Chapter 41

Firewall

Introduction

This chapter describes the router's built-in firewall facility, and how to configure and monitor it.

The Internet is not controlled and certain individuals use it destructively. These individuals attack other computer systems for entertainment and/or profit. A *firewall* is a security device that allows safe access to the Internet by enforcing a set of access rules between the various interfaces of the product.

A firewall typically has two interfaces that are attached to:

- a public network (Internet)
- an internal private network (intranet) that requires protection

A firewall prevents unrestricted access to the private network and protects computer systems behind the firewall from attack. Because a firewall provides a single link between the private network and the public network, it is uniquely positioned to provide a single point where all traffic entering and leaving the private network can be logged and monitored. This information is useful for providing a security audit trail.

The firewall facility is enabled with a special feature license. To obtain a special feature license contact an Allied Telesyn authorised distributor or reseller. Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

Firewall Technologies

The firewall affects IP-based protocols only. It does not affect IPX, DECnet, and AppleTalk network protocols. Key firewall technologies are:

- **Application Gateway**
  This is the traditional approach used to build a firewall, where every connection between two networks is made via an application program (called a *proxy*) specific for that protocol. A session from the private network is terminated by the proxy, which then creates another separate session to the end destination. Typically, a proxy is designed with a detailed knowledge of how the protocol works and what is and is not allowed. This approach is very CPU intensive and very restrictive. Only protocols that have specific proxies configured are allowed through the firewall; all other traffic is rejected. In practice most third-party proxies are transparent proxies, which pass all traffic between the two sessions without regard to the data.

- **Stateful Inspection**
  A more recent approach to firewall design uses a method called "*stateful inspection*". Stateful inspection is also referred to as *dynamic packet filtering* or *context-based access control* (CBAC). In this technology, an inspection module understands data in packets from the network layer (IP headers) up to the application layer. The inspection module checks every packet passing through the firewall and makes access decisions based on the source, destination and service requested. The term *stateful* refers to the firewall's ability to remember the status of a flow. For example, whether a packet from the public Internet is returning traffic for a flow originated from the private intranet. The TCP state of TCP flows is also monitored,

allowing inappropriate traffic to be discarded. The benefit of this approach is that stateful inspection firewalls are generally faster, less demanding on hardware, and more adaptive to new Internet applications.

The router's firewall implementation has the following features:

- Dynamic packet filtering (stateful inspection) technology.
- Application of dynamic filtering to traffic flows, using the base rule that all access from the outside (i.e., public interfaces) is denied unless specifically permitted and all access from the inside (i.e., private interfaces) is allowed unless specifically denied.
- The firewall opens only required ports for the duration of a user session. Configuration commands are required to allow access to internal hosts from a public interface.
- The firewall intercepts all TCP connections and completes the connection. This feature better tracks and defends against denial of service attacks by depletion of TCP slots. Any further out-of-sequence TCP frames are dropped.
- The firewall can be configured to limit internal access to the public network based on a policy setting.
- The generation of unreachable ICMP messages can be enabled or disabled.
- The firewall can be configured to block pings that are destined for firewall interfaces (by default, these pings are allowed).
- The firewall acts as an IDENT proxy (enabled by default, can be disabled).
- All firewall events can be selectively logged to the Logging Facility.
- Significant firewall events generate notifications to designated destinations, including SNMP traps, triggers which can be configured to activate scripts, an email address or an asynchronous port. The size of event required to generate a notification can be set.
- The firewall supports protocols such as FTP (active and passive mode), RealAudio from Progressive Networks, Streamworks from Xing Technologies, CuSeeMe from White Pines, VDOLive from VDOnet, QuickTime streaming video from Apple Computer, Microsoft NetShow, NETBIOS, GRE, OSPF, PPTP and RSVP.
- The firewall detects and logs a range of denial of service attacks including SYN and FIN flooding, Ping of death (illegal ping packet sizes, or an excessive number of ICMP messages), Smurf attacks (packets with an IP address of the private network and typically a broadcast address) and port scans.
- An accounting facility records, via the Logging Facility, the traffic flow for an individual session.

Policies

The first step in deploying a firewall is to determine exactly what traffic should be allowed and denied. This is called the *security policy*. The configuration of the firewall is based around the concept of a security policy. The security policy contains rules that specify the types of traffic allowed through the firewall.

To enable or disable a firewall, use the commands:

```
enable firewall
disable firewall
```

To display the current status and a configuration summary, use the command:

```
show firewall
```

To create or destroy a policy, use the commands:

```
create firewall policy=name
destroy firewall policy=name
```

The firewall does not become active until at least one public and one private interface have been assigned to the policy. A public interface is an interface attached to a public network such as the Internet. A private interface is an interface attached to a private network, such as a company intranet behind the firewall. The basic function of a firewall is to control the forwarding of traffic between the public interface and the private interface. Interfaces are added to or removed from a policy using the commands:

```
add firewall policy=name interface=interface type={public|
   private} [method={dynamic|passall}]
delete firewall policy=name interface=interface
```

An interface can be defined as private in only one security policy. An interface can be defined as public in up to two security policies. After at least one private interface and one public interface have been added, the firewall is functional and automatically implements default policy rules:

- All flows originating from inside (i.e., private interfaces) are allowed. When a session is initiated from a private interface to an outside IP host and has been allowed by the firewall, traffic for that session can flow in both directions. When the session completes, no further traffic is accepted to the private IP host on that port.
- All flows originating from the outside (i.e., public interfaces) are blocked, except ICMP echo requests (pings) to the router interfaces specified in the policy.
- Traffic is dropped when it comes from an interface not covered by policy and goes to an interface specified private in a policy.
- Traffic between interfaces not specifically covered by a policy is passed as normal.
- The firewall acts as an IDENT proxy. Protocols such as FTP and SMTP query the identity of the source of a new session using the IDENT protocol RFC 1413. The firewall proxies IDENT queries when necessary.

If you are configuring the router as a load balancer, it is essential that the firewall's policies allow traffic from clients to travel to and from the public interface and port on each configured virtual balancer. If the firewall blocks this traffic, the load balancer does not operate. If you already have a firewall configured on the routing device that acts as a load balancer, you must ensure

that existing policies allow this traffic flow. For more information, see Chapter 50, Load Balancer.

The current status and configuration of a policy or all policies can be displayed using the command:

```
show firewall policy=name [summary] [counter]
```

By default, the firewall accepts pings to the policy's interfaces, and blocks all other ICMP packets from public interfaces and from untrusted private interfaces. The processing of ICMP packets, IP packets with options set, and ping packets can be enabled or disabled on a per-policy basis using the commands:

```
enable firewall policy=policy-name [icmp_forwarding={all|
   parameter|ping|sourcequench|timeexceeded|timestamp|
   unreachable}] [options={all|record_route|security|
   sourcerouting|timestamp}] [ping]

disable firewall policy=policy-name [icmp_forwarding={all|
   parameter|ping|sourcequench|timeexceeded|timestamp|
   unreachable}] [options={all|record_route|security|
   sourcerouting|timestamp}] [ping]
```

For example, to prevent the router or switch from responding to pings from the public side of the firewall to either public or private interfaces, or from untrusted private interfaces use the command:

```
disable firewall policy=policy-name ping
```

By default, the IDENT proxy is enabled. This means that the router or switch accepts IDENT requests. To disable the IDENT proxy, use the command:

```
disable firewall policy=policy-name identproxy
```

To further refine the control over flows to and from the public network, rules are added to the policy to allow or deny specific types of traffic.

Rules

Policy rules refine the default security policy, which denies all access from hosts on the public network to hosts on the private network but allows access from hosts on the private network to hosts on the public network.

Policy rules define precisely when and how traffic can flow through the firewall based on IP addresses, port numbers, day of the week, time of day, or protocol. For example, if a mail server is running on the private network, a rule could be added to allow TCP traffic to port 25 (the SMTP port) on the mail server host.

To add or delete a rule from a policy, use the commands:

```
add firewall policy=policy-name rule=rule-id action={allow|
   deny} interface=interface protocol={protocol|all|egp|gre|
   ospf|sa|tcp|udp} [after=hh:mm] [before=hh:mm] [days={mon|
   tue|wed|thu|fri|sat|sun|weekday|weekend}[,...]]
   [gblip=ipadd] [gblport={all|port[-port]}]
   [ip=ipadd[-ipadd]] [list={list-name|radius}] [port={all|
   port[-port]|service-name}] [remoteip=ipadd[-ipadd]]
   [sourceport={all|port[-port]}] [encapsulation={none|
   ipsec}]

delete firewall policy=policy-name rule=rule-id
```

To modify an existing rule, use the command:

```
set firewall policy=policy-name rule=rule-id
   [protocol={protocol|all|egp|gre|ospf|sa|tcp|udp}]
   [after=hh:mm] [before=hh:mm] [days={mon|tue|wed|thu|fri|
   sat|sun|weekday|weekend}[,...]] [gblip=ipadd]
   [gblport={all|port[-port]}] [ip=ipadd[-ipadd]] [port={all|
   port[-port]|service-name] [remoteip=ipadd[-ipadd]]
   [sourceport={all|port[-port]}] [encapsulation={none|
   ipsec}]
```

To display currently configured rules for a policy, use the command:

```
show firewall policy=policy-name
```

Rules are processed in order from the lowest number to the highest number. If rules both deny and allow an activity, the rule with the lowest number takes precedence. Typically, rules specify the access to or from a particular IP address and port. Controlling access to many destinations could require a large number of commands. The firewall solves this problem by providing support for lists of addresses stored in files in the routers file subsystem or on a RADIUS server. A rule can be configured to allow or deny access to addresses in up to four lists or RADIUS servers. See "Access Lists" on page 41-9 for more information about configuring access lists.

Rules are processed as follows:

1. Based on the direction of the new flow or session, the default access result is set to the case of no matching rules. For new sessions or flows originating from a private network, access is set to *allowed*. For sessions and flows originating from a public network, access is set to *denied*. Each rule is then matched to the new flow or session until a match is found or all rules are rejected as not applicable, in which case the default access is used.
2. The protocol of the new flow is checked against the protocol field of the rule. If there is no match then the rule is rejected as not applicable.
3. The destination port is then matched to the rule port range. If there is no match then the rule is rejected as not applicable.
4. The source port is then matched to the rule's source port range if it is set. The source port used is dependent on the direction of the flow. For flows from a private network the source port of the flow is used. For flows from the public network, the destination port is used. If there is no match then the rule is rejected as not applicable.
5. The new flow's remote IP address is then matched to the rule's remote IP address or range if it is set. The remote IP address used is dependent on the direction of the flow. For flows from a private network the remote IP address used is the destination IP address of the flow. For flows from the public network, the source IP address of the flow is matched to the remote

IP address of the rule. If there is no match then the rule is rejected as not applicable.

6. The new flow's IP address is matched to the rule's IP range or global IP address. If there is no match then the rule is rejected as not applicable. The IP address used is dependent on the direction of the flow. For flows from a private network the IP address used is the source IP address of the flow. For flows from the public network, the destination IP address is matched either to the IP address of the rule or to the global IP address set for the rule, depending on whether NAT is being applied to the interface.
7. If the IP address matches the rule then the time period is checked against allowed times for the rule. If the current time is not within the specified time range for the rule then the rule is rejected as not applicable.
8. If a hardware list or lists have been specified for the rule and the rule has been applied to an Ethernet interface, then the hardware lists are checked for a match to the source MAC address of the new flow. If there is no match then the rule is rejected as not applicable.
9. When an IP list has been specified for the rule, then the list is checked for a match to either the destination IP address for new flows started from the private network, or for the source address for new flows started from the public network.

   When no IP lists or RADIUS servers are set, and the rule action is ALLOW, the new flow is allowed. When the rule action is to DENY, the flow is denied. Similarly, when there are IP lists and a match, and the rule action is ALLOW, the new flow is allowed. If the rule action is DENY, the flow is denied.
10. When there are IP lists and no match, RADIUS is not set, and the rule action is ALLOW, the new flow is denied. If the rule action is DENY and the flow originated from the private interface, the flow is allowed. Otherwise, the rule is rejected as not applicable.
11. When there are IP lists and there is no match and RADIUS is set, the new flow is placed in a queried state and a request is passed to a RADIUS server to determine if the new flow should be allowed or denied. RADIUS server responses are interpreted as follows:

    - When the rule action is ALLOW and the RADIUS server either rejects the request or returns an IP address of 0.0.0.0, the flow is denied.
    - When the rule action is ALLOW and the RADIUS server accepts the request and returns a valid IP address, the flow is allowed.
    - When the rule action is DENY, and the new flow is from the private network, and the RADIUS server either rejects the request or does not respond to the request, the flow is allowed.
    - When the rule action is DENY, and the new flow is from the public network, and the RADIUS server either rejects the request or does not respond to the request, the flow is denied.
    - When the rule action is DENY and the RADIUS server returns a valid IP address, the flow is allowed.
    - When the rule action is DENY and the RADIUS server accepts the request and returns an IP address of 0.0.0.0, the flow is denied.

