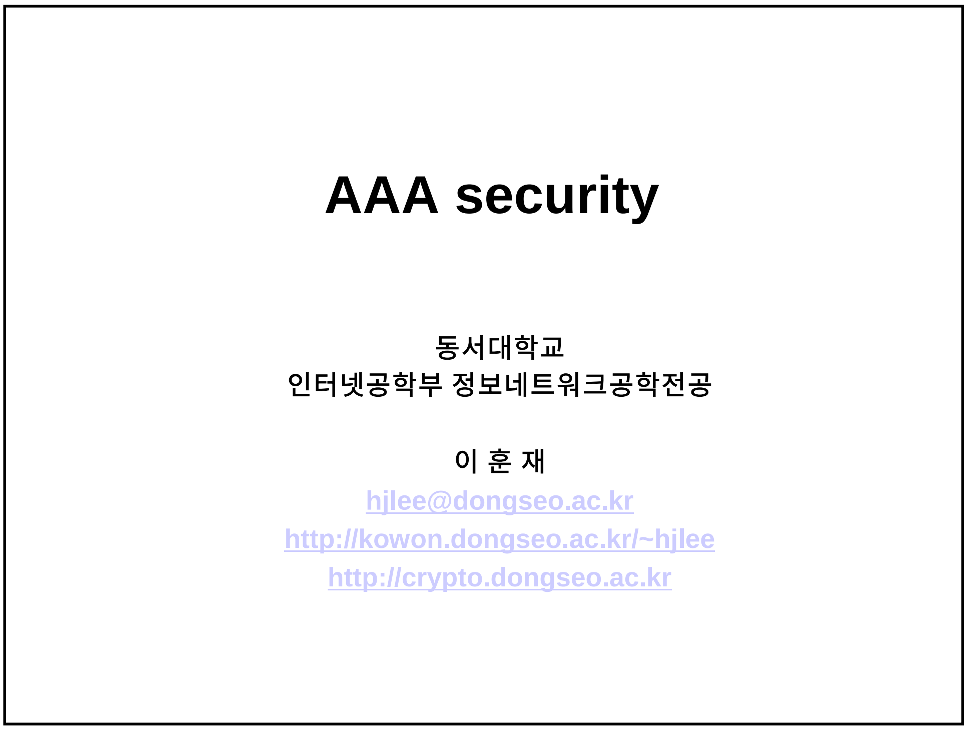
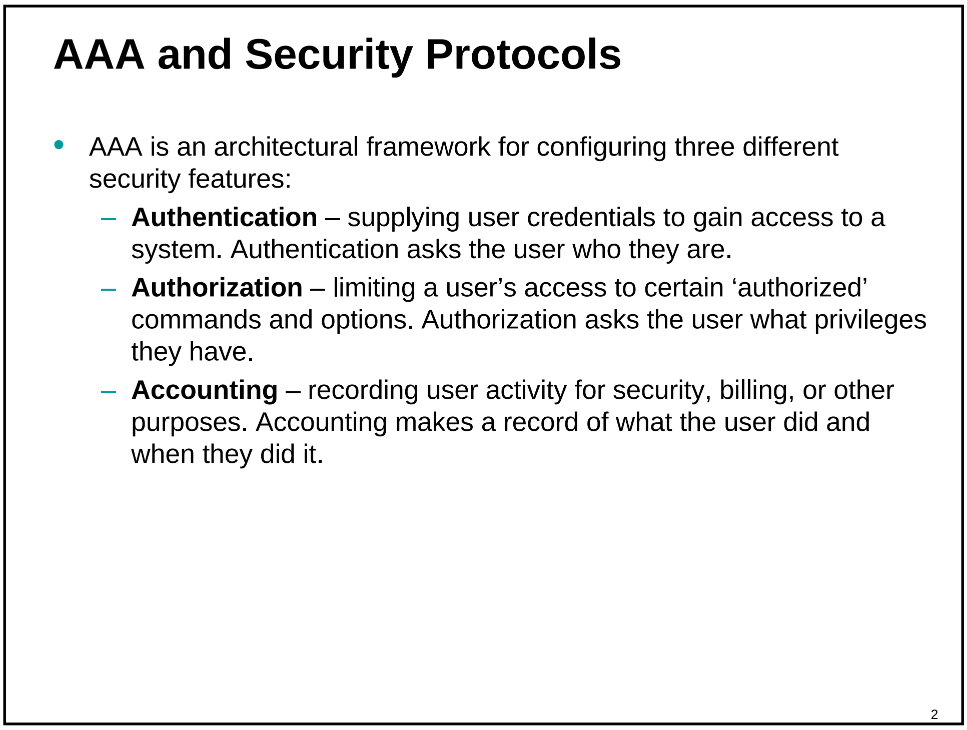


AAA security

동서대학교
인터넷공학부 정보네트워크공학전공

이 훈 재
hjlee@dongseo.ac.kr
http://kowon.dongseo.ac.kr/~hjlee
http://crypto.dongseo.ac.kr



AAA and Security Protocols

- AAA is an architectural framework for configuring three different security features:
  - **Authentication** – supplying user credentials to gain access to a system. Authentication asks the user who they are.
  - **Authorization** – limiting a user’s access to certain ‘authorized’ commands and options. Authorization asks the user what privileges they have.
  - **Accounting** – recording user activity for security, billing, or other purposes. Accounting makes a record of what the user did and when they did it.

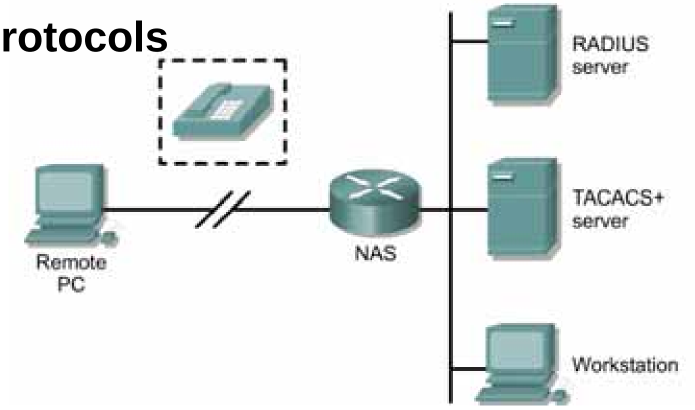
Advantages of AAA

- Using AAA for device logins offers three main advantages:
  - **AAA provides scalability** – many Cisco IOS devices can use AAA to refer to a common set of usernames and passwords on a central security server
  - **AAA supports standardized protocols** – Cisco IOS devices running AAA can communicate securely with security servers using the protocols covered later.
  - **AAA allows for multiple backup systems** – Cisco IOS devices can consult a second or third source of information if the primary source of security information is offline

Cisco IOS and Security Protocols

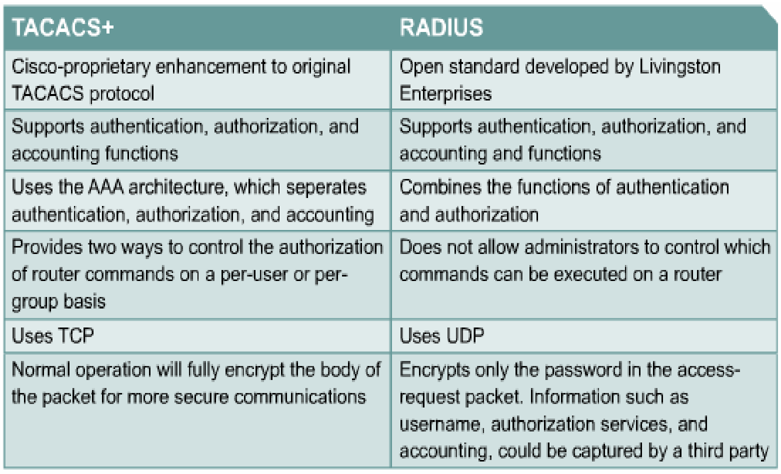
- The Cisco IOS supports three key security protocols:
  - Terminal Access Controller Access Control System Plus (TACACS+)
  - Remote Authentication Dial-In User Service (RADIUS)
  - Kerberos

Security protocols



- Hosts use a security protocol to communicate with a specialized security server.
- The security server maintains a password and username database.
- The security server also stores authorization configurations and accounting information.
- The Cisco IOS supports three key security protocols named TACACS+, RADIUS, and Kerberos.

TACACS+ vs. RADIUS

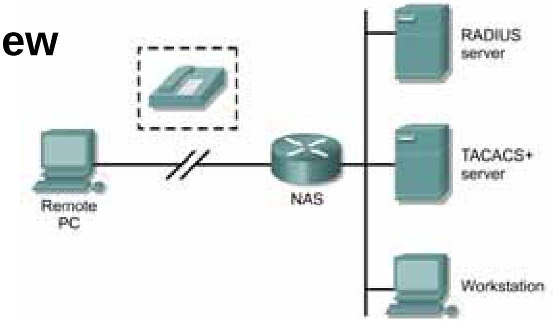


TACACS+	RADIUS
Cisco-proprietary enhancement to original TACACS protocol	Open standard developed by Livingston Enterprises
Supports authentication, authorization, and accounting functions	Supports authentication, authorization, and accounting and functions
Uses the AAA architecture, which seperates authentication, authorization, and accounting	Combines the functions of authentication and authorization
Provides two ways to control the authorization of router commands on a per-user or per-group basis	Does not allow administrators to control which commands can be executed on a router
Uses TCP	Uses UDP
Normal operation will fully encrypt the body of the packet for more secure communications	Encrypts only the password in the access-request packet. Information such as username, authorization services, and accounting, could be captured by a third party

TACACS+

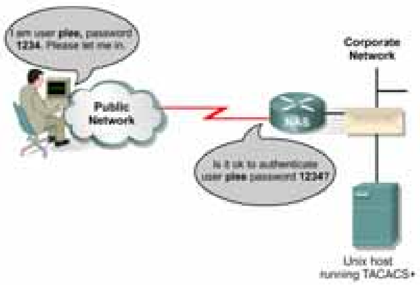
- **TACACS+** is a security application used with AAA that provides **centralized validation of users** attempting to gain access to a router or network access server.

