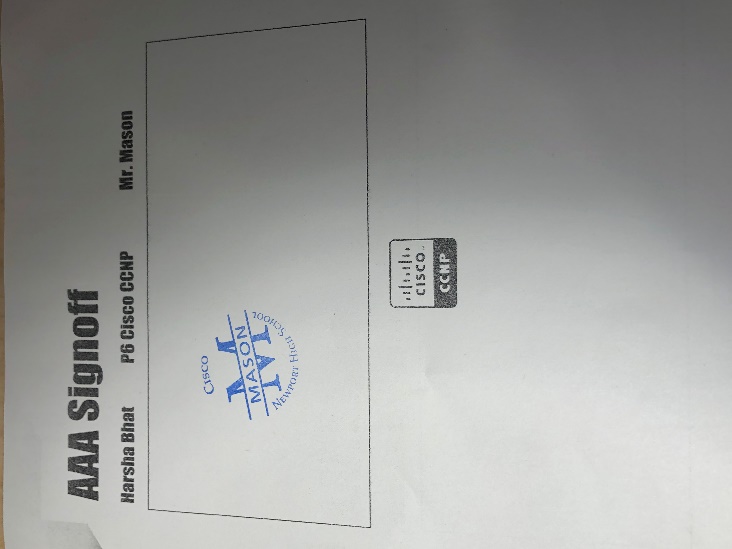
# AAA: Radius & Tacacs+

# 

# harsha Bhat

# CCNP 6/7

# Mr. Mason



The purpose of this lab is to configure AAA on a router. Users with random names and passwords will be denied access, while the right user with the right password will gain access.

### Background Information

Back in the dark ages, we learned to configure routers and switches with enable secret and console passwords individually. However, for loads and loads of devices, this is not a very scalable or secure solution. Enter Authentication, Authorization, and Accounting, an authentication service hosted on an external server to automatically provide security to all connected devices.

Authentication refers to the identification of a user. There are many ways you can identify a user: examples include a password, pin number, id card, fingerprint. Just something unique to the user that can’t be replicated. On a switch/router, this is the console/telnet and enable password. Credentials stored on the AAA server replace these: if a user’s credentials match with those stored on the server, the user gains access to the device.

Authorization refers to determining the privilege or access level of a user. For example, higher level employees might have access to all company devices, while lowly unpaid interns might only have access to the coffee machines. In AAA, this is done through router commands such as aaa authorization enable default group radius/tacacs+, which allows users logging into console/telnet mode to instantly go directly to enable mode. We accidentally configured this in our lab when it was not our intention, see the problems section for more information on that issue.

Accounting refers to the resources consumed by a user. If Bob hogs the coffee machine all day, he should be billed for binging, not Alice. Likewise, on host devices on an AAA network, employers and service providers would want to determine how much to charge for their services, productivity of their employees, etc.

There are many different protcols for configuring AAA, and we used both RADIUS and TACACS+. Having used both of them, I would argue TACACS+ is better as it provides more authorization features and uses TCP over RADIUS’s UDP, at the cost of more resource utilization. I would recommend RADIUS for Linux users and Windows users with foreknowledge of active directory (TACACS is losing support in later versions of Linux and is being emulated through docker containers which I feel is too complicated a process), and TACACS+ for Windows users and network administrators.

### Lab Summary\*

I configured RADIUS on Linux and TACACS on windows server. I used Kali Linux with Freeradius and TACACS+ provided by Tacacs.net. First, I sudo apt-get install freeradius freeradius-utils on the VM to install Freeradius on Kali; this is an automatic process. Next, I edited the clients.conf and “users” files. In clients.conf, I created the definition of a RADIUS client. The client has 4 fields, of which only IP and secret are required:

1. IP address of the client, e.g., ipaddr = 127.0.0.1. Optionally, you can add a subnet mask, e.g., netmask = 24.
2. “Secret” is a text value you will later configure on the server.
3. “Shortname” is an alternative identifier used in place of the IP address.
4. “Nastype” tells a script what common authentication database the device is from. In this lab, we are using nastype = cisco as our routers are ISR4321s. Other recognized databases are computone, livingston, max40xx, multitech, netserver, pathras, patton, portslave, tc, usrhiper, and “other”.