    See "RADIUS Servers" on page 41-10 for a detailed description of the format of RADIUS requests and RADIUS database entries.

Multicast Packet Handling

Multicast packets must be carefully handled by the firewall because it does not know on which interface the packet is to be forwarded. When several policies use the receiving interface, the packet cannot be associated with only one policy. The firewall uses the following logic to decide if a multicast packet should be allowed or denied:

1. When the interface on which the multicast packet is received is a public interface for one or more policies, the packet is discarded unless at least one policy has an allow rule for it. Therefore, when one policy allows a particular multicast packet, all other policies implicitly allow the packet. IP multicasting then decides on which interfaces the packet is forwarded.
2. When the interface on which the multicast packet is received is a private interface and not a public interface in any other policies, it is allowed.
3. Network Address Translation (NAT) of any kind cannot be applied to multicast packets.

Access Lists

Access lists are lists of addresses to which access is controlled by one or more policy rules. The firewall supports the following mechanisms for storing and managing access lists:

- List files stored in the router's file subsystem
- RADIUS servers

List Files

A list file is an ASCII text file with a `.TXT` stored on the router's file subsystem and containing a list of addresses. List files are more suited to small lists of addresses that remain relatively static. Two types of list files can be used—IP address lists and hardware address lists.

An IP list file contains a list of IP host and network addresses. Lines in an IP address file have the following format:

- A single IP address in dotted decimal notation; *or*
- A single IP address in dotted decimal notation, followed a space or tab and the name of the host; *or*
- A comment character "#" followed by comment text; *or*
- A range of IP addresses in dotted decimal notation separated by a hyphen, optionally followed by a text name.

For example, the file LISTIP.TXT might contain the following:

```
202.36.163.6
202.49.72.92 ftp.company.com # FTP host
# access for an entire network
202.36.163.0 - 202.36.163.255 dummy network
```

The destination IP address is checked for outbound (private to public) traffic and the source IP address is checked for inbound (public to private) traffic.

A hardware address list file contains a list of hardware addresses. Lines in a hardware address file have the following format:

- A single MAC address in standard notation.
- A single MAC address in standard notation, followed by a space or tab and the name of the host.
- The comment character "#" followed by arbitrary text.

For example, the file LISTMAC.TXT might contain the following:

```
00-00-cd-02-03-01
00-00-cd-02-03-01 pc1.company.com # Bobs PC
# a comment line
```

Only the source MAC address of the packet is compared to the list.

A list file is added to or deleted from a policy using the commands:

```
add firewall policy=policy-name list=list-name file=filename
   type={ip|address}
```

where *name* is the name of the policy, *list-name* is a user-defined name for the list and *filename* is the name of the file on the router's file subsystem. To create a rule to provide access control for the addresses in a list, use the command:

```
add firewall policy=policy-name rule=rule-id action={allow|
   deny} interface=interface protocol={protocol|all|egp|gre|
   ospf|sa|tcp|udp} list=list-name [other-options...]
delete firewall policy=policy-name list=list-name
```

To add up to four lists to a single rule, use multiple invocations of the command:

```
add firewall policy=name rule=rule-id list=list-name
```

RADIUS Servers

There are situations where it is necessary or desirable to control access to a large number of addresses, but impractical to store these addresses in a list file on the router. A typical example would be an organisation wanting to allow general access to the web but restrict access to specific web sites. There are companies that compile lists of web sites that individuals or groups may find objectionable and offer them as a paid service. These lists are usually very large and updated regularly by the service. A RADIUS server is an ideal place to store these lists.

The firewall can be configured to use one or more RADIUS servers to perform checks on user access rights. If a LIST of type RADIUS is specified for a rule, and a RADIUS server has been configured, the router makes RADIUS requests of the following format:

```
User-Name [ipadd]

User-Password allowdeny
```

where *ipadd* is the source or destination IP address of the new flow, depending on the direction of the flow.

The RADIUS server entry that specifically denies access looks like:

```
[ipadd] Password = "allowdeny", Framed-Address = 0.0.0.0
```

The RADIUS server entry that specifically allows access looks like:

```
[ipadd] Password = "allowdeny", Framed-Address = ipadd
```

Once the RADIUS server has been configured and the address added to the server's database, the router must be configured to generate RADIUS requests. To add or delete a RADIUS server, use the commands:

```
add radius server=ipadd secret=secret
delete radius server=ipadd
```

To display a list of known RADIUS servers, use the command:

```
show radius
```

See Chapter 1, Operation for a detailed description of the **add radius server**, **delete radius server**, and **show radius server** commands.

To create a rule to provide access control for the addresses in the RADIUS server, use the command:

```
add firewall policy=policy-name rule=rule-id action={allow|
   deny} interface=interface protocol={protocol|all|egp|gre|
   ospf|sa|tcp|udp} list=radius [other-options...]
```

Network Address Translation (NAT)

Network Address Translation (NAT) allows a single device to act as an agent between the Internet (public) network and a local (private) network.

The firewall uses its own built-in NAT service to translate a local IP address to a global IP address suitable for remote networks. The firewall has two ways of configuring NAT:

- interface-based
- rule based

Interface-based NAT

Interface-based NAT provides a simple address translation for traffic passing between a pair of interfaces. The following methodologies are supported by interfaced-based NAT:

- Standard NAT
  This translates the addresses of private side devices to addresses suitable for the public side of the firewall (source address is translated for outbound packets, and destination address for inbound packets).
- Enhanced NAT
  This translates many private or public side addresses into a single global or local address (source address is translated for outbound packets, and destination address for inbound packets).

Rule-based NAT

Rule-based NAT provides advanced address translation based on the properties of a packet received on a particular firewall interface. Selector values such as source address, destination address, protocol type, and port number (TCP/UDP) determine which packets undergo translation. The following methodologies are supported:

- Standard NAT
  This translates the addresses of private side devices to addresses suitable for the public side of the firewall (source address is translated for outbound packets, and destination address for inbound packets).
- Reverse NAT
  This translates the addresses of public side devices to addresses suitable for the private side of the firewall (destination address is translated for outbound packets, and source address for inbound packets).
- Double NAT
  This translates both the public and private side source and destination addresses.
- Enhanced NAT
  This translates many private or public side addresses into a single global or local address. If it is applied to a private interface, the rule matches the outbound sessions (source address is translated for outbound packets and destination address for inbound packets). If it is applied to a public interface, the rule matches the inbound sessions (source address is translated for inbound packets and destination address for outbound packets).
- Subnet Translation
  This translates IP addresses from one subnet into another subnet, for example all 192.168.xxx.xxx IP addresses can be translated into 202.36.xxx.xxx addresses. Subnet translation may be applied to Standard, Reverse, and Double NAT.

Adding and Removing a NAT Translation

To add or delete an interface-based NAT translation to a policy, use the commands:

```
add firewall policy=policy-name nat={enhanced|standard}
   interface=interface [ip=ipadd] gblinterface=interface
   [gblip=ipadd[-ipadd]]

delete firewall policy=policy-name nat={enhanced|standard}
   interface=interface gblinterface=interface [ip=ipadd]
```

If the GBLIP parameter is not specified, the IP address of the public interface is used as the global IP Internet address.

To add or delete a rule-based NAT translation for traffic received on a specific interface that matches the selector values specified, use the commands:

```
add firewall policy=policy-name rule=rule-id action=nat
   interface=interface protocol={protocol|all|egp|gre|icmp|
   ospf|sa|tcp|udp} [after=hh:mm] [before=hh:mm] [days={mon|
   tue|wed|thu|fri|sat|sun|weekday|weekend}[,...]]
   [encapsulation={none|ipsec}] [gblip=ipadd] [gblport={all|
   port[-port]}] [gblremoteip=ipadd[-ipadd]]
   [ip=ipadd[-ipadd]] [list={list-name|radius}]
   [nattype={double|enhanced|reverse|standard}]
   [natmask=ipadd] [port={all|port[-port]|service-name]
   [remoteip=ipadd[-ipadd]] [sourceport={all|port[-port]}]
   [ttl=hh:mm]
```

```
delete firewall policy=policy-name rule=rule-id
```

Ethernet Interfaces

WAN connections such as those used for connecting to the Internet, sometimes use Ethernet interfaces. When connected in this way, a router that is also acting as a NAT device must be able to respond to ARP requests for *any* of its global IP addresses. Failure to do this prevents upstream devices such as ISP servers from forwarding packets to these (global) addresses, even though the router may be correctly configured.

For example, if a NAT router acts as a firewall and translates the source address of an outgoing packet to an address other than that of its own IP interface, the firewall router needs to ARP respond for this source address in order to receive and translate returning packets.

This feature is always enabled when NAT rules and interface-based NATs are created, so no configuration is required. It is possible to enable and disable ARP debugging on a firewall policy, and use the **show firewall arp** command to display the addresses for which the firewall may respond to ARP requests.

The following additional constraints ensure that NAT configurations do not interfere with normal network operations:

- ARP requests must be received on an interface for which the interface-based NAT or NAT rule applies.
- The IP address in the ARP request must fall within the subnet of the logical IP interface configured on the interface that received the request.

To enable the display of debugging information relating to ARP requests that are processed by the firewall, use the command:

```
enable firewall policy=policy-name debug=arp
```

To disable the display of debugging information relating to ARP requests that are processed by the firewall, use the command:

```
disable firewall policy=policy-name debug=arp
```

To display the addresses for which the firewall may ARP respond, use the command:

```
show firewall arp [policy=policy-name]
```

Example output for the **show firewall arp** command is shown in Figure 41-1:

Figure 41-1: Example output from the **show firewall arp** command

```
IP               ARP Interfaces  NAT Type  Int        Gbl Int    Rule
(range)          Policy
-------------------------------------------------------------------------------
172.20.8.50      Public          Int based eth0-0     eth1-0     -
                 Office
172.20.8.57      All Public      Rule      eth0-1     -          1
 -172.20.8.62    LAN
-------------------------------------------------------------------------------
```

Table 41-1: Parameters in the output of the **show firewall arp** command

Parameter	Meaning
IP (range)	An IP address (or range of addresses) for which the device may need to make an ARP response.
Policy	The name of the policy whose NAT configuration the IP address (range) belongs to.
ARP interfaces	The interfaces within the policy that ARP requests for the IP address (range) respond to; either"Public", "All Public", "Private", or "All Private". "Public" means that ARP requests are permitted on the public interface listed in the "Gbl Int" field. "Private" means ARP requests are permitted on the private interface specified by the "Int" field. "All Public" means ARP requests are permitted on all of the policy's public interfaces. "All Private" means ARP requests are permitted on all of the policy's private interfaces.
NAT Type	The type of NAT that the IP address (range) is associated with; one of "Int based" or "Rule". "Int based" means that the address (range) was specified by an interface-based NAT configuration with the **add firewall policy nat** command. "Rule" means the address (range) was specified by a NAT rule configured with the **add firewall policy rule** command with action=nat.
Int	The private interface associated with the NAT configuration. If NAT Type is "Int based", this is the private interface specified by the INTERFACE parameter in the **add firewall policy nat** command. If the NAT type is Rule, then this is the name of the interface to which the rule is attached when it is a private interface. A dash (-) indicates the rule is attached to a public interface (see the "Gbl Int" parameter).
Gbl Int	The public interface associated with the NAT configuration. If NAT Type is "Int based" then this is the public interface specified by the GBLINTERFACE parameter in the **add firewall policy nat** command. If the NAT type is Rule, then this is the name of the interface to which the rule is attached when it is a public interface. A dash (-) indicates that the rule is attached to a private interface (see the Int parameter).
Rule	The number of the rule to which this entry is associated. If NAT type is "Int based", no value is displayed.

Web Redirection with Reverse NAT Rules

The implementation of reverse NAT allows the firewall to perform Web Redirection. A NAT rule can be created that redirects HTTP traffic and sends it to one particular web server defined in the rule, regardless of where it was originally destined. Selector parameters may also be included in the rule to fine tune how to direct the traffic.

This feature is useful for ISPs in the travel and hospitality industry with users probably unknown to the ISP who want to plug their laptops into the ISP's LAN. With web redirection, traffic from a user's PC or laptop can be redirected to the ISP's web server. This makes the user arrange payment for the service before being able to browse to other sites. With appropriate supporting "deny" rules, all other traffic types from the user's PC can be blocked until payment is made.

