Module 4 – Essential Router Configuration

Objective: To configure VTY filters, SSH, Out of Band Access, Network Time Protocol, TACACS+ and other Essential Features on the ISP Workshop Lab Network.

Prerequisite: Module 1, the IOS Essentials presentation, and Module 2

This module is an optional module which can be inserted into the ISP/IXP Workshop programme at any stage after the completion of Module One. However, it assumes aspects of the network design used in the OSPF Area module, and the BGP Route Reflector module.

The aim of the module is to introduce the student to certain essential aspects of Cisco's IOS which are not introduced as part of the standard modules. Most features introduced elsewhere in the lab are involved with routing – however, IOS has a rich set of features which are often overlooked, but required, in an ISP's network.

The structure of this module will follow the one worked on in the Advanced OSPF Module, and the Route Reflector Module. It is a typical design of ISP backbones, and gives good hints about how to configure Out of Band Management, structure an NTP hierarchy in the network, and deploy TACACS+ for system management access.

1. **VTY Security.** The first step is to demonstrate how to secure the vtys on a router. This is the most often forgotten security feature on service provider routers, and creates a huge security risk for these devices when connected to the public Internet. The vtys on the router will now be secured so that only connections from recognised addresses will be permitted.

2. **VTY Security – telnet source address.** The first step in doing this is to configure telnet so that it uses the loopback interface as the source address for all telnet packets originated by the router.

   ```
   ip telnet source-interface loopback 0
   ```

   To check that this has worked, telnet from your router to a neighbouring router and then enter the "who" command. You will see that you are logged in, and the source address will be displayed. For example, using telnet from Router1 to Router3 gives:

   ```
   Router3>who
       Line     User      Host(s)                  Idle Location
   *  2 vty 0   philip    idle                 00:00:00 10.0.15.241
   ```

3. **VTY Security – constructing an access-list.** Next, an access-list with all the loopback interfaces in the lab is constructed. Given we carefully constructed our addressing plan in Module 1 to ensure that the loopbacks all fitted into a single contiguous range of addresses, this access-list might be something like:

   ```
   access-list 3 permit 10.0.15.240 0.0.0.15 ! Router loopback range
   access-list 3 permit host 192.168.1.4     ! Workshop laptop
   access-list 3 deny any
   ```

   which covers all routers in the workshop lab and the workshop laptop (nameserver etc).

4. **VTY Security – applying the filter to the vtys.** Now that the access-list has been constructed it can be applied to the vtys on the router. To do this, use the access-class command:

```
line vty 0 4
 access-class 3 in
```

5. **Checking the filters.** Try using telnet to connect to a neighbouring router. Do you still get access? If not, why not? (If you don't, then the access-list is either wrong, or you have forgotten to do the step which set telnet source address to be that of the loopback interface.) Now remove the "ip telnet source-interface loopback 0" command set earlier and try using telnet to connect to a neighbouring router. If that team has implemented the access-list you will now find that access is barred.

   It is **strongly** recommended that all ISPs implement vty filters on **all** their routers in their network. Absence of vty filters has usually led to service provider networks being compromised by intruders.

   A final point: on the IOS images used in this workshop, there are 5 vtys available. On some releases or versions of software with extended features in them, a greater number of vtys may be available. Often these are not displayed in the configuration. It is worth checking how many vtys are supported by your version of IOS when you are configuring these filters. Only protecting the first 5 is not good security – people attempting to break into such a router will occupy the first 5 vtys, then gain a login prompt on the 6th vty. To check for extra vtys on your router, enter the following in configuration mode:

```
Router1#conf t
Router1(config)#line vty 0 ?
  <1-63>  Last Line number
  <cr>

Router1(config)#line vty 0 63
Router1(config-line)#access-class 3 in
```

   The **?** option will display how many vtys the router supports. When applying the access-list, make sure that all vtys are included as per the example above.