Next, I edited the users file and added user entries for both console and $enab15$ and launched debug mode with sudo freeradius -X. Now the server was up and running on the VM, it was time to move on to the router. First, I defined a new AAA model, and enabled RADIUS for the console and Finally, I defined the radius server that the router should use as the same one I had in clients.conf.

As Kali had no support for TACACS+, I made use of the Active Directory Windows Server I had previously abandoned. I edited Authentication.xml very similarly to the RADIUS users file: I made a user, then configured enable and console passwords. I’ll say it again: I liked this method of creating users better as it was a lot more straightforward. In Authorization.xml, I simply made sure users could use all commands, however there are additional features. In tacplus.xml, I configured the server to use the IP of the Windows server, which of course caused some connectivity issues. Finally, in configuring the router, I did everything I did for RADIUS except I changed the commands to authenticate console and enable mode to do so using TACACS.

\*problems section also reads a bit like a summary

***<****In my Local Cisco Lab, AAA has been likened to a wordless scream. This was justified through the extensive issues in VM configuration. As of the time of writing, I know of one CCNP who is still configuring AAA despite having started it January 20th****>***

### Lab Commands

Most configuration done in this lab was not on the CLI of devices such as a router, but rather in text files that changed properties of pre-existing downloadable server applications.

Reasons, reasons, reasons!

Kali:

sudo apt-get install freeradius freeradius-utils – installs freeradius

sudo freeradius -X – starts Freeradius in debug mode

Windows command prompt:

sc start TACACS.net – starts TACACS+

tacverify/tactest – troubleshoot for syntax and server errors

Router:

aaa new-model – enables AAA on the router

aaa authentication attempts login 5 – you get 5 tries to login before being disconnected from SSH.

aaa authentication login default group radius/tacacs+ – enables console login for either RADIUS or TACACS+.

aaa authentication enable default group radius/tacacs+ – enables enable mode login for either RADIUS or TACACS+.

radius server <name> – defines an arbitrary radius server

address ipv4 <ip> auth-port 1812 acct-port 1813 – ties the previously created server to the one you created on the conf/authentication file. Note that for this to work, you need the IP to be the one you defined in the file.

key x – enter this to match the secret “x” you configured under clients.conf or were prompted to make in the TACACS installation.

aaa authentication banner ‘x’**\*** – configures the login banner

aaa authentication fail-message ‘x’**\*** – configures the message you receive if you enter the wrong password.

**\***Gabe used these, but I chose not to for simplicity reasons. However, for security reasons, these should be similar to the MOTD banner on a router, for legal reasons generally a very nastily-worded message to deter hackers as shown in the example below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UNAUTHORIZED ACCESS TO THIS NETWORK DEVICE IS PROHIBITED

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

You must have explicit permission to access or configure this device.

All activities performed on this device may be logged.

Violations of this policy may result in disciplinary action,

and may be reported to law enforcement.

There is no right to privacy on this device.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Network diagram

“Server” is not an intimidating bare metal box, rather a PC running Kali and Windows server VMs

Server

### Configurations

## Radius

Clients.conf:

client 10.0.0.1 {

secret = nuts

nastype = cisco

shortname = router

}

*Note: my config is in the form*

*client <ip> {*

*<attribute> = <value>*

*}*

*When default is*

*client <short-name> {*

*<attribute> = <value>*

*}*

Users:

memey Cleartext-Password := "deez"

Service-Type = NAS-Prompt-User

***This*** *is the objectively horrible part of the RADIUS user config I was talking about.*

Cisco-AVPair = "shell:priv-lvl=15"

$enab15$ Cleartext-Password := "nuts"