The following gives a simple example of how a system such as this could be configured. The ISP has a switch configured with a firewall. The AR400 router's VLANs, vlan1 and vlan2, are private and public interfaces respectively. The ISP's web server has the IP address 205.1.28.6. The following rules perform the web redirection and the blocking of all non-web traffic:

```
add firewall policy=isp rule=298 interface=vlan1 action=nat
   nattype=reverse protocol=tcp port=80 gblremote=205.1.28.6

add firewall policy=isp rule=299 interface=vlan1 action=deny
   protocol=all
```

After a user has arranged payment, a rule can be added that specifies the IP address that the ISP assigns to the user, allowing the user full access to the service. The following is an example of such a rule. The user has been allocated the IP address 10.8.0.172. It is important that the rule number is lower than the blocking and redirecting rules because rules are tried in order from the lowest rule number until a match is found. A low number ensures that the allow rule is applied when appropriate, rather than other rules.

```
add firewall policy=isp rule=5 interface=vlan1 action=allow
   ip=10.8.0.172 protocol=all
```

If the ISP wants to take advantage of the time limited rules feature that allows a user access for 30 minutes, the following rule could be used instead.

```
add firewall policy=isp rule=5 interface=vlan1 action=allow
   ip=10.8.0.172 protocol=all ttl=0:30
```

UPnP

UPnP is an architecture that allows devices to automatically discover, negotiate, and request services. The UPnP implementation is closely related to the firewall. You must configure a firewall policy before you can use UPnP. Several of the commands needed to configure UPnP are firewall commands.

For details on UPnP, see Chapter 42, UPnP.

Dynamic Interfaces

The firewall supports dynamic interfaces as well as static interfaces. Adding dynamic interfaces to the firewall allows it to control incoming dynamically-created PPP connections (configured using the **create ppp template** command on page 9-62 of Chapter 9, Point-to-Point Protocol (PPP)).

If you have dynamic PPP connections and do not configure corresponding firewall dynamic interfaces, traffic sent via the dynamic PPP connections:

- are dropped when traffic is routed out a private interface
- bypass the firewall when traffic is routed out a public interface or an interface that is not attached to a firewall policy

Dynamic Interface Templates

Each firewall policy uses a *dynamic interface template* to process dynamic interfaces. The dynamic interface template name is used as a placeholder for adding dynamic interfaces to policies, NAT entries and rules, wherever an interface name is required.

To create a dynamic interface template and add it to a firewall policy, use the command:

```
create firewall policy=policy-name dynamic=template
```

where *template* is a 1 to 15-character string to conveniently identify this template. Note that the template name is not constrained by the name of the PPP template or the underlying physical interface.

When the remote device tries to open the dynamic PPP connection, PPP authenticates the link as required. Then the firewall needs to check whether that connection is allowed or denied, and to apply any rules or NAT settings to the traffic flow. To do this, the firewall uses a list of acceptable usernames that are associated with the policy. To specify these usernames individually, use the command:

```
add firewall policy=policy-name dynamic=template
   user=username
```

To create a text file with a list of usernames with one per line and associate it with the policy, use the command:

```
add firewall policy=policy-name dynamic=template
   file=filename.txt
```

When a dynamic interface is created by an incoming call, the username used to authenticate the incoming call is checked against the usernames assigned to each dynamic interface template. When a match is found, the dynamic interface inherits all the firewall attributes such as NATs and rules of the corresponding dynamic interface template.

Usernames are globally assigned to policies and dynamic interface templates. A single username should be assigned to one firewall dynamic interface template or policy. Two special usernames are reserved: NONE and ANY. The username NONE specifies dynamic interfaces that do not require authentication. The ANY username is used to match all authentication usernames. This lets you specify all PPP authenticated usernames by entering a single line of text.

To delete a single username or all names in a file from a dynamic interface template, use the commands:

```
delete firewall policy=policy dynamic=template user=username

delete firewall policy=policy dynamic=template
   file=filename.txt
```

To destroy a dynamic interface template, use the command:

```
destroy firewall policy=policy-name dynamic=template
```

Configuring Dynamic Interfaces

After a dynamic interface template is created and usernames assigned to it, the dynamic interface template can be specified as an interface in commands that add interfaces to firewall policies, NATs, and rules. The value DYN-*template* identifies the interface as a dynamic interface template, rather than a static interface. For example, if the dynamic interface template is called *remote,* the interface would be *dyn-remote*.

To add or remove dynamic interfaces from firewall policies, use the commands:

```
add firewall policy=policy-name interface=dyn-template
   type={private|public} [method={dynamic|passall}]

delete firewall policy=policy-name interface=dyn-template
```

To add or remove rules from dynamic interfaces, use the commands:

```
add firewall policy=policy-name rule=rule-id
   interface=dyn-template other-options...

delete firewall policy=policy-name rule=rule-id
```

To add or remove NATs from dynamic interfaces, use the commands:

```
add firewall policy=policy-name nat={enhanced|standard}
   interface=dyn-template [ip=ipadd] gblinterface=interface
   [gblip=ipadd[-ipadd]]

delete firewall policy=policy-name nat={enhanced|standard}
   interface=dyn-template gblinterface=interface [ip=ipadd]
```

A dynamic interface template cannot be added to a global interface in a NAT definition because a dynamic interface is never directly assigned an IP address. A global interface must have a global address, which must be a real globally unique Internet address.

Firewall SMTP Proxy (Application Gateway)

Abuse of email systems on the Internet is very common. Abuse can be as simple as someone sending unwanted emails, known as *spam*, and includes glutting an entire server with spam without permission from the owner of the server. The consequences of abuse range from minor inconvenience through to total failure of mail servers.

The firewall is a convenient place to attempt to shield a mail server on either a private intranet or the public Internet from the consequences of email system abuse. Adding an SMTP proxy to a firewall policy means that the SMTP proxy inspects SMTP packets that pass through the firewall and accepts or rejects sessions based on the source and destination email addresses involved.

Interaction Between the SMTP Proxy and Firewall Rules

It is not necessary to provide a rule in the firewall policy to permit traffic that passes in and out through the firewall using the SMTP policy. By default, all traffic using the proxy is "allowed" to pass. However, rules can be added to the policy, for example rules to deny SMTP sessions through the proxy from particular IP addresses.

Protecting the Email System

The SMTP application gateway protects against the following:

- Third party relaying of email.
  A third party mail relay occurs when a mail server processes a mail message where neither the sender or the recipient is a local user. The mail server is an entirely unrelated party to mail processing. If an email originates from the public side of the firewall, the firewall SMTP proxy rejects it if the address in the “RCPT TO” field has a different domain name to a mail server on the private side of the firewall. If an email originates from the private side of the firewall, the firewall SMTP proxy rejects it when the domain name:
  - in the MAIL FROM field is different to the domain name specified by the **set firewall policy smtpdomain** command, or
  - in the RCPT TO field is not consistent with IP address of the IP packet

  Note that for the latter to occur, a DNS server must be setup using the **add ip dns** command on page 14-65 of Chapter 14, Internet Protocol (IP). If a DNS server is not configured, the proxy checks the email based on the MAIL FROM field.
- Spam email.
  The firewall SMTP proxy rejects email sent from email addresses or domains specifically identified as spam sources in the firewall policy. The user maintains a list of spam sources in a text file on the router’s file subsystem. The text file consists of one or more single line entries each containing an email address or a domain name that has been identified as a source of spam. Messages are rejected that are received with one of the listed addresses or domains as its source. More detail on the format of the text file is in the **add firewall policy spamsources** command on page 41-50.

- Smurf Amp email attacks.
  In a Smurf Amp attack, the attacker broadcasts a TCP SYN packet (a TCP Synchronisation packet is the first packet in a TCP session) for an SMTP session with a source address that belongs to the intended victim. Any SMTP servers that receive the packet all respond to the source, potentially swamping the victim with SYN ACK (Synchronisation Acknowledge) packets. The responses of the SMTP servers amplify the original SYN from the attacker, hence the term *Amp*. While the attack does not have a serious impact on the router when running an SMTP proxy, it exploits the router in order to inconvenience the victim. To prevent such an attack, the router discards SYN packets received by the SMTP proxy that have a broadcast destination address.

Email Relaying

The firewall SMTP proxy can relay any email that originates from the private side of the firewall. This happens when the IP packets for the email are only destined to the private interface of the firewall. The proxy forwards the email to the final destination specified in the "RCPT TO" field. Note that the relaying function requires that a DNS server is setup using the**add ip dns** command on page 14-65 of Chapter 14, Internet Protocol (IP).

Firewall HTTP Application Gateway (Proxy)

The firewall's HTTP proxy (Application Gateway) filters outbound HTTP sessions based on the URLs requested, and block the setting of all cookies, or cookies requested from servers in a specific domain. The Firewall HTTP Application Gateway requires an HTTP Proxy special feature licence and an Application Gateway special feature licence in addition to the firewall licence.

Firewall HTTP Proxies and Firewall Policies

To add or delete a firewall HTTP proxy, use the HTTP option for the PROXY parameter in the commands:

```
add firewall policy=policy-name proxy={http|smtp}
   interface=interface gblinterface=interface direction={in|
   out|both} [ip=ipadd] [days=day-list] [after=hh:mm]
   [before=hh:mm]

delete firewall policy=policy-name proxy={http|smtp}
   interface=interface gblinterface=interface direction={in|
   out|both} [ip=ipadd]
```

HTTP Filters

To add to or delete from the HTTP filter for a firewall policy, use the commands:

```
add firewall policy=policy-name httpfilter=filename
    [direction={in|out}]

delete firewall policy=policy-name httpfilter=filename
    [direction={in|out}]
```

These commands add or delete the contents of a HTTP filter file from the HTTP filter of the specific firewall policy. The HTTP filter file contains a list of URLs, keywords, and cookie settings that filter traffic traversing the HTTP proxy. IP addresses may also be specified in the filter file.

HTTP Cookies

By default, HTTP cookie requests are allowed to pass through the HTTP proxy configured under the firewall policy. To discard cookie sets from particular domains or URLs, put entries in the filter file for the direction in which you want to filter, as described above. To configure the HTTP proxy to discard all HTTP cookie sets from all responses, use the command:

```
disable firewall policy=policy-name httpcookies
```

To re-enable HTTP cookie requests to pass through the HTTP proxy, use the command:

```
enable firewall policy=policy-name httpcookies
```

Logging

URL requests and cookies that are denied are logged by the firewall and for each denial an entry appears in the list of recent firewall "deny events". To display this list, use the command:

```
show firewall event=deny
```

If the event is for a denied URL request then up to 29 characters of the requested URL are displayed. If the event is for a blocked cookie then up to 18 characters of the name of the domain trying to set the cookie are displayed.

An entry similar to that for the **show firewall event** command is also placed in the router log (see Chapter 33, Logging Facility). To view this entry use the command:

```
show log
```

The firewall can be configured to send notification of "deny events" (see "Notifications" on page 41-22).

Disabling the TCP Set-up Proxy

The firewall's TCP setup proxy for TCP connections initiated from the public side of the firewall can be disabled for a specific firewall policy. This lets a permitted firewall TCP session initiated from a public host to connect directly to hosts on the private network.

The firewall's TCP setup proxy is enabled by default. When the TCP proxy is disabled, the load balancer cannot be used.

To disable the setup proxy for a specified firewall policy, use the command:

```
disable firewall policy=policy-name tcpsetupproxy
```

To enable the setup proxy for a firewall for which the setup proxy had been previously disabled, use the command:

```
enable firewall policy=policy-name tcpsetupproxy
```

Take care when using the **disable firewall policy tcpsetupproxy** command because it can reduce the security of the firewall and leave the private network vulnerable to attack such as an SYN flood.

Enabling the Secure Shell Server

If you have a firewall configured, and you want the router to act as a Secure Shell (SSH) server, you need to add a firewall rule that accepts Secure Shell connections. Once this rule has been added, you can enable the Secure Shell server.

For example, to add Secure Shell access over port 22 with a public IP address of 200.200.200.1, a private IP address of 192.168.1.1, a public interface of ppp0, and a remote IP address of 200.200.200.5, use the command:

```
add firewall policy=main rule=1 action=allow interface=ppp0
   protocol=tcp port=22 ipaddress=192.168.1.1
   gblip=200.200.200.1 gblport=22 remote=200.200.200.5
```

Monitoring Firewall Activity

The firewall provides a range of options for monitoring the configuration of the firewall itself, as well as firewall events, access control and attacks.

Notifications

The firewall can be configured to send notifications about significant firewall events to one or more of the following destinations:

- An email address. See "Mail Subsystem" on page 1-46 of Chapter 1, Operation for information about configuring the mail subsystem.
- All terminal and Telnet sessions logged in with Manager privilege.
- An asynchronous port
- An SNMP trap host. See Chapter 38, Simple Network Management Protocol (SNMP) for information about configuring SNMP trap hosts.

The size of event (what constitutes a "significant event") required to generate these notifications can be set.

To set the threshold levels at which notifications and triggers are generated for attack events, use the command:

```
set firewall policy=policy-name attack={dosfloow|fragment|
   hostscan|ipspoof|land|pingofdeath|portscan|smtrelay|
   smurf|smurfamp|spam|synattack|tcptiny|udpattack}
   [intrigger=count] [outtrigger=count] [detail=count]
   [time=minutes]
```

See the **set firewall policy attack** command on page 41-74 for more information.

