ePipe Advanced Setup

Defining the Internet Address and Subnet Mask

Use the DEFINE INTERNET ADDRESS command to define the server’s Internet address. Then reinitialise the Communications Server for the new Internet address to take effect. The following example shows how to define the Communications Server’s Internet address as 123.4.5.6, define the subnet mask as 255.255.0.0, and reinitialise the Communications Server.

```
Local 1>> DEFINE INTERNET ADDRESS 123.4.5.6 SUBNET
MASK 255.255.0.0
```

```
Local 1>>REBOOT
```

Use the SHOW INTERNET command to display the current Internet address and subnet mask.

Determining the Internet Address from the Network

If an IP address has not been defined, the Communications Server attempts to determine its IP address from the network, first by using BOOTP and, failing that, by using DHCP and RARP. DHCP, BOOTP and RARP are TCP/IP network protocols in which the Communications Server broadcasts its Ethernet address, and a network host returns a matching IP address. Note that with DHCP and BOOTP the responding network host can return the name of a default startup configuration file to be downloaded to the Communications Server using TFTP. See ‘Startup Configuration Files’ later in this section. Refer to your TCP/IP network host documentation for details about DHCP, BOOTP and RARP.

Manipulating the ARP Table

The Address Resolution Protocol (ARP) matches an IPaddress with an Ethernet address. For example, if the Communications Server wants to send an Ethernet packet to a remote host, and the remote host’s Ethernet address is not in the server’s ARP table, the Communications Server sends an ARP broadcast to all hosts on the network. A remote host sends an ARP response with its Ethernet address to the Communications Server. The server enters the remote host’s Ethernet address into its ARP table with a matching IP address, and sends the Ethernet packet and subsequent packets directly to the remote host. You can change, display, and clear entries in the Communications Server’s ARP table. Refer to the following commands in the ePipe on-line or WWW help:

SET/DEFINECHANGE INTERNET ARP ENTRY

LIST/SHOW INTERNET ARP ENTRY

PURGE/CLEAR INTERNET ARP ENTRY

Creating an Internet Host Table

An IP host table maps host names to IP addresses, allowing you to access remote hosts using a host name or alias instead of an IP address. You can create a host table on the Communications Server. Refer to the SET/DEFINE/CHANGE INTERNET HOST command in the ePipe on-line help or ePipe WWW Help section. Refer also to the next section, ‘Using the Domain Name System’ for an alternate method of mapping host names to IP addresses

Using the Domain Name System (DNS)

The Domain Name System (DNS) is an online distributed database system used to map host names to IP addresses. DNS servers in a TCP/IP network implement a hierarchical namespace that allows sites freedom to assign host names and IP addresses. Setting up a DNS server replaces the need to create local host tables in the Communications Server.

Using Web Configuration

Using a WWW browser, go to the ePipe Setup page, select the Setup Wizard link, and activate the Wizard “Domain Name Servers for the ePipe”.

Using Command Line Configuration

To set up a DNS server, enter the IP address of the DNS server in the Communications Server’s DNS table and specify the Communications Server’s domain name. Use the CHANGE INTERNET NAMESERVER command to add the DNS server’s name and IP address to the Communications Server’s DNS table. For example:

```
Local 1>> CHANGE INTERNET NAMESERVER enghost ADDRESS
123.4.5.8
```

The following example shows how to use the CHANGE

INTERNET NAME RESOLUTION DOMAIN command to specify the domain name.

Local 1>> CHANGE INTERNET NAME RESOLUTION DOMAIN eng.company.com

To display the Communications Server’s current DNS information, use the SHOW INTERNET NAME RESOLUTION command.

Using Downloadable Configuration Files

The ePipe stores its configuration in non-volatile memory, and will retain its configuration indefinitely without power. However, you may wish to store the configuration file on a network server, which allows ease of modification, as well as the ability to back-up the configuration and implement change-control. You can cause the ePipe to read a configuration file at startup in one of two ways:

1. Use the CHANGE SERVER STARTUPFILE command (see the ePipe online help or the help section of the ePipe WWW interface).

2. Configure your BOOTP or DHCP server to respond with a boot-file value when allocating an address for the ePipe.

   A boot-time configuration file is read from the server using the Trivial File Transfer Protocol (TFTP). TFTP ships standard with most UNIX systems and is available for MS-Windows and other server operating systems.

Capturing the existing configuration

A starting point for your configuration file is to extract the current configuration from your ePipe. To do this

1. Use the ePipe WWW interface, go to the Status page, and select Reports, then General Report. Cut out the results of “SHOW CHANGES ALL”.