6. **VTY Transport support.** The final step is to change the transports permitted on the vtys from the defaults in IOS to those which are applicable to a service provider backbone. We will also change the supported transports on the console port and the auxiliary port of the router. The only required input transport for an ISP backbone router is telnet (and Secure Shell if you have an image with SSH support). The only required output transport for an ISP backbone router is also telnet. And the preferred transport should be none (otherwise the router CLI will try and use all transports to resolve what has been typed in at the command prompt). For example:

```
line con 0
 transport output telnet
 transport preferred none
line aux 0
 transport output telnet
 transport input none
 transport preferred none
line vty 0 4
 transport input telnet
 transport output telnet
 transport preferred none
```

7. **Setting a domain name.** The next step takes the lab participants through setting up SSH on their routers. Before we can do that, we need to set a domain-name – this is used by SSH key generation to define the hooks for a suitable key to identify the router. We'll just set the domain as follows:

```
ip domain-name workshop.net
```

8. **Using SSH for router access.** The router software images have SecureShell support available in them, so this step will enable SSH support for access to and from the routers. You can recognise an image which has SSH in it as it will have either "k4" or "k9" in the name, signifying 3DES crypto support; for example, *c2600-advipservicesk9-mz.124-25a* is a crypto service provider image for the 2600XM series routers. If you do a *show version* at the command prompt, you'll see what IOS release the router is running.

   To enable support for SSH on the router, first the key needs to be set. To do this enter the following IOS command in configuration mode:

```
crypto key generate rsa
```

   which will generate an RSA crypto key for the router. The prompt asks what key modulus should be chosen, in a range from 360 (more or less useless) to 2048 (the best). This key will be automatically stored in a file in NVRAM – this file is not readable by any user on the router.

   Next, set the source ip address for all SSH connections from the router to be the loopback interface:

```
ip ssh source-interface Loopback 0
```

   SSH is now available for use on the router. Early IOS images only permitted SSH access to the router, but more recent images also allow outbound SSH connections. And from 12.3 onwards, SSHv2 is also supported.

9. **Adding SSH to VTY Transport.** We now need to change the transports permitted on the vtys to allow SSH inbound and outbound. This is a simple case of adding *ssh* to the lines which have telnet in them.

   NOTE WELL: ISPs who care about their infrastructure security completely disable telnet support on their routers – industry best practice considers telnet an obsolete and archaic protocol, ill-suited for use for management of public network infrastructure equipment.

   For example:

```
line con 0
 transport output telnet ssh
 transport preferred none
line aux 0
 transport output telnet ssh
 transport input none
 transport preferred none
line vty 0 4
 transport input ssh
```

```
        transport output telnet ssh
        transport preferred none
```

10. **Using SSH.** Finally, test SSH on your router. This is simply a case of using the ssh command at the command prompt, as you would use the telnet command to connect to a remote neighbour.

Checkpoint #1: *Check the configuration and operation of the vty filters as suggested in the previous steps. Check that SSH connections work, and demonstrate this to the lab assistant.*

11. **Out of Band Management.** One of the often forgotten aspects of any operational network is out of band access to the router and switch equipment. To manage remote sites, ISPs never rely on simply being able to telnet to the router – they configure some other means of access, usually connecting the console port of the router to a modem or a device called a "terminal server". With this facility it is possible to get access to the remote equipment when network connectivity is unavailable (due to misconfiguration, or failure of WAN links) or console access is required to support functions which are incompatible with telnet access (such as upgrading the operating system images on some devices).

    The lab instructors have configured an access router to be used as the out of band management system for the lab network. The router is at IP address 192.168.1.253, connected to the secondary LAN on Router6. The router has several asynchronous connections on it (the 2511RJ has 16, the AS2611 has 32). In most circumstances, the access routers are used to connect modems. However, this feature can be reversed so that the router provides a means of accessing the devices connected to each port, the so-called "terminal server" function.

12. **Configuring Out of Band Access.** The standard lab set up has the COM1: port of the supplied computer connected to the console port of the router. This connection should now be moved to the auxiliary port (usually labelled AUX). Locate the cable from the access router which has been run to your station and connect it to the console port of the router. The layout is represented in Figure 1.