To enable or disable notification destinations, use the commands:

```
enable firewall notify={all|mail|manager|port|snmp}
disable firewall notify={all|mail|manager|port|snmp}
```

To display a history of recent events, use the command:

```
show firewall event
```

Debugging

To enable or disable debugging on a per-policy basis, use the commands:

```
enable firewall policy=name debug={all|packet|pkt|process}
disable firewall policy=name debug={all|packet|pkt|process}
```

Event Triggers

The firewall forwards the following events to the Trigger Facility:

- DOSATTACK—A denial of service attack in which a remote user continually sends unwanted traffic.
- FRAGATTACK—An attack using TCP fragments that are either too large or can never be reassembled.
- HOSTSCAN—A scan of the hosts of the private network.

- PORTSCAN—A portscan of the firewall or private network.
- SMTPATTACK—An attack where email is received that is unwanted either because it is from a source identified as a source of spam, it is attempting to use a mail server as a third party relay, or it has a broadcast reply address.
- SMURFATTACK—An *Internet Control Message Protocol* (ICMP) echo request with a broadcast destination address.
- SYNATTACK—An attack on a host using multiple opening TCP SYN packets to exhaust a host's available sessions or memory.
- TCPATTACK—An attack on a host using TCP tiny fragments.

The Trigger Facility can be configured to respond to these events by running management-defined scripts. Triggers can be activated by the start or end of an event. See Chapter 30, Trigger Facility for more information about creating triggers to respond to firewall events.

To set the threshold levels at which notifications and triggers are generated for attack events, use the command:

```
set firewall policy=policy attack={dosflood|fragment|
   hostscan|ipspoof|land|pingofdeath|portscan|smtpattack|
   smurfattack|synattack|tcptiny|udpattack}
   [intrigger=count] [outtrigger=count] [detail=count]
   [time=minutes]
```

To display the firewall trigger threshold levels, use the command:

```
show firewall policy attack
```

Logging

The firewall can be configured to log an extensive range of events to the router's Logging Facility (Table 41-2).

Table 41-2: Log types and subtypes for firewall events .

Option	Meaning
INATCP	Logs the start of TCP sessions initiated from the public Internet.
INAUDP	Logs the start of a UDP flow initiated from the public Internet.
INAICMP	Logs a ICMP request initiated from the public Internet.
INAOTHER	Logs the start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet.
INALLOW	Logs the start of all incoming allowed sessions and flows, and is the sum of the previous four values.
OUTATCP	Logs the start of TCP sessions initiated from the private Intranet.
OUTAUDP	Logs the start of a UDP flow initiated from the private Intranet.
OUTAICMP	Logs a ICMP request initiated from the private Intranet.
OUTAOTHER	Logs the start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the private Intranet.
OUTALLOW	Logs the start of all allowed outgoing sessions and flows, and is the sum of the previous four values.
ALLOW	Logs the start of all allowed flows and sessions both in and out of the firewall.
INDTCP	Logs the failed start of TCP sessions initiated from the public Internet.

Table 41-2: Log types and subtypes for firewall events (continued).

Option	Meaning
INDUDP	Logs the failed start of a UDP flow initiated from the public Internet.
INDICMP	Logs a failed ICMP request initiated from the public Internet.
INDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet.
INDENY	Logs the failed start of all denied incoming sessions and flows, and is the sum of the previous four values.
OUTDTCP	Logs the failed start of TCP sessions initiated from the private Intranet.
OUTDUDP	Logs the failed start of a UDP flow initiated from the private Intranet.
OUTDICMP	Logs a failed ICMP request initiated from the private Intranet.
OUTDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the private Intranet.
OUTDENY	Logs the failed start of all denied outgoing sessions and flows, and is the sum of the previous four values.
DENY	Logs the failed start of all flows and sessions both in and out of the firewall.
INDDTCP	Logs the failed start of TCP sessions initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDUDP	Logs the failed start of a UDP flow initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDICMP	Logs a failed ICMP request initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP or ICMP) initiated from the public Internet. Up to 192 bytes of the IP packet are also logged.
INDDUMP	Logs the failed start of all denied incoming sessions and flows, and is the sum of the previous four values. Up to 192 bytes of the IP packet are also logged.
OUTDDTCP	Logs the failed start of TCP sessions initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDUDP	Logs the failed start of a UDP flow initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDICMP	Logs a failed ICMP request initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDOTHER	Logs the failed start of an IP protocol flow (other than TCP, UDP, and ICMP) initiated from the private Intranet. Up to 192 bytes of the IP packet are also logged.
OUTDDUMP	Logs the failed start of all denied OUT sessions and flows, and is the sum of the previous four values. Up to 192 bytes of the IP packet are also logged.
DENYDUMP	Logs the failed start of all flows and sessions both in and out of the firewall. Up to 192 bytes of the IP packet are also logged.
EVERYDENY	If EVERYDENY is enabled, every instance of a deny that matches one of the deny LOG options that are enabled is logged. This may result in a large number of log entries. If EVERYDENY is disabled, only the first instance of a deny for a given source IP, destination IP, and protocol combination is logged in a two minute period if a matching deny LOG option is enabled. The EVERYDENY option by itself does not cause any logging to occur. The default is for EVERYDENY to be disabled.

Logging specific firewall events can be enabled or disabled on a per-policy basis by using the commands:

```
enable firewall policy=name log={allow|deny|denydump|
    everydeny|inaicmp|inallow|inaother|inatcp|inaudp|
    inddicmp|inddother|inddtcp|inddudp|inddump|indeny|
    indicmp|indother|indtcp|indudp|outaicmp|outallow|
    outaother|outatcp|outaudp|outddicmp|outddother|outddtcp|
    outddudp|outddump|outdeny|outdicmp|outdother|outdtcp|
    outdudp}
disable firewall policy=name log={allow|deny|denydump|
    everydeny|inaicmp|inallow|inaother|inatcp|inaudp|
    inddicmp|inddother|inddtcp|inddudp|inddump|indeny|
    indicmp|indother|indtcp|indudp|outaicmp|outallow|
    outaother|outatcp|outaudp|outddicmp|outddother|outddtcp|
    outddudp|outddump|outdeny|outdicmp|outdother|outdtcp|
    outdudp}
```

Several options can be enabled or disabled in a single invocation by specifying the options as a comma-separated list, for example:

```
enable firewall policy=office log=indeny,outdeny
```

To minimise the number of log messages generated by the firewall, for some events the first four packets are logged, then the first packet is repeated with the text "`(x number)`" appended to indicate the number of repeat messages.

Firewall log messages are processed by the default TEMPORARY special log output definition. The TEMPORARY log output definition contains a log message filter that matches log messages with severity 3 or greater. However, the severity level for some firewall events is less than 3. Therefore, while the logging of a particular firewall event may be enabled in a firewall policy, the log messages generated by that event are processed by the TEMPORARY log output definition when their severity is 3 or greater (see Chapter 33, Logging Facility).

Further configuration is required to log firewall events whose log messages are assigned a severity of less than 3. The log types requiring additional configuration are listed in Table 41-3 on page 41-25.

Table 41-3: Log Types and Subtypes Requiring Additional Configuration .

Option	Additional Configuration
OUTATCP	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTATCP
OUTAUDP	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAUDP
OUTAICMP	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAICMP
OUTAOTHER	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAOTHER
OUTALLOW	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTATCP ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAUDP ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAICMP ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAOTHER
CONFCHNG	ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=CONFCHNG

Enhanced Packet Fragment Handling

The default firewall policy behaviour is that fragmented packets are only permitted by the policy if there are no more than 8 fragments and the combined protocol data consists of 1780 bytes, or less.

When enhanced packet fragment handling is enabled, the firewall policy permits the forwarding of IP packets of the specified protocol type that have been fragmented into more than 8 fragments. Packet fragment handling can be performed on UDP, ICMP, and other protocol types, excluding TCP. If packet fragment handling is enabled, the default maximum number of fragments that an IP packet may consist of is 20. The maximum number of fragments able to be specified is 50.

Enhanced packet fragment handling is disabled by default.

To set the maximum number of fragments that a fragmented IP packet may consist of when packet fragment handling is enabled, use the command:

```
set firewall maxfragments=8..50
```

To enable packet fragment handling, use the command:

```
enable firewall policy=policy-name
   [fragments={icmp|udp|other}[,...]]
```

To disable packet fragment handling, use the command:

```
disable firewall policy=policy-name
   [fragments={icmp|udp|other}[,...]]
```

Accounting

The firewall maintains accounting information that enables the firewall manager to determine the effect that various firewall policies are having on traffic flow. Accounting can be enabled or disabled on a per-policy basis using the commands:

```
enable firewall policy=policy-name accounting
```

```
disable firewall policy=policy-name accounting
```

To display the currently stored accounting records, use the command:

```
show firewall accounting [policy=policy-name]
   [reverse=number] [tail=number]
```

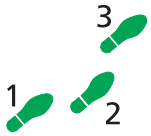
Accounting records are also written to the Logging Facility, with a severity of 3. The log can be displayed with the **show log** command on page 33-34 of Chapter 33, Logging Facility. This logging information can also be sent to a syslog daemon if required. See Chapter 33, Logging Facility.

Configuration Examples

The following examples illustrate the steps required to configure the firewall for a range of applications. The configurations provides very good firewall protection for a number of common router configurations. In particular, when a host on a network connected to a private interface initiates a session (TCP) or flow (UDP) to a host reachable by a public interface, then only context sensitive traffic relating to that session or flow is allowed back through the firewall. The firewall drops traffic initiated from hosts reachable by a public interface. The exception is when special filter rules have been added (see the fourth example below). Further, most common denial of service attacks are logged and combated by the firewall.

Minimum Configuration for a Small Office

This example illustrates how to configure the most basic firewall for a small office wanting to be as secure as possible without restricting access to the public Internet. The office computers are connected to the router via an Ethernet interface. The Ethernet interface has been assigned the global IP addresses 202.49.74.0 to 202.49.74.255. The PPP interface has been assigned a single global Internet address 202.49.72.2.



To configure a firewall without restricting access to the public Internet

1. **Create the security policy.**

   To create a policy named "office", use the command:

   ```
   create firewall policy=office
   ```

2. **Add the interfaces to the security policy.**

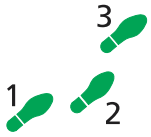
   To add Ethernet and PPP interfaces to the policy, use the commands:

   ```
   add firewall policy=office interface=eth0 type=private
   add firewall policy=office interface=ppp0 type=public
      method=dynamic
   ```

   Since externally initiated access to hosts on the private network is not required, no further configuration is necessary. When at least one private and one public interface are added to a policy, the policy is operational.

A Firewall with an ISP-Assigned Internet Address

This example illustrates how to configure a firewall for a small office that is dynamically assigned a single global Internet address by its ISP when the router connects to the ISP and negotiates an IP option for the PPP link. NAT must be used on the private network for this reason. The office computers are connected to the router via an Ethernet interface, and there is a connection to the Internet over a PPP interface. The Ethernet interface uses the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface is dynamically assigned a single global Internet address by the ISP. For more information about configuring PPP, see Chapter 9, Point-to-Point Protocol (PPP).



To configure Firewall with a single global Internet address from an ISP

1. **Create the security policy.**

   To create a policy named "office", use the command:

   ```
   create firewall policy=office
   ```

2. **Add the interfaces to the security policy.**

   To add the Ethernet and PPP interfaces to the policy, use the commands:

   ```
   add firewall policy=office interface=eth0 type=private
   add firewall policy=office interface=ppp0 type=public
      method=dynamic
   ```

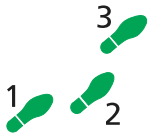
3. **Add the NAT mapping to the private interface.**

   To add NAT mapping to the Ethernet interface to translate private IP addresses to the dynamically assigned global IP address, use the command:

   ```
   add firewall policy=office nat=enhanced interface=eth0
      gblinterface=ppp0
   ```

A Firewall with a Single Global Internet Address

This example is similar to the previous example, except that the ISP has assigned a single static global Internet address to the office. NAT must be used on the private network to translate private IP addresses to the global IP address. The office computers are connected to the router via an Ethernet interface, and there is a connection to the Internet over PPP. The Ethernet interface uses the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface has been assigned the global Internet address 202.49.72.2.



To configure Firewall with a single global Internet address

1. **Create the security policy.**

   To create a policy named "office", use the command:

   ```
   create firewall policy=office
   ```

2. **Add the interfaces to the security policy.**

   To add the Ethernet and PPP interfaces to the policy, use the commands:

   ```
   add firewall policy=office interface=eth0 type=private
   add firewall policy=office interface=ppp0 type=public
      method=dynamic
   ```

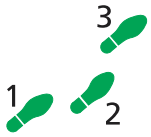
3. **Add the NAT mapping to the private interface.**

   To add a NAT mapping to the Ethernet interface to translate private IP addresses to the statically assigned global IP address, use the command:

   ```
   add firewall policy=office nat=enhanced interface=eth0
      gblinterface=ppp0 gblip=202.49.72.2
   ```

Allowing Access to a WWW Server

This example builds on the previous example by allowing access from the public Internet to a WWW server on the private network. The office has been assigned a single global Internet address by its ISP. For this reason NAT must be used on the private network. The office computers are connected to the router via an Ethernet interface, and there is a connection to the Internet over PPP. The Ethernet interface uses the private IP network addresses 192.168.10.0 to 192.168.10.255. The PPP interface has been assigned the single global Internet address 202.49.72.2. The office wants to provide access to a WWW server on the private network to advertise its products.