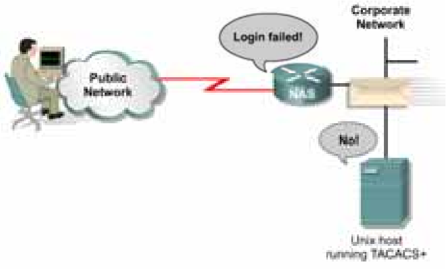
TACACS+ Overview



- TACACS+ is a **security application** used with AAA that provides centralized validation of users attempting to gain access to a router or network access server.
- TACACS+ services are maintained in a database on a TACACS+ daemon running on a UNIX, Windows NT, or Windows 2000 workstation.
- TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

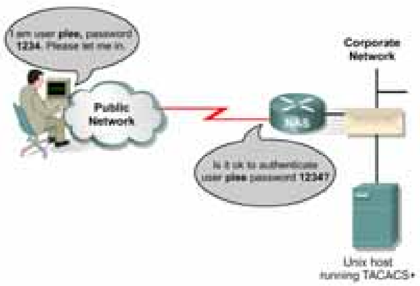
TACACS+

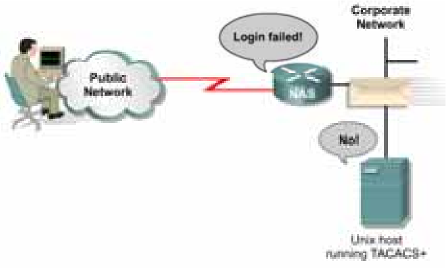

I am user plee, password 1234. Please let me in.
Public Network
Corporate Network
NAS
Is it ok to authenticate user plee password 1234?
Unix host running TACACS+



- TACACS+ provides the **most comprehensive and flexible security configurations** when using **Cisco** routers and switches.
- TACACS+ originated from the TACACS and extended TACACS protocols.
- Neither of these older protocols is seen as a viable solution.
- A **Cisco-proprietary protocol, TACACS+** is **not compatible** with **TACACS** or extended TACACS.

TACACS+


I am user plee, password 1234. Please let me in.
Public Network
Corporate Network
NAS
Is it ok to authenticate user plee password 1234?
Unix host running TACACS+

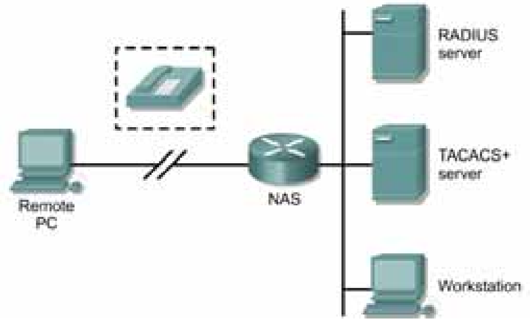


- TACACS+ uses **TCP** to communicate between a TACACS+ server and a TACACS+ client.
- Unlike RADIUS, TACACS+ separates the functions of authentication, authorization, and accounting.
- Use TACACS+ to take advantage of all of the features supported by AAA.

RADIUS

- RADIUS is a **distributed client/server system** used with AAA that secures networks against unauthorized access.
- In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server.
  - The RADIUS server contains all user authentication and network service access information.
  - RADIUS can also be used for 802.1x implementations.

RADIUS Overview

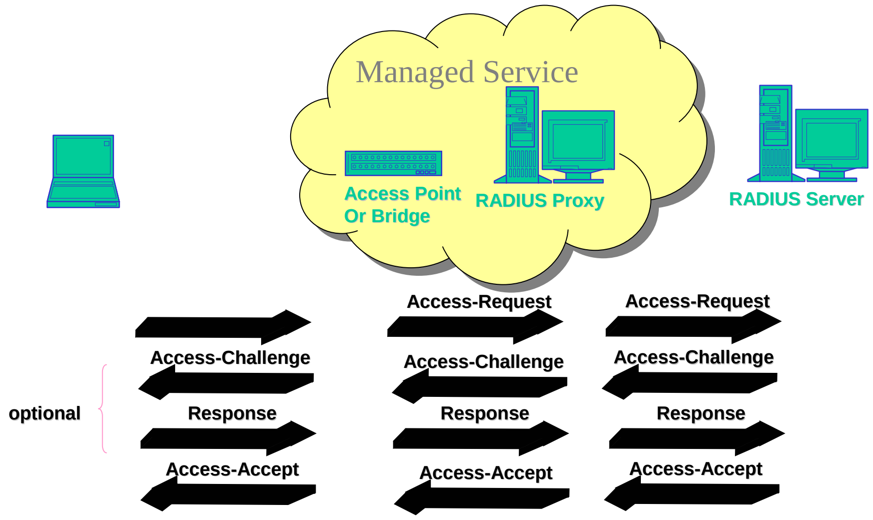


- RADIUS is a **distributed client/server system** used with AAA that secures networks against unauthorized access.
- In the **Cisco implementation**, RADIUS **clients run on Cisco routers** and send authentication requests to a **central RADIUS server**.
- This central server contains all user authentication and network service access information.

RADIUS

- Remote Authentication Dial In User Service
  - Originally developed for dial-up access
- Widely implemented client/server network protocol
  - Implemented in transport layer (using UDP)
  - Clients are all types of Network Access Servers (NAS)
  - Provides 3A (authentication, authorization, accounting)
  - Example: NT4.0 IAS
- Supports mobile and remote users
  - physical ports (modems, DSL, wireless)
  - virtual ports (extranets, VPNs)
- Allows centralized/remote control and accounting
- Proxy RADIUS protocol allows distributed authentication

How it works



Managed Service

Access Point Or Bridge

RADIUS Proxy

RADIUS Server

Access-Request

Access-Request

optional

Access-Challenge

Access-Challenge

Access-Challenge

Response

Response

Response

Access-Accept

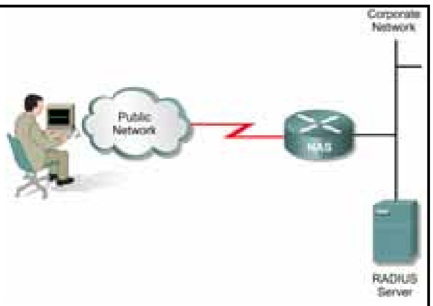
Access-Accept

Access-Accept

RADIUS Security Mechanisms

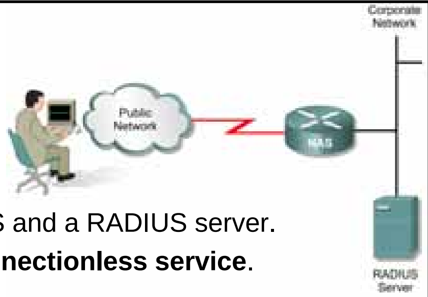
- RADIUS client and server share a secret (usually entered as a string password)
- Each request receives an authenticator (nonce)
- Messages are encrypted using a stream cipher, generated using MD5 applied to the secret and authenticator
  - Plaintext (user and password fields) are XORed with stream
  - Chained CBC-style if password is too large
- A few weaknesses were discovered
  - MD5 was not meant to be a stream cipher
  - By XORing two captured ciphertexts, the eavesdropper gets the XOR of the two plaintexts; if one password is shorter, the suffix of the other appears in plaintext
  - Similarly, enables an offline attack on the shared secret
- A few improvements were suggested, including use of symmetric encryption
- Better yet, RADIUS exchange can be encrypted via VPN (IPSec)

RADIUS



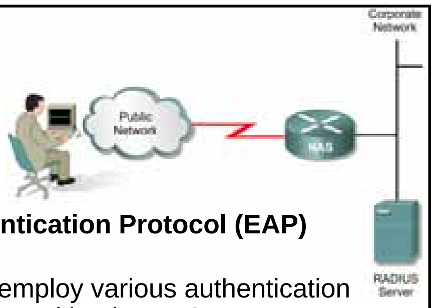
- The RADIUS protocol was developed by Livingston Enterprises as an authentication and accounting protocol for use with access servers.
- RADIUS is specified in RFCs 2865, 2866, and 2868.
- Even though TACACS+ offers more flexible AAA configurations, RADIUS is a popular AAA solution.
- RADIUS is an open standard and typically uses fewer CPU cycles.
- RADIUS is less memory intensive than the proprietary TACACS+.
- Currently, RADIUS is the only security protocol supported by emerging wireless authentication protocols.

RADIUS



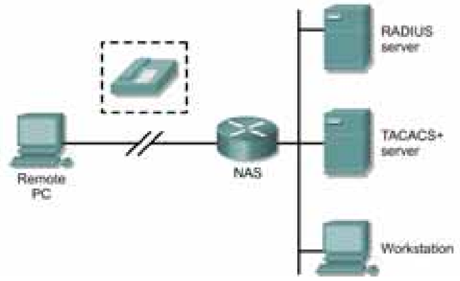
- **UDP** communications between a NAS and a RADIUS server.
- RADIUS protocol is considered a **connectionless service**.
- RADIUS is a client/server protocol.
- The RADIUS client is typically a Network Access Server (NAS).
- The RADIUS server is usually a daemon process running on a **UNIX or Windows** machine.
- The client passes user information to designated RADIUS servers and acts on the response that is returned.
- RADIUS servers receive user connection requests, authenticate the user, and return the configuration information necessary for the client to deliver service to the user.
- A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

RADIUS



- Today, the emerging **Extensible Authentication Protocol (EAP)** relies on **RADIUS services**.
- EAP with RADIUS makes it possible to employ various authentication methods on a network that are not supported by the NAS.
- As a result, customers can use standard support mechanisms for authentication schemes, such as token cards and public key, to strengthen end-user and device-authenticated access to their networks.
- There are several variants of EAP, such as the Cisco proprietary Lightweight Extensible Authentication Protocol (**LEAP**) and the standards-based Protected Extensible Authentication Protocol (**PEAP**).
- These authentication protocols provide dynamic per-user, per-session Wired Equivalent Privacy **(WEP) key enhancements** to decrease a variety of wireless network attacks.

Kerberos Overview



- Kerberos is a **secret-key network authentication protocol** used with AAA that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication.
- Kerberos was designed to authenticate requests for network resources.
- Kerberos is based on the concept of a **trusted third party** that performs secure verification of users and services.
- The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be.
- To accomplish this, a trusted Kerberos server **issues tickets to users**.
- These tickets, which have a limited lifespan, are stored in a user's credential cache.
- These tickets are then used in place of the standard username and password authentication mechanism.

Kerberos

- **Kerberos** is a secret-key network authentication protocol used with AAA that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication.
- Kerberos issues tickets to users which are stored in a user's credential cache and can be used in place of standard username and password authentication.
  - The tickets have a limited life span and alleviate the need to send username and password information over the network.

