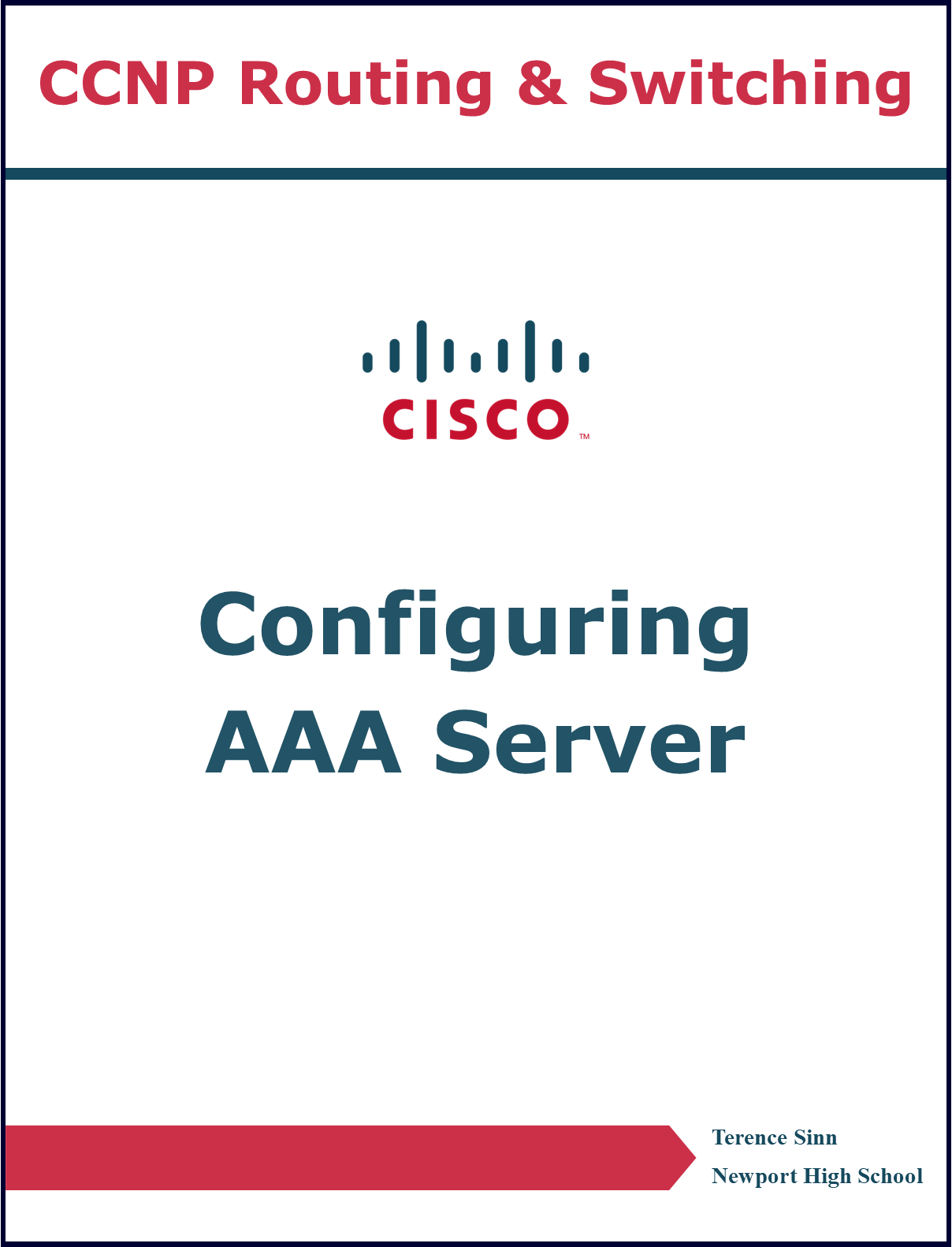
****

**Purpose**

Cybersecurity is a massive consideration in modern networks, and requirements are always increasing to meet the needs of network users. The design of using a centralized server to handle login information is beneficial because passwords will be inaccessible to intruders on unrestricted networking devices like routers. This model of keeping passwords on a separate server is called AAA (triple A), authentication, authorization, and accounting. Through this, the access to configure and monitor networking devices can be greatly controlled to improve security of administrative tools.

**Background Information**

AAA stands for authentication, authorization, and accounting. The authentication part refers to logging in or providing proof of who you are. Authorization comes after authentication and refers to allowing access based off who you are. For example, a network administrator and network support technician will both be able to authenticate into the system but be authorized to use different tools on the network. Authorization is the way to differentiate between different people. Accounting refers to keeping track of all the access to the network, including saving who has logged in and what changes have been made.

AAA is commonly implemented separately from network devices on a centralized server. The purpose is that the AAA server will act as the central database for all usernames and passwords so that they do not have to be stored on the networking devices themselves. This makes it so that if a threat actor gets hold of a networking device, they will be unable to crack usernames and passwords on it because none are stored in the device. A centralized server also avoids the need of configuring the username and password for each networking device, instead requiring the configuration of access to the AAA server. This simplifies configuration so that adding new devices to the network can be configured for access control quicker.

