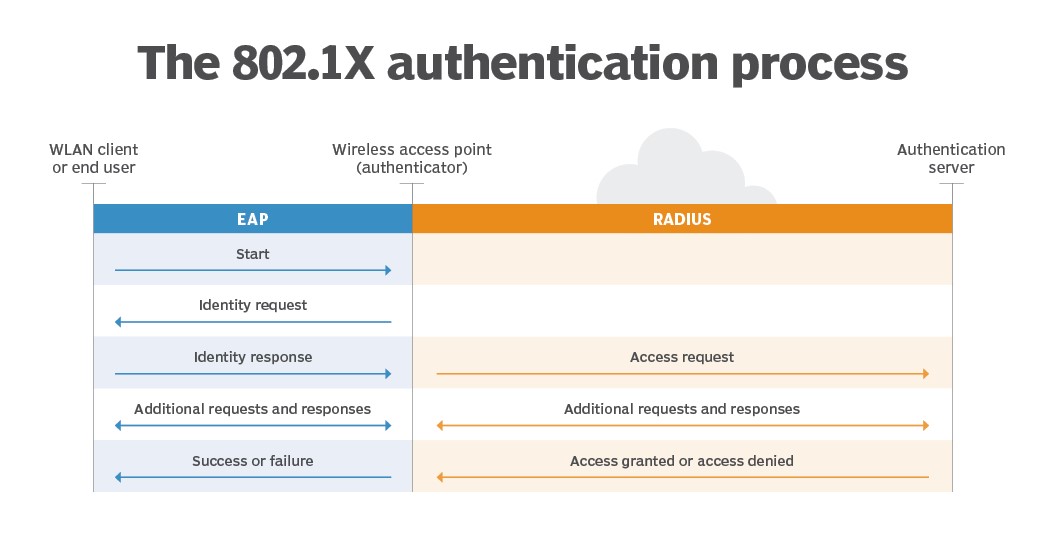
To specify the method to be used for authentication, there is a command during AAA configuration for the AAA router side. That command is a default or customized authentication method list (see *Lab Commands*). Without the command, the router will not have any visible methods of authentication. On the contrary, this also implies that multiple authentication protocols, including RADIUS and TACACS, can be configured at the same time and used in the same method list. This improves on fault tolerance as if either server is down, it is ensured that another method will be available. However, if the authentication server replies with a reject message, it will not go to the next method.

The open-source Remote Authentication Dial-In User Service (RADIUS) is the most popular and widely used authentication protocol standard for traditional LANs and Internet Service Providers. Using UDP ports 1645 and 1812, it runs at the application layer and can use either TCP or UDP for the transport. Specifics in RADIUS can be referred back to the main packet flow topology.



Upon receiving the user request with their access credentials, the NAS sends a “RADIUS Access Request” message with that information to the RADIUS server. Using an authentication scheme such as Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), or any Extensible Authentication Protocol (EAP), the RADIUS server checks if the information is correct. Used over PPP communication, these protocols have lots of depth which will not be covered in this document. After determining validity, the server responds with one of three Access responses to the NAS: Accept, Reject, or Challenge. A Challenge message is a request for further information, such as another password or PIN. It is important to know that all information is being sent in cleartext by default. For the accounting aspect of RADIUS, an “Accounting Start” message sent by the NAS begins tracking the user’s network access. When the user session is over, the NAS issues a final accounting record to the server. Also, note that “Diameter” is a planned replacement for RADIUS, so topologies may reference it.

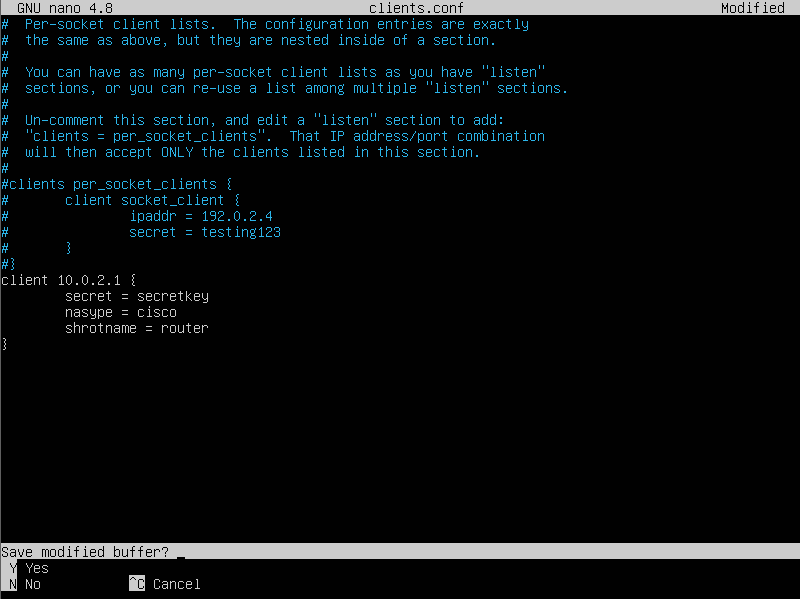
Terminal Access Controller Access-Control System (TACACS) is the Cisco proprietary security protocol with services that parallel those of RADIUS. TACACS+ is the newer version of the original TACACS protocol, and also no long compatible with those older versions. Using TCP and port 49, a TACACS+ host, or server, processes AAA requests from the NAS and replies accordingly. Providing the same features as RADIUS, functionality and background are incredibly similar.

 As these protocols are mainly used for remote access purposes, it is very common to see the sphere of Wireless communication and authentication in the same conversation as these protocols. For the wireless 802.1X authentication process, RADIUS is often used as the authentication protocol. The processes are the same and it’s simply important to recognize it’s applications and use.

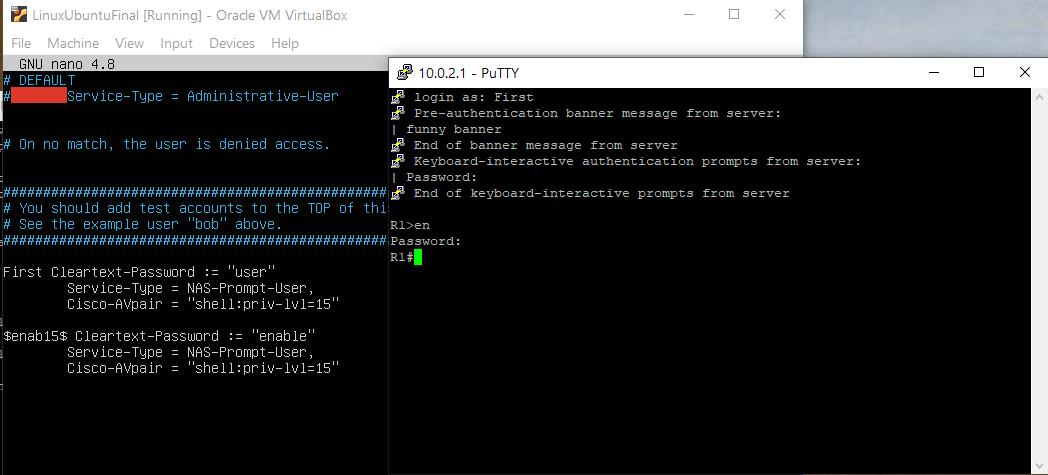
**Lab Summary**

As this was not a routing scenario, the topology is quite unique. A router, acting as the authentication client and NAS, is connected to a PC. The PC hosts the RADIUS and TACACS+ servers using Virtual Machine software to represent separate devices, although physically connected through a single link. The switch has no important configuration and simply reflects the physical wiring of the lab (see *Network Diagram with IPs*).

These authentication protocols are considered operational when user credentials can be made and changed on the authentication server, and changes can be visible from the user device. For the RADIUS server, a Linux version called FreeRadius was used. All FreeRadius configuration was done on the Ubuntu distribution of Linux in the 20.04 version, which is a Command Line Interface (CLI). Upon installing FreeRadius 3.0 in Ubuntu, users and clients can be configured in the “Users” and “Clients.conf” files, respectively, using the default NANO editor. Some important configurations include the username and password, but also the shared secret, also known as the key. This key can be any simple text but the one configured on the client file of the Authentication Server must be identical to the key of the NAS. Any faults in this key identification would reject all service and communication between the server and client. Then, IP addresses should be configured to be in the subnet and the VM should have the network adapter listed as “Bridged Adapter.” Lastly, restart the service to update all changes.

 After navigating server-side configurations, the client, being the router, requires changes as well. AAA commands include the method list and banner messages, as well as describing radius server info, such as its IP, key, and retransmit count. The method-list should contain backup methods in the case that the server is inactive. The enable method uses the Enable Secret password and was put as the last method in the list for backup purposes.

Using the default NANO text editor within Linux, clients.conf file was entered and the client/NAS was added.

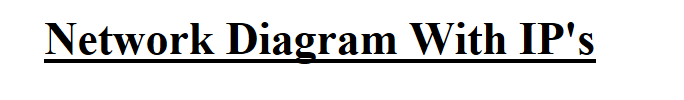
 Upon successful client and server configuration, RADIUS should be fully functional. Attempting to access the router via either console or SSH (remote access), it prompts a login username and password. Any correct username and password pair is visible in the users file of the Linux server. The password to reach privileged EXEC mode is also configured there, under the username “$enab15$.” New users were added afterwards to prove that verification was from the server rather than a local configuration.

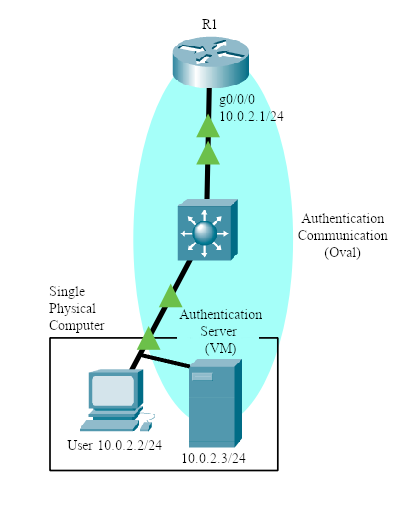
Login prompt upon SSH remote access should appear like this. Entering the corresponding username and password should allow login.