13. **Using Out of Band Access.** With the connections in place, each team should log into their router through the auxiliary port. Check that the OOB router is pingable. If not, try and work out why not. Each tty line on the OOB router is referenced through a particular TCP port. In general, line X is referenced through port 200X. To access a particular line, telneting to the router and specifying port 200X will give access to line X. For the purpose of this lab, Router1 has been connected to Line1, Router2 to Line2, etc. The following table lists these assignments:

Router1	**Line1**	**telnet 192.168.1.253 2001**
Router2	**Line2**	**telnet 192.168.1.253 2002**
Router3	**Line3**	**telnet 192.168.1.253 2003**
Router4	**Line4**	**telnet 192.168.1.253 2004**
…etc…		

    Each router team should now access the console of their router via the OOB router. They should enter the appropriate telnet command and check that the login functions as expected.

The instructors will log into the Out of Band router and show the class the configuration. Observe especially the configuration used for each physical line on the router.

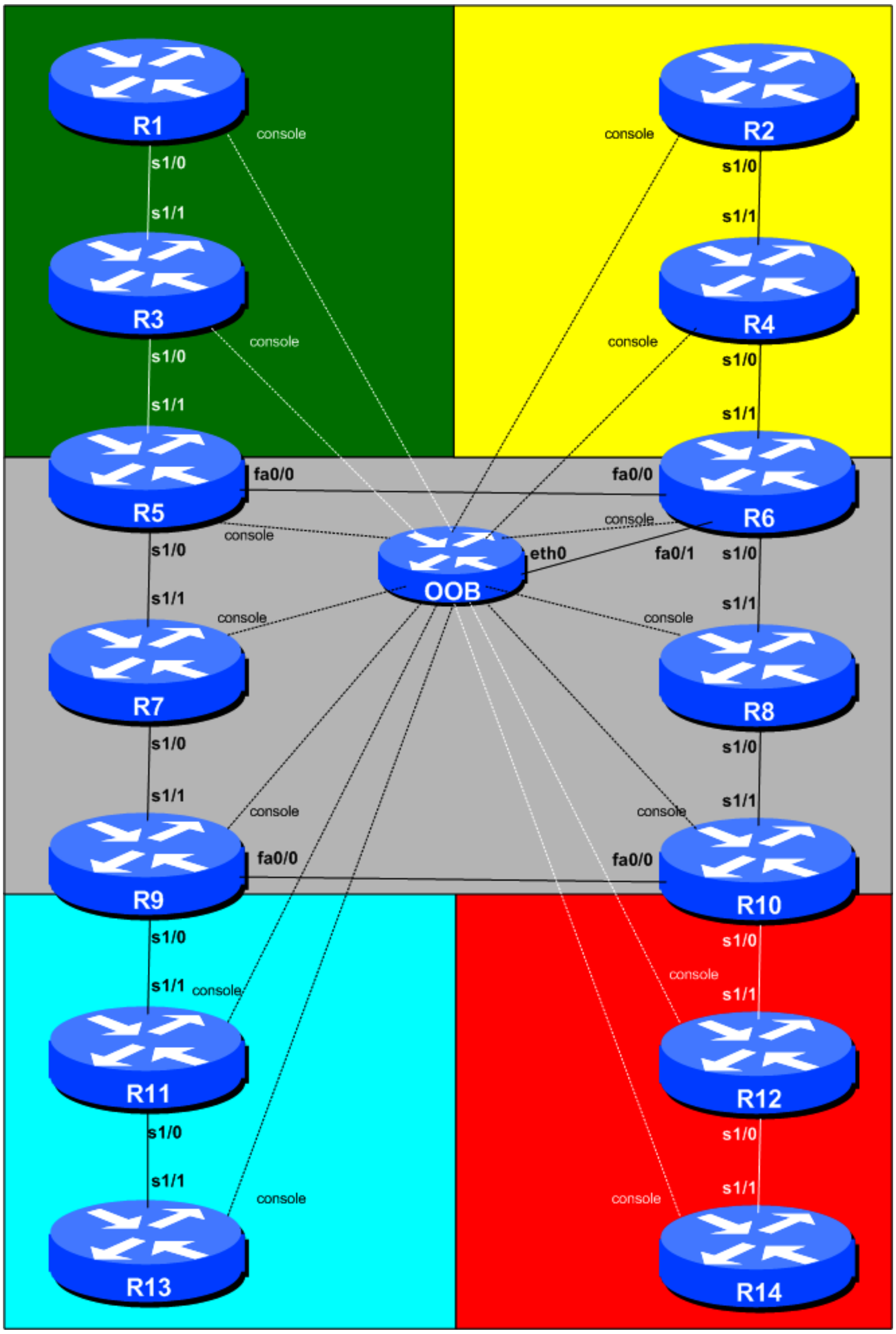

Figure 1 – Out of Band connections

Checkpoint #2: *Check the configuration and operation of the OoB access. Each router team should use this for the rest of this module. The lab instructor will explain the OoB router configuration to the class.*