AAA is defined by two separate protocols, RADIUS and TACACS+. RADIUS is an open vender neutral protocol whereas TACACS+ is a propriety protocol. As a result, RADIUS is a more flexible protocol with more options to extend and customize because of its open nature. However, out of the box, TACACS+ comes with more features if the network is only using Cisco devices. One such feature is that TACACS+ encrypts all AAA data whereas RADIUS only encrypts the passwords sent. Another difference is that TACACS+ uses TCP ports 1645 and 1646 for reliability whereas RADIUS uses UDP ports 1812 and 1813.

**Lab Summary**

A router was connected to a PC running a VirtualBox instance of Ubuntu. The Ubuntu instance had FreeRADIUS installed and acted as the AAA server. On the Ubuntu instance, under root/etc/freeradius/3.0, the clients.conf file and users file were edited to configure the RADIUS server. The router was then configured to use the radius server for authentication.

**Lab Commands**

**Cisco**

**Router (config) # aaa new-model**

Enables the configuration of AAA servers on the router.

**Router (config) # radius server <server name>**

Creates a new radius server with the given server’s name. For best practices, use full capitalization for the server’s name to differentiate from Cisco key words.

**Router (config-radius-server) # address <IP version> <IP address> auth-port <port number> acct-port <port number>**

Configures the IP address of the radius server as well as the authentication, authorization, and accounting ports to use. IP version can be either IPv4 or IPv6. IP address is the corresponding IP address of the server. For radius, the auth-port will be 1812 and the acct-port will be 1813.

**Router (config-radius-server) # key <shared key>**

Configures the shared key between the router and the AAA server to communicate and establish a connection.

**Router (config) # aaa authentication login default group radius local**

Configures the router to use the radius server as the main authentication source when logging in. It also configures a backup login with locally stored passwords.

**Router (config) # aaa authorization exec default group radius if-authenticate**

Configures the router to use the radius server to authorize creating an privilege 15 exec shell if the user is authenticated.

**Router# test aaa group radius <username> <password> new-code**

Tests and returns authentication information of the given username and password on the configured radius server.

**Linux**

**sudo apt install freeradius**

Installs FreeRADIUS on Ubuntu.

**systemctl restart freeradius**

Restarts the FreeRADIUS process to apply any recent changes to the configuration files.

**freeradius -X**

Starts FreeRADIUS in debug mode for troubleshooting.

**radtest <username> <password> <server IP address> <port number> <secret key>**

Tests the FreeRADIUS server with the given information. The username and password must be a valid user configured in the users file. The IP address should refer to itself if the FreeRADIUS server is installed no the same device, that is 127.0.0.1 or localhost. The port number should be 1812. The secret key should be a key configured in the clients.conf file.

**Network Diagram with IP’s**

**Diagram

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| **Device Name** | Interface | IP Address |
| **Router** | G0/0/0 | 192.168.0.1/24 |
|  |  |  |
| **AAA Server** | Enp0s3 | 192.168.0.2/24 |
|  |  |  |

**Configurations**

**Router**

**Router# show running-config**

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname Router

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

aaa new-model

aaa authentication login default group radius local

aaa authorization exec default group radius if-authenticate

aaa session-id common

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21482HYV

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

ip address 192.168.0.1 255.255.255.0

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

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

radius server SERVER

address ipv4 192.168.0.2 auth-port 1812 acct-port 1813

key test

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

end

**AAA Server**

**etc/freeradius/3.0/clients.conf**

client 192.168.0.1 {

ipaddr = 192.168.0.1

secret = test

nastype = cisco

shortname = router

require\_message\_authenticator = no

}

**etc/freeradius/3.0/users**

bob Cleartext-Password := "password"

Service-Type = NAS-Prompt-USER,

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

**Problems**

Throughout the entire configuration process, a persistent problem was that the AAA server was unable to ping the router. The root cause for the problem has not been determined, but it is hypothesized that it’s a bug with VirtualBox’s ARP process. When running **show arp** on the router, the ARP entry for the AAA server is there. However, when running **arp -a** on the AAA server, there are no entries. A static ARP entry was configured on the AAA server with the command **arp -i enp0s3 -s 192.168.0.1 <MAC address>** but the devices could still not ping each other. The workaround was to restart the VirtualBox machine and connectivity is restored.

Another problem was that the router could not reach the AAA server for authentication when using the **test aaa group radius <username> <password> new-code** command. The cause of this is that the default ports 1645 and 1646 were still used. Changing the **address ipv4 192.168.0.2** command to **address ipv4 192.168.0.2 auth-port 1812 acct-port 1813** fixed this problem.

Another unresolved problem was the configuration of authorization. The initial goal was to restrict the ability of an authenticated user to enter privileged exec mode. The first attempt was to implement the Cisco-AVPair = “shell:priv-lvl=1” line under a user in the etc/freeradius/3.0/users file. To enable authorization, the command **aaa authorization exec default group radius if-authenticated.** Despite this being the suggested way to configure privilege levels from the Cisco configuration website, this did not work. Another attempt was to change the Service-Type to Administrative-User. This also did not work. Restrictive authorization remains unresolved.

**Conclusion**

AAA servers are a simple, yet effective way to control access to a network. It’s modularity and centralization make both expanding networks both easier and more secure to govern. AAA’s wide availability through open-source tools such as FreeRADIUS make it accessible to be implemented on all networks. However, with the rise of cloud-based services, the usage of on-site AAA servers may be replaced with AAA servers hosted on the cloud. Regardless, configuration of AAA remains a crucial part of network security best practices.