To configure firewall to allow access to a WWW server

1. **Create the security policy.**

   To create a policy named "office", use the command:

   ```
   create firewall policy=office
   ```

2. **Add the interfaces to the security policy.**

   To add the Ethernet and PPP interfaces to the policy, use the commands:

   ```
   add firewall policy=office interface=eth0 type=private
   add firewall policy=office interface=ppp0 type=public
      method=dynamic
   ```

3. **Add the NAT mapping to the private interface.**

   To add a NAT mapping to the Ethernet interface to translate private IP addresses to the statically assigned global IP address, use the command:

   ```
   add firewall policy=office nat=enhanced interface=eth0
      gblinterface=ppp0 gblip=202.49.72.2
   ```

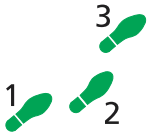
4. **Add a rule to allow access to the WWW server.**

   The basic firewall configuration does not allow hosts on the private network to be accessed from the public network. To allow access to the office WWW server behind the firewall, add a rule to allow access to the WWW server at IP address 192.168.10.12 from the public Internet. Web browsers and web servers interact using the HTTP protocol, which is a TCP/IP-based protocol using a well-known port, so the rule must allow TCP traffic to the HTTP port to pass from the public interface to the private interface:

   ```
   add firewall policy=office rule=1 action=allow
      interface=ppp0 ip=192.168.10.12 protocol=tcp port=http
      gblip=202.49.72.2 gblport=http
   ```

A Firewall with TCP Setup Proxy Disabled for File Sharing

This example illustrates how to configure a firewall for file sharing between public and private hosts that use different Windows operating systems. When a public host attempts to connect to a private host using a Windows operating system for the purpose of file sharing, two protocols may be used, depending upon the Windows version in use. If the public host is Windows 2000 or higher, a connection using direct hosting of SMB is attempted. If the private host is Windows 95/98, direct hosting of SMB is not supported and is rejected by the private host. If direct hosting of SMB is rejected, the Windows 2000 or higher public host tries to connect using NetBIOS, which is acceptable by the private Windows 95/98 host.



To configure the Firewall to forward TCP connections on port 139 or port 445 to the destination private host

1. **Create the security policy.**

   To create a policy named *zone1*, use the command:

   ```
   create firewall policy=zone1
   ```

2. **Add the interfaces to the security policy.**

   To add Ethernet and PPP interfaces to the policy, use the commands:

   ```
   add firewall policy=zone1 interface=eth0 type=private
   add firewall policy=zone1 interface=ppp0 type=public
   ```

3. **Add TCP rules to for the selected ports.**

   To create rules that allow connections using direct hosting of SMB (TCP port 445) and SMB using NetBIOS (TCP port 139) access to the private network, use the commands:

   ```
   add firewall policy=zone1 rule=1 action=allow
      interface=ppp0 protocol=tcp port=445
   add firewall policy=zone1 rule=2 action=allow
      interface=ppp0 protocol=tcp port=139
   ```

4. **Disable the TCP setup proxy.**

   To disable the TCP setup proxy so that TCP connections on port 445 or port 139 are forwarded to the destination private host for acceptance or rejection, use the command;

   ```
   disabled firewall policy=zone1 tcpsetupproxy
   ```

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor the firewall. The firewall requires IP to be enabled and configured correctly. See Chapter 14, Internet Protocol (IP) for the commands required to enable and configure IP.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page xcv of Preface in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

add firewall policy apprule

Syntax

```
ADD FIREwall POLIcy=policy-name APPRule=app-rule-id
   ACtion={ALLOw|DENY} INTerface=interface
   APPlication={FTP|TELnet|SMTP|TIME|DNS|BOOTPS|BOOTPC|
   TFTP|GOPHer|FINGer|WWW|HTTP|KERBeros|RTELnet|POP2|POP3|
   RTSP|SNMPTRap|SNMP|VDOlive|REALAudio|REALVideo|CUSEeme|
   XING|QUICktime|MMS|BBMS} [COMmand={GET|PUT}]
   [POrt=port]
```

where:

- *app-rule-id* is a number from 1 to 299.
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).
- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *port* is an Internet service port number or name.

Description

This command defines rules for managing application traffic between interfaces covered by the firewall policy, and is modelled on the **add firewall policy rule** command.

The APPRULE parameter specifies both an identifier for the rule and the position of the rule in the list of rules for this policy. Rules are processed in order, from the lowest to the highest numbered rule. The identifier is used to refer to this rule in other commands.

The ACTION parameter specifies what the firewall should do with traffic that matches the selectors defined for this rule. If ALLOW is specified, the traffic is permitted to pass through the firewall. If DENY is specified, the traffic is prevented from passing through the firewall.

The INTERFACE parameter specifies a valid interface. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The APPLICATION parameter specifies the name of an application for which the session flows are to be modified by this rule. The REALAUDIO, REALVIDEO, MMS and BBMS applications all listen on port 7070. If REALAUDIO or REALVIDEO is specified, port 7070 is used for the streaming protocol PNA. If any other option is specified, ports 7070, 554 and 7071 are used for Real Time Streaming Protocol (RTSP). For any option except MMS or BBMS, at least one of the parameters COMMAND or PORT is required.

The COMMAND parameter specifies a comma-separated list of keywords, dependent on the application. GET and PUT are currently supported, representing the FTP STOR and RETR commands (RFC 959), respectively. The COMMAND parameter is valid when APPLICATION is set to FTP. Application protocol packets containing these commands are allowed through the firewall or removed from the flow, depending on how the ACTION parameter is set.

The PORT parameter allows an alternate port to be used for the application, and for flows to the specified port to be treated as flows for that application.

The **show firewall policy** command displays information about application rules that have been defined (Figure 41-8 on page 41-87, Table 41-12 on page 41-88).

Examples

To remove FTP STOR commands from FTP application flows originating on public interface ppp0 (and therefore preventing public users from uploading files to an internal FTP server), use the command:

```
add fire poli=admin appr=1 ac=deny int=ppp0 app=ftp com=put
```

To identify all flows destined for TCP port 688 as KERBEROS sessions so that the firewall applies KERBEROS application rules to flows destined for port 688, use the command:

```
add fire poli=admin appr=2 ac=deny int=ppp0 app=kerberos
   po=688
```

To identify flows destined for TCP port 1755 as MMS sessions so that the firewall applies MMS application rules to flows destined for that port, use the command:

```
add fire poli=admin appr=2 ac=allo int=ppp0 app=mms
```

Related Commands

delete firewall policy apprule
show firewall policy

add firewall policy dynamic

Syntax

```
ADD FIREwall POLIcy=policy-name DYnamic=template
   {FIle=filename.txt|USer={username|ANY|NONE}}
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *template* is a character string 1 to 15 characters long. Valid characters are any printable character. If *template* includes spaces, it must be in double quotes.
- *username* is a character string 1 to 63 characters long. Valid characters are any printable character. If *username* includes spaces, it must be in double quotes.
- *filename.txt* is the name of a file on the router.

Description

This command adds one user or a list of users from a file to the specified dynamic interface template for a policy.

The FILE parameter specifies the file containing a list of users to be added. The text must have one username per line.

The USER parameter specifies the user to be added. Two special usernames are reserved, NONE and ANY. The username NONE is used to specify dynamic interfaces that do not require authentication. The ANY username is used to match all authentication usernames. This allows the one catch-all for all authenticated usernames. A single username can be assigned to only one firewall dynamic interface template or policy.

Example

To add user *anna* to the dynamic interface template *remote* for the *management* policy, use the command:

```
add fire poli=management dy=remote us=anna
```

Related Commands

delete firewall policy dynamic
show firewall policy

add firewall policy httpfilter

Syntax

```
ADD FIREwall POLIcy=policy-name HTTPFilter=filename
   [DIrection={IN|OUT}]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *filename* is the name of a file on the router.

Description This command adds the contents of a HTTP filter file to the HTTP filter of the specified firewall policy. The HTTP filter file contains a list of URLs, keywords and cookie settings that are used to filter the traffic traversing the HTTP proxy.

The POLICY parameter specifies the policy where the HTTP filter file is to be added. It must already exist.

The HTTPFILTER parameter specifies the name of the HTTP filter file. The filter file is a file type with a TXT extension containing zero or more single line entries. The string `keywords:` must be placed at the beginning of the file and is used to start the keyword section. Keywords can be placed on the same line if they are separated by a space or placed on separate lines. The URL section is indicated by a `URLS:` keyword as the first word on the line. URL entries must contain full domain, directory, and folder names. IP addresses may also be specified in the filter file. Only one domain is allowed per line. Options are supplied after the entry and a colon. Each option is separated by a space.

The option keywords that are allowed for each entry are "allow" and "nocookies". The "allow" option explicitly allows the URL, or part of it, given on the line. This is useful for exceptions to a deny filter or a given keyword. The "nocookies" option specifies that the proxy should not accept cookie requests from the domain or URL given, and implicitly allows the URL. Comments may be placed in the file using a `#` character on the beginning of the line. White space before and after an entry does not affect the parsing of the file but there must be white space between the URL and colon for the options. After the colon, white space is not needed but there must be white space between each option specified. Empty lines are also allowed. Note that all URL entries without options are considered to be denied.

How specific the URLs are determines the order of precedence of the entries in the file. For example, `www.plant.com/this/is/a/url/grow.html` would have more precedence than a entry containing `www.plant.com/this`. Also, if the allow option is specified, it takes greater precedence than a similar entry with deny. If there is an allow rule in the filter file for `www.somewhere.com/somepage.html` and the IP address for `www.somewhere.com (192.168.1.13)` is in the filter file, the request is denied because the domain name server lookup for `www.somewhere.com/somepage.html` returns the following IP 192.168.1.13/somepage.html: allow is placed in the filter file and the request is allowed. Finally, keywords in the file take the least precedence. They are applied to sections of the URL, not part of the closest fitting URL entry.

Figure 41-2 on page 41-35 contains an example of a URL filter file.

Figure 41-2: Example of a HTTP filter file.

```
# The keywords section starts with the string "keywords:".
keywords:
# The keywords can match any part of the URL. URLs containing these entries are
# denied unless specifically allowed by an entry later in the file.
sex
plants
toys
.nz
# Putting a * in front of the keyword indicates that the string must appear at
# the end of the URL, for the URL to be denied. The following entry would match
www.anything.com/this/is/an/example, but not www.example.com
*example
# The * operator can be used to specify the type of file.
*.mp3
*.jpg

# The URLs section starts with the string "URLS:", and specifies particular URLs
# to deny, allow or cookie filter.
URLS:

# If no explicit deny is put on the end then the URL is denied.
# Note the implicit /* on the end of the domain.
www.plant.com
www.nude.com

# Specific sections of websites can be matched. The sections must be complete
# folder/directory names, so the following entry would match
# www.hacker.com/dosAttack/dos.html but not www.hacker.com/dosAttacks/dos.html
www.hacker.com/dosAttack

# The "nocookies" option denies cookie requests from the domain, and makes an
# implicit allow.
www.acompany.com: nocookies

# The "allow" option can be used to override general URL exclusions.
www.nude.com/this/is/not/porn : allow

# The "allow" option can also be used to override general keyword exclusions.
www.sexy.plants.com : allow

# The "allow" and "nocookies" options can be combined to allow a URL that is
# forbidden by the keywords, but deny cookie requests.
www.acompany.co.nz : allow nocookies
```

In order to edit the contents of the list generated from the HTTP filter file held in the firewall policy it must be deleted from the firewall policy (using the **delete firewall policy httpfilter** command), edited and then added to the firewall policy again. Alternatively, the file may be edited. Optionally, restarting the device reloads the filter file. Editing alone does not alter the configuration held in the policy. No more than 5 URL filter files may be attached to a policy at one time.

The DIRECTION parameter specifies the direction of HTTP sessions to which the filter is to be applied. If IN is specified, the filter applies to HTTP requests that originate on the public side of the firewall (inbound). If OUT is specified, the filter applies to HTTP requests that originate on the private side of the firewall (outbound). The default is OUT.

URL filters have no effect unless the specified policy also has an HTTP proxy configured with a direction that matches the direction specified for the URL filter.