For TACACS+, the server ran on the computer itself rather than a VM, making the user device the TACACS host. The startup exe file was installed from the official TACACS website. Of the files downloaded after the wizard installation, “authentication,” “clients,” and “authorization” were edited for the goal of this lab. These files can be directly edited through a text editor, like Notepad++. User templates have already been made and by removing the comment tags, the text is rendered as readable code. There is also a pre-existing template for the clients. The corresponding information was filled in for all the files. This included IP addresses, usernames, passwords, and enable passwords.

From the installation wizard, a couple other useful features were installed. TacVerify looks for errors in the configuration, and admin command prompt can run TacTests that attempt to login with information without ever contacting the NAS. Restarting the service is also done on the command prompt.

Changes of configuration afterwards are documented as evidence of functionality. After all elements of configuration, the protocols are operational and AAA is in use.



****

**Lab Commands**

Most commands are new, including both router-side CLI and Linux commands.

Router side:

**AAA new model –** Enables AAA.

**AAA authentication login [default | *list name*] [*method list*…] –** Creates a login authentication method list. Under the unnamed, default list, each specified method is checked.

**AAA authentication enable [default | *list name*] [*method list*…] –** Creates an enable authentication list. Under a list, each specified method is checked. Default is used.

Additional methods are used only if previous method is absent or returns an error, not if it fails. This provides flexibility to the lab as multiple methods can used, such as both group RADIUS and group tacacs+. Methods include:

* **Group RADIUS –** Checks with RADIUS server**.**
* **Group TACACS+ -** Checks with TACACS+ server**.**
* **Enable –** Uses enable secret password**.**
* **Local –** Uses local username database**.**
* **None –** No authentication for login**.**

**AAA authentication {banner | fail-message} “*message*” –** Create a banner and/or a fail-message for the router login interface.

**RADIUS/TACACS server [*RADIUS server name*] –** Creates RADIUS/TACACS server name and enters RADIUS/TACACS server configuration mode.

**Key [*shared secret key*] –** Assigns the shared secret key to the RADIUS server. Keys need to be identical on both the server and the router for RADIUS authentication to function.

**Address ipv4 [*IP address*] auth-port [*port number*] acct-port [*port number*]–** Assigns the ipv4 address to the RADIUS server, and also assigns authentication port numbers. RADIUS should use port 1812 and 1813 for “auth” and “acct,” respectively, for best practice.

**Retransmit [*number of retransmits*] –** Indicates the number of retry attempts before it determines failure.

**Timeout [*number of seconds*] –** Indicates the number of seconds of requesting before it determines to retry transmission.

**radtest [username] [password] [servername] [port] [secret]–** Attempts RADIUS login test and shows results of whether the login credentials would work.

Linux side:

**LS –** Lists folders and files in current directory.

**CD [*Folder name*] –** Enter new directory of new folder.

**Sudo su –** Enters super admin mode. “Sudo…” syntax for admin permissions is not necessary on future commands.

**Sudo apt-get install FreeRadius {FreeRadius-utils} –** Installs FreeRadius. {As well as other FreeRadius utilities, optional.}

**Sudo apt policy FreeRadius –** Checks FreeRadius version. v3.0 was used for this lab.

**Service FreeRadius restart –** Restarts the FreeRadius service. Updates changes.

**FreeRadius -CX –** Checks FreeRadius operation and seeks errors in file configuration.

**Nano [*File name*]–** Using the default file editor called “nano,” opens and edits file contents.

**Client [*Client IP*] {**

**secret = [*Secret name*]**

**nastype = [*NAS type*]**

**shortname = [*Device-type*]**

**}**

Under “nano clients.conf,” the following creates the FreeRadius client. This set of commands determines its IP, shared secret key, NAS type, and a local nickname for the device. The key has to be IDENTICAL to the router-side configuration. NAS type of “cisco” and a shortname of “router” was used.

**[*username*] Cleartext-Password := “[*password*]”**

**Service-Type = NAS-prompt-user,**

**Cisco-AVpair = shell:priv-lvl=[*privilege number 1-15*]**

Under “nano users,” the following creates the FreeRadius user credentials. This set of commands can be repeated and is used to create a user on this server. These usernames and passwords are the ones the router will check with upon each login request.

**$enab15$ Cleartext-Password := “[*enable password*]”**

**Service-Type = NAS-prompt-user,**

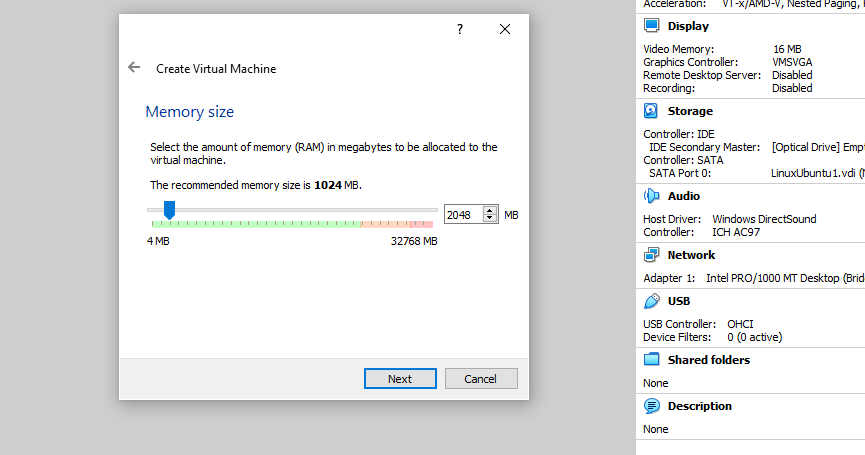
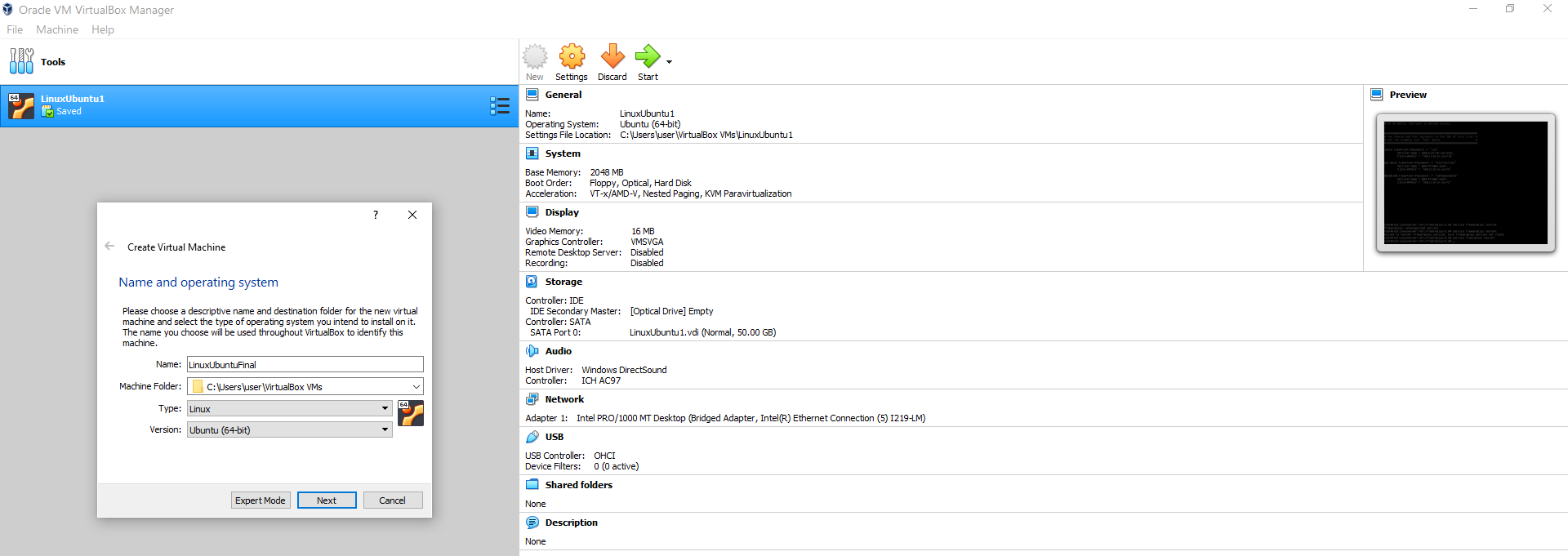
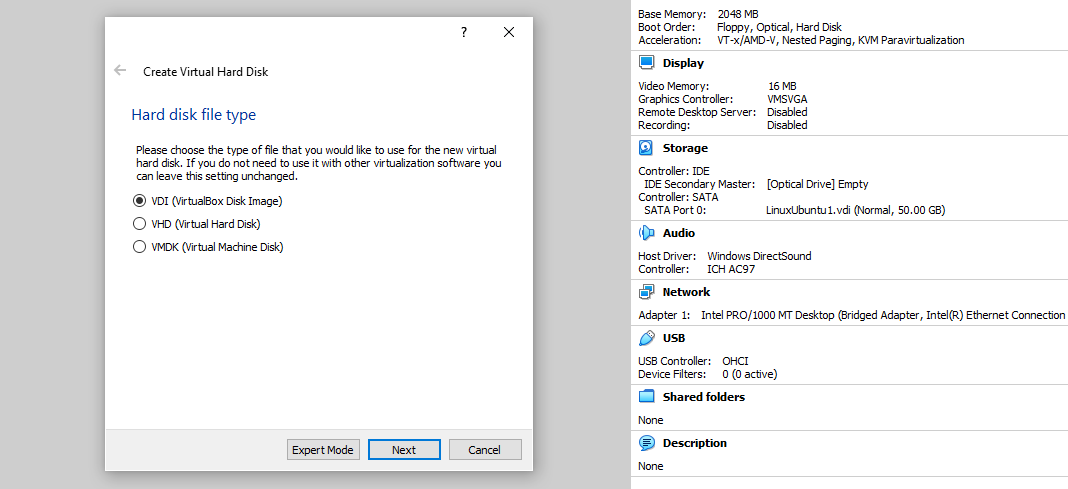
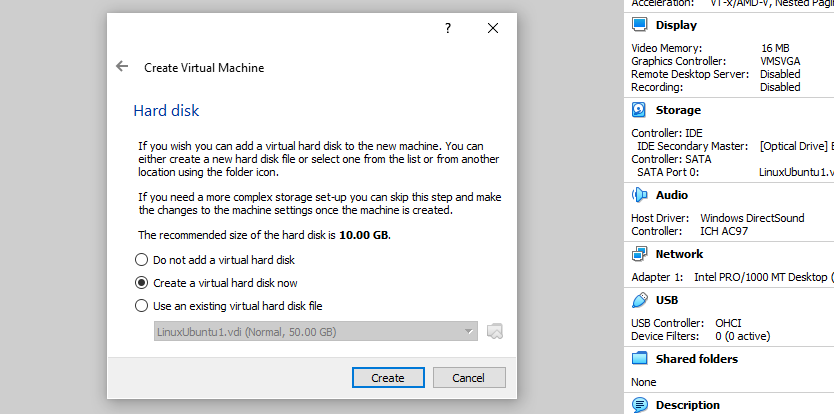
**Cisco-AVpair = shell:priv-lvl=[*privilege number 1-15*]**