2. Create an empty file on your TFTP server, and ensure it is writable. On the ePipe console issue the command: SHOW CHANGES > filename HOST hostname where filename is the name of the empty file on your TFTP server, and hostname is the name of your TFTP server.

If you make configuration changes using the ePipe WWW configuration tools, you should repeat the above process, and merge your custom configuration with the new configuration generated by the configuration wizard(s).

Creating a Configuration File

Once you've captured your current configuration, you can use a text editor to amend or extend the configuration as required. Save the file in your TFTP directory using the name you specified for SERVER STARTUPFILE or BOOTP/DHCP boot-file.

Creating a Server-Specific Configuration File

If you have more than one ePipe it may be advantageous to separate out the configuration items that are common to all ePipe units into a "default configuration file". Specify this file as the ePipe startup file. A special startup file notation "%@" is supported to allow you to specify a device-specific config file as well as a standard config file. The "%@" notation expands to become the hexadecimal value of the ePipe's IP address. The hexadecimal value is what's used for the file name of the server-specific config file. For example, if your ePipe's IP address is 132.237.6.51 then the server-specific file name would be 84ED0633. Append the following command to your standard config file. "EXECUTE %@ HOST host_name"

Downloading the Configuration File

As well as executing the configuration file at boot time, you can load a new configuration file at any time by executing the command EXECUTE filename HOST hostname where filename is the name of the file on your TFTP server, and hostname is the name of your TFTP server machine

Using the eConfig Utility

eConfig is a utility that can be used to back up and restore the configuration of an ePipe. The configuration is written to a text file that can be saved on any PC on the network. These files can also be opened and edited in any text editor by experienced ePipe users.

Get eConfig from the ePipe miscellaneous utilities download page.

Why Backup or Restore Configurations?

There are several reasons for backing up and/or restoring ePipe configurations, including:

- Replication of a single configuration to multiple ePipes. This will likely require the configuration file to be edited to prevent duplication or conflicts of IP addresses and other parameters.

- Failure of the ePipe. The saved configuration can then be restored to the replacement unit.
- Protection from accidental errors during setup of the ePipe. The ePipe configuration can be rolled back to its previous state if a backup configuration exists.

When doing a restore, care must be taken to ensure that the configuration to be loaded into the ePipe is correct, in terms of both the syntax of the commands and the accuracy of information provided (e.g. ISP phone numbers). Thorough testing of a configuration is strongly recommended prior to replicating the configuration to other ePipe units. Testing of the new configuration/s, once restored to ePipe units, is also recommended to ensure a trouble-free installation of many ePipes using a similar or replicated configuration.

How to use eConfig

eConfig is a Windows 32-bit executable designed to run on Windows 95/98/ME/NT/2000. Simply start eConfig as you would any other Windows program using Windows explorer or click on Start > Run.

In the “ePipe Server” text window, type the DNS name or IP address of the ePipe you wish to configure or backup. Underneath, type in a username and password for a privileged user on that ePipe. The default privileged user is ‘root’ and the default password for this user is ‘system’.

The Backup button is used to backup the current configuration of the selected ePipe to the specified file.

The Restore button is used to download the configuration stored in the specified file to the selected ePipe. During Restore you will be given the option to erase the current configuration of ePipe prior to performing the restore.

Note:

- Erasing the configuration of the ePipe is equivalent to reinitialising the ePipe to its factory default settings. This is a non-reversible operation. Ensure you have a backup of the current configuration prior to doing this.
- If you do not erase the ePipe prior to restoring a configuration, the restore operation may replace, modify or add to the existing configuration of the ePipe. This may result in an undesirable configuration, depending on the current ePipe configuration and the configuration commands in the file being restored.

Specifying Dedicated or Preferred Services

A dedicated service is most useful for inexperienced users who are unfamiliar with Communications Server commands, and users who will always use one particular service exclusively. Specifying a dedicated service has some security benefits too; it keeps users from accessing the Communications Server command set. The following example shows how to specify Telnet as the default protocol of port 3 and server1 as the dedicated host.

```
Local 1>> CHANGE PORT 3 DEFAULT PROTOCOL TELNET
Local 1>> CHANGE PORT 3 DEDICATED SERVER1
```

To establish a session with a dedicated service, press the “Enter” key; you don’t have to log in to the Communications Server or type any Communications Server commands. A preferred service is most useful for more experienced users who are familiar with Communications Server commands, and users who will use one particular service frequently, but might occasionally use other services. The following example shows how to specify Telnet as the default protocol of port 3 and docdock as the preferred host.

```
Local 1>> CHANGE PORT 3 DEFAULT PROTOCOL TELNET
Local 1>> CHANGE PORT 3 PREFERRED unixhost
```