References: *IOS Essentials* – the section on Network Time Protocol (NTP). *Cisco Documentation* – NTP Commands.

14. **Setting up NTP**. A hierarchy will be used for setting up NTP. This avoids fully meshing the NTP connections on the routers, and is just as effective. The objective is to have a synchronised time source throughout the network. This way every router's time is synchronised with their peers – among other things, it makes comparison of log files so much easier.

    Most ISPs choose a few of their core routers as those which will act as the main NTP peers for the rest of the routers in their network. In other words, a hierarchy is used. The steps are this:
    - The ISP chooses a few external sources to synchronise time against. This can either be by asking permission from the external sources, or by using public servers.
    - The ISP chooses a few internal routers or other systems which will provide the main time synchronisation for their networks. These are usually core routers.
    - The ISP then configures the remaining routers in the network to peer with these "core time" routers.

    The instructor will have configured some systems to be NTP stratum 1 devices. These devices will have the "official" time in the workshop lab, and will be the devices which all other routers will synchronise their time with. The Advanced OSPF Module introduced OSPF areas – the Area Border Routers (ABRs) will be the only routers which peer directly with "official" workshop time sources. The remaining routers will get their time from the ABRs. (Notice that the ABRs are also configured as Route Reflectors in the Route Reflector Module.)

15. **Initial Preparation.** The lab time sources are located on the ethernet LAN connected to Router6. Router6 will announce this network to the rest of the classroom using OSPF. Configuration required for Router6 is:

    ```
    router ospf 41
     network 192.168.1.0 0.0.0.255
    ```

    The LAN uses the network 192.168.1.0/24 – the time sources have IP addresses 192.168.1.1/24 and 192.168.1.2/24. All routers in the lab should check that they can ping both IP addresses. If you cannot, please inform the lab instructors.

16. **Set up NTP Security on all routers.** MD5 will be used to encrypt the NTP sessions between all NTP peers in the network. The word "*cisco*" will be the MD5 key for this lab exercise.

    ```
    ntp authenticate
    ntp authentication-key 1 md5 cisco
    ntp trusted-key 1
    ```

17. **Set NTP to use the router's loopback interface** as the NTP source used for the peering session.

    ```
    ntp source loopback 0
    ```

18. **Set the NTP Server Session** to the systems acting as the master stratum 1 clock source. (This command is applied to the routers in Area 0 only – in other words, Routers 5, 6, 7, 8, 9 and 10 only.)

```
ntp server 192.168.1.1 key 1
ntp server 192.168.1.2 key 1
```

19. **Use the "*show ntp*" commands** to check the status of the NTP synchronisation and the clock setting on the router.

```
show clock
show ntp associations
show ntp status
```

20. **Set the NTP peer connections between the routers in Area 0.** Use the loopback interface for your router as the NTP peering point. Notice the choice of loopback interface rather than any other active interface on the router. Routers 5, 6, 7, 8, 9 and 10 should all configure each other as NTP peers. An example configuration for Router 5 might be:

```
ntp peer 10.0.15.246 key 1
ntp peer 10.0.15.247 key 1
ntp peer 10.0.15.248 key 1
ntp peer 10.0.15.249 key 1
ntp peer 10.0.15.250 key 1
```

    Doing this ensures that the Area 0 routers are all in sync with each other at the same stratum, and allows the lab to build a hierarchy.

21. **Again use the "*show ntp*" commands** to check the status of the NTP synchronisation and the clock setting on the router.

Checkpoint #3: *The teams with the routers in Area 0 should demonstrate their configuration and its operation to the lab instructors.*

22. **Set NTP synchronisation for the remaining routers in the network.** Again use the loopback interface for each router as the NTP synchronisation source.