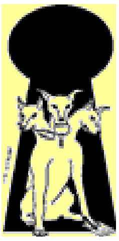
Client/Server Authentication

Kerberos

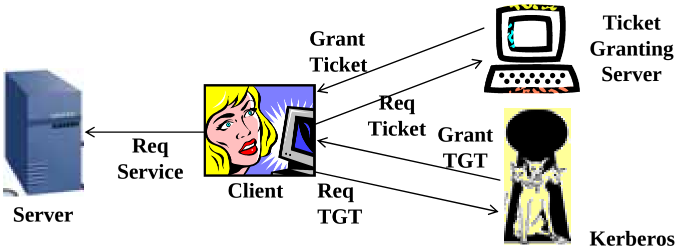
Main sources: Stallings, Schneier, Kaufman et al

Kerberos

- Client / Server Authentication service
  - Deployed as a network service that allows users and servers to mutually authenticate
  - Uses conventional symmetric key as proof of identity (DES)
  - Developed in MIT by Project Athena.
- Types of concerns addressed
  - User impersonation
  - Alteration of a device identity
  - Replay attacks
- Requirements
  - Security:
    - eavesdropper cannot get enough information
    - Kerberos itself should be secure
  - Reliability and high availability
  - Transparency to the User

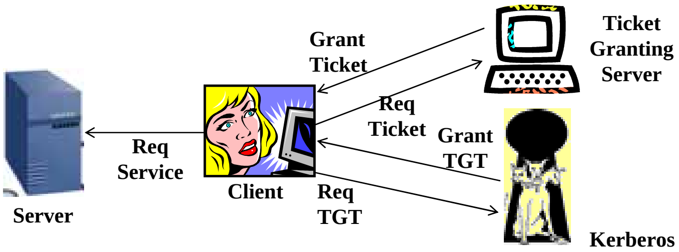


Kerberos Protocol

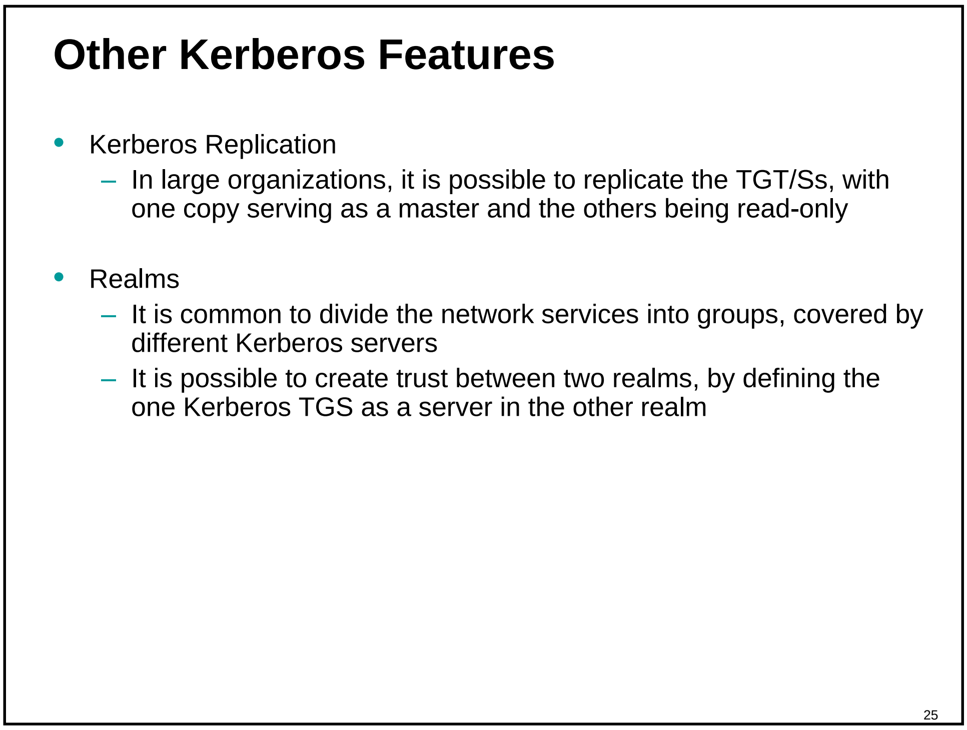


- Ticket: T(c,s) = s,E_{Ks}(c,a,v,$K_{c,s}$)
  - c-client, s-server, a-client address, v-validity time
  - Used as a “pass” until expiration
- Authenticator: A(c,s) = E $_{Kc,s}$(c,t,k)
  - t-time stamp, k-additional session key
  - Used once, but the client can generate as many as she wishes

Kerberos Protocol

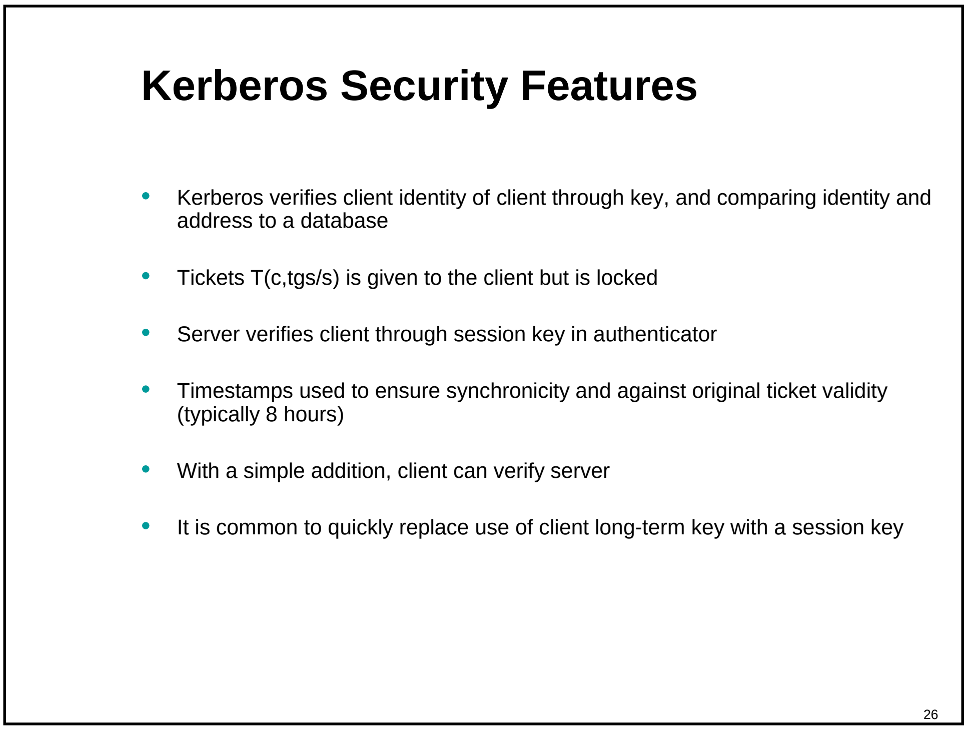


- Req TGT: Send c,tgs
  - Grant TGT: Gen $K_{c,tgs}$; Send E_{Kc}($K_{c,tgs}$), E_{Ktgs}(T(c,tgs))
  - Req Ticket: Send $E_{Kc,tgs}$(A(c,tgs)), E_{Ktgs}(T(c,tgs)), s
  - Grant Ticket: Gen $K_{c,s}$; Send $E_{Kc,tgs}$($K_{c,s}$), E_{Ks}(T(c,s))
  - Req Service: $E_{Kc,s}$(A(c,s)), E_{Ks}(T(c,s))



Other Kerberos Features

- Kerberos Replication
  - In large organizations, it is possible to replicate the TGT/Ss, with one copy serving as a master and the others being read-only
- Realms
  - It is common to divide the network services into groups, covered by different Kerberos servers
  - It is possible to create trust between two realms, by defining the one Kerberos TGS as a server in the other realm



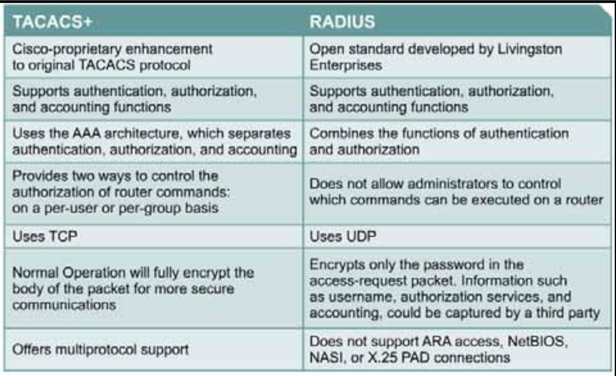
Kerberos Security Features

- Kerberos verifies client identity of client through key, and comparing identity and address to a database
- Tickets T(c,tgs/s) is given to the client but is locked
- Server verifies client through session key in authenticator
- Timestamps used to ensure synchronicity and against original ticket validity (typically 8 hours)
- With a simple addition, client can verify server
- It is common to quickly replace use of client long-term key with a session key

Attacks on Kerberos Security

- Kerberos itself stores many keys and should be protected
- Tickets may be replayed within allowed lifetime. Server should store recent requests and check for replays
- Adversary may cache many TGTs and work offline to decrypt them. Clients shall use safe passwords
- By changing server clocks, adversary may replay tickets. Hosts shall synchronize clocks often
- Kerberos will be enhanced with public-key cryptography and smart card-based key management

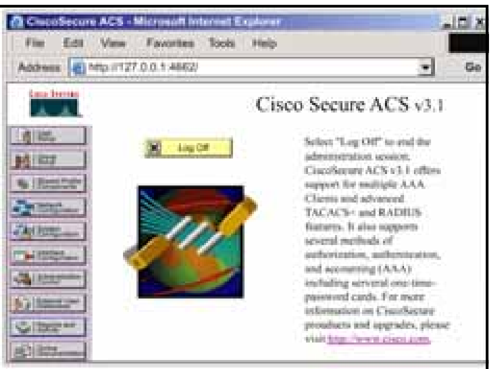
Comparison



TACACS+	RADIUS
Cisco-proprietary enhancement to original TACACS protocol	Open standard developed by Livingston Enterprises
Supports authentication, authorization, and accounting functions	Supports authentication, authorization, and accounting functions
Uses the AAA architecture, which separates authentication, authorization, and accounting	Combines the functions of authentication and authorization
Provides two ways to control the authorization of router commands: on a per-user or per-group basis	Does not allow administrators to control which commands can be executed on a router
Uses TCP	Uses UDP
Normal Operation will fully encrypt the body of the packet for more secure communications	Encrypts only the password in the access-request packet. Information such as username, authorization services, and accounting, could be captured by a third party
Offers multiprotocol support	Does not support ARA access, NetBIOS, NASI, or X.25 PAD connections

- Of the three protocols, **TACACS+ and RADIUS** offer the most **comprehensive AAA support**.
- Kerberos provides a highly secure method of authentication, in which passwords are never sent over the wire.
- However, **Kerberos does not support the authorization and accounting** components of AAA.
- Therefore, Kerberos will **not be covered** in any detail in this module.
- While the two protocols have much in common, there are several key differences between TACACS+ and RADIUS that set the two apart.

