Module 3...

IOS Message Logging

>> Log information is simply status data, such as changes in the router's interface status, modifications to running configuration, and debugging output.

>> Less useful when everything is smooth. But valuable when a problem comes up.

>> IOS uses UNIX's syslog logging system.

>> IOS provides four methods for viewing logging information:

1) Console—The router's console port
2) Monitor—The router's system monitor, a VTY "console" message display
3) Trap (Syslog logging)—Output to a remote syslog server running on UNIX or NT
4) Buffer—A place to store a list of logging events in the router's DRAM

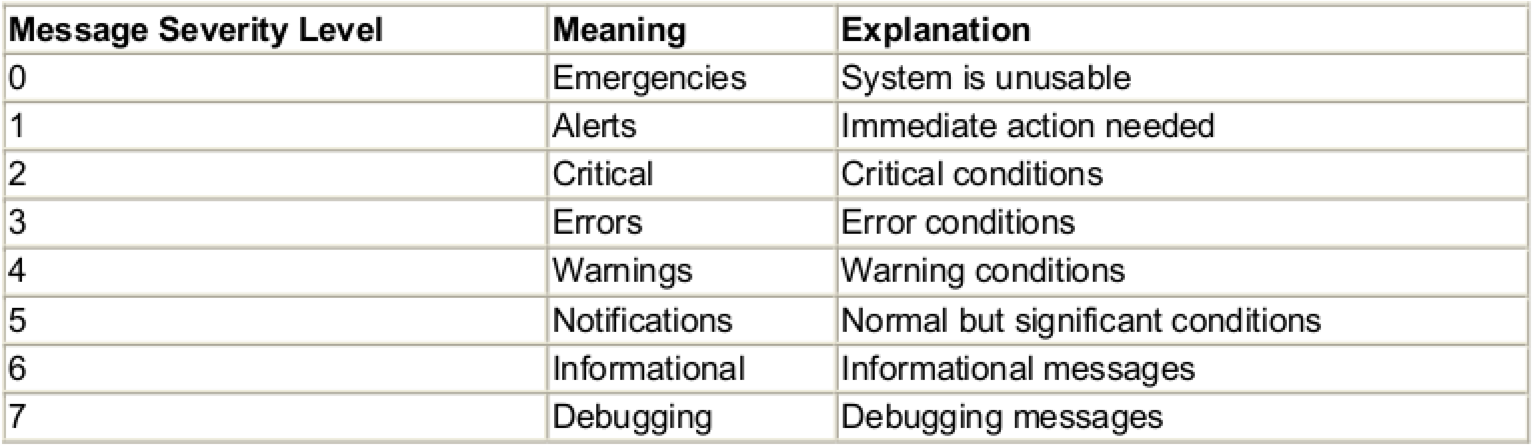
>> By default, all console messages are enabled, monitor and buffered is disabled, and trap needs to be configured to know where to send its messages.

IOS Message Logging...

>> Messages will have the common format given below.

```
[seq no:timestamp:] %facility-severity-MNEMONIC:description
```

>> Cisco uses the syslog level classification to define the severity of logging messages:



Message Severity Level	Meaning	Explanation
0	Emergencies	System is unusable
1	Alerts	Immediate action needed
2	Critical	Critical conditions
3	Errors	Error conditions
4	Warnings	Warning conditions
5	Notifications	Normal but significant conditions
6	Informational	Informational messages
7	Debugging	Debugging messages

>> Eg: (for console message) Whenever an interface is turned on or off (*shutdown* or *no shutdown* commands).

```
Router(config-if)#shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
```

>> The logging information can be seen at the Privileged mode by using the command <*show logging*>

>> Terminal messages can be enabled by the privileged mode command <*terminal monitor*> and disabled by <*terminal no monitor*> command.

Setting Up Buffered Logging

>> Buffer system is less efficient because the data is lost when the power is off, and buffering reduces system performance.

>> First, eneble buffered logging.

>> Can be enabled in global config mode as follows;

```
Router(config)#logging on
Router(config)#logging buffered 64000
```

Here 64000 refers the maximum buffer (DRAM) size to be used for logging.

>> <*logging history size n*> command was used earlier to set the number of messages to be stored (here n; maximum 500).

>> To view logging information, use the privileged EXEC command <*show logging*>*;* and the last line shows the buffer size.

>> The buffer logfile is a rotating one, so once the message count has reached its limit, it starts to overwrite itself, deleting the old message in the file as new messages are added.

Setting Up Syslog (trap) Logging

>> First configure a host to process the syslog messages.

Router(config)#logging 192.168.0.2

>> This will enable trap logging.

>> But the destination system must have the service *syslog*. This is done by starting the syslogd daemon and setting the */etc/syslog.conf* file.

>> Each message has a facility level and severity level associated with it.

>> Besed on the severity value, a syslog-defined action can be done. This can be defined in thr *conf* file.