    The remaining router teams should now configure *ntp server* statements for their ntp synchronisation. The Area Border Routers should be the time source for each non-zero area. For example, the teams in OSPF Area 10 should set up synchronisation with the ABR of Area 10. Teams in OSPF Area 30 should set up synchronisation with the ABRs of Area 30. For example, Router 11 might be configured with:

```
ntp server 10.0.15.249 key 1        ! Router 9
```

23. **Set the NTP peer connections between the remaining routers in each area.** In this case, Router1 and Router3 should set up an ntp peering with each other, Router2 and Router4 should set up an ntp peering with each other, Router11 and Router13 should set up ntp peering with each other, and Router12 and Router14 should set up ntp peering with each other. An example for Router14:

```
ntp peer 10.0.15.252 key 1          ! Router 12
```

This is done to ensure that if the ntp server for these routers disappears, the time synchronisation within the OSPF area remains consistent.

24. **Use *show ntp* commands** to check the NTP status, the NTP time synchronisation, and the setting of the clock on the router**.**

```
show clock
show ntp associations
show ntp status
```

25. **Set the time zone on the router.** The router can be set with a time zone offset from GMT. The timezone command takes a string of characters – obviously set it to the local timezone. Note that only the first seven characters are used in any time display. The following sets the time zone for Singapore with a GMT offset of +8.

```
clock timezone SST 8
```

or

```
clock timezone GMT+8 8
```

26. **(Optional) Daylight savings time setting.** If the local time zone has daylight savings time (called summer time), this can be configured on the router too. For example, in the UK:

```
clock timezone GMT 0
clock summer-time BST recurring last Sun Mar 2:00 last Sun Oct 3:00
```

UK standard time zone is GMT (Greenwich Mean Time). However, in the summer months, the UK changes to BST (British Summer Time), where the time becomes GMT+1. The “summer-time” configuration tells the router this, and between 1am on the first Sunday in March and 1am on the last Sunday in October, it will automatically add one hour to the time and time zone set on the router. Alter the start and finish of summertime according to local conditions. Notice that in the Southern Hemisphere, daylight savings time is usually between November and April.

27. **Set time stamps for all logs on the router.**

```
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
```

28. **(Optional) Additional security can be added with ACLs.** An Access List (ACL) and Access Group can be created and applied to all NTP peer sessions. Since the loopback interface is used, the ACLs will mirror the NTP peer statements. For example, Router 14 would have the following ACL #5 and Access Group created. Notice that the 4 routers being used as core NTP peers need an ACL which will allow access for all routers in the network.

```
access-list 5 permit 10.0.15.250
access-list 5 permit 10.0.15.252
!
ntp access-group peer 5
```

29. **(Optional for routers with a separate real time clock.) Set the real time clock.**
Use the *ntp update-calendar* command to update the router’s real time clock.

```
ntp update-calendar
```

30. **(Optional) Set up NTP for the ISP's customers.** NTP broadcast can be set on any interface to allow an ISP's customers to use NTP or Simple NTP (SNTP) to pick up time packets and synchronise their systems.

```
interface ser 0/0
 ntp broadcast
```

Checkpoint #4: *Demonstrate your configuration and its operation to the lab instructors. Save the configuration to NVRAM. Do you notice any extra information at the top of the configuration saved in NVRAM?*

31. **Login Banner.** IOS by default has a simple welcome message when a new administrative connection to the router is opened. Most ISPs tend to customise this banner to be appropriate to their business – see the example given in the ISP Essentials presentation. We will now set up a login banner for the routers in the workshop lab. Use an appropriate greeting – consult the ISP Essentials presentation to see examples of appropriate and inappropriate greetings. If you use an inappropriate greeting, expect the lab instructors to ask you to change it. Use the following example:

```
login banner ^
Cisco Systems ISP/IXP Workshop Lab
^
```

32. **Logging.** Routers by default capture syslog data produce locally by various features in the IOS. However, the default logging set up is probably not optimal for ISPs. Each router team should configure logging defaults on their router to be as follows:

```
no logging console
logging source-interface Loopback 0
logging trap debugging
logging buffered 16384 debugging
logging facility local4
logging 192.168.1.4
```

This command set will set the log source interface to the Loopback 0 interface, trap level to debug (i.e. most detailed), create a 16K buffer on the router and store the most detailed logs there, and any logs sent to the 192.168.1.4 loghost should be sent using syslog facility local4.