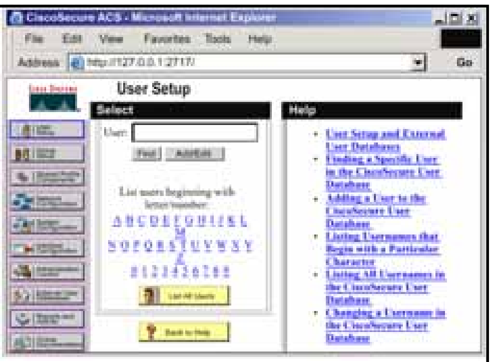
CiscoSecure Access Control Server


CiscoSecure ACS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://127.0.0.1:4662/
Go
Cisco Systems
Cisco Secure ACS v3.1
Log Off

Select "Log Off" to end the administration session. CiscoSecure ACS v3.1 offers support for multiple AAA Clients and advanced TACACS+ and RADIUS features. It also supports several methods of authorization, authentication, and accounting (AAA) including several one-time-password cards. For more information on CiscoSecure products and upgrades, please visit http://www.cisco.com.

- The CiscoSecure Access Control Server (ACS) is specialized security software that runs on Windows 2000.
- The software simplifies and centralizes access control and accounting for dialup access servers, virtual private networks (VPNs) and firewalls, voice-over-IP (VoIP) solutions, broadband access, content networks, and wireless networks.
- Cisco ACS uses a web-based graphical interface and can distribute the AAA information to hundreds or even thousands of access points in a
- network. The CiscoSecure ACS software uses either the TACACS+ or the RADIUS protocol to provide network security and tracking.

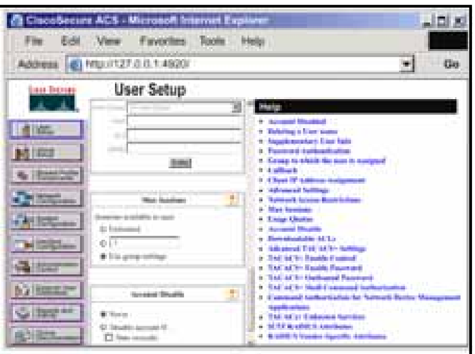
CiscoSecure Access Control Server


CiscoSecure ACS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://127.0.0.1:2717/
Go
Cisco Systems
User Setup
Select
Help
User:
Find
Add/Edit
List users beginning with letter/number:
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
List All Users
Back to Help

- User Setup and External User Databases
- Finding a Specific User in the CiscoSecure User Database
- Adding a User to the CiscoSecure User Database
- Listing Usernames that Begin with a Particular Character
- Listing All Usernames in the CiscoSecure User Database
- Changing a Username in the CiscoSecure User Database

- Using the Web-based interface, an administrator can log in to the CiscoSecure ACS database and easily:
  - create user accounts
  - group accounts
  - set passwords
  - configure access control

CiscoSecure Access Control Server


CiscoSecure ACS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://127.0.0.1:4920/
Go
User Setup
Help
Account Disabled
Deleting a User name
Supplementary User Info
Password Authentication
Group to which the user is assigned
Callback
Client IP Address Assignment
Advanced Settings
Network Access Restrictions
Max Sessions
Usage Quotas
Account Disable
Downloadable ACLs
Advanced TACACS+ Settings
TACACS+ Enable Control
TACACS+ Enable Password
TACACS+ Outbound Password
TACACS+ Shell Command Authorization
Command Authorization for Network Device Management Applications
TACACS+ Unknown Services
IETF RADIUS Attributes
RADIUS Vendor-Specific Attributes
Max Sessions
Account Disable

- Each of the devices on a network can be configured to communicate with ACS.
- Service providers can use ACS to centralize control of dialup access.
- With a CiscoSecure ACS, system administrators may use a variety of authentication methods that are aligned with a varying degree of authorization privileges.
- CiscoSecure ACS also acts as a central repository for accounting information.
- This accounting information can be used for billing, capacity planning, and security audits.

Configuring AAA

The aaa new-model command

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

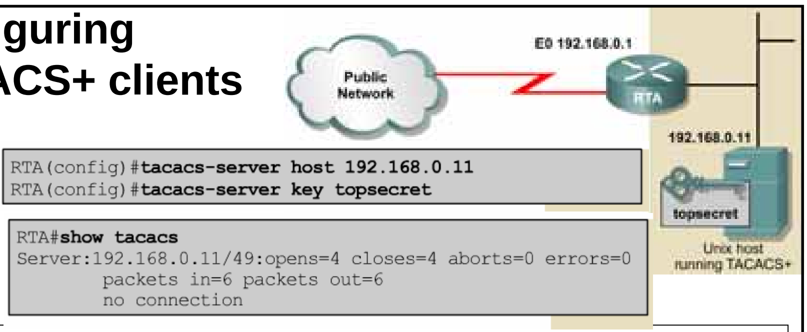
- AAA configurations can be complex.
- It is crucial that an organization plans out the security policies before beginning to configure AAA.
- The **`aaa new-model`** command enables the AAA feature so that other AAA commands can be entered.
- **Warning:** Do not issue the **`aaa new-model`** command without first preparing to configure AAA authentication.
- In some cases, just issuing this command will force Telnet users to authenticate with a username.
- This can happen even if no username database or authentication method is configured.
- Unless the router console is accessible, no one will be able to access the router.

The aaa new-model command

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

- As a technique, always configure the **local database before** issuing any AAA commands.
- The following sections describe how to configure the three elements of AAA using TACACS+, RADIUS, and local databases.
- **Suggestion**: If possible be physically at or nearby the router. Many times AAA has been misconfigured locking out the remote network administrator from making the corrections.

Configuring TACACS+ clients



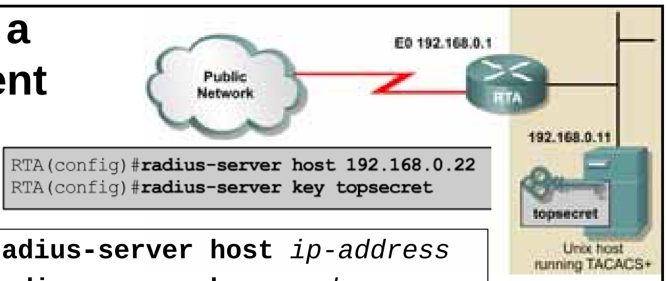
```
RTA(config)#tacacs-server host 192.168.0.11
RTA(config)#tacacs-server key topsecret
```

```
RTA#show tacacs
Server:192.168.0.11/49:opens=4 closes=4 aborts=0 errors=0
        packets in=6 packets out=6
        no connection
```

```
Router(config)#tacacs-server host ip-address
Router(config)#tacacs-server key word
```

- If a router uses a TACACS+ server or group of servers in combination with AAA, then the router must be configured with all the addresses.
- Searches for the hosts in the order in which they are specified.
- The router must also be configured with the TACACS+ **encryption key**.
- The key must be the **same on both** the TACACS+ server and its clients.
- In this simple TACACS+ configuration, a router is configured to communicate with a TACACS+ server at 192.168.0.11, using the shared key **`topsecret`**.

Configuring a RADIUS Client



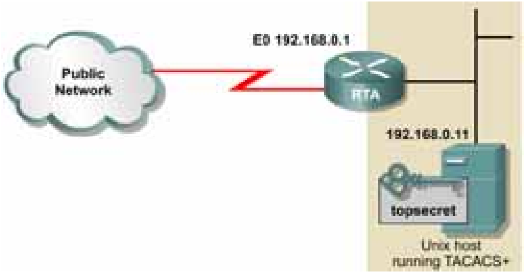
```
RTA(config)#radius-server host 192.168.0.22
RTA(config)#radius-server key topsecret
```

```
Router(config)#radius-server host ip-address
Router(config)#radius-server key word
```

- As with TACACS+, a server address and shared key must be configured for a router to use RADIUS.
- Searches for the hosts in the order in which they are specified.
- In this simple RADIUS configuration, a router is configured to communicate with a RADIUS server at 192.168.0.22, using the shared key **`topsecret`**.

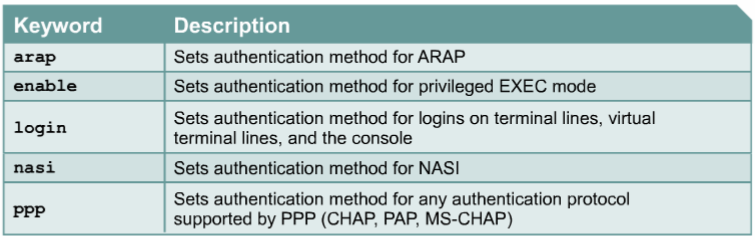
Configuring AAA authentication

Configuring AAA authentication



- There are several different types of authentication on a router.
  - When a router detects an incoming Telnet connection, the router authenticates.
  - When privileged EXEC mode is accessed, the router authenticates.
  - When a router detects an incoming PPP connection, the router authenticates.
  - A username and password that successfully authenticates for one type of access may not work for another.

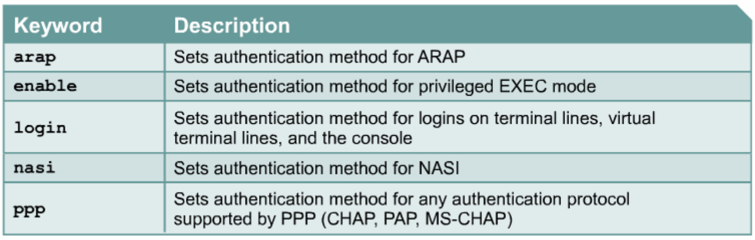
Configuring AAA authentication



Keyword	Description
`arap`	Sets authentication method for ARAP
`enable`	Sets authentication method for privileged EXEC mode
`login`	Sets authentication method for logins on terminal lines, virtual terminal lines, and the console
`nasi`	Sets authentication method for NASI
`ppp`	Sets authentication method for any authentication protocol supported by PPP (CHAP, PAP, MS-CHAP)

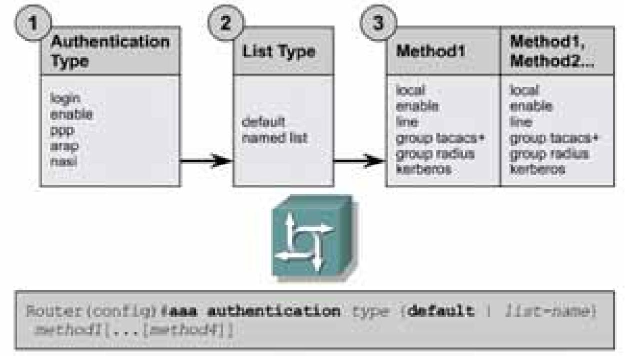
- AAA can be used to authenticate several types of connections, including the following:
  - Access to privileged EXEC mode (enable mode)
  - Access to virtual terminals
  - Access to the console
  - CHAP and PAP authentication for PPP connections
  - NetWare Asynchronous Services Interface (NASI) authentication
  - AppleTalk Remote Access Protocol (ARAP) authentication
- This module does not cover the legacy dialup protocols NASI or ARAP.