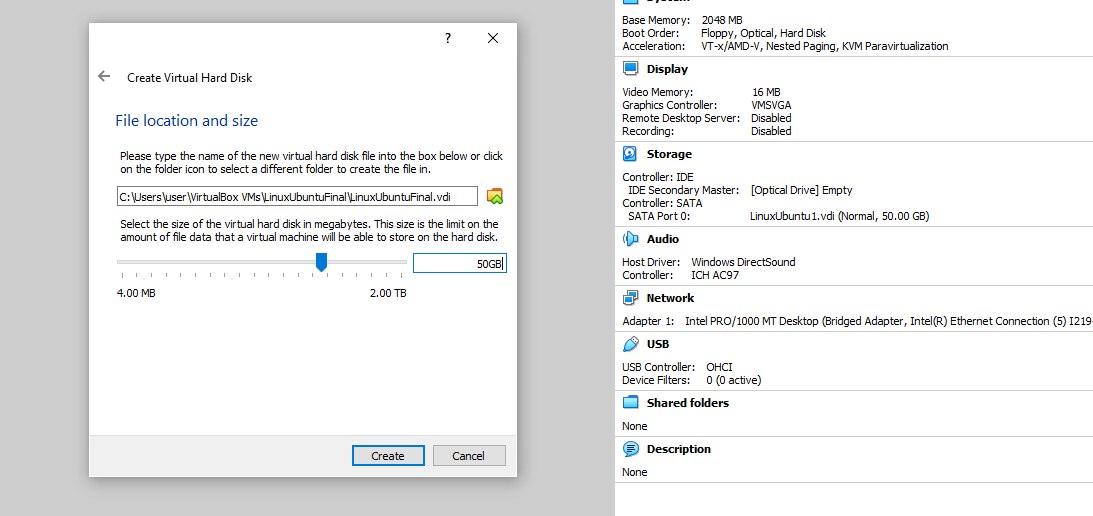
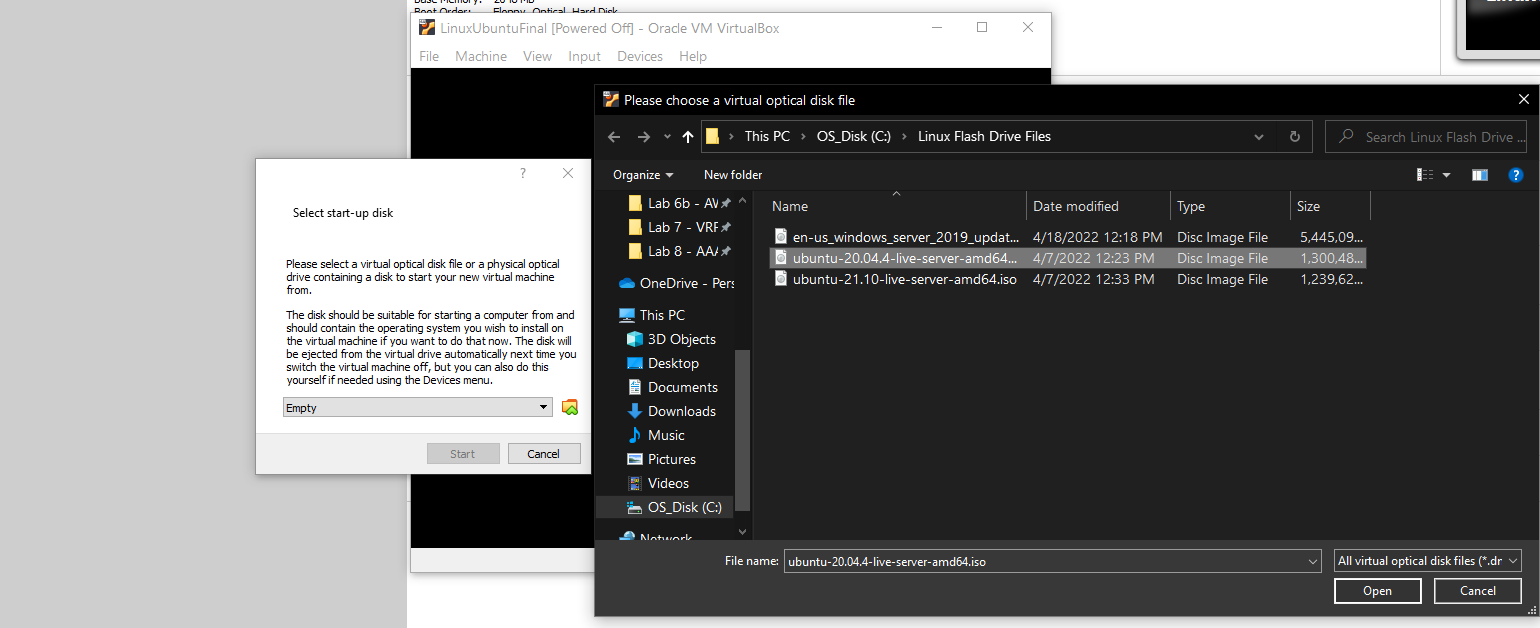
Creates password upon users’ requests for entering privileged mode.

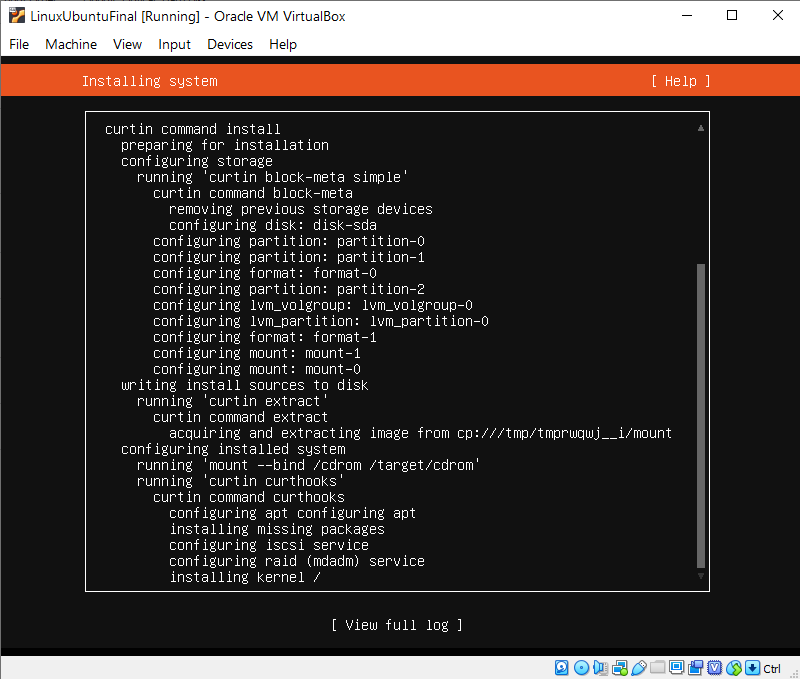
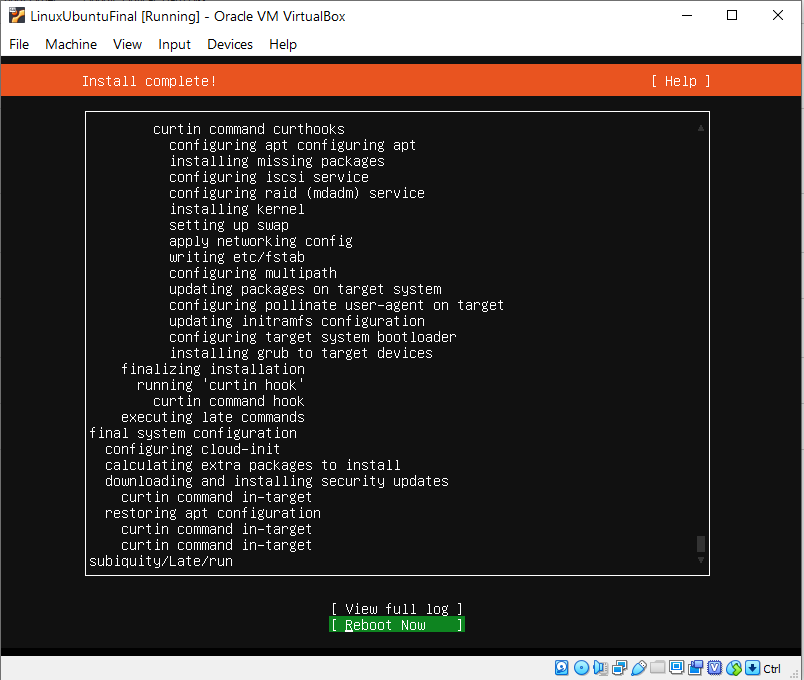
**Procedure**