As mentioned in Module 1, it is highly desirable (if not best practice) to disable logging to the router console. If you still haven't done this then the command to do so is `no logging console`. Console logging is on by default in IOS.

NOTE: For ISP operations, it is strongly recommended to DISABLE console logging – router consoles are usually connected to terminal servers as we have just seen, and if the router is under some stress due to events taking place on it, or on the network, it will waste considerable CPU cycles by sending log messages to the 9600baud console port, thereby further slowing it down. Virtually all ISPs disable console logging, and alternatively have the syslog messages sent to the local syslog host, as per the configuration above.

Checkpoint #5: *The lab instructors will show the log information gathered on the loghost PC. They will also show how to configure a Linux box to gather router loghosts – for example, files such as syslog.conf and the set up of syslogd.*

33. Configuring TACACS+ – background.

The lab instructors have configured a TACACS+ server in the network. Its IP address is 192.168.1.4. They have also configured usernames and passwords on the TACACS+ server for each router team.

The first account created simulates a general ISP staff account. The person can log into the router and check the system status, but he or she cannot configure the router or enter privilege (enable) mode. The naming convention is "router#" where # is the number of the router. For example, Router2's staff's username would be "***router2***".

The second account created simulates a NOC or Engineering staff account. The person can log into the router and have full access to all commands. The naming convention is "nocrouter#" where "#" is the number of the router. For example, Router 10's NOC username would be "***nocrouter10***".

34. Configuring TACACS+ – router configuration.

To configure your router to use the TACACS+ server for user authentication, enter the following sequence of commands.

```
ip tacacs source-interface loopback 0
!
tacacs-server host 192.168.1.4
tacacs-server key cisco
!
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication enable default tacacs+ enable
aaa authorization commands 0 default tacacs+ none
aaa accounting commands 15 default start-stop tacacs+
aaa accounting exec default start-stop tacacs+
!
```

The first *aaa* command tells the router to use the AAA model of system security – basically this tells the router to use the "authentication, authorisation, accounting" system rather than the default legacy system.

The next two commands "*aaa authentication*" define the process a router goes through to authenticate a connecting user. "*login default"* says that by default a user logging in to the router must first be passed to the *tacacs+* server for the user's name and password before looking for the local *enable secret* configuration. The *enable* configuration is necessary just in case the TACACS+ server is unavailable. Notice that if a user fails to authenticate on the TACACS+ server, the session is terminated. The router only fails over to the *enable* secret in the event of the TACACS+ server being unavailable.

Likewise, *enable default* tells the router that by default a user requesting to go into enable mode must first be passed to the *tacacs+* server before checking the local *enable* secret for a password.

The advantage of checking a remote server for the enable password is that the enable password can be changed for an entire network, on the server, without having to log into each router. This greatly simplifies administration.

The *aaa authorization* command enables command authorisation on the router. The *commands 0* directive states that standard commands when a user logs in are checked to see if the user is authorised to use them. The authorisation is specified by the *tacacs+ none* commands – the former tells the router to check the TACACS+ server, while *none* tells the router not to use command authorisation when the TACACS+ server is unavailable.

The final two commands are *aaa accounting* instructions, which sends aaa accounting records to the TACACS+ server. The *start-stop* option logs to the TACACS+ server when commands were executed, and completed.