Examples To add the contents of the file `banned.txt` to the HTTP filter of firewall policy zone1 for filtering outbound HTTP sessions, use the command:

```
add fire poli=zone1 httpf=banned.txt
```

Related Commands

add firewall policy proxy
create firewall policy
delete firewall policy httpfilter
delete firewall policy proxy
disable firewall policy httpcookies
enable firewall policy httpcookies
show firewall policy

add firewall policy interface

Syntax

```
ADD FIREwall POLIcy=policy-name INTerface=interface
   TYpe={PUBlic|PRIVate} [METhod={DYnamic|PASsall}]
   [TRUSTPRIVATE=FALSE|NO|YES|TRUE] [UPNPTYPE={LAN|WAN}]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).

Description This command adds an interface to the specified policy. The completed policy must contain at least one private interface and at least one public interface. An interface can only be specified as "private" in one policy. An interface can be specified as "public" in multiple policies. Multiple interfaces specified in a policy as "private" exchange packets without intervention from the firewall.

The POLICY parameter specifies the policy to which the interface is to be added. The specified policy must already exist.

The INTERFACE parameter specifies an existing IP interface to be added to the policy. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The TYPE parameter specifies whether the interface is to be treated as a private interface (inside the firewall) or a public interface (outside the firewall).

The METHOD parameter specifies the method to be used by the firewall to pass packets between private and public interfaces, and is only valid if TYPE is set to PUBLIC. If PASSALL is specified, the firewall does not interfere with packet flow. This option should only be selected to allow an interface to run 1:1 NAT translation as defined in RFC 1631. If DYNAMIC is specified, dynamic packet filtering is used. The default is DYNAMIC.

The TRUSTPRIVATE parameter specifies whether devices connected to the interface are trusted enough to have access to the router via the private interface IP address that is unrestricted by the firewall policy. This parameter may only be specified when TYPE is PRIVATE. (Access to the router by devices connected to public interfaces is always restricted by the firewall.) If YES or TRUE is specified traffic from devices connected to the interface and destined for the interfaces IP address or the address of another private interface within the same policy is always permitted regardless of any rules that may be defined for the interface or values specified for the ICMP_FORWARDING and PING parameters of the **enable firewall policy** command. If NO or FALSE is specified traffic from devices connected to the interface and destined for the interfaces IP address or the address of another private interface within the same policy is subject to the interfaces rules and the values specified for the ICMP_FORWARDING and PING parameters of the **enable firewall policy** command. The default is YES.

The UPNPTYPE parameter adds the specified interface to UPnP as either a LAN or WAN interface. A maximum of 64 interfaces may be specified as LAN interfaces for UPnP on the policy. One interface can be defined as WAN interface for UPnP. An interface added with TYPE set to PUBLIC cannot have the UPNPTYPE set to LAN, and if TYPE is PRIVATE, UPNPTYPE cannot be WAN. Use the **set firewall policy upnp** command to specify a firewall policy name that has already been enabled for UPnP or one that will be enabled.

Examples

To add an interface to an existing policy named "zone1", use the command:

```
add fire poli=zone1 int=eth0 ty=priv
```

To add a WAN interface operating over PPP0 to the policy named "zone1", use the command:

```
add fire poli=zone1 int=ppp0 ty=pub met=pas
```

To add *vlan2* as a private interface to the firewall policy *dmz*, and use the firewall policy to restrict access to the router for devices connected to this interface, use the command:

```
add fire poli=dmz int=vlan2 ty=priv trustprivate=no
```

To add eth0 as a public interface for the firewall policy called "upnp", use the command:

```
add fire poli=upnp int=eth0 ty=pub upnptype=wan
```

Related Commands

create firewall policy
create firewall policy dynamic
delete firewall policy interface
show firewall policy

add firewall policy list

Syntax

```
ADD FIREwall POLIcy=policy-name LISt=list-name
   FIle=filename TYpe={IP|ADDRESS}
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *list-name* is a character string 1 to 15 characters long. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9), and the characters ~ ' ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = " | \ [ ] ; : ? / , < >.
- *filename* is the name of a file on the router.

Description

This command adds a list of either IP addresses and networks or Ethernet MAC addresses to the specified policy. These lists are used in policy rules.

The POLICY parameter specifies the policy to which the list is added. The specified policy must already exist.

The LIST parameter specifies a name for the list. The name is used in other commands to refer to the list.

The TYPE parameter specifies the type of information in the file. If IP is specified, the file contains IP host and network address information. If ADDRESS is specified, the file contains Ethernet MAC addresses.

The FILE parameter specifies the name of a file on the router's file subsystem containing the list. The filename must have a TXT extension and be a text file. See "List Files" on page 41-9 for a detailed description of the format of list files.

Examples

To add a list of IP addresses named "firstfloor" from the file LISTIP.TXT to the firewall policy named "zone1", use the command:

```
add fire poli=zone1 lis=firstfloor ty=ip fi=listip.txt
```

Related Commands

create firewall policy
delete firewall policy list
show firewall policy

add firewall policy nat

Syntax

```
ADD FIREwall POLIcy=policy-name NAT={ENHanced|STAndard}
   INTerface=interface [IP=ipadd] GBLINterface=interface
   [GBLIP=ipadd[-ipadd]]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *ipadd* is an IP address in dotted decimal notation.
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the interface may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).

Description

This command adds a NAT translation to the specified policy. If an interface or global interface is specified, then that interface must have already been added to the security policy.

The POLICY parameter specifies the policy where the NAT translation is to be added. The specified policy must already exist.

The NAT parameter specifies the type of NAT translation to perform. If STANDARD is specified and the IP parameter is specified, there is a one-to-one translation between a private IP address and the specified global IP address. If the IP parameter is not specified, then global IP addresses are used dynamically from the supplied pool of addresses as required. The pool of IP addresses consists of one or more global IP addresses. When a pool of global addresses is specified and all sessions are complete for a particular global IP mapping, then that global IP address is returned to the pool for reuse. If ENHANCED is specified, Enhanced NAT (ENAT) is used and both the private IP address and protocol dependent port numbers are translated. The benefit of ENAT is that only a single global Internet address is required to map an entire private network.

The INTERFACE parameter specifies the private interface from which all received traffic is translated before being passed to the public interface specified by the GBLINTERFACE parameter. Both interfaces must already be defined and belong to the same policy. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

A dynamic interface template cannot be added to a global interface in a NAT definition because a dynamic interface is never directly assigned an IP address. A global interface must have a global address, which must be a real globally unique Internet address.

The IP parameter specifies the private IP address used when a single public IP address is mapped to a single private IP address, and is only valid when NAT is set to STANDARD. This parameter is not valid if a range is specified for the GBLIP parameter.

The GBLINTERFACE parameter specifies the public interface from which all received traffic is translated before being passed to the private interface specified by the INTERFACE parameter. Both interfaces must already be defined and belong to the same policy.

The GBLIP parameter specifies a single global IP address or a range of global IP addresses to be used by the NAT translation. If NAT is set to STANDARD and a pool of global IP addresses is required, then a range must be specified. If NAT is set to ENHANCED, then generally only a single global IP address is required. However, there are situations where it is necessary to allow sessions to be initiated from a public interface to private hosts via more than one public IP address. An example would be WWW traffic for two public IP addresses that must be passed through to two private hosts. In this case, a range of global IP addresses is required; however, only the first address of the range would be used as a source address for packets in outgoing sessions.

If the GBLIP parameter is not specified, the IP address of the global interface is used as the global IP internet address. This is useful in configurations where the public interface does not have a static IP address, for example, a dial-up user who is dynamically allocated an IP address by the ISP.

If NAT is set to STANDARD and a pool of global IP addresses is required, then a range must be specified.

If NAT is set to ENHANCED, then generally only a single global IP address is required. However, there are situations where it is necessary to allow sessions to be initiated from a public interface to private hosts via more than one public IP address. An example of this situation would be WWW traffic for two public IP addresses that must be passed through to two private hosts. In this case, a range of global IP addresses is required.

Examples

To add an enhanced NAT mapping to the firewall policy named "zone1", use the command:

```
add fire poli=zone1 nat=enh int=eth0 gblin=ppp0
   gblip=202.36.163.2
```

Related Commands

create firewall policy
create firewall policy dynamic
delete firewall policy nat
show firewall policy

add firewall policy proxy

Syntax

```
ADD FIREwall POLIcy=policy-name PROXy={HTTP|SMTP}
   INTerface=interface GBLINterface=interface
   DIrection={IN|OUT|BOTH} [IP=ipadd] [DAYs=day-list]
   [AFTer=hh:mm] [BEFore=hh:mm]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).
- *ipadd* is an IP address in dotted decimal notation.
- *day-list* is one or more of the keywords: mon, tue, wed, thu, fri, sat, sun, weekday, weekend, or all, which are separated by commas.
- *hh:mm* is a time in hours and minutes.

Description

This command adds a specific application proxy to the security policy. If application proxies are present in a firewall policy then it is not necessary to add a rule to allow traffic into the public interface. The presence of a proxy with DIRECTION=IN or BOTH is equivalent to an "allow" rule for that type of traffic. It is possible to specifically deny access to the proxy by adding a deny rule to the public interface. It is also possible to bypass the proxy by adding an appropriate allow rule. For example, by adding an allow rule to the private interface HTTP traffic from certain IP addresses on the private network could be allowed to bypass an HTTP proxy.

Take care when adding "allow" rules that bypass firewall proxies so that only the desired traffic is not processed by the proxy.

The POLICY parameter allows a specific security policy to be selected.

The PROXY parameter specifies the application proxy that is added to the security policy. Available application proxies are described in Table 41-4 on page 41-41.

Table 41-4: Application Proxies

Proxy	Functions
HTTP	Filtering of requested URLs.
	Blocking/filtering of cookies.
SMTP	Provides filtering of spam email from known spam sources.
	Blocking of third party relay attacks.
	Blocking of email smurf amp attacks.

The INTERFACE parameter specifies the private interface from which all received traffic is translated before being passed to the public interface specified by the GBLINTERFACE parameter. Both interfaces must already be defined and belong to the same policy. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The GBLINTERFACE parameter defines the public interface from which all public traffic is received before being passed to the private interface specified by the INTERFACE parameter. The interfaces must be defined before issuing this command and both belong to the same security policy.

The DIRECTION parameter sets the direction that the proxy works. A direction of IN means that the proxy allows the session to be initiated from the public Internet. If the direction is set to IN, the IP parameter must be set. A DIRECTION of OUT means that the proxy allows sessions to be initiated from the private Intranet. A DIRECTION of BOTH allows sessions to be initiated from either the private Intranet or public Internet. The default is OUT.

The IP parameter defines the destination private host for a proxy where DIRECTION is set to IN or BOTH. Traffic arriving at the public interface to be passed through the proxy has the IP address of the public interface as its destination. This parameter defines the private IP with which the proxy establishes a connection.

The BEFORE and AFTER parameters specify the time period when the proxy is active.

The DAYS parameter specifies in a comma-separated list the days when the proxy is active. This allows rules to be active on certain days of the week. The value WEEKDAY covers Monday to Friday. The value WEEKEND covers Saturday and Sunday.

Examples To add an SMTP proxy to the firewall policy called *zone1* that allows the SMTP sessions to be initiated from the private or public side of the firewall between the interfaces eth0vlan1 (private) and ppp0 (public) where the IP of the SMTP server on the private intranet is 192.168.1.10, use the command:

```
add fire poli=zone1 prox=smtp int=eth0vlan1 glbin=ppp0
   ip=192.168.1.10 di=both
```

Related Commands **delete firewall policy proxy**
disable firewall policy smtprelay