>> The actions are;

1) Log to a file (/usr/log/<filename>)
2) Forward the message to another syslog process on another host (@hostname or @ip address)
3) Write the message to a specified user's operator window. (user,username)
4) Write the message to all users'operator windows (*)

>> Log messages with a high severity level are sent to the master syslog server and to users'operator windows, whereas general router information and debug messages are sent to files.

IOS Authentication and Accounting

>> IOS supports two modes of system authentication: **old-mode** and **new-mode.**

>> Old-mode uses a *static, locally stored user table* or *Terminal Access Controller Access Control System (TACACS)* for user authentication.

>> TACACS is a security protocol suite in UNIX for providing remote user authentication.

>> The service is provided by running a service or process daemon on a Windows NT or UNIX system.

>> There are three versions of TACACS supported by IOS:

- TACACS (available in old-mode)
- Extended TACACS (XTACACS. Available in old-mode and new-mode)
- TACACS+ (CiscoSecure 2.x) (By Cisco. Available only in new-mode and preferred by Cisco)

>> New-mode is also referred to as AAA (authentication, authorization and accounting).

>> AAA also facilitates IOS command accounting and authorization for restricted EXEC command access.

>> AAA also provides the capability to use Remote Authentication Dial-In User Service (RADIUS) and Kerberos V5 (a trusted authentication system) as an alternative to TACACS.

IOS Authentication and Accounting...

>> **Kerberos**: Both users and systems must have an identity key, known as a *principle*, stored on a centralized authentication and database server, known as the **Key Distribution Center (KDC).** All access verification questions are deferred to the KDC (which is why Kerberos is called a "trusted third-party" system). The collection of users and systems that trust the Kerberos KDC to authenticate is known as a **Kerberos realm**.

How Kerberos Work?

1) A user logs in, and a request is sent to the KDC, where the user's access rights are verified.
2) If the user is valid, a ticket-granting ticket (TGT) and a session key (SK) (which is used as a one-time password [OTP] to decrypt the TGT) are sent back in encrypted form, using the user's password as the source of the encryption key. The user is now authenticated locally.
3) To get a ticket (which has a limited life span), the user generates a ticket request for the remote host it wants to access. The TGT and an authenticator are sent to the Ticket Granting Server (TGS). The request is verified, and a ticket and session key (an OTP to decrypt the ticket) for the requested system are returned.
4) After the ticket is acquired, the user logs in to the system. The system then verifies that the ticket and authenticator match, and if so, access is permitted.

Using Old-Mode Authentication: Local

>> By default, the IOS uses AAA old-mode authentication.

>> VTY username and password creation for telnet...

```
Router(config)#line vty 0 4
Router(config-line)#login local
Router(config)#
Router(config)#username admin password 1234
```

>> Now vty connection will ask for username and password.

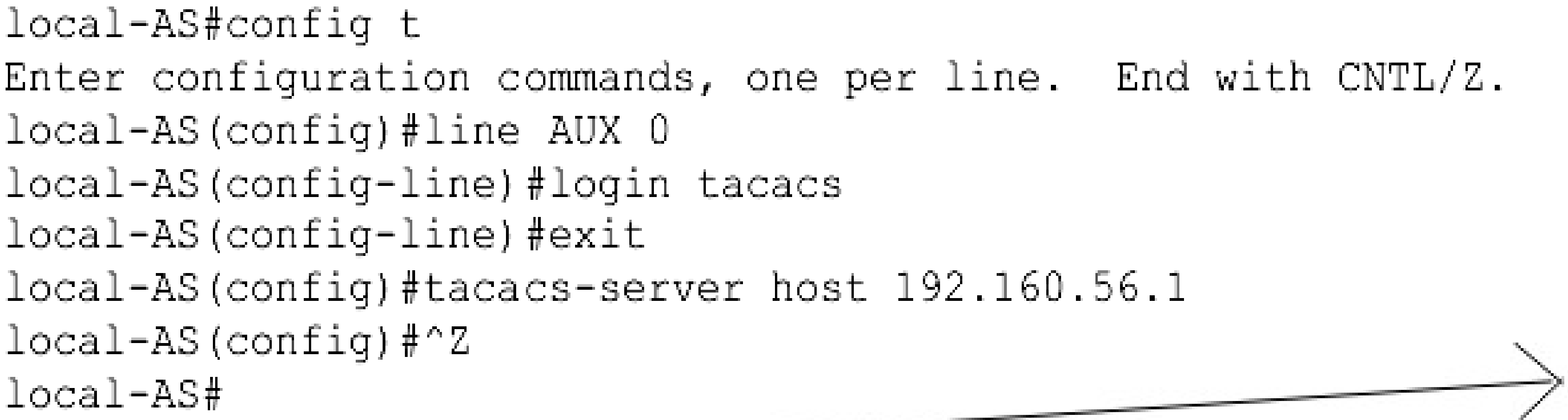
```
PC>telnet 192.168.1.1
Trying 192.168.1.1 ...Open
User Access Verification
Username: user
Password:
Router>
```