Checkpoint #6: *Demonstrate your configuration and its operation to the lab instructors. The lab instructors will show the TACACS+ information gathered on the TACACS+ server PC. They will also show how to configure a Linux box to be a TACACS+, where to get the software, and the various configuration files.*

35. **Setting up DNS.** The lab instructors will have configured a name server for the workshop network. This step will now configure name and address resolution for the workshop network. Recall that in Module 1, *ip domain-lookup*s were turned off as there was no nameserver for the lab network. Domain lookups can now be turned back on. We already set the domain name earlier on in the module when we activated SSH (Step 7). The remaining commands required to enable DNS on the router are:

```
ip name-server 192.168.1.4
!
ip domain-lookup
!
```

36. **Testing DNS Configuration.** The domain name used is "workshop.net" – the nameserver has been configured to provide name and address resolution for this domain.Try running a traceroute from your router to the other side of the workshop network. Notice how fully qualified domain names are now present in the traceroute, rather than addresses only. (Fully qualified domain name (or FQDN) is a complete hostname and domain name pairing.) An example of a traceroute from Router1 to Router13 is given here (based on the topology of the OSPF Area Module).

```
Router1>traceroute router13

Type escape sequence to abort.
Tracing the route to router13.workshop.net (10.0.15.253)

  1 ser0-1.router3.workshop.net (10.0.15.10) 0 msec 4 msec 4 msec
  2 ser0-1.router5.workshop.net (10.0.15.14) 4 msec 4 msec 4 msec
  3 ser0-1.router7.workshop.net (10.0.15.34) 8 msec 4 msec 4 msec
  4 ser0-1.router9.workshop.net (10.0.15.42) 8 msec 8 msec 4 msec
  5 ser0-1.router11.workshop.net (10.0.15.58) 8 msec 4 msec 4 msec
  6 ser0-1.router13.workshop.net (10.0.15.62) 8 msec *  8 msec
Router1>
```

Notice from the traceroute what the format of the names is. The reverse DNS has been configured such that the addresses map into a name with format InterfaceNumber.RouterNumber.Domain. This is very common practice on the Internet today, and makes debugging of network problems much easier.

Important: It is strongly recommended that all ISPs populate the reverse DNS for their infrastructure equipment. It makes it extremely hard to debug your own backbone if you do not use the reverse DNS. There is no security risk for doing this – router security is implemented via access-lists on the interfaces and vtys, not by omitting content from the reverse DNS.

Checkpoint #7: *Demonstrate your configuration and its operation to the lab instructors. The instructors will demonstrate a traceroute through the network. Notice the detailed use of interface name as part of the router name – this is common practice amongst many ISPs.*

37. **Netflow (Local).** One of the very important features which has been available on Cisco IOS for many years is Netflow. Netflow is basically an information gatherer on the router – it records source and destination address, source and destination ports, the protocol types, and the amount of traffic between those destinations.

    As can be imagined, this type of information is invaluable for the service provider. They can monitor traffic flows in their network, spot popular destinations, deal with attacks on the network, do sensible bandwidth and capacity planning, and manage the operational network more effectively. The ISP Security Workshop covers the use of Netflow as a Network Security Tool in much more detail.

    The Router Teams should now enable Netflow on their routers. Netflow is enabled per interface, and captures flow information inbound on that interface. Netflow should be enabled on active interfaces only (obviously). An example might be:

    ```
    interface fastethernet 0/0
     ip flow ingress
     ip flow egress
    !
    ```

    which will gather flow statistics for traffic inbound (ingress) and outbound (egress) on fastethernet 0/0.

    Once this has been done, display the flow statistics of the router on the console. The command to do this is `show ip cache flow`. What do you see? Output will look similar to:

    ```
    Beta-7200-2>sh ip cache flow
    IP packet size distribution (17093 total packets):
       1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
       .000 .735 .088 .054 .000 .000 .008 .046 .054 .000 .009 .000 .000 .000 .000

        512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
       .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    IP Flow Switching Cache, 1257536 bytes
      3 active, 15549 inactive, 12992 added
    ```

```
  210043 ager polls, 0 flow alloc failures
  last clearing of statistics never
Protocol         Total     Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows      /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          35       0.0        80    41      0.0      14.5      12.7
UDP-DNS             20       0.0         1    67      0.0       0.0      15.3
UDP-NTP           1223       0.0         1    76      0.0       0.0      15.5
UDP-other        11709       0.0         1    87      0.0       0.1      15.5
ICMP                 2       0.0         1    56      0.0       0.0      15.2
Total:           12989       0.0         1    78      0.0       0.1      15.4

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr SrcP DstP  Pkts
Et1/1          144.254.153.10  Null           144.254.153.127 11 008A 008A     1
Et1/1          144.254.153.112 Null           255.255.255.255 11 0208 0208     1
Et1/1          144.254.153.50  Local          144.254.153.51  06 701D 0017    63
```

The first section highlighted above shows a summary of the IP packet size distribution of traffic crossing the router. This example is from a lab, so is a bit unexciting. But it shows that 73.5% of all traffic is between 33 and 64 bytes in size.