To establish a session with a preferred service, log in to the Communications Server and use the CONNECT/OPEN command. However, if AUTOCONNECT is enabled, you are connected automatically to the preferred service when you log in to the Communications Server. Refer to the following commands ePipe on-line help or ePipe WWW Help section.

```
CONNECT/OPEN
SET/DEFINE/CHANGE PORT DEDICATED
SET/DEFINE/CHANGE PORT DEFAULT PROTOCOL
SET/DEFINE/CHANGE PORT PREFERRED
```

Setting Up Devices & Services

All the following can be achieved using the ePipe WWW configuration interface, and selecting the Setup Wizards in the Direct Connection Services section. See the WWW configuration page and its on-line help for details on how to do this. This section describes how to configure devices and services using the command line interface (or downloadable configuration files).

Note you will need to have activated the Direct Connection Services feature (DCS) in order to use these features of the ePipe 2100.

Setting up a Connection to a Modem

To allow connection of a modem to a port, use the following commands:

CHANGE PORT n MODEM CONTROL ENABLED

CHANGE PORT n SPEED 115200 (or whatever speed you prefer)

(choose one of the following)

CHANGE PORT n ACCESS LOCAL (for incoming calls only)

CHANGE PORT n ACCESS REMOTE (for outgoing calls only)

CHANGE PORT n ACCESS DYNAMIC (for bidirectional use)

Initializing a modem using a port initialization script

A port initialisation command can be used to initialise a modem at boot time. The command syntax is: CHANGE PORT n INITIALIZATION SCRIPT script where script is the initialisation string. The script is a series of send/expect strings separated by spaces—as used by chat-scripting utilities on Unix or MS-Windows.

Configuring a telnet listner port for a modem

To make devices attached to Communications Server ports available to users or systems using svr_tty on the network, configure Telnet listeners and associate them with Communications Server ports. You can configure up to 32 Telnet listeners on a single Communications Server. When configuring a Telnet listener for a modem, the port you are configuring must not be in use. You cannot be logged in to the port or it cannot be in use by any other user or service. In the example below, Telnet listener 2003 is added to the permanent database and associated with port 3 on the Communications Server. It is identified as a modem and enabled for connections.

Local 1>> CHANGE TELNET LISTENER 2003 PORTS 3 ENABLED

Local 1>> CHANGE TELNET LISTENER 2003 IDENTIFICATION “MODEM 987-6543”

Local 1>> CHANGE TELNET LISTENER 2003 CONNECTIONS ENABLED

Local 1>> CHANGE PORT 3 ACCESS REMOTE

Telnet listener 2003 is now ready to accept connections for modem service. A few parameters must be set, however, before the port can handle modem connections. The sample configuration below shows how to configure port 3 for dial-out modem access.

```
Local 1>> CHANGE PORT 3 MODEM CONTROL ENABLED
```

```
Local 1>> CHANGE PORT 3 SPEED 9600
```

Now a user can access the modem via Telnet from a remote host by giving the name or IP address of the Communications Server and the TCP port number of the Telnet listener. For example, to access a Communications Server named termserv on Telnet listener 2003, from the remote host type:

```
telnet termserv 2003
```

Note: Do not manually configure Telnet listeners for use with TruPort or Extend. Both of these products will automatically create the required Telnet listeners.

Configuring for Dialin Login

For dialin login (or dumb-terminal access) use the following commands:

```
CHANGE PORT n ACCESS LOCAL CHANGE PORT n SPEED baudrate
```

```
CHANGE PORT n LOGIN ACCOUNT ENABLED
```

```
CHANGE PORT n DEDICATED NONE
```

See the on-line or WWW-based help for the following commands for more ways to customized dumb-terminal access.

```
CHANGE ACCOUNT
```

```
CHANGE PORT DEDICATED
```

```
CHANGE PORT DEFAULT
```

```
CHANGE PORT PREFERRED
```

Configuring a PPP Port for incoming modem calls

For dial-in PPP access, use the following commands

```
CHANGE PORT n ACCESS LOCAL
```

```
CHANGE PORT n SPEED baudrate
```

```
CHANGE PORT n DEDICATED PPP
```

(choose one of):

```
CHANGE PORT n PPP PAP ENABLED
```

CHANGE PORT n PPP CHAP ENABLED

If you choose PAP, you should create user accounts for each user (using CHANGE ACCOUNT). eg.

CHANGE ACCOUNT username PROTOCOL PPP password “passwd”

If you choose CHAP, you should add secrets (CHAP passwords) using CHANGE SECRET. eg.

CHANGE SECRET HOST username SECRET “passwd” (where name is the username of the incoming caller).

