Purpose

The purpose of this lab was for us to set up TACACS+ and Radius authentication through a Virtual Machine. Through this lab, we learned more about how to secure our routers in better ways than we knew before. Instead of just putting in a password that the router checks, we set it up so the remote Radius or TACACS+ server checks the username and password which makes it more secure.

Background Information

In this lab, we used Radius and TACACS+ to remotely authenticate users in a router. Radius and TACACS+ are both forms of authentication used to permit users into a router. We setup Radius and TACACS+ on a remote Linux and Windows Server for this lab. We set these up through VMware, an application which lets us create virtual environments for testing. We also set up AAA in the routers, which is a protocol used to secure access to a cisco network device. It stands for Authentication, Authorization, and Accounting.

Lab Summary

In this lab, we configured Radius through a Windows Server and TACACS+ through a Linux Server. We started setting up TACACS+ by creating a virtual environment for Ubuntu and configuring it in there. We first installed TACACS+ through the terminal, and then we opened up the config file where we setup the groups and the users which we would use to sign into the router. We then boot up the router and configured it with AAA for TACACS+, and then the authentication was setup. Next, we started to setup Radius by creating a Windows Server and installing the Active Directories. We then setup a Network Policy and Radius server and added a group and user into the active directory. After this, we went into the router and setup AAA, and then the Radius authentication worked.

Lab Commands

In this lab, we learned a few new commands for AAA. We needed different commands for Radius and TACACS+. For both of them, we used the command “aaa new-model” which created a new AAA model. For TACACS+, we first used the command “aaa authentication login default group tacacs+ local none” which tells the router to check any login attempts against the TACACS+ server, and if it isn’t available to check the local user database. Next, we used the commands “aaa authorization exec default group tacacs+ local none,” “aaa authorization commands 0 default group tacacs+ local none,” and “aaa authorization commands 15 default group tacacs+ local none.” These three commands tell the router to contact the TACACS+ server and make sure if the user is allowed to run commands at that privilege level. The next three commands are “aaa accounting exec default start-stop group tacacs+,” “aaa accounting commands 0 default start-stop group tacacs+,” and “aaa accounting commands 15 default start-stop group tacacs+.” These three commands tell the router to log activity into the TACACS+ server. Finally, for TACACS+, there is the “tacacs-server host …” command, and the “tacacs-server key …” command. The host command lets the router know where the host for the TACACS+ server is, and the key command sets the TACACS+ encryption key.

Next, for Radius, we used only four commands, “aaa new-model,” “aaa authentication login default group radius local,” “aaa authorization exec default group radius if-authenticated,” and “radius-server host … key …” The first command is used to create a new AAA model. The next command tells the router to check for all login attempts against the Radius server first, and then check the locally configured user database if the server isn’t available. The next command is used to contact the Radius server to determine if the user is allowed, and then lets them into the router. The final command lets the router know the IP and secret key of the Radius server so they can connect.