Service-Type = NAS-Prompt-User

Cisco-AVPair = "shell:priv-lvl=15"

Router:

aaa new-model

aaa authentication attempts login 5

aaa authentication login default group radius

aaa authentication enable default group radius

radius server deez

address ipv4 10.0.0.1 auth-port 1812 acct-port 1813

timeout 30

retransmit 3

key nuts

## TACACS+

Authentication.xml:

*Is it just me, or does the text seem bigger in XML tags?*

<User>

<Name>memey</Name>

<LoginPassword ClearText="deez" DES="">

<EnablePassword ClearText="nuts" DES="">

</User>

Authorization.xml

<Permit>.\*</Permit>

Tacplus.xml

<LocalIP>10.0.0.2</LocalIP>

Router:

aaa new-model

aaa authentication login default group tacacs+

aaa authentication enable default group tacacs+

tacacs server deez

address ipv4 10.0.0.2

key nuts

*Note: the key this time does not come from a configuration file, rather it was preconfigured by you as part of the TACACS+ installation. It can be viewed/edited in clients.xml if forgotten.*

Problems

The main issue in AAA was my relative inexperience with VMs, despite extensive usage of a Kali VM in learning pentesting at home.

First, I tried to configure RADIUS on Windows Active Directory. I received Active Directory documentation from fellow CCNP Myles Robertson, and successfully configured Active Directory. However, I was having trouble with the windows VM not bridging correctly with the local network, so I opted to install RADIUS on a physical device. Little did I know I would go back to other VMs, each with their fair share of connectivity issues.

Originally, I wanted to set up the server on a laptop which had Kali as its primary OS rather than use a VM. However, in installing packages, a major error occurred: the package installer broke, and nothing was downloadable anymore. We asked fellow CCNP Manmeet Ranu to help us with the issue which he successfully fixed after many hours, a service which we are grateful for. However, this caused us to move on to alternative forms of configuring RADIUS while waiting for the laptop’s Kali to be fixed: Gabe tried a Raspberry Pi while I used a Kali VM. Eventually, each of us stuck with our respective configurations.

I had some issues with configuration files so Gabe finished his configuration ahead of me (and I’m told, everyone else working on the lab). However, while he felt his configuration was technically correct, he had an issue where he went directly to privileged exec mode in logging in, bypassing user exec mode completely. ***This was the famous Authentication vs. Authorization issue you’ve all been waiting for*.** Essentially, I figured out he had used an authorization command, aaa **authorization** enable default group radius, instead of the equivalent authentication command, aaa **authentication** enable default group radius/tacacs+. Moreover, as he didn’t know how to configure an enable password on RADIUS (he likely thought the “secret” entered under clients.conf was the enable password, though I am not entirely sure of this), I went into his users file and added an entry under $enab15$. This was a feature of RADIUS I found imprudent, as the format suggested that a user logging into the console was entirely different from the same user logging into enable mode. I had to advise this 2nd step of the process in helping CCNP Jason Liu configure an enable password on RADIUS later.

After Gabe finished, I had some minor issues with the router but successfully implemented RADIUS. However, in TACACS+ configuration I led the class by installing TACACS+ from TACACS.net first, and by distributing it to the others. But then, I missed Cisco 2 days in a row for AP testing, so I had to then catch up to Gabe, who finished TACACS with the resources and guides I gave him and had gone so far even as to create his own documentation, which I must say is very good, much better than what I gave him and even what I found online.

Finally, an issue that plagued the completion of this lab was the Kali website being down/blocked by Mr. Mason’s firewall. Thankfully, we had downloaded all the needed packages by the time this occurred and were able to complete the lab with minimal delay.

### Conclusion

AAA was a troublesome lab well worth the reward. If at all possible, I will try to figure out Bridging adapters for VMs or use physical devices. As a side note, it was interesting seeing TACACS+ configuration files being written in XML, which we hear about all the time in CCNA questions.