You can set the addresses (and other paramters) used for PPP dialin users with the commands

CHANGE PORT n PPP LOCAL ADDRESS xx.xx.xx.xx

CHANGE PORT n PPP REMOTE ADDRESS xx.xx.xx.xx

(see the on-line or WWW help for CHANGE PORT PPP for details on further PPP configuration commands.)

Configuring a PPP port for outgoing modem calls

To allow PPP dial-out you must create a modem DIALER on the ePipe which is configured to use PPP. If the remote host starts PPP immediately after answering the modem, use the command (substituting the appropriate phone number):

CHANGE DIALER dialername SCRIPT “ATDT555-XXXX CONNECT”

If you are calling a system which requires a textual login/ password to be entered before starting PPP, use the command:

CHANGE DIALER dialername SCRIPT “atdt555-XXXX CONNECT \r\d login: username word: password” (where “username” and “password” are the appropriate strings, and the apprpriate phone number is inserted).

Next, execute the commands:

CHANGE DIALER dialername PROTOCOL PPP

If the remote host does not automatically assign an IP address, use the commands:

CHANGE DIALER dialername REMOTE ADDRESS host_IP_addr

CHANGE DIALER dialername LOCAL ADDRESS local_IP_addr

To start the PPP connection from the Communications Server, execute the command:

CONNECT DIALER dialername

Once the modem dialer is set up, you don't have to reconfigure the port each time you want to start a PPP connection. Just use the CONNECT DIALER command to start a PPP connection from the Communications Server.

Setting up Dial on Demand (DOD)

All the following can be achieved using the ePipe WWW configuration interface, using the Setup Wizards in the Shared Internet Access section. See the WWW configuration page and its on-line help for details on how to do this. This section describes how to configure devices and services using the command line interface (or downloadable configuration files). The CHANGE INTERNET DOD command allows you to configure DOD network connections which are connections that are only made when required and usually disconnect after a period of idle time. More information on advanced set-up is also available on the Stallion WWW site: http://www.stallion.com/html/support/appnotes/

Overview

The DOD configuration consists of three main parts.

- The dialup protocol and network addresses to be used for the connection.

  The protocol and addresses determine the network that will be established by DOD. These values are normally provided by the administrator of the site you will be connecting to. Values such as the local IP address, the remote IP address and the subnet mask are part of this information.

- The DIALER Configuration

  The DIALER configuration enables DOD to call a remote system, log in and start the appropriate networking protocols. DIALER configuration is crucial to the functionality of DOD. You should read and understand the tutorial on DIALERS before starting to configure DOD connections.

- The FILTER configuration

  FILTERS allow DOD to determine which packets force a connection to be established, which packets maintain a link, and how long the link remains connected. Filters can be as simple or complicated as the particular application requires. For this tutorial only the most simple of filters will be used in order to test DOD connections. For more information on how to design and build filters see the tutorial on “INTERNET FILTERS”. If a DOD setup has no filter, it will connect on any activity and remain connected for one day after the last network activity.

Creating a Dialer

Using a Terminal or Terminal emulator, go through all the steps to log onto the system you will be calling. Take careful note of the text that appears just before you need to enter information. Also take careful note of any information you need to enter. These two pieces of information are basically the chat script for your DIALER. The following example sets up a dialer called “isp” for PPP using a typical login style chat script.

CHANGE DIALER isp PROTOCOL PPP

CHANGE DIALER isp CHARACTER MAP 0 CHANGE DIALER isp SUBNET MASK 255.255.255.0

CHANGE DIALER isp SCRIPT “atdt\T ogin:—ogin: nppp assword: nppp”

CHANGE DIALER isp PHONENO “0,1234”

Once we have created a dialer, and configured a port for use with a modem (see above, or the on-line/WWW MODEM tutorial), we are ready to test the dialer. From the servers command prompt enter:

CONNECT DIALER isp DIAGNOSTICS ENABLED

It is best to test your dialer with diagnostics enabled so that you may see why it is failing and correct any errors in the script.

Creating a Simple Filter

To continue with the DOD setup it is best to create a simple filter just for testing. This filter can then be refined once the mechanics of the connection have been confirmed. For this purpose we create a filter that will start the connection whenever a TCP connection is used on the network. This simple filter will hangup the call after 30 seconds of idle time. This will allow us to check that connecting and re-establishing a connection are working correctly before we introduce more complicated filtering. To create a simple filter enter these commands

CHANGE INTERNET FILTER simple ENTRY 1 RULE “tcp accept 30”

CHANGE INTERNET FILTER simple ENTRY 2 RULE “udp accept”

```
CHANGE INTERNET FILTER simple ENTRY 3 RULE "icmp accept"
```

For more information see the INTERNET FILTERS tutorial in the ePipe on-line help, or the Help section of the ePipe WWW interface.

