- AAA -

AAA

Securing access to Cisco routers and switches is a critical concern. Often, access is secured using *enable* and *vty/console* passwords, configured locally on the device.

For large networks with many devices, this can become unmanageable, especially when passwords need to be changed. A centralized form of access security is required.

AAA is a security system based on **Authentication**, **Authorization**, and **Accounting**.

Authentication is used to grant or deny access based on a user account and password. **Authorization** determines what level of access that user has on the Router/router when authenticated. **Accounting** can keep track of who logged into what device, and for how long.

AAA must be enabled globally on a router/Router. By default, it is disabled.

Router(config)# *aaa new-model*

Privilege Levels

IOS devices have a total of **16 privilege levels**, numbered 0 through 15. **User Exec** mode is privilege level 1. **Privileged Exec** mode is privilege level 15.

We can create a custom Privilege level, including the commands users are allowed to input at that mode:

Router(config)# *privilege exec all level 3 show interface*
Router(config)# *privilege exec all level 3 show ip route*
Router(config)# *privilege exec all level 3 show reload*

To then enter that privilege level from User Mode:

Router> *enable 3*

* * *

Configuring Authentication

Authentication can be handled several different ways. We can use a username and password configured locally on the router/Router:

Router(config)# *username MYNAME password MYPASSWORD*

Or we can point to a centralized RADIUS or TACACS+ server, which can host the username/password database for all devices on the network:

Router(config)# *radius-server host 172.16.10.150*
Router(config)# *radius-server key MYKEY*

Router(config)# *tacacs-server host 172.16.10.151 key MYKEY*
Router(config)# *tacacs-server key MYKEY*

The above commands point to a *host* server. A measure of security is maintained by using a shared *key* that must be configured both on the router and the RADIUS/TACACS+ server.

We can also create groups of RADIUS or TACACS+ servers to point to:

Router(config)# *aaa group server radius MYGROUP*
Router(config-sg-radius)# *server 172.16.10.150*
Router(config-sg-radius)# *server 172.16.10.152*
Router(config-sg-radius)# *server 172.16.10.153*

There are several key differences between RADIUS and TACACS+ servers:

- RADIUS is an industry standard protocol, while TACACS+ is Cisco proprietary
- RADIUS utilizes UDP, while TACACS+ utilizes TCP
- RADIUS encrypts only the password during the authentication process, while TACACS+ encrypts the entire packet

There is one additional key difference: TACACS+ allows for the *authorization* of a user, in addition to the *authentication* of a user. Thus, TACACS+ allows us to control what commands a particular user can input. RADIUS provides only authentication services.

* * *

Configuring Login Authentication

On the previous page, we directed our router to a specific RADIUS or TACACS server. Next, we must specify which methods of authentication we want our router to *consider* when a user logs in. We can actually configure the router to use *multiple* forms of authentication (up to **four**):

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

The above command creates an *authentication* profile for router *login* named *default,* directing the router to use the *RADIUS* server(s), *TACACS+* server(s), and *local* forms of authentication, **in that order**.

Thus, the RADIUS server(s) will always be used, unless they fail. Then the TACACS+ server will be used and then finally local authentication. This provides fault-tolerance and automatic failover.

You should *always* include local at the end of this command. Otherwise, if all RADIUS and TACACS+ servers are down, you won't be able to log into the router.

Multiple authentication profiles can be created. Each must have a unique profile name. Obviously, *default* is the default profile name. If we wanted a separate profile named ONLYLOCAL:

Router(config)# *aaa authentication login ONLYLOCAL local*

The last step in configuring authentication is to apply the profile to a "line," such as the console or telnet ports.

Router(config)# *line vty 0 15*
Router(config-line)# *login authentication default*

Notice we referenced the authentication profile's name of *default*.

* * *

Configuring PPP Authentication

The previous page illustrates the use of AAA Authentication to control user login to routers and switches. Additionally, we can use AAA to authenticate both ends of a PPP connection.

Point-to-Point Protocol (PPP) is a standardized WAN encapsulation protocol that can be used on a wide variety of WAN technologies, including:

- Serial dedicated point-to-point lines
- Asynchronous dial-up (essentially dialup)
- ISDN

To specify the authentication methods for PPP:

Router(config)# *aaa authentication ppp MYPROFILE radius local*

Notice the new keyword of *ppp*, as opposed to *login*. Once we have specified the desired authentication methods, we must apply this profile to the appropriate interface:

Router(config)# *interface serial 0*
Router(config-if)# *encapsulation ppp*
Router(config-if)# *ppp authentication pap MYPROFILE*

Or:

Router(config)# *interface serial 0*
Router(config-if)# *encapsulation ppp*
Router(config-if)# *ppp authentication chap MYPROFILE*

Notice that the top example uses PAP (*Password Authentication Protocol*), while the bottom example uses CHAP *Challenge Handshake Authentication Protocol*. PAP sends the password in **clear text**, whereas CHAP encrypts the password with an MD5 hash. Thus, CHAP is far more secure.

* * *

Configuring Authorization

Authorization allows us to dictate what rights a user has to the router once they have logged in:

```
Router(config)# aaa authorization commands default radius
Router(config)# aaa authorization config-commands default radius
Router(config)# aaa authorization exec default radius
Router(config)# aaa authorization network default radius
Router(config)# aaa authorization reverse-access default radius
```

The Router will consult the RADIUS server to "authorize" access to specific privilege modes (or in the case of TACACS+, even specific commands). A user trying to access Global Configuration mode must be authorized to do so on the RADIUS server.

Explanations of the above "sections" we can authorize:

- **commands –** access to any Router command at any mode
- **config-commands –** access to any Router configuration command
- **exec –** access to privileged mode
- **network –** access to network-related commands
- **reverse-access** – ability to reverse telnet from the Router

We can then apply this authorization to a line:

```
Router(config)# line vty 0 15
Router(config-line)# authorization default
```

* * *

Configuring Accounting

We can configure accounting to log access to routers and switches:

```
Router(config)#  aaa accounting system default stop-only
Router(config)#  aaa accounting exec default start-stop
Router(config)#  aaa accounting commands 3 default start-stop
Router(config)#  aaa accounting commands 15 default start-stop
```

We can configure accounting on three separate functions:

- **System** – records system-level events, such as reloads
- **Exec** – records user authentication events, including duration of the session
- **Commands** (1-15) – records every command typed in at that privilege level. In our above example, we're logging our custom Privilege Level 3

We can then specify *when* these functions should be recorded:

- **Start-stop** – recorded when the event starts and stop
- **Stop-only** – recorded only when the event stops

Finally, we must apply this to a line:

```
Router(config)#  line vty 0 15
Router(config-line)#  accounting default
```

Troubleshooting AAA

To debug the various functions of AAA:

```
Router#  debug aaa authentication
Router#  debug aaa authorization
Router#  debug aaa accounting
Router#  debug radius
Router#  debug tacacs
```

* * *