Problems

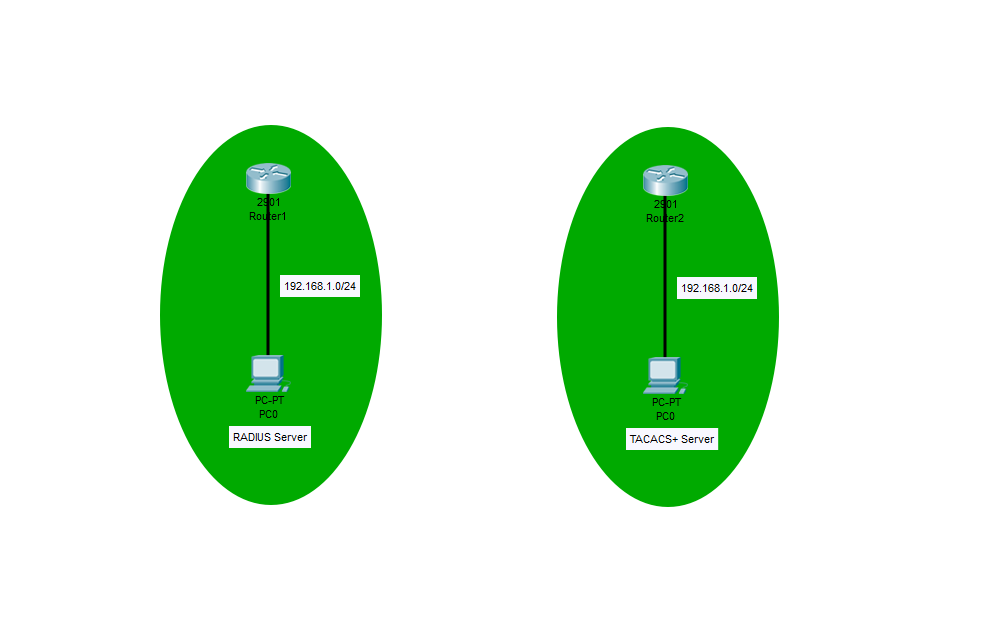
While we worked on this lab, we came across multiple difficult problems that would cause our Radius and TACACS+ servers to not work. We first worked on TACACS+ and had multiple problems there. One of our first problems was we didn’t know how to configure the server. After researching online, we figured out that we had to use the terminal and open a configuration file. Our next problem was how to configure the file. We had no idea how to set it up, but after researching it, we were able to add groups and users so the file would be configured. Afterwards, we setup the router configs and had another problem, it wasn’t working. After spending some time to fix it, we realized that the file was slightly misconfigured, and we hadn’t started up the TACACS+ server so the router could read it. After fixing these problems by fixing the file and starting up the server through the terminal, the TACACS+ server started working.

Our next problems were with Radius. After we researched how to set up Radius and finished installing the active directories, we had our first problem in what to do next. After we researched even more, we realized we had to configure network policies and the Radius server. After we set those up, we configured the router, and it did not work. After some troubleshooting, we found the problems. The first problem was that we did not create a user. We had to go into the active directory and create a user to add to the group we used for Radius. Our next problem was that the policies were misconfigured and caused the server not to connect to the router. We fixed this by going into the policy are changing the conditions we had before. Then, we also realized that the users were not configured correctly. We figured out that we had assigned them to the wrong group and quickly fixed it. The next problem that was causing the router and the Radius server to not communicate was that we needed to restart the NPS in order for it to take effect. Our final problem was simply that we were misspelling the password. We tested it for a while and wondered why it wasn’t working, but we created a new user with a different password, and it worked, so we determined we had put in the work password. Finally, after we fixed all these problems, we tested it on the router and the authentication worked. Overall, we had a lot of trouble getting this lab to work, but we did learn a lot about more secure ways of authentication and how to troubleshoot problems that weren’t as simple as missing commands in the router config.

Conclusion

Overall, I learned a lot during this lab about more secure methods of authentication such as Radius and TACACS+. After having many problems in setting them up whether it was as simple as misspelling in the TACACS+ config file, or putting in the wrong password when using Radius, we were able to pull through and get them both working. In the end, we learned how to correctly configure TACACS+, Radius, and AAA in order to setup a more secure router.