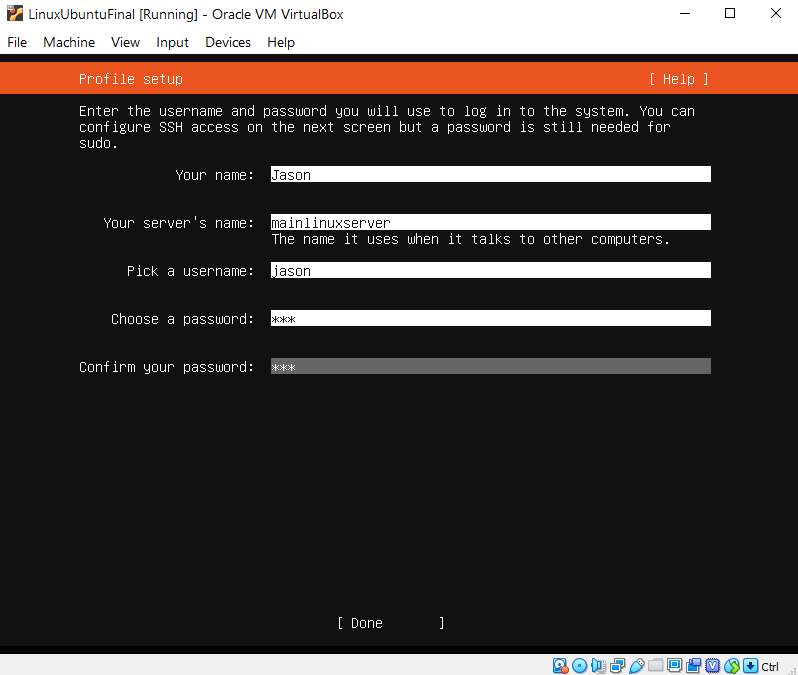
Here is the step-by-step installation and configuration for both RADIUS and TACACS.

**RADIUS**:

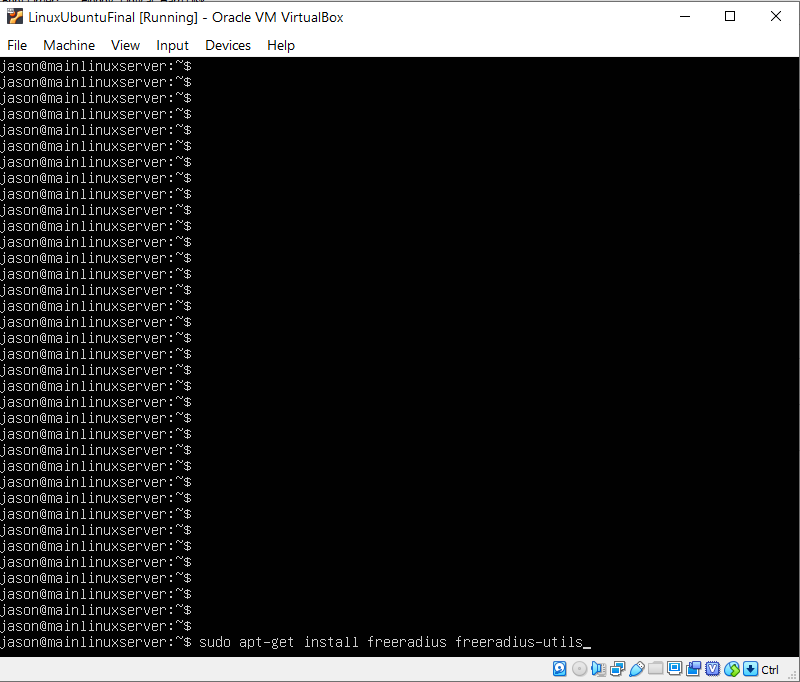
1. Create the VM. Oracle VirtualBox version 6.0 was used. Continue through the settings, such as setting the memory size (2048 MB), creating a virtual hard dis(50GB), and choosing the start-up disc file.



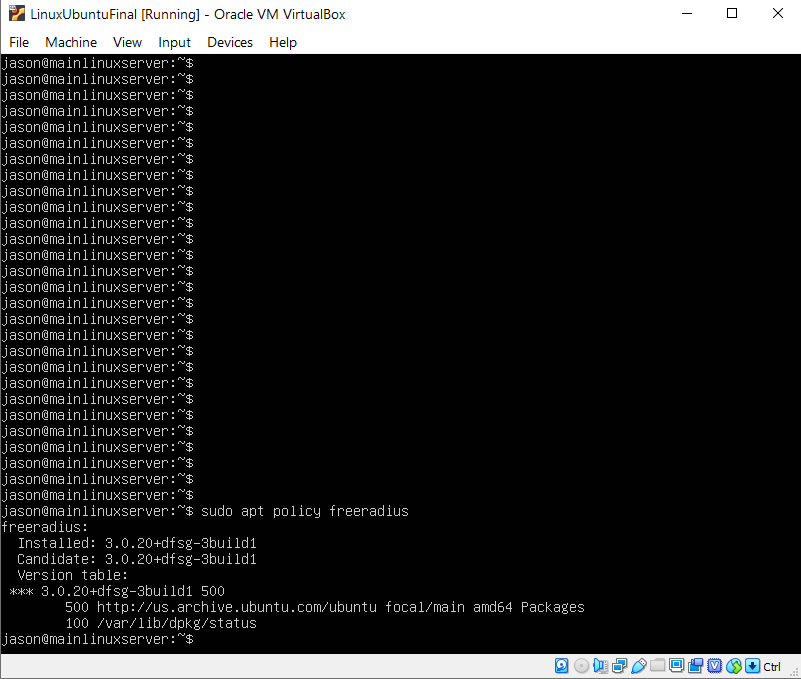
1. Entering the VM, create the Linux login info and continue default configurations until you reach the main interface. It will ask for a reboot.

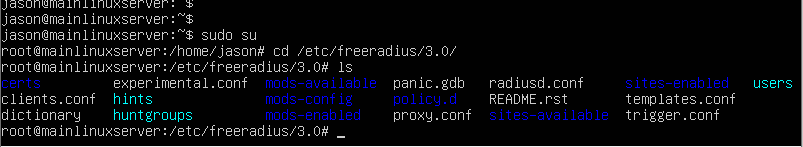


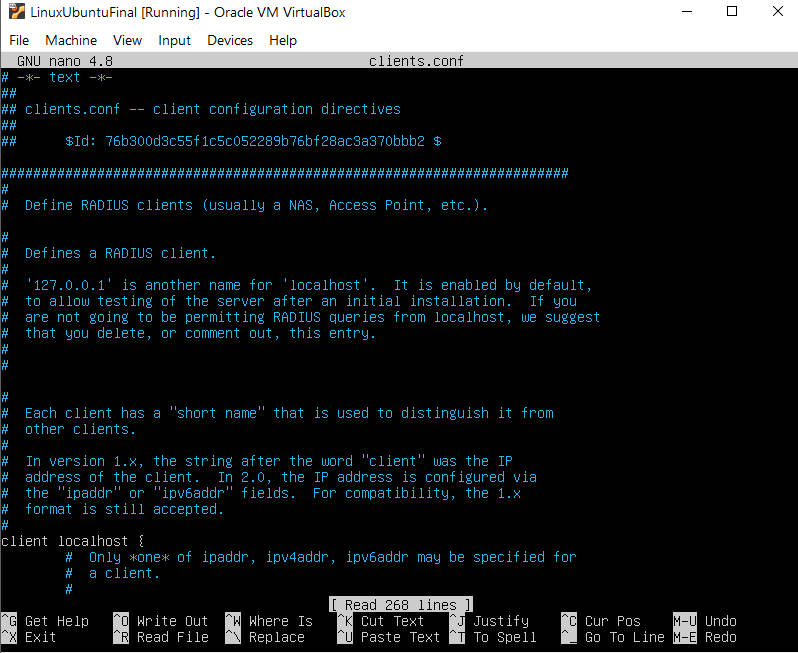
Create Linux Profile Credentials.

1. Install FreeRadius and check its version. 3.0 was used for this lab.

Installing Freeradius and checking policy version.

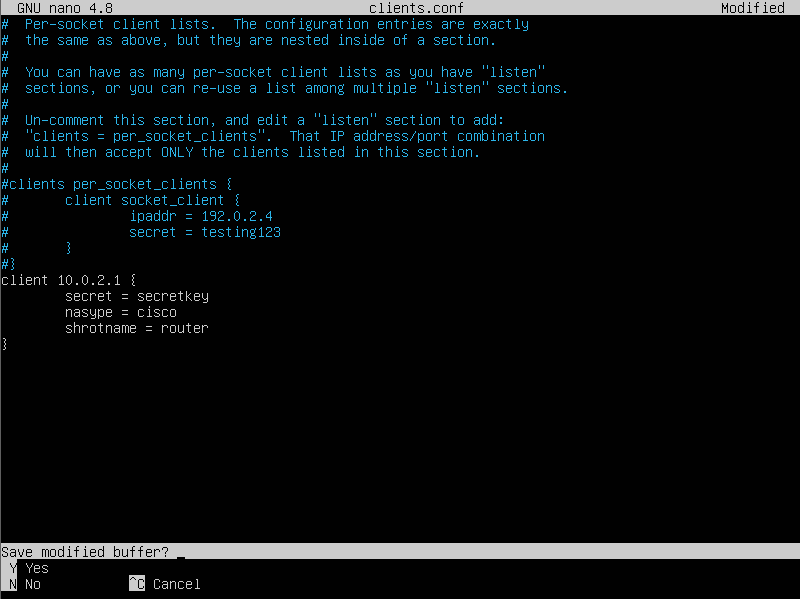


1. Enter the /etc/freeradius/3.0/ folder and check the folders within. Look for Clients.conf and Users. Then use the nano editor to enter these files.