Configuring AAA authentication



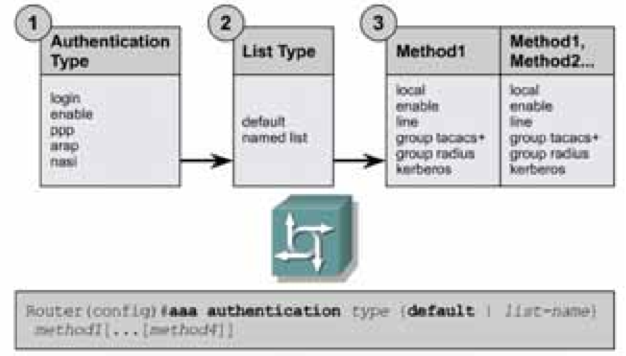
Keyword	Description
`arap`	Sets authentication method for ARAP
`enable`	Sets authentication method for privileged EXEC mode
`login`	Sets authentication method for logins on terminal lines, virtual terminal lines, and the console
`nasi`	Sets authentication method for NASI
`ppp`	Sets authentication method for any authentication protocol supported by PPP (CHAP, PAP, MS-CHAP)

- AAA authentication is configured with the **`aaa authentication`** command.
- When using this command, the type of authentication that is being configured must be specified as to whether it is login, enable, PPP, and so on.
- Once an authentication type has been specified, either a default method list or a named method list must be defined.



```
Router(config)#aaa authentication type {default | list-name}
 method1[...[method4]]
```

- These lists are called "**method lists**" because they **list the types of authentication to be performed and the sequence in which they will be performed.**
- Authentication methods include the following:
  - Using a password already configured on the router, such as the enable password or a line password
  - Using the local username/password database
  - Consulting a Kerberos server
  - Consulting a RADIUS server, or a group of RADIUS servers
  - Consulting a TACACS+ server or a group of TACACS+ servers

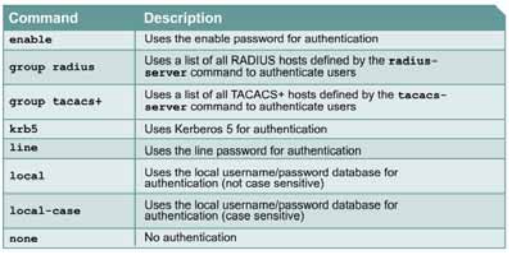


```
Router(config)#aaa authentication type {default | list-name}
 method1[...[method4]]
```

- AAA commands can be confusing because there are so many syntax possibilities.
- When configuring AAA authentication, use the following three-step process for each `aaa authentication` command.
  - **Step 1** Specify the **authentication type** whether login, enable, PPP, and so on.
  - **Step 2** Specify the **method list** as default or give it a name.
  - **Step 3** List, in order, the **authentication methods** to be tried.
- The sections that follow focus specifically on configuring three types of AAA authentication: login, enable, and PPP.

Configuring login authentication

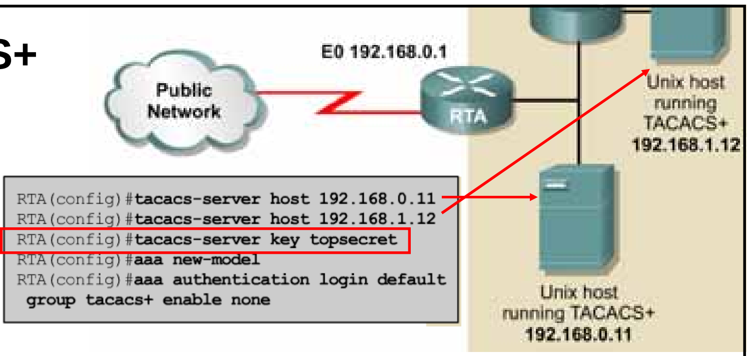
```
Router(config)#aaa authentication login {default | list-name}
   method1 [...[method4]]
```



Command	Description
`enable`	Uses the enable password for authentication
`group radius`	Uses a list of all RADIUS hosts defined by the **`radius-server`** command to authenticate users
`group tacacs+`	Uses a list of all TACACS+ hosts defined by the **`tacacs-server`** command to authenticate users
`krb5`	Uses Kerberos 5 for authentication
`line`	Uses the line password for authentication
`local`	Uses the local username/password database for authentication (not case sensitive)
`local-case`	Uses the local username/password database for authentication (case sensitive)
`none`	No authentication

- The **`aaa authentication login`** command enables AAA authentication for logins on terminal lines (TTYs), virtual terminal lines (VTYs), and the console (con 0).
- The **default list** is applied to all lines.
- A named list must be applied to a specific line or group of lines using the **`aaa login authentication`** command.

TACACS+

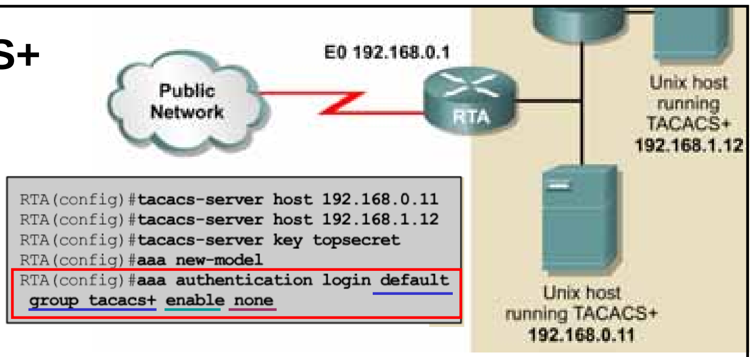


E0 192.168.0.1
Public Network
RTA
Unix host running TACACS+ 192.168.1.12
Unix host running TACACS+ 192.168.0.11

```
RTA(config)#tacacs-server host 192.168.0.11
RTA(config)#tacacs-server host 192.168.1.12
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default
 group tacacs+ enable none
```

- The **`aaa authentication login`** can be used together with the other AAA commands covered in this module to create and apply a default authentication list.
- Because this authentication method list specifies TACACS+ as the first method, the **`tacacs-server host`** and **`tacacs-server key`** commands are used to configure RTA as a TACACS+ client.
- Two TACACS+ servers are specified, 192.168.0.11 and 192.168.1.12. The server specified first, 192.168.0.11, is tried first.

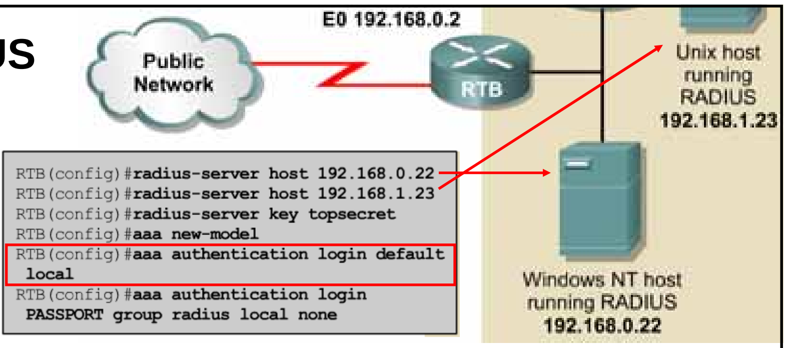
TACACS+


E0 192.168.0.1
Public Network
RTA
Unix host running TACACS+ 192.168.1.12
Unix host running TACACS+ 192.168.0.11

```
RTA(config)#tacacs-server host 192.168.0.11
RTA(config)#tacacs-server host 192.168.1.12
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default
  group tacacs+ enable none
```

- The **`aaa new-model`** command enables the AAA feature.
- Finally, the **`aaa authentication login`** command defines the method list.
- The method list configures RTA to attempt to **contact the TACACS+ servers first.**
- **If neither server is reached**, this method returns an ERROR and AAA tries to **use the second method**, the **enable** password.
- If this attempt also returns an ERROR, because no **enable** password is configured on the router, **the user is allowed access with no authentication.**

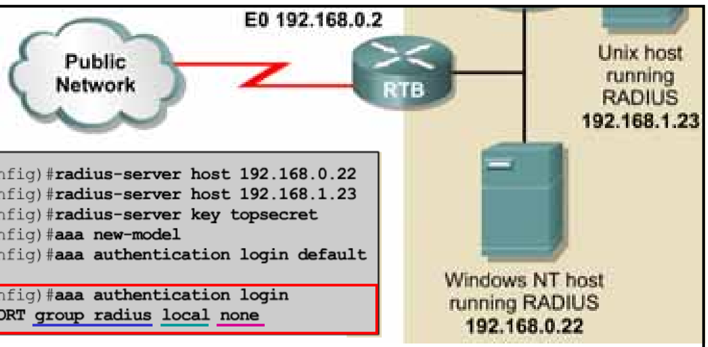
RADIUS


E0 192.168.0.2
Public Network
RTB
Unix host running RADIUS 192.168.1.23
Windows NT host running RADIUS 192.168.0.22

```
RTB(config)#radius-server host 192.168.0.22
RTB(config)#radius-server host 192.168.1.23
RTB(config)#radius-server key topsecret
RTB(config)#aaa new-model
RTB(config)#aaa authentication login default
 local
RTB(config)#aaa authentication login
 PASSPORT group radius local none
```

- The **default list** is applied to the console (con 0), all TTY lines including the auxiliary line or AUX port, and all VTY lines.
- To override the default method list, apply a named list to one or more of these lines.
- RTB is configured with the **`radius-server host`** and **`radius-server key`** commands because the named method list relies on RADIUS.
- The **`aaa authentication login default local`** command configures the default method as the local username/password database.
- This method is applied to all TTYs, VTYs, and the console by default.

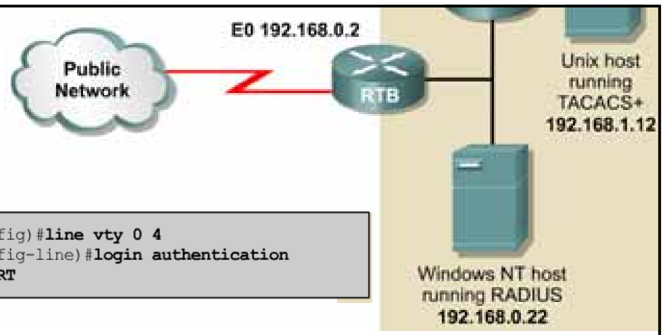
RADIUS


E0 192.168.0.2
Public Network
RTB
Unix host running RADIUS 192.168.1.23
Windows NT host running RADIUS 192.168.0.22

```
RTB(config)#radius-server host 192.168.0.22
RTB(config)#radius-server host 192.168.1.23
RTB(config)#radius-server key topsecret
RTB(config)#aaa new-model
RTB(config)#aaa authentication login default
 local
RTB(config)#aaa authentication login
 PASSPORT group radius local none
```

- The **`aaa authentication login PASSPORT group radius local none`** command creates a named method list called PASSPORT.
- The first method in this list is the **group of RADIUS servers**.
- If RTB cannot contact a RADIUS server, then RTB will **try and contact the local username/password database.**
- Finally, the **none** keyword assures that if no usernames exist in the local database, the user is granted access.