Network Diagram

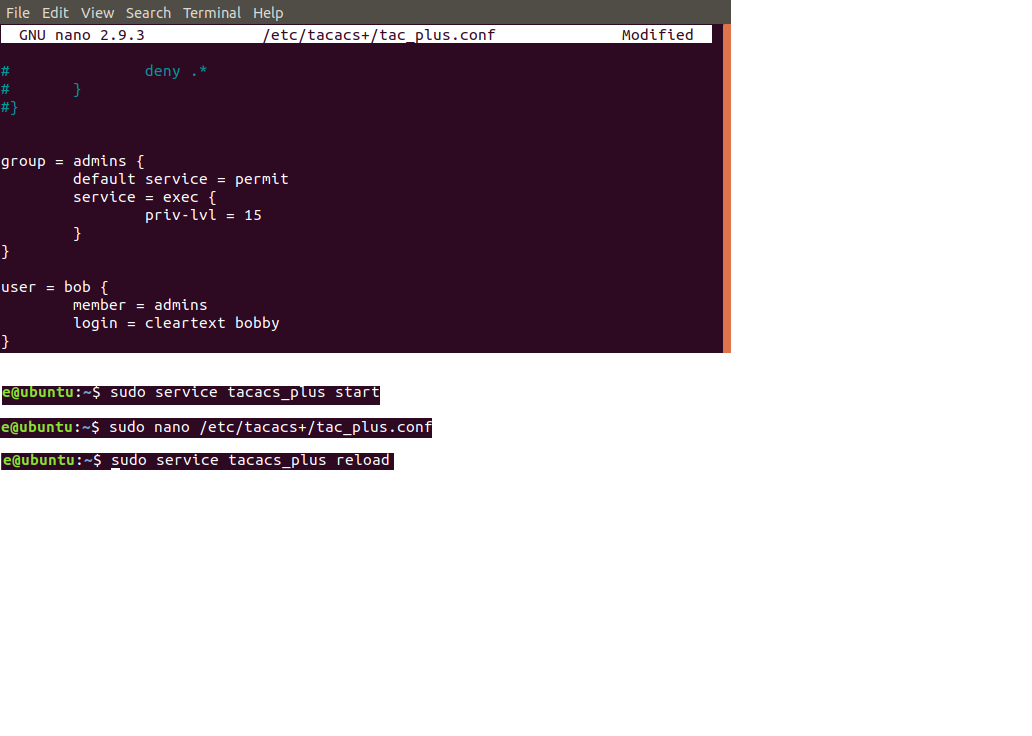


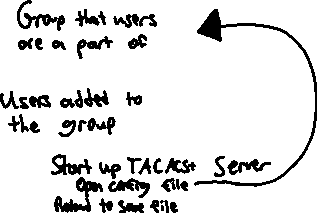
Configurations

---------

TACACS+

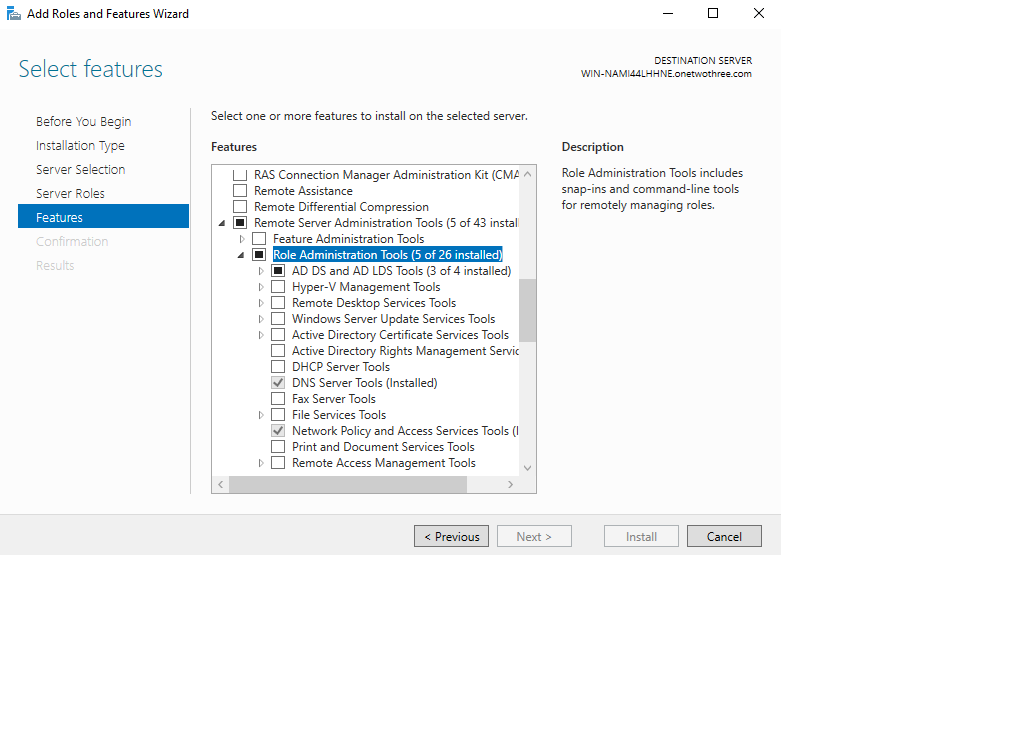
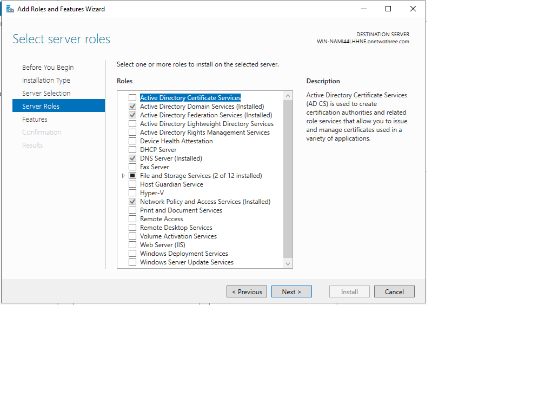
---------