add firewall policy rule

Syntax

```
ADD FIREwall POLIcy=policy-name RUle=rule-id
   ACtion={ALLOw|DENY|NAT|NONat} INTerface=interface
   PROTocol={protocol|ALL|EGP|GRE|ICmp|OSPF|SA|TCP|UDP}
   [AFTer=hh:mm] [BEFore=hh:mm] [DAYs={MON|TUE|WED|THU|
   FRI|SAT|SUN|WEEKDAY|WEEKEND}[,...]]
   [ENCapsulation={NONE|IPSec}] [GBLIP=ipadd]
   [GBLPort={ALL|port[-port]|service-name}]
   [GBLRemoteip=ipadd[-ipadd]] [IP=ipadd[-ipadd]]
   [LISt={list-name|RADius}] [NATType={DOuble|ENHanced|
   REVerse|STAndard}] [NATMask=ipadd] [POrt={ALL|
   port[-port]|service-name}] [REMoteip=ipadd[-ipadd]]
   [SOurceport={ALL|port[-port]}] [TTL=hh:mm]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *rule-id* is a number from 1 to 299.
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).
- *protocol* is an Internet IP protocol number.
- *hh:mm* is a time in hours and minutes.
- *ipadd* is an IP addresses in dotted decimal notation.
- *port* is an Internet service port number or name.
- *list-name* is a character string 1 to 15 characters long. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9), and the characters ~ ' ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = " | \ [ ] ; : ? / , < >.
- *service-name* is a pre-defined name for an IP service (Table 41-5 on page 41-46).

Description

This command adds a rule defining the access allowed between private and public interfaces of the specified policy. By default all access from public interfaces (outside the firewall) is denied and all access from private interfaces (inside the firewall) is allowed. To refine the security policy additional rules can be added to allow or deny access based on IP addresses, port numbers, day of the week, or time of day. Each rule for a specific interface in a policy is processed in order, starting with the lowest numbered rule and proceeding to the highest numbered rule, or until a match is found.

When a firewall policy rule is modified, new firewall sessions apply the modified rule. Firewall sessions that existed before the modification continue according to the unmodified rule until:

- they timeout due to a lack of traffic
- the firewall session is deleted using the **delete firewall session** command
- the firewall is reset using the **disable firewall** then **enable firewall** commands.

In addition to rules based on IP address, port, protocol, date and time, the processing of ICMP packets, IP packets with options set and ping packets can be enabled or disabled on a per-policy basis using the **enable firewall policy** command on page 41-67 and the **disable firewall policy** command on page 41-62.

The POLICY parameter specifies the policy to which the rule is to be added. The specified policy must already exist.

The RULE parameter specifies both an identifier for the rule and the position of the rule in the list of rules for this policy. Rules are processed in order, from the lowest to the highest numbered rule. The identifier refers to this rule in other commands.

The ACTION parameter specifies what the firewall should do with traffic that matches the selectors defined for this rule. If ALLOW is specified, the traffic is permitted to pass through the firewall. If DENY is specified, the traffic is prevented from passing through the firewall. If NONAT is specified, traffic that matches the rule does not undergo NAT translation should a NAT relationship exist for the interfaces involved. If NAT is specified, the NATTYPE parameter should be used to specify whether the NAT rule performs DOUBLE, ENHANCED, REVERSE, or STANDARD NAT translation. The values NONAT and NAT implicitly allow traffic through the firewall.

A rule specified with **action=nat** takes precedence over NAT relationships specified by the **add firewall policy nat** command. A rule specified with **action=nat** implicitly allows traffic that matches the rule. Take care when defining the rule so that only the desired traffic is permitted through the firewall.

The INTERFACE parameter specifies the interface to which the rule is applied. The interface must already exist and belong to the policy. Valid interfaces are dynamic ones (see "Dynamic Interfaces" on page 41-16) as well as the following:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

To see a list of all interfaces currently available, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The PROTOCOL parameter specifies the IP protocol number or the name of a predefined protocol type to match (Table 41-5 on page 41-46). If TCP or UDP is specified, then the PORT parameter must also be specified.

The AFTER and BEFORE parameters specify the time period when the rule is active. Time periods begin and end at midnight, and should be entered in 24-hour format. The AFTER and BEFORE parameters can be used together, but the time specified for AFTER must be earlier than the time specified for BEFORE. If AFTER is specified without BEFORE, the rule is active from AFTER until midnight. If BEFORE is specified without AFTER, the rule is active from midnight until BEFORE.

The DAYS parameter specifies the days as a comma-separated list on which the rule is to apply. This allows rules to be active on certain days of the week. The value WEEKDAY covers Monday to Friday. The value WEEKEND covers Saturday and Sunday.

When set to IPSEC, the ENCAPSULATION parameter specifies that traffic can match the rule only when it has been decapsulated from an IPsec tunnel. This is useful for selecting traffic arriving from an IPsec tunnel that uses a dynamically assigned IP source address. If NONE is specified, encapsulation does not take place.

The GBLIP parameter specifies a single IP address that is matched to the destination address of packets received on a public interface. The GBLIP parameter also specifies the global IP address to be used as the public IP address for private side devices if NAT is active on the interface, or if the value specified for the ACTION parameter is NAT.

The GBLPORT parameter specifies the TCP port number, service name (e.g. FTP, HTTP), or range of port numbers that apply to the rule if NAT has been configured on the interface by using the **add firewall policy nat** command on page 41-39. The GBLPORT parameter cannot be used when ACTION=NAT is specified. In this situation, the PORT parameter should be used instead.

The application of the GBLREMOTEIP parameter changes depending on the type of interface it is applied to. If the INTERFACE parameter specifies a public interface and the NATTYPE is REVERSE or DOUBLE, the GBLREMOTEIP parameter specifies a single IP address that is matched to the source IP address of packets received on that interface. If the INTERFACE parameter specifies a public interface, and the NATTYPE parameter is ENHANCED, then the GBLREMOTEIP parameter specifies a single IP address or a range of IP addresses that is matched to the source IP address of packets received on that interface. If the INTERFACE parameter specifies a private interface, the GBLREMOTEIP parameter is substituted as the destination address for packets received on the interface. This parameter should only be specified when the ACTION parameter is NAT and the NATTYPE is ENHANCED, REVERSE or DOUBLE.

The IP parameter specifies a single IP address or a range of IP addresses that match the source address of packets received on a private interface. The IP parameter also specifies the IP address to be used as the private IP address for private side devices if NAT is active on the interface, or if the value specified for the ACTION parameter is NAT.

The LIST parameter specifies a list of addresses to be checked for a match against the source or destination address of the new flow. The value may be the name of a predefined list of IP or MAC addresses, or the keyword RADIUS. If RADIUS is specified and a RADIUS server has been defined, a RADIUS lookup is performed to check the source or destination address of the new flow. Up to four lists can be add to a rule by repeated invocations of this command.

The NATTYPE parameter may only be used when the value specified by the ACTION parameter is NAT. It specifies whether the NAT rule performs DOUBLE, ENHANCED, REVERSE or STANDARD NAT. DOUBLE NAT translates both the public and private side source and destination addresses. ENHANCED NAT defined for a private interface translates the private side source address (specified using the IP parameter) and protocol dependent ports to a single source address (specified by the GBLIP parameter), suitable for the public side of the firewall. ENHANCED NAT defined for a public interface translates the public side source address (specified using the GBLREMOTEIP parameter) and protocol dependent ports to a single source address (specified by the REMOTEIP parameter), suitable for the private side of the firewall. REVERSE NAT translates the addresses of public side devices (specified using the GBLREMOTEIP parameter), to addresses suitable for the private side of the firewall (specified using the REMOTEIP parameter), so translates source address for inbound traffic and destination address for outbound traffic. STANDARD NAT translates the addresses of private side devices (specified using the IP parameter) to addresses suitable for the public side of the firewall (specified by the GBLIP parameter), so translates source address for outbound traffic and destination address for inbound traffic.

The NATMASK parameter specifies an IP address mask that translates IP addresses from one subnet to another. The MASK parameter must be specified when the rule action is NAT and the NATTYPE is DOUBLE, REVERSE, or STANDARD. The NATMASK parameter can be used when translating entire subnets from one address to another. If DOUBLE NAT is specified, the NATMASK is applied to the IP, GBLIP, REMOTEIP and GBLREMOTEIP parameters. If REVERSE NAT is specified, the NATMASK is applied to both the REMOTEIP and GBLREMOTEIP parameters. If STANDARD NAT is specified, the NATMASK is applied to both the IP and GBLIP parameters. The IP, GBLIP, REMOTEIP and GBLREMOTEIP parameters must specify a single IP address if the NATMASK parameter is used.

The PORT parameter specifies a port number, a range of port numbers, or a pre-defined service name (Table 41-5 on page 41-46) to match. If ALL is specified, the rule matches any port number. If dynamic NAT is active on the interface it is possible to re-map a global port number to a different internal port number. For rules applied to a private interface, PORT is the destination port on the public network. For rules applied to a public interface, PORT is either the destination port on the private network or, in the case of NAT being applied, the destination port on the private network where the traffic is mapped. When ACTION=NAT is specified, the PORT parameter specifies the port number or range of port numbers to match.

Table 41-5: Pre-defined IP protocol service names .

Service Name	Port Number	Standard Protocol
ECHO	7	TCP or UDP
DISCARD	9	TCP or UDP
FTP	21	TCP
TELNET	23	TCP
SMTP	25	TCP
TIME	37	TCP or UDP
DNS	53	UDP
BOOTPS	67	UDP
BOOTPC	68	UDP

Table 41-5: Pre-defined IP protocol service names (continued).

Service Name	Port Number	Standard Protocol
TFTP	69	UDP
GOPHER	70	TCP
FINGER	79	TCP
WWW	80	TCP
HTTP	80	TCP
KERBEROS	88	TCP
RTELNET	107	TCP
POP2	109	TCP
POP3	110	TCP
SNMPTRAP	162	UDP
SNMP	161	UDP
BGP	179	TCP
RIP	520	TCP
L2TP	1701	UDP
PPTP	1723	TCP
VDOLIVE	7000	TCP
REALAUDIO	7070	TCP
REALVIDEO	7070	TCP

The REMOTEIP parameter specifies a single IP address or a range of IP addresses that match the destination address of packets received on a private interface. If the value specified for the ACTION parameter is not NAT, the REMOTEIP parameter also specifies a single IP address or range of IP addresses that match the source address of packets received on a public interface. If the value specified for the ACTION parameter is NAT, the REMOTEIP parameter also specifies the IP address to be used as the private IP address for public side devices.

Table 41-6 on page 41-47 summarises the required parameters for the firewall NAT Rules which were explained in the IP, REMOTEIP, GBLIP, GBLREMOTEIP and NATMASK paragraphs above.

Table 41-6: Required parameters for firewall NAT rules.

		Parameters				
NAT Rule Type	Direction	IP	REMOTEIP	GBLIP	GBLREMOTEIP	NATMASK
Standard	I	T		S	X	X
	O	S		T	X	X
Standard subnet	I	T		S	X	T
	O	S		T	X	T
Enhanced	I		T	X		X
	O			T	X	X
Reverse	I		T	X	S	X
	O		S	X	T	X

		Parameters				
Reverse subnet	I		T*	X	S	T*
	O		S	X	T	T*
Double	I	T	T*	S	S	X
	O	S*	S	T	T	X
Double subnet	I	T	T*	S	S	T*
	O	S*	S*	T	T	T*

Key to table

- Direction
  I = in. The rule is applied to a public interface.
  O = out. The rule is applied to a private interface.
- S = Selector. The value supplied for this parameter is compared to the corresponding field in a packet.
- T = Translator. The value supplied for this parameter is substituted into the packet to bring about the address translation.
- * The parameter is required for the rule to function correctly, but can be put into a **set firewall policy rule** command if the **add** command line has become too long.
- X = Not permitted. This parameter is not permitted in this type of NAT rule.
- Empty table entry = an optional selector.

The SOURCEPORT parameter specifies a source port to match for a TCP or UDP flow. This allows rules to be made based on the source port of the IP flow.

The TTL (time to live) parameter specifies the time duration in hours and minutes that the rule exists. The rule is active from the creation of the rule and is deleted when the specified time expires. All entries created from this rule are destroyed when the rule expires. Rules defined with a TTL value do not appear in router-generated configuration scripts because they are dynamic.

Examples To allow WWW access to an internal server at IP address 202.36.163.12 that is attached to a private interface defined in the policy named "zone1" via the public interface PPP0, use the command:

```
add fire poli=zone1 ru=1 ac=allow int=ppp0 ip=202.36.163.12
   prot=tcp pr=www
```

If the company's business hours are from 8 a.m. to 5 p.m. and no external access is permitted outside these hours, use the command:

```
add fire poli=zone1 ru=2 ac=allo int=ppp0 ip=202.36.163.12
   prot=tcp pr=www aft=08:00 bef=17:00
```

To deny staff WWW access during the company's business hours from 8 a.m. to 5 p.m., use the command:

```
add fire pol=zone1 ru=3 ac=deny int=eth0vlan1 prot=tcp pr=WWW
   aft=08:00 bef=17:00
```

To allow DNS information from a server at 192.168.12.2 to a private DNS server at IP address 192.168.34.1, which uses UDP originating on port 53, use the command:

```
add fire poli=zone1 ru=5 ac=allo int=ppp0 prot=udp
   ip=192.168.34.1 rem=192.168.12.2 so=53
```

To allow Telnet access to a UNIX server on a private network with NAT configured to use the public interface PPP0 with the global IP address 202.49.72.1, use the command:

```
add fire poli=zone1 ru=6 ac=allo int=ppp0 ip=192.168.1.1
   prot=tcp pr=tel gblip=202.49.72.1 gblp=tel
```

To add a list to limit the destinations that users of the private network can access based on the list file LISTIP.TXT and also a RADIUS lookup, use the commands:

```
add fire poli=zone1 lis=listallow ty=IP fi=listip.txt

add fire poli=zone1 ru=7 ac=allo int=eth0vlan1 lis=listallow
   prot=all

add fire poli=zone1 ru=7 lis=rad
```

To translate the source IP address of traffic with a source address of 192.168.2.100 and destination IP addresses from 192.168.1.1 to 192.168.1.100 that are received on the private interface to the global IP address 192.168.1.53 using Enhanced NAT, use the command:

```
add fire poli=zone1 ru=7 ac=nat natt=enh int=eth0 prot=all
   IP=192.168.2.100 rem=192.168.1.1-192.168.1.100
   gblip=192.168.1.53
```

To translate the source address of traffic received on the private interface and destined for addresses from 204.22.3.1 to 204.22.3.99 to the global subnet 210.25.4.0, use the command:

```
add fire poli=zone1 ru=10 ac=nat natt=sta int=eth0 prot=all
   gblip=210.25.4.0 natm=255.255.255.0 rem=204.22.3.1-
   204.22.3.99
```

To provide a corresponding rule on the public interface to translate to the private subnet 10.1.2.0, use the command:

```
add firewall policy=zone1 ru=11 ac=nat natt=sta int=eth1
   prot=all gblip=210.25.4.0 IP=10.1.2.0 natm=255.255.255.0
   rem=204.22.3.1-210.22.3.99
```

To translate both the source and destination addresses of traffic received on the private interface with a source address of 192.168.0.74 to a destination address of 210.25.7.1 and new source address of 210.25.4.1, use the command:

```
add fire poli=zone1 ru=50 ac=nat natt=do int=eth1 prot=all
   ip=192.168.0.74 GBLIP=210.25.4.1 gblr=210.25.7.1
```

To redirect all traffic received on a private interface to a destination of 210.25.7.1, without changing the source address, use the command:

```
add fire poli=zone1 ru=51 ac=nat natt=rev int=eth1 prot=all
   gblr=210.25.7.1
```

Related Commands

create firewall policy
create firewall policy dynamic
delete firewall policy rule
set firewall policy rule
show firewall policy

add firewall policy spamsources

Syntax `ADD FIREwall POLIcy=`*`policy-name`* `SPAMsources=`*`filename`*

where:

- *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *filename* is the name of a file on the router.

Description This command adds a file containing a list of email addresses and domain names identified as sources of spam email to the specified policy.

The POLICY parameter specifies the name of the firewall policy where the SMTP configuration is to be added. The specified policy must already exist. There is no default.

The SPAMSOURCES parameter specifies the name of a file that contains a list of identified spam sources that are to be blocked by the SMTP proxy. The specified file must have a ".spa" suffix. The file is a text file and contains one or more single line entries each containing an email address or domain name in the usual format. Lines may be "commented out" by placing a "#" at the start of the line. Figure 41-3 on page 41-50 shows an example of an SMTP spam source file.

Figure 41-3: Example of an SMTP proxy spam sources file.