VTYs


E0 192.168.0.2
Public Network
RTB
Unix host running TACACS+ 192.168.1.12
Windows NT host running RADIUS 192.168.0.22

```
RTB(config)#line vty 0 4
RTB(config-line)#login authentication
 PASSPORT
```

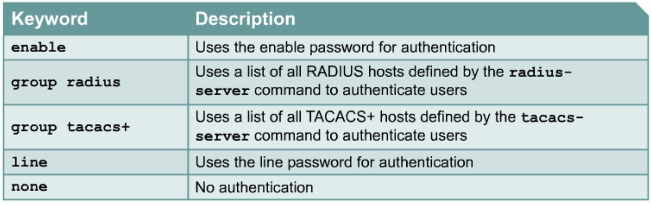
- Named method lists for login authentication are applied using the **`login authentication`** command.

  `Router(config-line)#`**`login authentication`** *`listname`*

- The **`login authentication`** command can be used to apply the PASSPORT method list to all five VTYs.

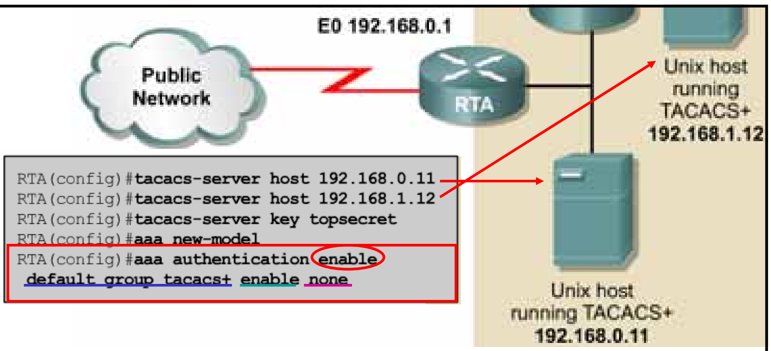
Enabling password protection at the privileged level

```
Router(config)#aaa authentication enable default method1
   [...[method4]]
```



Keyword	Description
`enable`	Uses the enable password for authentication
`group radius`	Uses a list of all RADIUS hosts defined by the **`radius-server`** command to authenticate users
`group tacacs+`	Uses a list of all TACACS+ hosts defined by the **`tacacs-server`** command to authenticate users
`line`	Uses the line password for authentication
`none`	No authentication

- The **`aaa authentication enable`** command enables AAA authentication for privileged EXEC mode access.
- This authentication method is applied when a user issues the **`enable`** command in user EXEC mode as follows.
- A named list cannot be specified with the **`aaa authentication enable`** command because authentication for privileged EXEC mode is the same for all users on all lines.
- The default list is the only privileged mode method list that can exist.
- Therefore, the privileged mode method list does not need to be applied to a line or interface.

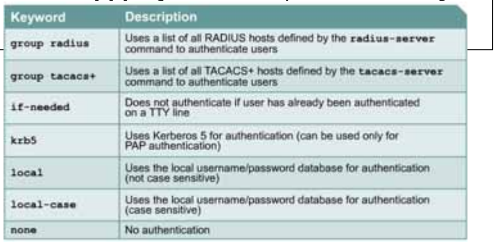

E0 192.168.0.1
Public Network
RTA
Unix host running TACACS+ 192.168.1.12
Unix host running TACACS+ 192.168.0.11

```
RTA(config)#tacacs-server host 192.168.0.11
RTA(config)#tacacs-server host 192.168.1.12
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication enable
 default group tacacs+ enable none
```

- The commands create a method list that **first tries to contact a TACACS+ server.**
- If neither server can be contacted, AAA tries to use the **enable** password.
- This attempt may return an error because no enable password is configured on RTA.
- However, the **user is allowed access with no authentication**.

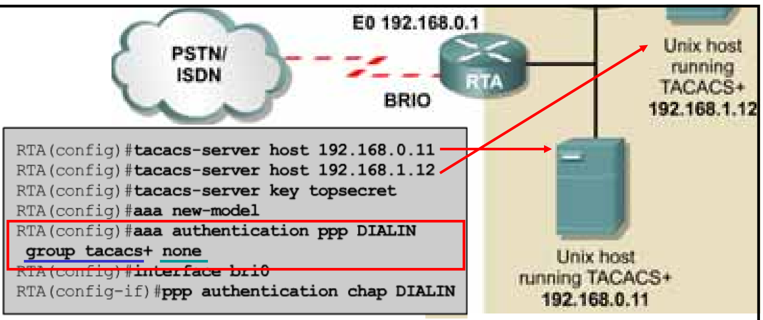
Configuring PPP authentication using AAA

```
Router(config)#aaa authentication ppp {default | list-name}
  method1 [...[method4]]
```



Keyword	Description
`group radius`	Uses a list of all RADIUS hosts defined by the **radius-server** command to authenticate users
`group tacacs+`	Uses a list of all TACACS+ hosts defined by the **tacacs-server** command to authenticate users
`if-needed`	Does not authenticate if user has already been authenticated on a TTY line
`krb5`	Uses Kerberos 5 for authentication (can be used only for PAP authentication)
`local`	Uses the local username/password database for authentication (not case sensitive)
`local-case`	Uses the local username/password database for authentication (case sensitive)
`none`	No authentication

- Many **remote users** access networks through a router using PPP over asynchronous dialup or ISDN BRI.
- These remote users can completely bypass the command line interface on that router.
- Instead, PPP starts a packet session as soon as the connection is established.
- A router can be configured to use AAA with the **aaa authentication ppp** command to authenticate these users.
- The router uses any available PPP authentication method such as CHAP, PAP or MS-CHAP.



PSTN/ ISDN
E0 192.168.0.1
RTA
BRIO
Unix host running TACACS+ 192.168.1.12
Unix host running TACACS+ 192.168.0.11

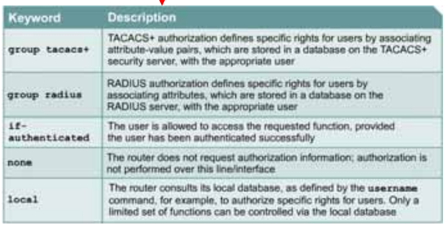
```
RTA(config)#tacacs-server host 192.168.0.11
RTA(config)#tacacs-server host 192.168.1.12
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication ppp DIALIN
 group tacacs+ none
RTA(config)#interface bri0
RTA(config-if)#ppp authentication chap DIALIN
```

- Remember to specify **none** as the final method in the method list to have authentication succeed even if all methods return an error.
- The **aaa authentication ppp** and **ppp authentication chap** commands can be used to apply a named AAA authentication list called DIALIN for PPP.
- This authentication method list **first tries to contact a TACACS+ server**.
- If this action returns an error, the user is allowed access with no authentication.
- The **ppp authentication** command is used to apply an AAA authentication method list for PPP.
- In this example, CHAP authentication will use the method list.

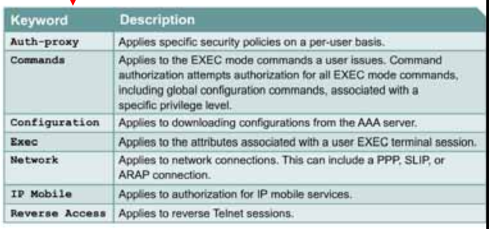
Configuring AAA authorization

Configuring AAA authorization

```
Router(config)#aaa authorization type {default | list-name}
   [method1 [...[method4]]
```



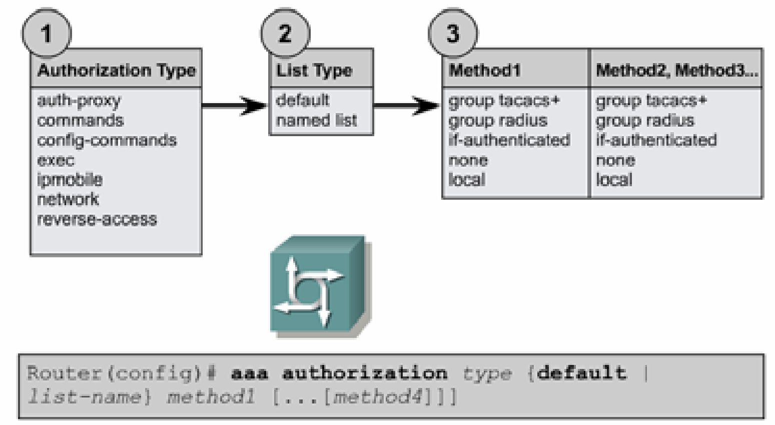
Keyword	Description
`group tacacs+`	TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user
`group radius`	RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user
`if-authenticated`	The user is allowed to access the requested function, provided the user has been authenticated successfully
`none`	The router does not request authorization information; authorization is not performed over this line/interface
`local`	The router consults its local database, as defined by the **username** command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database



Keyword	Description
`Auth-proxy`	Applies specific security policies on a per-user basis.
`Commands`	Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
`Configuration`	Applies to downloading configurations from the AAA server.
`Exec`	Applies to the attributes associated with a user EXEC terminal session.
`Network`	Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
`IP Mobile`	Applies to authorization for IP mobile services.
`Reverse Access`	Applies to reverse Telnet sessions.

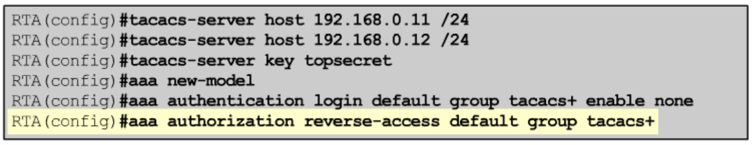
- **AAA authorization** limits the services available to a user.
- When AAA authorization is enabled, the router uses information retrieved from the user's profile to configure the session.
- This profile is located either in the local user database or on the security server.
- Once this authorization is done, the user will be granted access to a requested service only if the information in the user profile will allow it.