Using Old-Mode Authentication: TACACS server

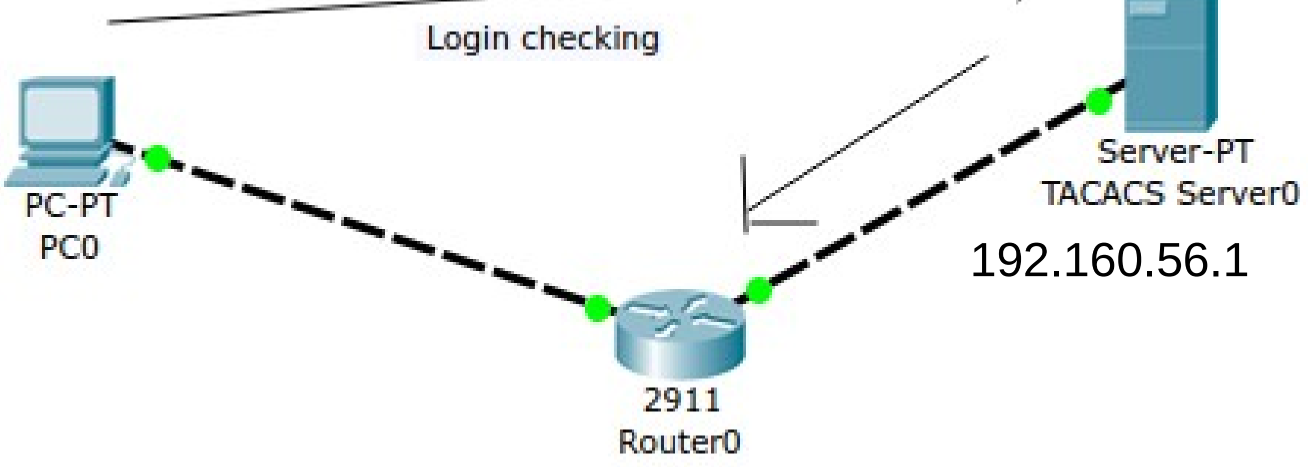
>> A TACACS server must be configured in the network for this router and it must hold all the user names and passwords of the users who need to get access to the router.

>> A Key should be defined in the TACACS server so that the server can use it for router authentication.

>> The IP address of the TACACS server must be defined in the router and TACACS authentication has to be enabled.



```
local-AS#config t
Enter configuration commands, one per line.  End with CNTL/Z.
local-AS(config)#line AUX 0
local-AS(config-line)#login tacacs
local-AS(config-line)#exit
local-AS(config)#tacacs-server host 192.160.56.1
local-AS(config)#^Z
local-AS#
```



Using New-Mode Authentication: AAA

>> New-mode authentication supports both a local static user table and the security authentication protocols.

>> Privileged EXEC and configuration EXEC modes authenticated remotely.

>> It can apply authentication to all line interfaces (including the console) when using the "default" list.

>> To enable AAA mode, use the command <*aaa new-model*>

Setting Up Login Authentication

>> Always create the local administrative account first. Then, enable <aaa new-model>. After AAA has been enabled, the authentication service type and list are defined:

```
Router(config)#username student password student
Router(config)#aaa new-model
Router(config)#aaa authentication login default group radius local
```

Here the system first use RADIUS for authentication. If it is not available, the system will use local authentication.

>> The radius server and key for authentication is defined by,

```
Router0(config)#radius-server host 10.0.0.10 key abcd
```

Similary, TACACS+ server is defined by

```
Router0(config)#tacacs-server host 10.0.0.12 key pqrs
```

Configuring Accounting

>> Accounting is used to track service abuse and login failures, generate usage statistics, and so on.

>> There are two types of IOS accounting: user and operations.

>> User Accounting: IOS provides access information on network sessions (such as PPP) and outbound connections (such as Telnet, rlogin).

>> Operational Accounting: Tracks the Information pertaining to router-centric activities.

>> Opeartional Accounting can be of two types:

<*system*> accounting—Provides system event information (similar to logging information).

<*command*> accounting—Keeps track of EXEC shell commands usage>

>> IOS accounting requires a TACACS+ or RADIUS server to process the accounting records and the client must be configured with <*aaa new-model*>.

>> The records are collections of attribute-value pairs.

>> TACACS+ server helps to collect "system accounting" information about the router by using <system> and <command> options.

>> RADIUS server helps to account only network, outbound connections, and EXEC sessions.

Configuring Accounting...

>> There are three accounting service commands;

- <start-stop> Record start and stop without waiting.

  Sends a notice to the accounting server when the session starts and ends.
- <wait-start> performs the same function as <start-stop>, except the actual session does not start until the accounting notice is acknowledged by the accounting server.
- <stop-only> Record stop when service terminates.

  Provides the most basic accounting information and only sends a notice when the session is completed.

>> The <show accounting> privileged EXEC command shows all the active sessions on the router.