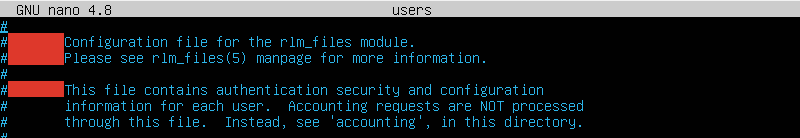
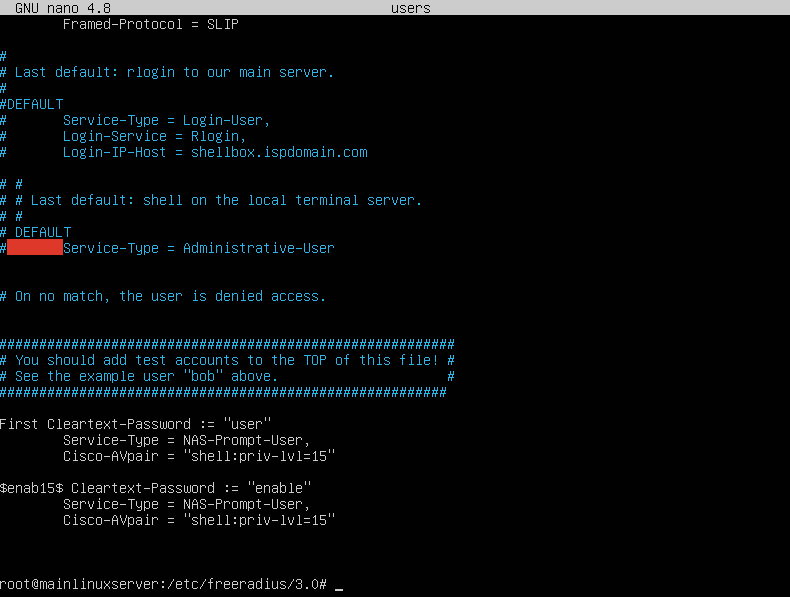


List of files in the /etc/freeradius/3.0/ folder.

Enter editor with “NANO clients.conf.”

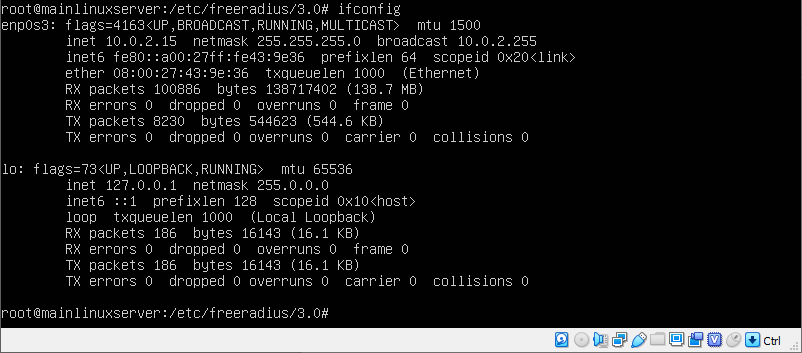
1. Create the client and enter related information within the Clients.conf file. Create the users and its related information in the Users file. See *Lab Commands* for more details.

Create clients on clients.conf file.



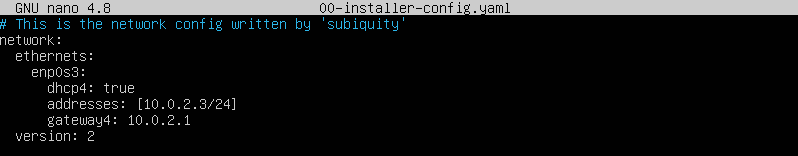
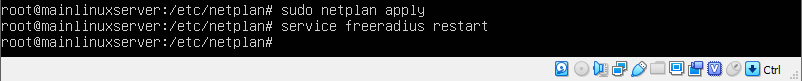
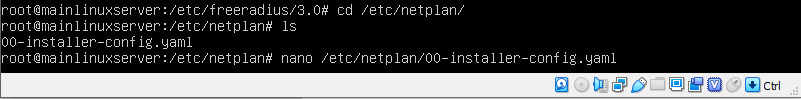
Create user profiles.

Enter editor with “NANO users.”

1. Use “ifconfig” to see ethernet address, and change the static address as needed to fit the topology by editing the netplan file. Use “gateway 4: [*IP*]” and “Addresses: [*IP*]” to add corresponding addresses. Apply configurations with “Sudo netplan apply.”

Check current IP with ifconfig.

It is 10.0.2.15/24 by default.



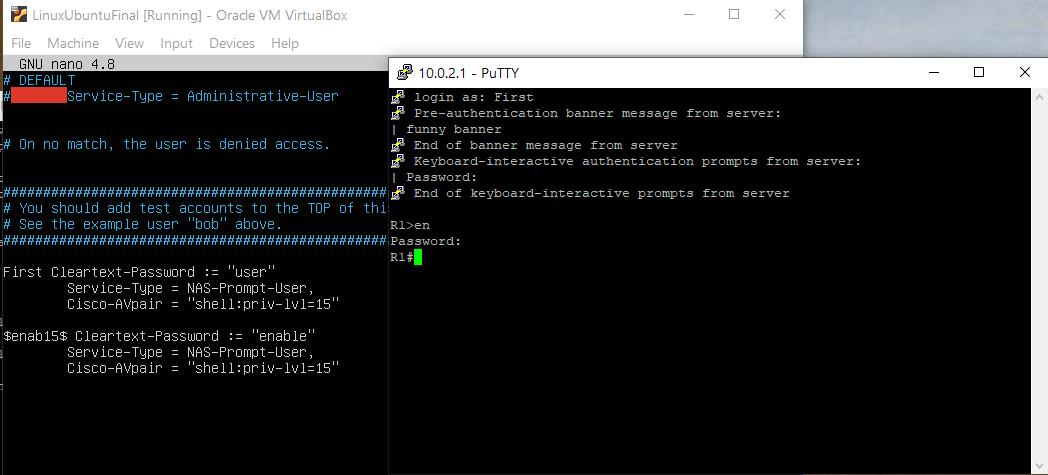
Use “sudo netplan apply” to save the static IP configuration.

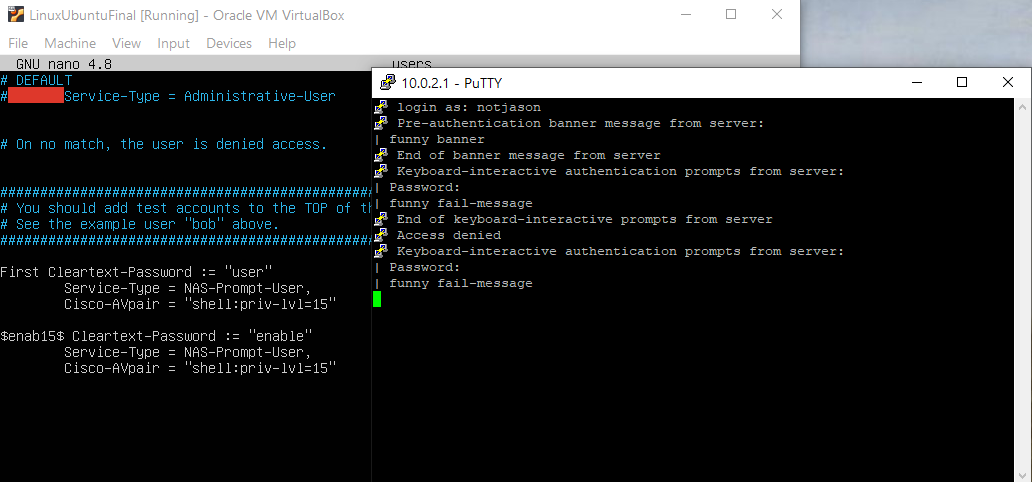
Add addressing information.

Find the “netplan” file and enter NANO editor for that file.

1. Restart the FreeRadius service and check the RADIUS configuration integrity to finalize RADIUS server-side configuration. Do a radtest to test if login would work.

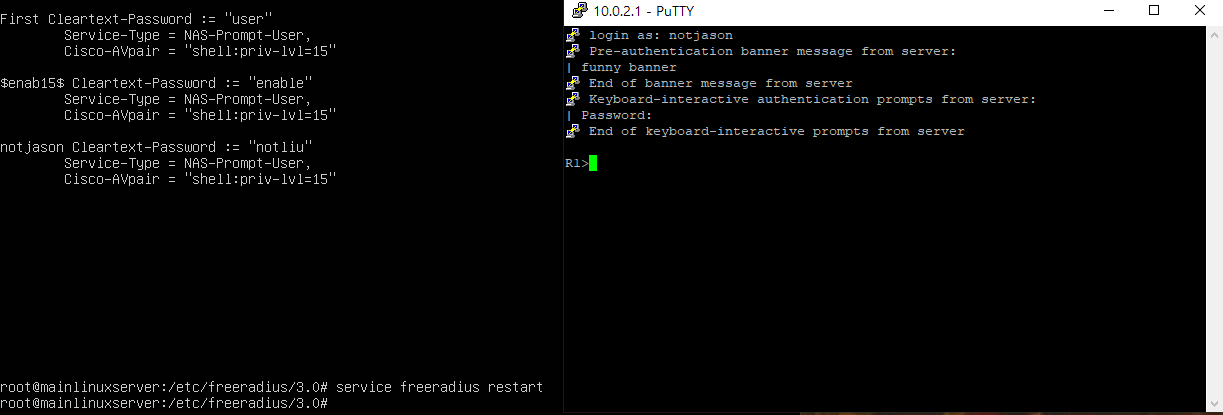


1. Finally, configure the router side for AAA and set it for RADIUS. (See *Configurations*).
2. Test a remote access login (PuTTY SSH was used). See if the entered user credentials work and if it denies unauthorized logins.