Configuring AAA authorization



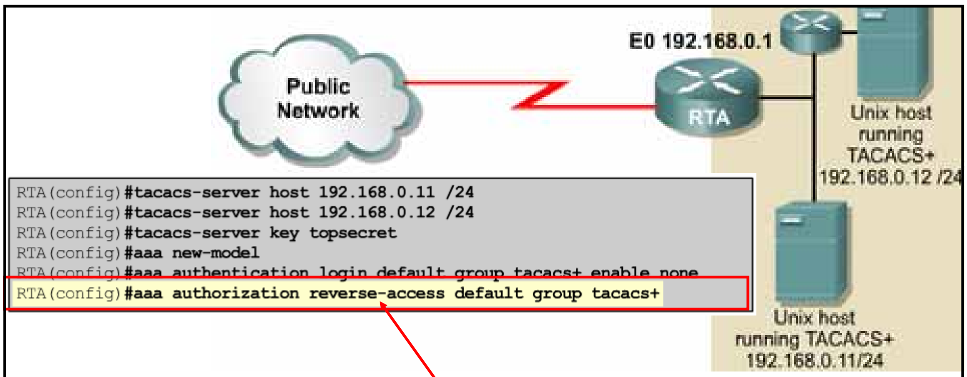
```
Router(config)# aaa authorization type {default |
list-name} method1 [...[method4]]]
```

Configuring AAA authorization



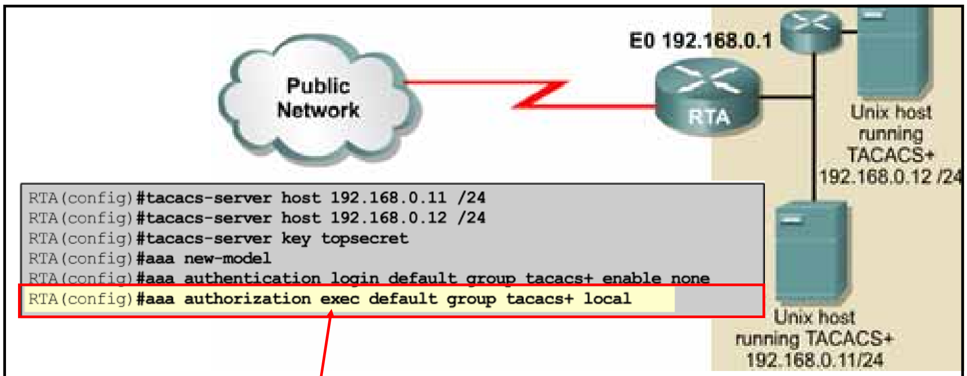
```
RTA(config)#tacacs-server host 192.168.0.11 /24
RTA(config)#tacacs-server host 192.168.0.12 /24
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default group tacacs+ enable none
RTA(config)#aaa authorization reverse-access default group tacacs+
```

- Before AAA authorization can be configured, the following tasks must be performed:
  - **Enable AAA** using the **`aaa new-model`** command.
  - **Configure AAA authentication.** Authorization generally takes place after authentication and it relies on authentication to work properly.
  - **Configure the router as a TACACS+ or a RADIUS client**, if necessary.
  - **Configure the local username/password database, if necessary**.
    - Use the **`username`** command to define the rights associated with specific use


Public Network
E0 192.168.0.1
RTA
Unix host running TACACS+ 192.168.0.12 /24
Unix host running TACACS+ 192.168.0.11/24

```
RTA(config)#tacacs-server host 192.168.0.11 /24
RTA(config)#tacacs-server host 192.168.0.12 /24
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default group tacacs+ enable none
RTA(config)#aaa authorization reverse-access default group tacacs+
```

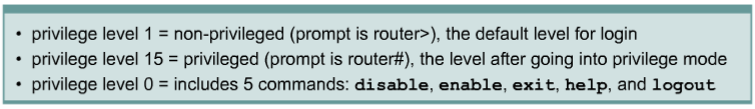
- The **`aaa authorization reverse-access`** command configures authorization for reverse Telnet sessions.
- Users attempting to reverse Telnet from the router must be authorized to issue the command first by a TACACS+ server.


Public Network
E0 192.168.0.1
RTA
Unix host running TACACS+ 192.168.0.12 /24
Unix host running TACACS+ 192.168.0.11/24

```
RTA(config)#tacacs-server host 192.168.0.11 /24
RTA(config)#tacacs-server host 192.168.0.12 /24
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default group tacacs+ enable none
RTA(config)#aaa authorization exec default group tacacs+ local
```

- The **`aaa authorization exec`** command configures authorization for EXEC sessions.
- The router will contact a TACACS+ server to determine if users are permitted to start an EXEC shell when they log in.

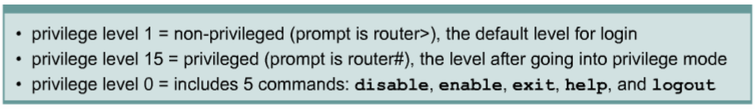
IOS command privilege levels



- privilege level 1 = non-privileged (prompt is router>), the default level for login
- privilege level 15 = privileged (prompt is router#), the level after going into privilege mode
- privilege level 0 = includes 5 commands: **disable**, **enable**, **exit**, **help**, and **logout**

- The **aaa authorization** command can also be used to control exactly which commands a user is allowed to enter on the router.
- Users can only enter commands at or beneath their privilege level.
- All **IOS router commands** are assigned a privilege level from 0 to 15.
  - There are **three privilege levels** on the router by default.
- Routers use privilege levels even when AAA is not configured.
- When a user opens an EXEC session using the console or a VTY, the user can issue any command in **privilege level 1** and/or **privilege level 0** by **default**.
  - **privilege level 1** – user mode
  - **privilege level 15** – priviledged (enable) mode
- Once the user authenticates using the **enable** command and **enable** password, that user has **privilege level 15**.

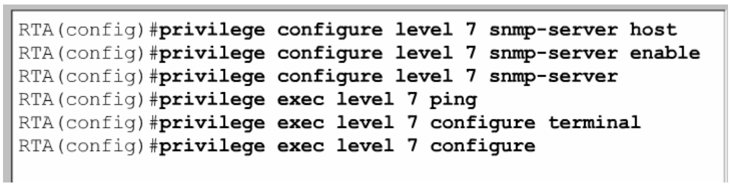
IOS command privilege levels



- privilege level 1 = non-privileged (prompt is router>), the default level for login
- privilege level 15 = privileged (prompt is router#), the level after going into privilege mode
- privilege level 0 = includes 5 commands: **disable**, **enable**, **exit**, **help**, and **logout**

- **Levels 2 to 14** are **not** used in a default configuration.
- However, commands that are normally at level 15 can be moved down to any level between 2 and 14.
- Commands that are normally at level 1 can be moved up to one of those levels.
- This security model involves some administration on the router.
- To determine the privilege level as a logged in user, the **show privilege** command is used.
- The commands that are available at a particular privilege level for the Cisco IOS Software Release being used can be determined.
- Enter a "**?**" at the command line when logged in at that privilege level to show those commands.
- **Note:** Instead of assigning privilege levels, command authorization can be done if the authentication server supports TACACS+. The RADIUS protocol does not support command authorization.

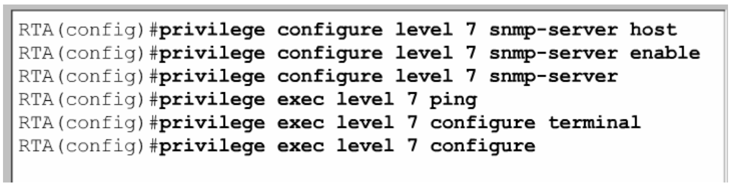
Configuring command authorization



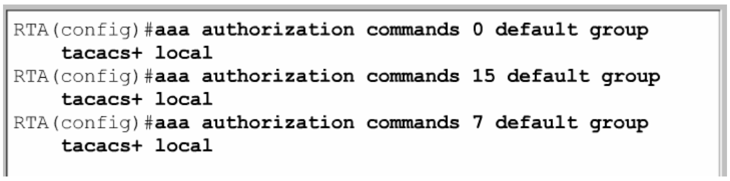
```
RTA(config)#privilege configure level 7 snmp-server host
RTA(config)#privilege configure level 7 snmp-server enable
RTA(config)#privilege configure level 7 snmp-server
RTA(config)#privilege exec level 7 ping
RTA(config)#privilege exec level 7 configure terminal
RTA(config)#privilege exec level 7 configure
```

- The **`privilege`** command can be used to configure precisely which commands belong to which privilege levels, including user-defined levels.
- The commands entered on RTA move the **`snmp-server`** commands from privilege level 15, the default, to privilege level 7.
- The **`ping`** command is moved up from privilege level 1 to privilege level 7

Configuring command authorization

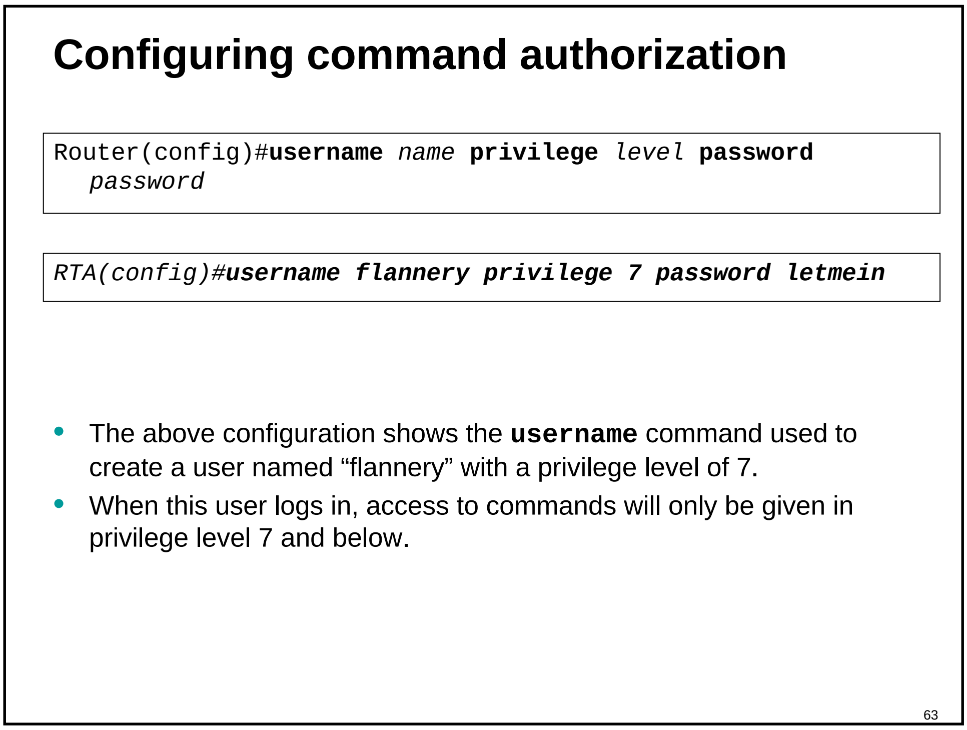


```
RTA(config)#privilege configure level 7 snmp-server host
RTA(config)#privilege configure level 7 snmp-server enable
RTA(config)#privilege configure level 7 snmp-server
RTA(config)#privilege exec level 7 ping
RTA(config)#privilege exec level 7 configure terminal
RTA(config)#privilege exec level 7 configure
```



```
RTA(config)#aaa authorization commands 0 default group
    tacacs+ local
RTA(config)#aaa authorization commands 15 default group
    tacacs+ local
RTA(config)#aaa authorization commands 7 default group
    tacacs+ local
```

- Once privilege levels have been defined, the **`aaa authorization`** command can be used to give access to commands by privilege level.
- The user who logs in with level 7 privileges can **`ping`** and do **`snmp-server`** configuration in configuration mode.
- Other configuration commands are not available.
- The security server or the local username/password database can determine a user’s privilege level.