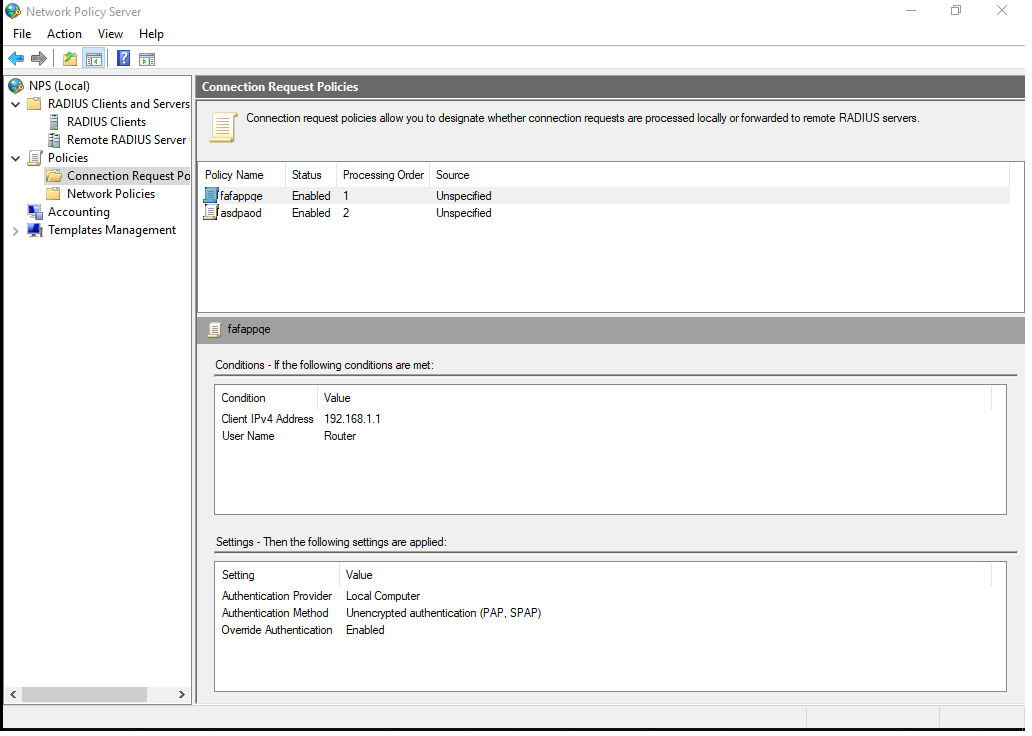


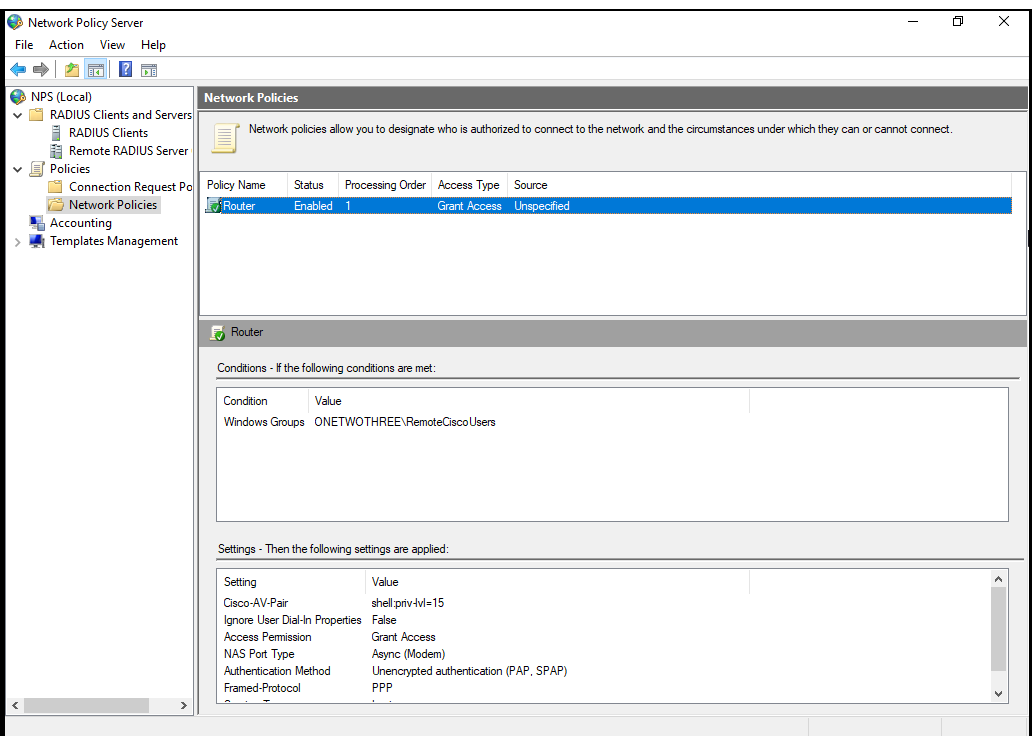
--------

Radius

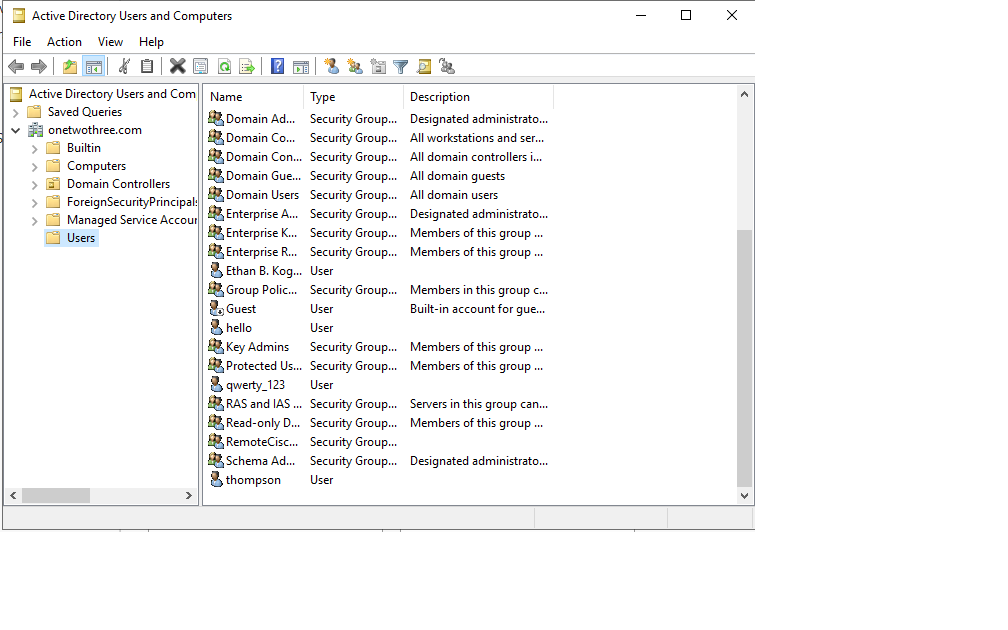














-------------

R1 RADIUS

-------------

Last configuration change at 22:15:45 UTC Mon Oct 28 2019

version 15.2

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

hostname Radius

boot-start-marker

boot-end-marker

aaa new-model

aaa authentication login default group radius local

aaa authorization exec default group radius if-authenticatedS

aaa session-id common

memory-size iomem 10

ip cef

no ipv6 cef

multilink bundle-name authenticatedS

voice-card 0

license udi pid CISCO2901/K9 sn FTX1704Y038

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 aafant privilege 15 password 0 aafant