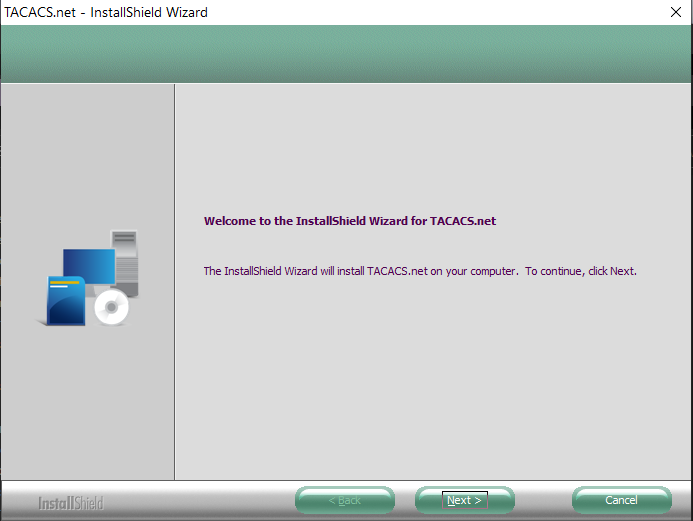
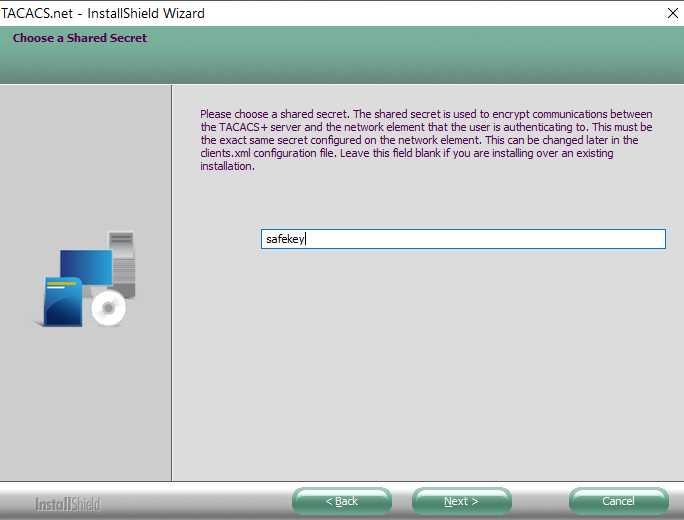
However, an unstated or incorrect user credential would prevent access and lead to a fail-message.

Using a correct username and password, RADIUS allows the user for access.

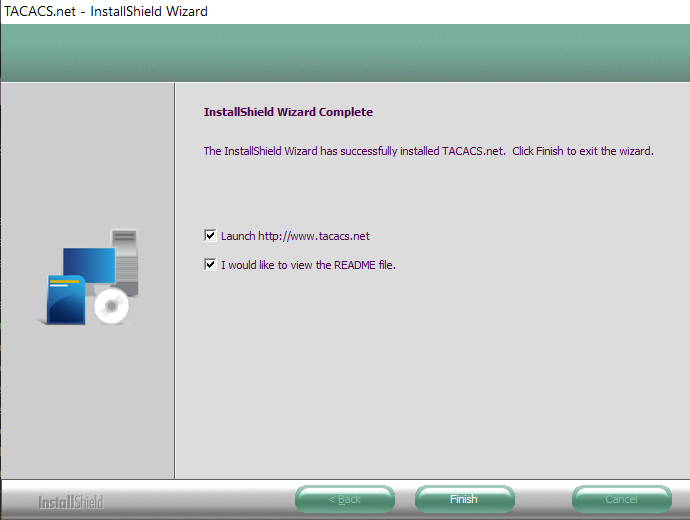


By creating the user on the server, the same credentials are now valid and RADIUS gives access.

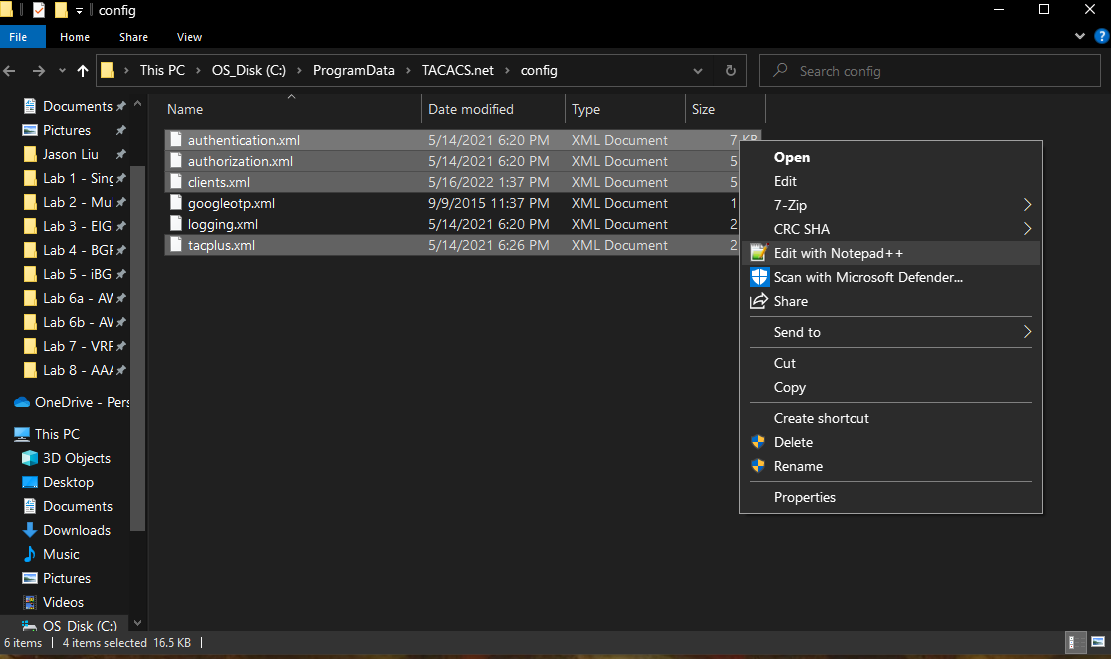
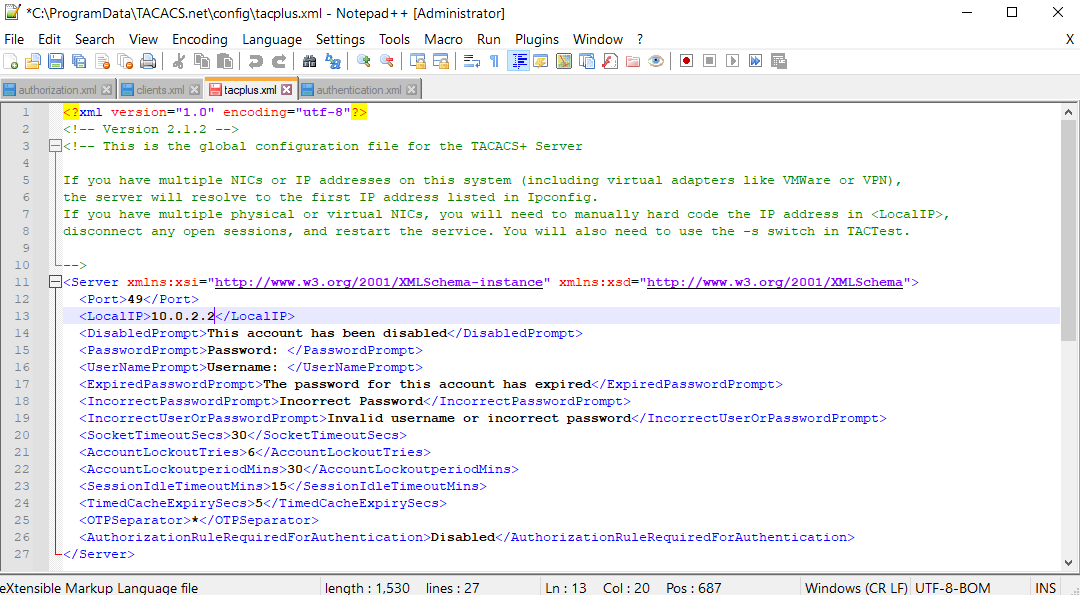
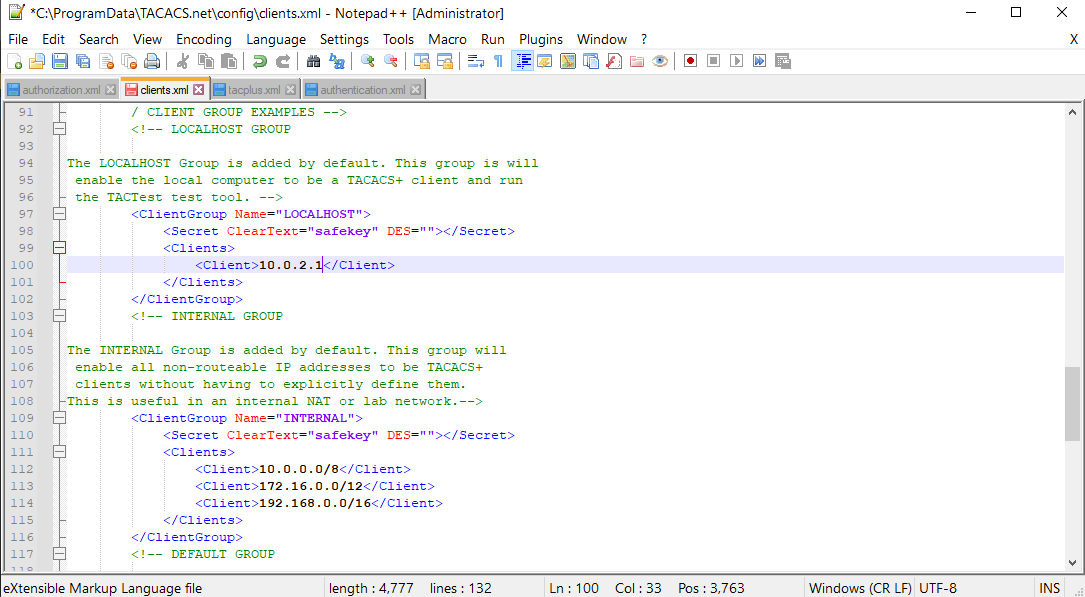
**TACACS+**:

1. Download and open the TACACS setup exe file from the official website. Going through the wizard, enter the secret key and remember it.