Creating the DOD bundle Setup

Now that we have a simple filter and a dialer, we are ready to create a DOD setup. The following commands will provide us with such a setup:

```
CHANGE INTERNET DOD dod1 DIALER 1 DIALER NAME isp
```

```
CHANGE INTERNET DOD dod1 FILTER simple
```

```
CHANGE INTERNET DOD dod1 ENABLED
```

You can confirm that you DOD setup is ready for operation by running the command:

```
SHOW INTERNET DOD dod1 STATUS
```

If everything has been configured correctly, you should now see that its status is Monitoring.

Testing the DOD Setup

Now that you have your setup ready to go it is time to test it. Based on the configuration above, the simplest way to test the connection is to telnet from the ePipe to the remote host. If possible you should also monitor the connection on another screen. The best way to accomplish this is to have two terminals or logins to the ePipe. On the first login, run the following command\:

```
MONITOR INTERNET DOD STATUS
```

This will allow you to watch the progress of the connection. On the second login run the command:

```
TELNET 192.168.0.1
```

The address given here is the address of the remote host, or the host address provided in the PPP configuration for our dialer. This command should start the connection. Once the connection is established, log in to the remote host and do nothing more for about 30 seconds. You should see the time count down on the monitoring screen to zero, then the connection should be disconnected. After the connection is disconnected the status of the link should return to "Monitoring". Press on the telnet session, the link should be reconnected and the telnet should remain logged in. Now you have that working you may spend more time refining your filters.

Routing Considerations

To use a DOD connection effectively it is quite often necessary to add routes to either the remote host or the ePipe, or both. The reason for this is that their may be hosts at either end which you want to be able to access through the dialup connection. The most common route you will want to add is on the ePipe. This is a default route to the remote network. For the above examples, and a remote network of 192.168.1.0, you would configure the default gateway as

```
CHANGE INTERNET GATEWAY 192.168.0.1 NETWORK ANY
```

This will allow the ePipe and any machines connected to the local network side of the ePipe to find their way to the remote network. For machine to understand this route the must be capable of processing RIP packets. Most unix systems and Windows NT are capable of this, however, Windows 95 is not. For Windows 95 systems you should configure the default gateway as either the ePipe, or another machine on the network that does understand routing protocols. Similarly, for machines on the remote network to access those on the local network, other than the ePipe itself, a suitable route for the local network needs to be installed on and propagated throughout the remote network. As this may be configured in many different ways, you should consult your operating system documentation on how to do this.

Bidirectional DOD

To configure a Bi-directional DOD (ie., a link which can be started from either end of the connection) is very simple. Configure DOD as shown above, then configure a login or a dedicated PPP/SLIP port for the remote host to connect to the local ePipe. Use the same network parameters for this login as for the dialer configuration to dial out, and DOD will automatically install the filters on the link when it is brought up.

RADIUS

Using the RADIUS Protocol

Support for the Remote Authentication Dial In User Service (RADIUS) protocol is part of the Secure Remote Access (SRA) Feature Set. RADIUS enables the ePipe, a RADIUS client or

Network Access Server (NAS), to communicate with a RADIUS server to authenticate incoming connections to the ePipe. Working on a client/server model, RADIUS permits a more sophisticated and uniform method of authentication and accounting than generic authentication services.

Note: RADIUS support can be configured and tested on the ePipe either by using the GUI or by using the command line interface.

RADIUS GUI Configuration

Step 1. Check both the “Enable RADIUS Support” and “Enable Fallback Mode” checkboxes. This will turn RADIUS on in the ePipe, as well as allowing changes to be made in case the configuration details are incorrect.

Step 2. Select a RADIUS secret. This secret must be the same as the secret on the RADIUS server.

Step 3. Select which ports will use RADIUS authentication by checking the port boxes.

Step 4. Enter the names or IP addreses of the RADIUS authentication and accounting servers. Change the port numbers if the RADIUS server uses different ports.

Step 5. Change the Retransmit and Retry counters if desired.

Step 6. Click on Configure to save and enable the RADIUS configuration.

Command Line Configuration

To configure and test RADIUS support on the ePipe, the following steps will need to be performed. Before these steps can be performed, a known working RADIUS server must be installed and configured on a host or hosts that are accessible by the ePipe.

1. Simple RADIUS configuration.

Enable RADIUS support with FALLBACK mode. This allows you to access the ePipe in the event that the RADIUS configuration is incorrect or the RADIUS server is not accessible for some reason.

```
CHANGE RADIUS SUPPORT ENABLED
```

```
CHANGE RADIUS FALLBACK ENABLED
```

Next, the information on the hosts to be used for RADIUS accounting and authentication needs to be configured, as does the shared secret. Usually the same host will be used for both authentication and accounting, although this is not mandatory.