The second section highlighted above shows a break down of the number of flows, their size and their activity for certain well known ports.

And the final section shows the current snapshot of traffic crossing the router, source and destination interfaces, source and destination addresses, protocol, source and destination ports, and the number of packets.

Note: Netflow does create a 5-10% CPU hit on all the software based routers. In normal circumstances ISPs adding Netflow shouldn’t cause too much impact on the router’s forwarding speed. Adding Netflow to a router running at 90% CPU isn’t a good idea, but then again industry experience deems that a router running at 90% CPU is very definitely performing suboptimally anyhow.

38. **Netflow (Export).** Netflow records saved by the router can also be exported. While we can look at this configuration in the lab, it’s fairly unlikely that the instructors will have a working Netflow collector system available for testing. If they do, consider yourself lucky!

    Netflow records are exported using UDP to the remote device. The router exports the flow information in a stream of UDP packets to a device specified as flow collector. If the instructors have a Netflow collector device in the lab, you can configure your router to export records there. A sample configuration for this might be:

    ```
    ip flow-export source loopback 0
    ip flow-export version 5 origin-as
    ip flow-export destination 192.168.1.5 2055
    ```

    The first line makes the source address of flow records the IP address of the loopback interface of the router. The second line specifies version 5 flow records (used by IOS routers) and that the origin-as should be exported (the other option is the peer-as, which isn’t really much use in an ISP backbone). The third line specifies that flow records should go to UDP port 2055 (fairly widely used port for Netflow collectors) and the host 192.168.1.5.

If the collector system is running cflowd[1] it's possible, using analysis tools such as Flowscan, to produce graphs and statistical summaries of what is happening on the ISP's network. If the lab instructors have a computer with such software running they will demonstrate the results from the lab. Otherwise, they will demonstrate typical examples of data analysis from Netflow statistics.

39. **Saving configurations on the router.** There are a variety of ways of storing router configurations. In Module One you will have used the command "*write mem*" or simply "*write*" to save the configuration to the router's non-volatile memory (NVRAM).

    However, most ISPs also choose to store their router configurations on a system in the NOC or engineering departments. There are many reasons, but the major one is to have a back up just in case the router NVRAM is somehow damaged or deleted. Other reasons include having an audit trail of changes made to the router configuration, back out in case of configuration error, and so on.

    The lab instructors have set up a TFTP server with an IP address of 192.168.1.4. This system can be used to store the current router configuration. But first, enter the following configuration command:

    ```
    ip tftp source-interface loopback 0
    ```

    This command tells the router that all tftp packets originating from the router will have source IP address of the Loopback 0 interface.

40. **Saving the configuration.** The commands to save the configuration are of the format *copy <source> <destination>* where the source and destinations can be any of the following options: *ftp, lex, null, nvram, rcp, running-config, startup-config, system, tftp.* To save the configuration to the TFTP server, use the "*copy system:/running-config tftp:*" command sequence. If the TFTP server is unreachable, "."s followed by an error message will be displayed rather than "!"s. (Note that the previous "*write net*" is still supported but may be removed at a future release.)

    An example of saving the configuration for Router 1 might be:

    ```
    Router1#copy system:/running-config tftp:
    Address or name of remote host[]? 192.168.1.4
    Destination filename [running-config]? router1-confg
    !!
    2259 bytes copied in 2.920 secs (1129 bytes/sec)
    Router1#
    ```

Checkpoint #8: *Demonstrate to the lab instructors how you saved your router configuration on the TFTP server. Ask the instructors and assistants to check that your router configuration is on the server.*

[1] Cflowd is a software package available from CAIDA (not supported by them any more, but it is still available for download from www.caida.org). Flowscan is also on their archive. The most common Netflow collection tool used today is one called nfsen (http://nfsen.sourceforge.net).