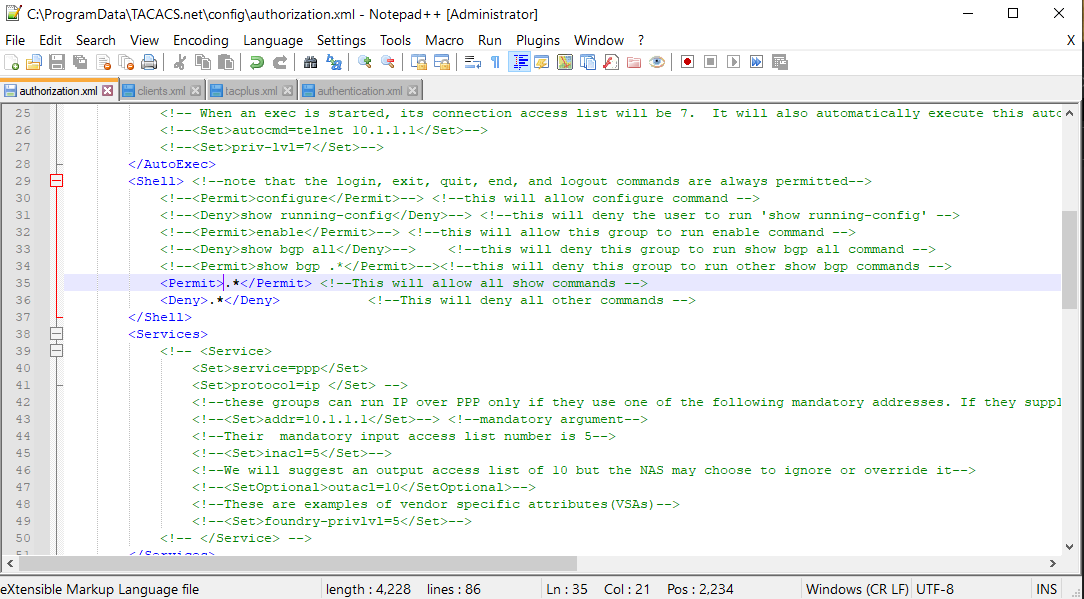
Create and remember the secret key.



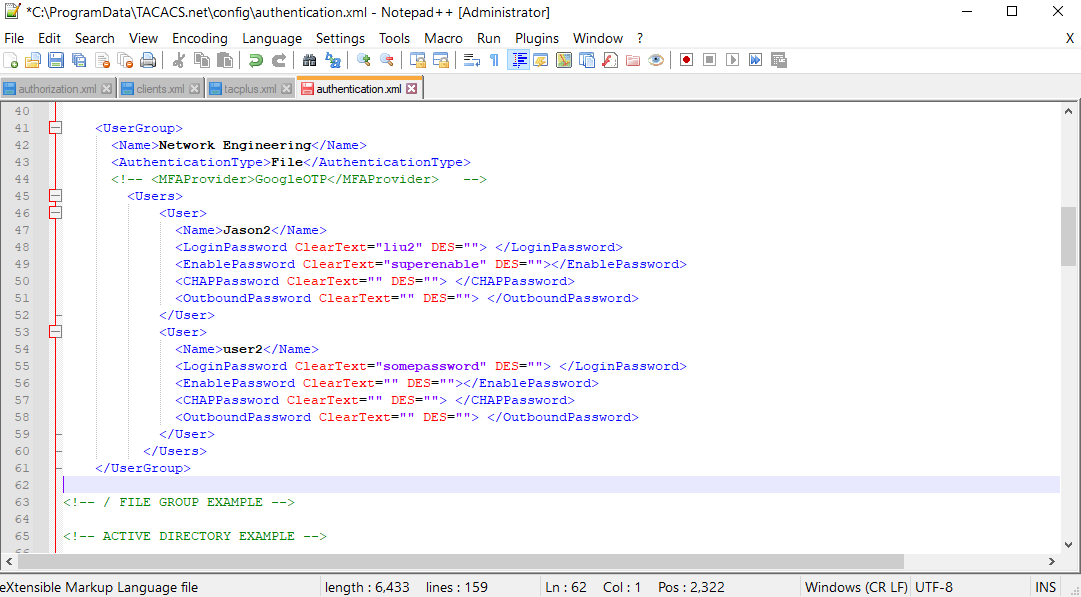
TACACS has finished installing. Files can be found under “TACACS.net” on the file list.

1. Find the TACACS.net file and open “authentication,” “authorization,” “tacplus,” and “clients” xml documents with Notepad++. Try to make any edit to prompt an option to enter administrator editor mode. Edit the server configurations to edit the IP address.
2. Under Authentication, uncomment and edit user configs. Under Authorization, change permissions to <.\*> to allow all. Under Clients, enter client info.

Create client and add its IP addresses.



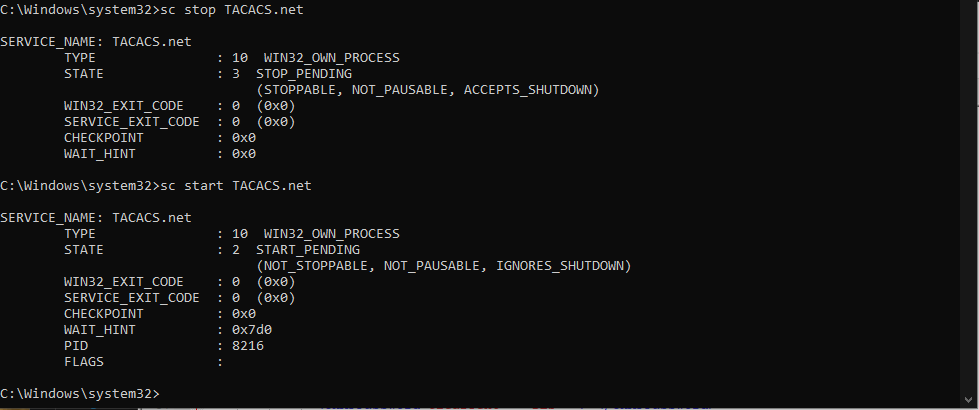
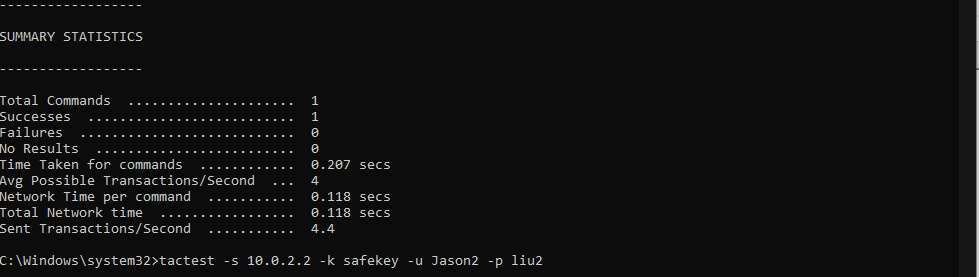
Allow all commands. Full authorization is given.



Create users and enter details.

1. Service stop and start to restart TACACS and update configs. Use TacVerify and TacTest command to verify configurations. The command is below should be entered int the admin command prompt:

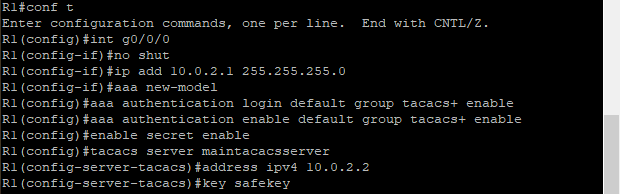
**tactest -s [*server IP address*] -k [*key*] -u [*username*] -p [*password*] –** Attempts TACACS login test and shows results of whether the login credentials would work.