The PORT option allows the use of non-standard TCP port numbers. The standard TCP port numbers for RADIUS are 1812 and 1813. These port numbers supercede the previously allocated port numbers of 1645 and 1646 which were already in use. Multiple authentication and accounting servers may be configured.

CHANGE RADIUS AUTHENTICATION SERVER host_name [PORT tcp_port]

CHANGE RADIUS ACCOUNTING SERVER host_name [PORT tcp_port]

CHANGE RADIUS SECRET ‘your-secret’

Determine which ports on the ePipe are to be used with RADIUS and enable the RADIUS facility on those ports.

CHANGE PORT [port_list] RADIUS ENABLED

Ensure that the serial port on the ePipe is correctly configured for the device that will be attached to that port.

2. RADIUS Server configuration.

Add the ePipe to the RADIUS server’s client database. Add the shared secret (the same secret as above) for the ePipe to the RADIUS server configuration.

Configure a user (e.g. administrator) in the user database of the RADIUS server so that when an authentication request with:

Service-Type=NAS-Prompt

is received by the RADIUS server, a response with

Service-Type=Administrative-User

is returned to the ePipe. If necessary start or restart your RADIUS server.

3. Test the RADIUS Configuration.

Connect a terminal or PC with a terminal emulator to a port on which RADIUS has been enabled and login as administrator.

If successful, the logon is authenticated via RADIUS. As the Service-Type that is returned by the RADIUS server is Administrative-User, the user administrator will immediately be granted access to Privileged Mode.

Also, an accounting record will be sent to the RADIUS accounting server.

If the logon was not successful, examine the log file(s) from your RADIUS server to determine why access was denied.

If access is denied, but not immediately, it is possible that the shared secret is not correct. Check whether the hostname or IP address of the ePipe is correctly specified in the RADIUS server client configuration database.

Once successfully logged on as administrator, you may wish to turn off FALLBACK mode and add more users/user types to the RADIUS server configuration.

Syntax of Network Address Translation (NAT) Rules

There are two types of NAT rules:

Mappings

A mapping rule controls how outgoing connections are modified to appear to originate from the ePipe. When the destination computer receives traffic, it seems to come from the ePipe. When the response is sent to the ePipe, the ePipe forwards it to the originating computer on your internal network.

Mapping rules allow your internal computers to access the internet without requiring globally unique addresses, and protects those computers from outside attack.

Redirections

A redirect rule allows incoming connections to the ePipe be “tunneled” to a computer on your local network.

This allows you to have a server (probably for WWW or email) on your internal network where it would normally be accessible. However, using NAT redirection, you can (eg.) arrange for WWW connections to your ePipe to actually connect to your WWW server. Only the connections you allow can access internal hosts, which makes management of security concerns much easier.

Mapping Rules

A NAT mapping rule looks something like:-

```
map <interface-name> <internal-address/length> ->
<external-address/length>
```

Any connections originating from a computer matching the internal address will be modified to have a source address matching the external address (usually there is only one external address, unless you have a very large network).

For example:

```
map ppp1 0.0.0.0/0 -> 192.159.80.1/32
```

would change any network connections via the interface ppp1 (here assumed to be an internet link, 0.0.0.0/0 is a network address that matches everything) to appear to originate from 192.159.80.1/32 (which in this example is the address of the ePipe. An address with a 32-bit length is a host address, rather than a subnet address).

That is, Mapping rules act on a *destination* interface, and change any traffic matching the rule to have a source address of the the ePipe’s network interface.

Advanced Mappings

The full form of a mapping rule is:-

```
map ifname internal/mask -> external/mask options
```

options can be the protocol type (TCP, UDP etc.), or a range of ports which restricts the altered source port of the outgoing connection.

For example

```
map ethernet2 10.1.1.0/24 -> 192.159.80.1/32 portmap
tcp/udp 10000:20000
```

Will map all TCP or UDP connections coming from the network 10.1.1.XXX and destined to a network connected to ethernet2 to be altered so they appear to come from the ePipe with a source port between 10000 and 20000.

For more details on advanced mapping rules see the CLI help for the IPNAT command (HELP IPNAT). Administrators familiar with BSD UNIX will note that the syntax is the same as the BSD “ipnat” command.

Redirect Rules

A NAT redirect rule has the form:

```
rdr <interface-name> <source-address/length> port
<number> -> <destination-address> port <number>
<protocol>
```

Any connections originating (via the named interface) from a computer matching the source and with the given destination port will be transparently redirected to the destination address and port.