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

no shut

interface Serial0/0/0

no ip address

shutdown

clock rate 2000000

interface Serial0/0/1

no ip address

shutdown

clock rate 2000000

ip forward-protocol nd

no ip http server

no ip http secure-server

radius-server host 192.168.1.3 key Cisco123$

control-plane

mgcp profile default

gatekeeper

shutdown

line con 0

line aux 0

line 2

no activation-character

no exec

transport preferred none

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

----------

R2 TACACS+

----------

Current configuration : 2035 bytes

Last configuration change at 20:06:55 UTC Wed Oct 23 2019 by ethan

version 15.2

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

hostname Tacacs

boot-start-marker

boot-end-marker

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

aaa authorization commands 15 default group tacacs+ local none

aaa accounting exec default start-stop group tacacs+

aaa accounting commands 0 default start-stop group tacacs+

aaa accounting commands 15 default start-stop group tacacs+

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 FTX1704Y038

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

ip address 192.168.1.1 255.255.255.0

duplex auto

speed auto

no shut

interface GigabitEthernet0/1

no ip address

shutdown

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

ip forward-protocol nd

no ip http server

no ip http secure-server

tacacs-server host 192.168.1.2

tacacs-server host 192.168.1.3

tacacs-server key testing123

control-plane

mgcp profile default

gatekeeper

shutdown

banner motd ^Cthe user and pass is ethan^C

line con 0

line aux 0

line 2

no activation-character

no exec

transport preferred none

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

stopbits 1

line vty 0 4

transport input all

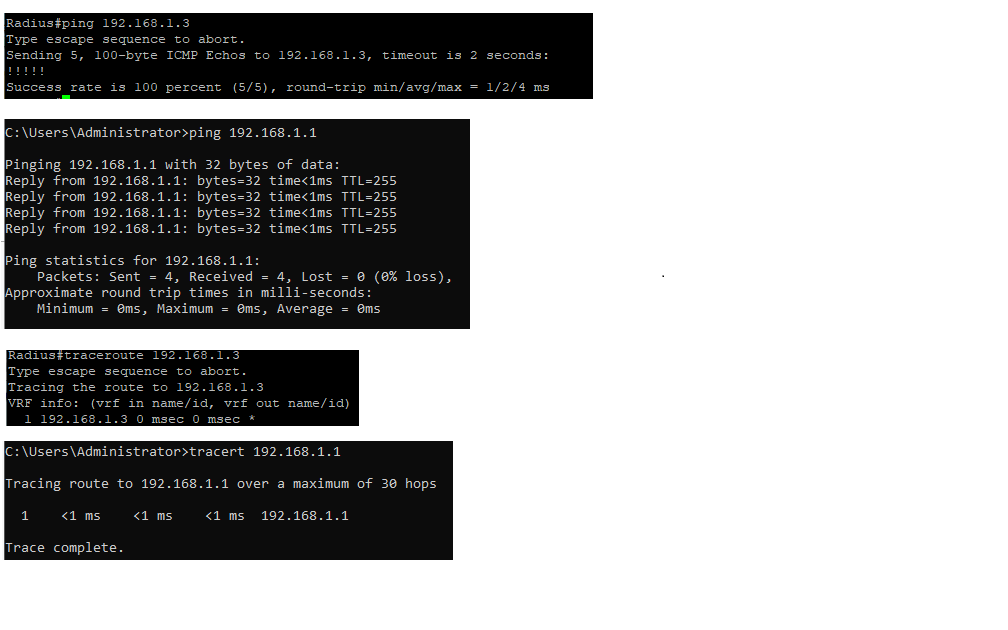
scheduler allocate 20000 1000

End

------------------------

RADIUS Pings/Traceroutes

------------------------



------------------------

TACACS+ Pings/Traceroutes

------------------------