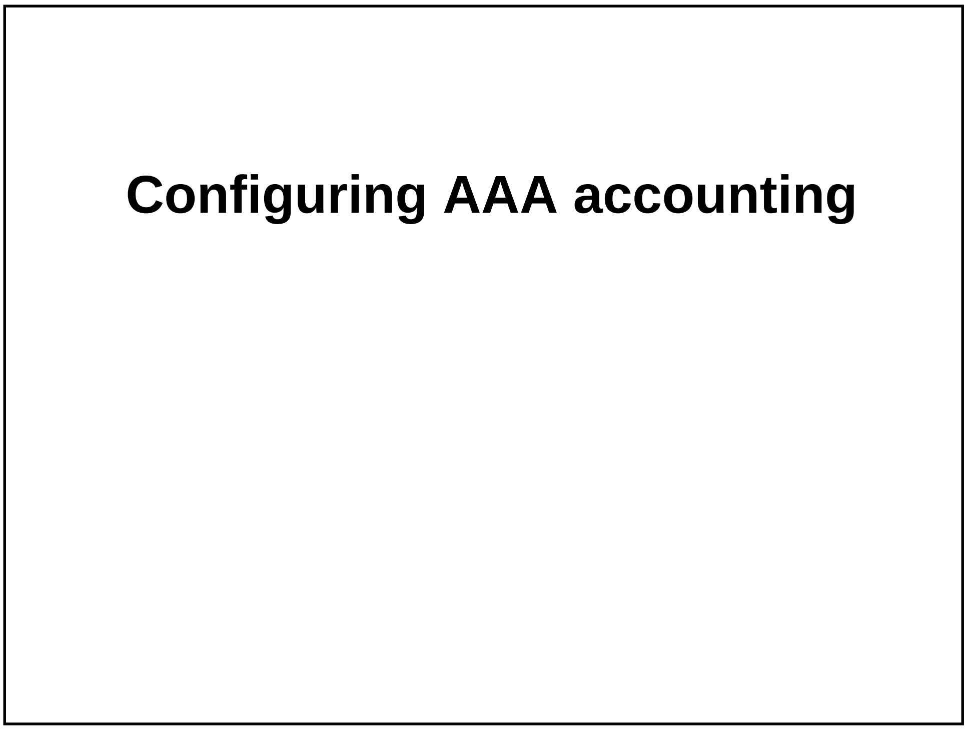


Configuring command authorization

```
Router(config)#username name privilege level password
   password
```

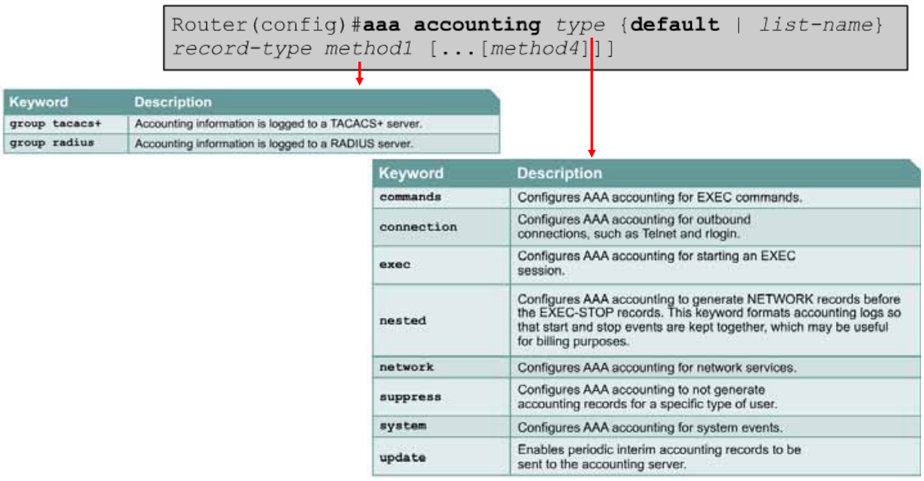
```
RTA(config)#username flannery privilege 7 password letmein
```

- The above configuration shows the **username** command used to create a user named “flannery” with a privilege level of 7.
- When this user logs in, access to commands will only be given in privilege level 7 and below.



Configuring AAA accounting

Configuring AAA accounting



```
Router(config)#aaa accounting type {default | list-name}
record-type method1 [...[method4]]]
```

Keyword	Description
`group tacacs+`	Accounting information is logged to a TACACS+ server.
`group radius`	Accounting information is logged to a RADIUS server.

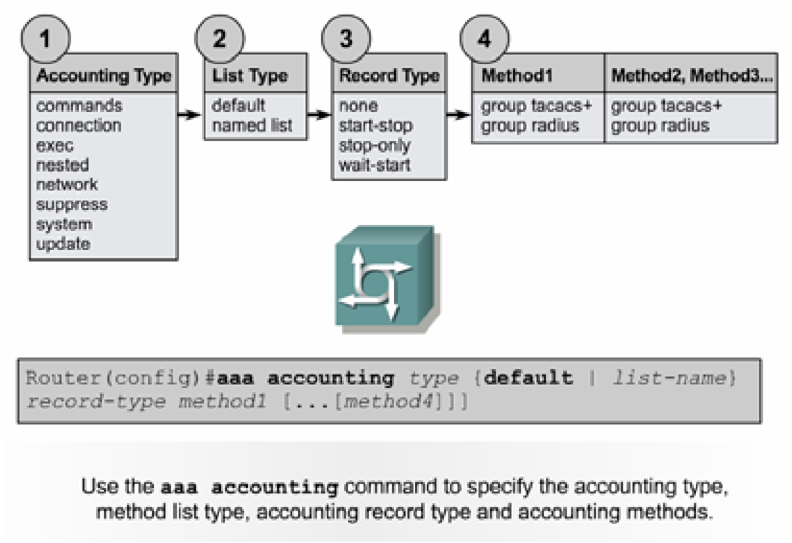
Keyword	Description
`commands`	Configures AAA accounting for EXEC commands.
`connection`	Configures AAA accounting for outbound connections, such as Telnet and rlogin.
`exec`	Configures AAA accounting for starting an EXEC session.
`nested`	Configures AAA accounting to generate NETWORK records before the EXEC-STOP records. This keyword formats accounting logs so that start and stop events are kept together, which may be useful for billing purposes.
`network`	Configures AAA accounting for network services.
`suppress`	Configures AAA accounting to not generate accounting records for a specific type of user.
`system`	Configures AAA accounting for system events.
`update`	Enables periodic interim accounting records to be sent to the accounting server.

- Method lists for accounting define the way accounting will be performed and the sequence in which these methods are performed.

Configuring AAA accounting

- Accounting method lists are specific to the type of accounting being requested.
- AAA supports the follow six different types of accounting.
  - Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
  - EXEC accounting provides information about user EXEC terminal sessions of the network access server.
  - Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
  - Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
  - System accounting provides information about system-level events.
  - Resource accounting provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.

Configuring AAA accounting



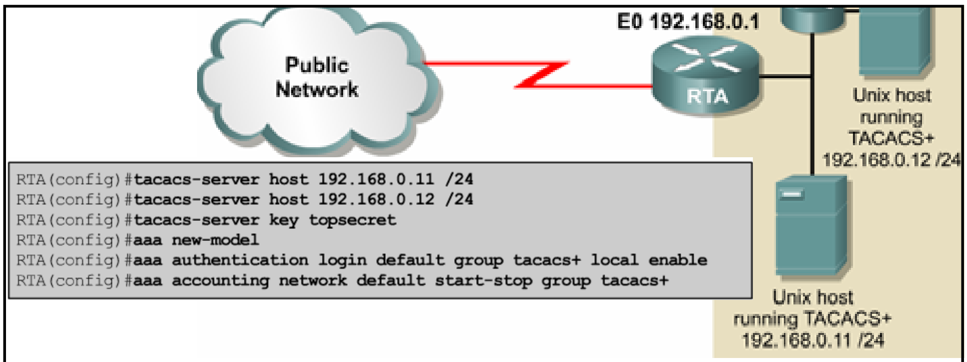
1 Accounting Type	2 List Type	3 Record Type	4 Method1	Method2, Method3...
commands connection exec nested network suppress system update	default named list	none start-stop stop-only wait-start	group tacacs+ group radius	group tacacs+ group radius

```
Router(config)#aaa accounting type {default | list-name}
record-type method1 [...[method4]]]
```

Use the **aaa accounting** command to specify the accounting type, method list type, accounting record type and accounting methods.

Configuring AAA accounting

- After specifying a named or default list, the accounting record type must be specified. The following are the four accounting record types:
  - none
  - start-stop
  - stop-only
  - wait-start
- For minimal accounting, use the **stop-only** keyword.
- This keyword instructs the specified method, RADIUS or TACACS+, to send a stop record accounting notice at the end of the requested user process.
- For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event.
- Wait-start sends both a start and a stop accounting record to the accounting server.
- However, if the **wait-start** keyword is used, the requested user service does not begin until the start accounting record is acknowledged.
- A stop accounting record is also sent.
- To stop all accounting activities on this line or interface, use the **none** keyword.


Public Network
E0 192.168.0.1
RTA
Unix host running TACACS+ 192.168.0.12 /24
Unix host running TACACS+ 192.168.0.11 /24

```
RTA(config)#tacacs-server host 192.168.0.11 /24
RTA(config)#tacacs-server host 192.168.0.12 /24
RTA(config)#tacacs-server key topsecret
RTA(config)#aaa new-model
RTA(config)#aaa authentication login default group tacacs+ local enable
RTA(config)#aaa accounting network default start-stop group tacacs+
```

- RTA is configured with the **`aaa accounting network`** command.
- This command enables accounting for network services, such as PPP, SLIP, and ARAP sessions.
- RTA will send accounting information for PPP sessions to a TACACS+ server.
- The format of the output stored on the server varies depending on the TACACS+ or RADIUS implementation.