For example:

```
rdr ethernet2 0.0.0.0/24 port 80 -> 192.168.1.80 port 80 tcp
```

Will redirect all incoming (via ethernet 2) connections destined for port 80 to port 80 on the host 192.168.1.80. In other words any attempt to connect to the ePipe's WWW port will be forwarded to a WWW server inside the secure network.

Redirect rules are a powerful way to provide internet services, while using an ePipe as a security gateway. Only the connections you allow through to the internal servers will be permitted.

Testing NAT rules (temporary rules)

You can use the IPNAT command to add, delete and show active NAT rules.

Rules added using IPNAT are not permanent—they will disappear if the ePipe is rebooted. For permanent rules see the next section. It is often best to fine-tune your setup using temporary rules entered by hand, before copying the final rule configuration to a NAT rule-set.

To add a NAT rule:-

```
IPNAT <rule>
```

```
eg. IPNAT map ethernet1 0.0.0.0/0 -> 192.159.80.1/32
```

To delete a NAT rule:-

```
IPNAT -r <rule>
```

To delete all active NAT rules:-

```
IPNAT -C
```

NAT Rule-sets (rules that are always present)

Each dialout bundle may have a “rule-set” associated with it. When the bundle is established, the NAT rule-set is installed into the NAT rule table. Note that (unlike filter rules) changes to a NAT rule table will not take effect until the bundle is restarted.

To create a rule-set:-

```
CHANGE INTERNET NAT ruleset ENTRY <number> RULE
<rule>
```

For example:

```
CHANGE INTERNET NAT “internet rules” ENTRY 1 RULE “map
ethernet1 192.168.1.0/24 -> 192.159.80.1/32 tcp
10000:20000”
CHANGE INTERNET NAT “internet rules” ENTRY 2 RULE “map
ethernet1 192.168.2.0/24 -> 192.159.80.1/32 tcp
20000:30000”
CHANGE INTERNET DOD “internet bundle” NAT “internet rules”
```

You can view all rule sets using the command:-

```
SHOW INTERNET NAT ALL
```

To view a single ruleset:-

```
SHOW INTERNET NAT <ruleset>
```

Upgrading ePipe Firmware

Firmware is the ePipe operating system software that is stored in the ePipe's non-volatile flash upgradeable memory.

Stallion has written an application note with full details on upgrading ePipe firmware, monitoring the upgrade process and troubleshooting upgrade problems. Please refer to the Upgrading ePipe Firmware (PDF - 238KB) application note for full details. Click here for a list of ePipe application notes. Please refer to the application notes page for more information on application notes.

Using Syslog

Introduction

To make debugging simple and effective the Stallion ePipe has been designed with System Event Logging Capabilities, or as it is more broadly referred to, Syslog. Syslog provides a means for processes, (i.e. daemons) to output message information to the user when an event occurs. These events range from standard informational messages all the way to important error conditions.

The ePipe uses Syslog messages to monitor the performance of the various system modules and to assist both the user and Stallion Support to diagnose and then correct, possible configuration flaws. Syslog capabilities are only implemented in the Command Line Interface (CLI) of the ePipe and as such are not available through the ePipe Management Assistant.

SYSLOG STATUS
To see the current status of the ePipe's SYSLOG setup enter the following command from the ePipe CLI:

```
SHOW SERVER SYSLOG
```

ACTIVATING SYSLOG OUTPUT
Typically Syslog messages are passed to an external PC or host that is running a Syslog daemon. The Syslog daemon receives

incoming Syslog messages and stores them in a log file that can be viewed at any time. The ePipe can be setup to output System event messages to a Syslog daemon.

To send Syslog output from the ePipe to an external Syslog Daemon enter the following command on the ePipe CLI:

```
CHANGE SERVER SYSLOG HOST [Host_Name]
```

Where [Host_Name] is the IP Address of the Syslog Daemon.

Alternatively the ePipe also allows users to pass Syslog information directly to the console port for immediate viewing.

If you are using the console port to monitor the ePipe operation you can enable Syslog output by entering:

```
CHANGE SERVER SYSLOG CONSOLE ENABLED
```

Alternatively if you are running a TELNET session to the ePipe, run the same command and then enter:

```
WATCHLOG
```

These CLI Instructions will enable SYSLOG output from the ePipe. However the user must still specify the processes to log and the level of detail that is required.

SYSLOG OUTPUT LEVELS

The ePipe has 4 levels of Syslogging: NONE, SUMMARY, DETAIL and DEBUG.

The default Syslog output level is set to NONE, which means that no Syslog information is currently being generated from that process.

The SUMMARY level provides minimal logging of events and is used to output only significant events to the user.

The DETAIL level is the most commonly used Syslog level from the ePipe. This level provides significant amounts of relevant information from the required process. This enables detailed monitoring without excessive output.