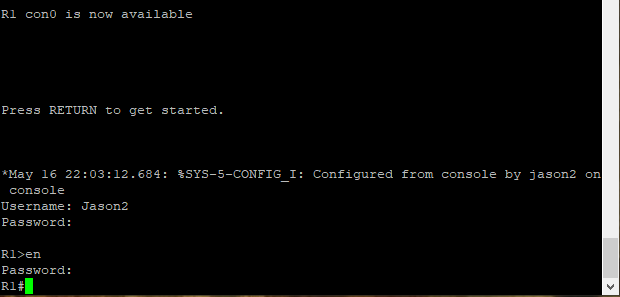


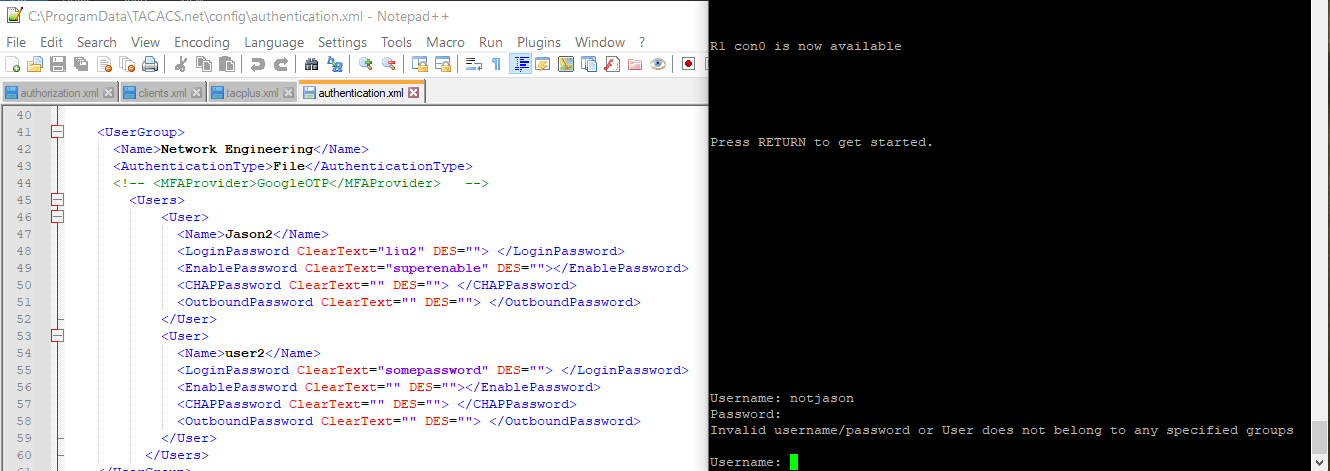
TacTest results. See success and failure numbers.

Restart TACACS service to update all information.

1. Configure the router side for AAA and set it for TACACS (see *Configurations*).



1. Test a remote access login (PuTTY SSH was used). See if the entered user credentials work and if it denies unauthorized logins.



However, an unstated or incorrect user credential would prevent access and lead to a fail-message.

Note that the username is “notjason,” which is not listed as a user.

Using a correct username and password, TACACS allows the user for access.

**Configurations**

Show Running-Configurations:

**R1 RADIUS**

R1#**show run**

interface GigabitEthernet0/2/1

no ip address

shutdown

negotiation auto

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

ip forward-protocol nd

ip http server

ip http authentication local

ip http secure-server

ip tftp source-interface GigabitEthernet0

**radius server mainlinuxserver**

**address ipv4 10.0.2.3 auth-port 1812 acct-port 1813**

**timeout 2**

**retransmit 2**

**key secretkey**

control-plane

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login authentication defauly

wsma agent exec

wsma agent config

wsma agent filesys

wsma agent notify

end

Building configuration...

Current configuration : 2056 bytes

Last configuration change by **notjason**

version 16.7

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

no platform punt-keepalive disable-kernel-core

**hostname R1**

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

**enable secret** **5 $1$drdy$3cVfwBxsTuOPryy7FYWA9/**

**aaa new-model**

**aaa authentication banner ^Cfunny banner^C**

**aaa authentication fail-message ^Cfunny fail-message^C**

**aaa authentication login default group RADIUS enable**

**aaa authentication enable default group RADIUS enable**

aaa session-id common

no ip domain lookup

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO220523GF

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

**interface GigabitEthernet0/0/0**

**ip address 10.0.2.1 255.255.255.0**

**negotiation auto**

interface GigabitEthernet0/0/1

no ip address

shutdown

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0/2/0

no ip address

shutdown

negotiation auto

**R1 TACACS+**

R1#**show run**

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

ip forward-protocol nd

ip http server

ip http authentication local

ip http secure-server

ip tftp source-interface GigabitEthernet0

**tacacs server maintacacsserver**

**address ipv4 10.0.2.2**

**key safekey**

control-plane

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login authentication defauly

wsma agent exec

wsma agent config

wsma agent filesys

wsma agent notify

end

Building configuration...

Current configuration : 1904 bytes

Last configuration change by jason2

version 16.7

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

no platform punt-keepalive disable-kernel-core

**hostname R1**

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

**enable secret 5 $1$drdy$3cVfwBxsTuOPryy7FYWA9/**

**aaa new-model**

**aaa authentication login default group tacacs+ enable**

**aaa authentication enable default group tacacs+ enable**

aaa session-id common

no ip domain lookup

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO220523GF

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

**interface GigabitEthernet0/0/0**

**ip address 10.0.2.1 255.255.255.0**

**negotiation auto**

interface GigabitEthernet0/0/1

no ip address

shutdown

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0/2/0

no ip address

shutdown

negotiation auto

interface GigabitEthernet0/2/1

no ip address

shutdown

negotiation auto

Show Commands:

R1 RADIUS:

R1#**show run | include RADIUS**

aaa authentication login default group RADIUS enable

aaa authentication enable default group RADIUS enable

RADIUS server mainlinuxserver

R1#**show run | include aaa**

aaa new-model

aaa authentication banner ^Cfunny banner^C

aaa authentication fail-message ^Cfunny fail-message^C

aaa authentication login default group RADIUS enable

aaa authentication enable default group RADIUS enable

aaa session-id common

R1 TACACS+:

R1#**show run | include tacacs**

aaa authentication login default group tacacs+ enable

aaa authentication enable default group tacacs+ enable

tacacs server maintacacsserver

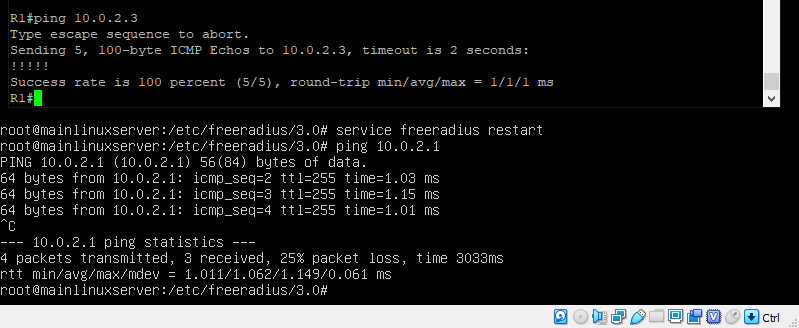
R1#**show run | include aaa**

aaa new-model

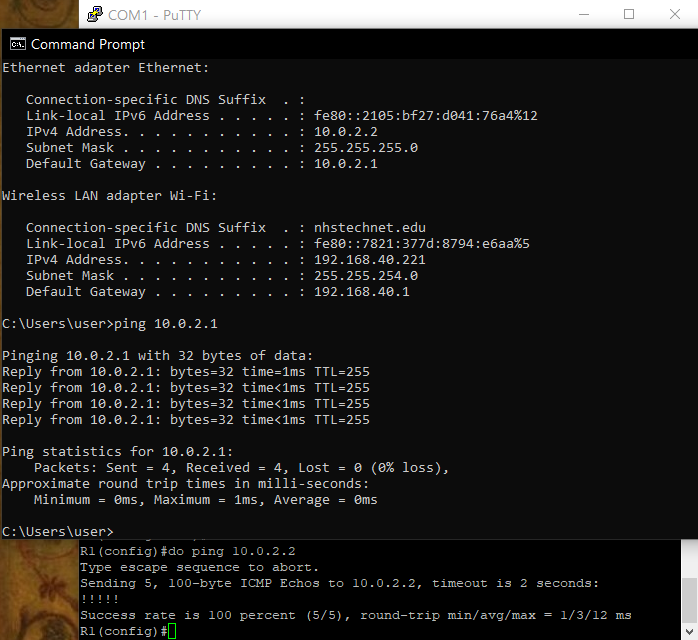
aaa authentication login default group tacacs+ enable

aaa authentication enable default group tacacs+ enable

aaa session-id common

Verification of connectivity via pings:

Connection between FreeRadius server (VM of Linux) and Client (Router) is successful.



Connection between TACACS+ server (hosted on PC) and Client (Router) is successful.

**Problems**

There were, expectedly, many problems during configuration, especially those stemming from the inexperience of the new interfaces. The one that would end up causing the greatest confusion is the necessity to restart the protocol service after making major changes, such as changing the shared secret key or an IP address. This caused many understanding conflicts as configurations do not update and apply instantaneously, making certain changes and configurations not display properly from a user device perspective. This interfered with expectations of certain commands and confused the general understanding of whether a command functioned or not. Ultimately, the “service FreeRadius restart” command for RADIUS and the “sc start/stop TACACS.net” for TACACS+ became go-to commands after any edit.

For FreeRadius v3 and Oracle VirtualBox, there were relatively small issues in regard to downloading the proper bootstrap version for USB integration. This was done to download operating disk images and .exe files for the virtual machine. All versions and options are found on the Oracle VirtualBox website. Version 6.0 was used for this lab. An important configuration directed towards VirtualBox is the necessity to change the network adapter to a *bridged adapter*. This allows the virtual machine to share and connect their information via ethernet, where it would otherwise be isolated. This was the key solution to more than a couple pinging problems.

Other quicker issues relating to TACACS+ included the lack of permission to run TacTest commands, which could be fixed by using the *admin* command prompt, done by right-clicking the command prompt application. When TACACS+ needed to be restarted but the start command says that it is already running, stop the TACACS before trying to restart it.

**Conclusion**

This revealing lab was indisputably valuable in the new interfaces to be familiar with. As my first useful application with Linux and Virtual Ware, I learned and navigated a wide array of commands, specifically those of Ubuntu Linux, Oracle VirtualBox, and Notepad++. In the foreign environment, it was an achievement to be able to understand and execute the protocols fully and functionally.