The DEBUG level provides the maximal Syslog output from the ePipe. Typically this is not useful to the end user as it generates too much information to process in a short time. DEBUG level is only used in cases where the DETAIL level is not providing sufficient output, which is quite rare.

SUPPORTED SYSLOG PROCESSES

The ePipe enables system logging of all the key modules within the ePipe. These modules are:

AUTH
COMMAND
DIALER
DNSPROXY
DOD
E2B
FILTER
HTTPD
I2B
ROUTED
SERVER
PPP
PPTP
PPPOE
DHCP

Consequently the user can setup the ePipe to generate Syslog output from any or all of these processes.

DEFINING SYSLOG OUTPUT

The command to generate Syslog output from any given process and level is:

```
CHANGE SERVER SYSLOG [MODULE] [LEVEL]
```

Where the module and level can be any of the above types. For example, to enable DETAIL level syslog on PPP and DIALER, you would use the following command:

```
CHANGE SERVER SYSLOG PPP DETAIL DIALER DETAIL
```

If Syslog output has been activated, then running this command will start the ePipe Syslog Output. Any messages generated from the monitored processes will now be output to the Syslog destination (either the console port or a remote host).

DE-ACTIVATING SYSLOG OUTPUT

To de-activate Syslog output to the Syslog Daemon on a network host, enter the following command:

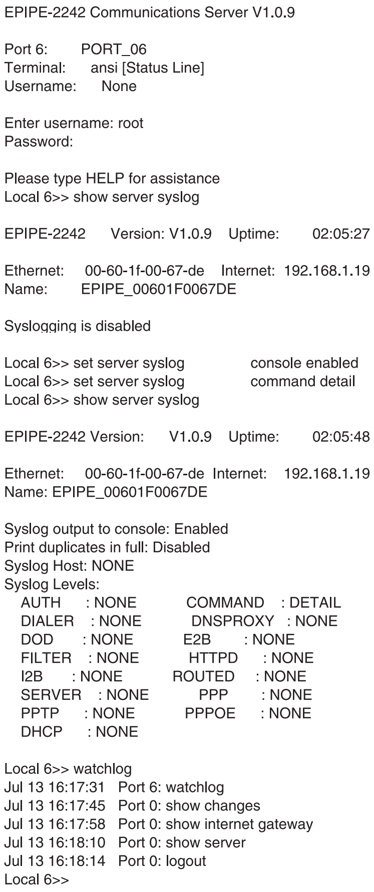
```
CHANGE SERVER SYSLOG HOST NONE
```

To turn Syslog off on the console port enter the following command:

```
CHANGE SERVER SYSLOG CONSOLE DISABLED
```

EXAMPLE SYSLOG SESSION

An example syslog session may look something like this:



```
EPIPE-2242 Communications Server V1.0.9

Port 6:        PORT_06
Terminal:      ansi [Status Line]
Username:      None

Enter username: root
Password:

Please type HELP for assistance
Local 6>> show server syslog

EPIPE-2242      Version: V1.0.9    Uptime:       02:05:27

Ethernet:     00-60-1f-00-67-de    Internet:  192.168.1.19
Name:         EPIPE_00601F0067DE

Syslogging is disabled

Local 6>> set server syslog              console enabled
Local 6>> set server syslog              command detail
Local 6>> show server syslog

EPIPE-2242 Version:      V1.0.9    Uptime:       02:05:48

Ethernet:     00-60-1f-00-67-de  Internet:    192.168.1.19
Name: EPIPE_00601F0067DE

Syslog output to console: Enabled
Print duplicates in full: Disabled
Syslog Host: NONE
Syslog Levels:
    AUTH      : NONE           COMMAND    : DETAIL
    DIALER    : NONE           DNSPROXY   : NONE
    DOD       : NONE           E2B        : NONE
    FILTER    : NONE           HTTPD      : NONE
    I2B       : NONE           ROUTED     : NONE
    SERVER    : NONE           PPP        : NONE
    PPTP      : NONE           PPPOE      : NONE
    DHCP      : NONE

Local 6>> watchlog
Jul 13 16:17:31   Port 6: watchlog
Jul 13 16:17:45   Port 0: show changes
Jul 13 16:17:58   Port 0: show internet gateway
Jul 13 16:18:10   Port 0: show server
Jul 13 16:18:14   Port 0: logout
Local 6>>
```

This example starts by the user starting a telnet session to the ePipe and logging in. Once the CLI is accessible, the user checks the status of syslog, then enables syslog to the console for the COMMAND module. The status of syslog is then re-checked to confirm the changes to syslog. WATCHLOG is executed to view the syslog output. When finished viewing the output, pressing [ENTER] will return the user to the CLI.