```
# SMTP Proxy spam sources file spam.spa
spambandit@hotmail.com
spammerzone.com.au
wesayspam@spamcentral.com
buymystuff@rubbish.com
```

In order to edit the contents of the SMTP spam sources file held in the firewall policy it must be deleted from the firewall policy (using the **delete firewall policy spamsources** command on page 41-58), edited and then added to the firewall policy again. Alternatively the file may be edited, then deleted from the policy and then added to the policy again. Editing alone does not alter the configuration held in the policy. No more than five spam-source files may be attached to a policy at one time.

Examples To add an SMTP proxy configuration file named *spamfile.spa* to the firewall policy name *zone1*, use the command:

```
add fire poli=zone1 spam=spamfile.spa
```

Related Commands

add firewall policy rule
delete firewall policy proxy
delete firewall policy spamsources
disable firewall
enable firewall
show firewall

create firewall policy

Syntax `CREate FIREwall POLIcy=`*`policy-name`*

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen (-) and the underscore character ("_").

Description This command creates a new firewall policy. The POLICY parameter specifies the name of the policy to be created, and is used in other commands to refer to the policy. The specified policy must not already exist.

A new policy does not become active until at least one private and one public interface have been added. The policy can be customised to handle specific traffic by adding interfaces, address lists, NAT translations and/or rules, using the commands:

```
add fire poli dynamic
add fire poli interface
add fire poli list
add fire poli nat
add fire poli rule
```

Examples To create a firewall policy named *area1*, use the command:

```
cre fire poli=area1
```

Related Commands
add firewall policy interface
add firewall policy list
add firewall policy nat
add firewall policy rule
create firewall policy dynamic
destroy firewall policy
disable firewall policy
enable firewall policy
show firewall policy

create firewall policy dynamic

Syntax `CREate FIREwall POLIcy=`*`policy-name`* `DYnamic=`*`template`*

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen (-) and the underscore character ("_").
- *template* is a character string 1 to 15 characters long. Valid characters are any printable character. If *template* includes spaces, it must be in double quotes.

Description This command creates a dynamic interface template and adds it to the specified policy. The dynamic interface template name is used as a placeholder for adding dynamic interfaces to policies, NAT entries and rules.

Example To create the dynamic interface template *remote* and add it to the *management* policy, use the command:

```
cre fire poli=management dy=remote
```

Related Commands
add firewall policy dynamic
delete firewall policy dynamic
destroy firewall policy dynamic
show firewall policy

delete firewall policy apprule

Syntax `DELete FIREwall POLIcy=policy-name APPRule=app-rule-id`

where:

- *app-rule-id* is a number from 1 to 299.
- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command deletes defined rules for managing application traffic between interfaces covered by the firewall policy, and is modelled on the **delete firewall policy rule** command.

Example To delete apprule number 1 from the policy named "zone1", use the command:

```
del fire poli appr=zone1 ru=1
```

Related Commands
add firewall policy apprule
show firewall policy

delete firewall policy dynamic

Syntax
```
DELete FIREwall POLIcy=policy-name DYnamic=template
    {FIle=filename.txt|USer={username|ANY|NONE}}
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *template* is a character string 1 to 15 characters long. Valid characters are any printable character. If *template* includes spaces, it must be in double quotes.
- *username* is a character string 1 to 63 characters long. Valid characters are any printable character. If *username* includes spaces, it must be in double quotes.
- *filename.txt* is the name of a file on the router.

Description This command deletes one user or a list of users from a file from the specified dynamic interface template for a policy.

The FILE parameter specifies the file containing a list of users to be deleted. The text must have one username per line.

The USER parameter specifies the user to be deleted. Two special usernames are reserved, NONE and ANY. The username NONE is used to specify dynamic interfaces that do not require authentication. The ANY username is used to match all authentication usernames. This allows the one catch-all for all authenticated usernames.

Example To delete user *anna* from the dynamic interface template *remote* for the *management* policy, use the command:

```
del fore poli=management dy=remote us=anna
```

Related Commands **add firewall policy dynamic**
show firewall policy

delete firewall policy httpfilter

Syntax

```
DELete FIREwall POLIcy=policy-name HTTPFilter=filename
   [DIrection={IN|OUT}]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *filename* is the name of a file on the router.

Description This command deletes all entries originally contained in the specified HTTP filter file from the policy's HTTP filter. The HTTP filter file contains a list of URLs and cookie sites that are not permitted through the HTTP proxies configured under the firewall policy.

The POLICY parameter specifies the policy from which the URL filter file is to be deleted. It must already exist.

The HTTPFILTER parameter specifies the name of the URL filter file that originally contained the filter entries that are to be deleted. The entries are identified within the policy by the name of the file they originally came from. It is not necessary for the file to currently exist on the device.

The DIRECTION parameter specifies the direction of HTTP sessions to which the filter applies. The default is OUT.

Examples To delete the entries associated with the file banned.url from the URL filter of firewall policy zone1, use the command:

```
del fire poli=zone1 httpf=banned.url
```

Related Commands **add firewall policy proxy**
create firewall policy

delete firewall policy proxy
disable firewall policy httpcookies
enable firewall policy httpcookies
set firewall policy

delete firewall policy interface

Syntax

```
DELete FIREwall POLIcy=policy-name INTerface=interface
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).

Description

This command deletes an interface from the specified policy. The resulting policy must contain at least one private interface and at least one public interface to remain operational.

The POLICY parameter specifies the policy from which the interface is to be deleted. The specified policy must already exist.

The INTERFACE parameter specifies an assigned and configured interface that is to be deleted from the policy. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). To see a list of current valid interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

Examples

To delete the Ethernet interface from a policy named "zone1", use the command:

```
del fire poli=zone1 int=eth0
```

Related Commands

add firewall policy interface
create firewall policy dynamic
show firewall policy

delete firewall policy list

Syntax

```
DELete FIREwall POLIcy=policy-name LISt=list-name
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *list-name* is a character string 1 to 15 characters long. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9), and the characters ~ ' ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = " | \ [ ] ; : ? / , < >.

Description

This command deletes a predefined list of IP addresses, networks or Ethernet MAC addresses from the specified policy.

The POLICY parameter specifies the policy from which the list is to be deleted. The specified policy must already exist.

The LIST parameter specifies the name of the list to be deleted. The specified list must already exist and be assigned to the policy.

Examples

To delete the list named "firstfloor" from the policy named "zone1", use the command:

```
del fire poli=zone1 lis=firstfloor
```

Related Commands

add firewall policy list
show firewall policy

delete firewall policy nat

Syntax

```
DELete FIREwall POLIcy=policy-name NAT={ENHanced|STAndard}
   INTerface=interface GBLINterface=interface [IP=ipadd]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).
- *ipadd* is an IP address in dotted decimal notation.

Description

This command deletes a NAT translation from an interface, or IP address associated with an interface.

The POLICY parameter specifies the policy from which the NAT translation or IP address is to be deleted. The specified policy must already exist.

The NAT parameter specifies the type of NAT translation to be deleted. If STANDARD is specified, an IP address is not specified with the IP parameter, and a pool of global IP addresses exists, then the global IP address pool and the associated NAT translation are deleted. If STANDARD is specified and an IP address is specified with the IP parameter, the NAT translation for the specified private IP address is deleted. If ENHANCED is specified, the IP parameter may not be specified.

The INTERFACE parameter specifies the private interface associated with the NAT translation that is to be deleted. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). To see a list of current valid interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The GBLINTERFACE parameter specifies the public interface associated with the NAT translation that is to be deleted. Valid interfaces are listed in the INTERFACE parameter description.

The IP parameter specifies a previously defined private IP address used when a single public IP address is mapped to a single private IP address associated with the NAT translation that is to be deleted. The IP parameter is valid when NAT is set to STANDARD.

Examples

To delete a NAT mapping defined in the policy named "zone1", use the command:

```
del fire poli=zone1 nat=enh int=eth0 gbli=ppp0
```

Related Commands

add firewall policy nat
create firewall policy dynamic
show firewall policy

delete firewall policy proxy

Syntax

```
DELete FIREwall POLIcy=policy-name PROXy={HTTP|SMTP}
   INTerface=interface GBLINterface=interface
   DIrection={IN|OUT|BOTH} [IP=ipadd]
```

where:

- *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *interface* is a valid interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

  Alternatively, the *interface* may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote).
- *ipadd* is an IP addresses in dotted decimal notation.

Description

This command deletes a specific application proxy from the security policy.

The POLICY parameter allows a specific security policy to be selected.

The PROXY parameter specifies the application proxy that is to be deleted from the security policy.

The INTERFACE parameter specifies the private interface from which all received traffic is translated before being passed to the public interface specified by the GBLINTERFACE parameter. Both interfaces must already be defined and belong to the same policy. Valid interfaces are:

- eth (e.g. eth0, eth0-1)
- PPP (e.g. ppp0, ppp1-1)
- VLAN (e.g. vlan1, vlan1-1)
- FR (e.g. fr0, fr0-1)
- X.25T (e.g. x25t0, x25t0-1)

and dynamic interfaces (see "Dynamic Interfaces" on page 41-16). The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The GBLINTERFACE parameter defines the public interface from which all public traffic is received before being passed to private interface specified by the INTERFACE parameter. The interfaces must be defined before issuing this command and both must belong to the same security policy.

The DIRECTION parameter sets the direction that the proxy works. If the direction is set to IN, then the IP parameter must be set, and optionally the GBLIP may also be set. A direction of IN means that the proxy allows session to be initiated from the public Internet. A DIRECTION of OUT means that the proxy allows sessions to be initiated from the private Intranet. A DIRECTION of BOTH allows sessions to be initiated from either the private Intranet or public Internet.

The IP parameter defines the destination private host for a proxy with DIRECTION set to IN.

Examples To delete the SMTP access proxy defined in the firewall policy called *zone1*, use the command:

```
del fire poli=zone1 prox=smtp int=eth0vlan1 gblin=PPP0
```

Related Commands **add firewall policy proxy**

delete firewall policy rule

Syntax `DELete FIREwall POLIcy=`*`policy-name`* `RUle=`*`rule-id`*

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *rule-id* is a number from 1 to 299.

Description This command deletes a rule from the specified policy. The POLICY parameter specifies the policy from which the rule is to be deleted. The specified policy must already exist. The RULE parameter specifies the rule to be deleted from the policy.

Examples To delete rule number 1 from the policy named "zone1", use the command:

```
del fire poli=zone1 ru=1
```

Related Commands **add firewall policy rule**
set firewall policy rule
show firewall policy

delete firewall policy spamsources

Syntax `DELete FIREwall POLIcy=`*`policy-name`* `SPAMsources=`*`filename`*

where:

- *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *filename* is the name of a file on the router.

Description This command deletes a list of email addresses and domain names from the specified firewall policy. Once the file has been deleted, the addresses and domains listed in the file are no longer treated as spam sources and mail from them is allowed through the firewall (unless specified in another spam sources file used by the policy).

The POLICY parameter specifies the name of the firewall policy from which the file is to be deleted. There is no default.

The SPAMSOURCES parameter specifies the name of the file with the list of spam sources that is to be deleted.

This command does not delete the file from the router, but from the policy. In order to edit the contents of the spam sources file, it must be deleted from the firewall policy, edited and then added to the firewall policy using the **add firewall policy spamsources** command on page 41-50. Alternatively the file may be edited, then deleted from the policy, and added to the policy again. Editing alone does not alter the configuration held in the policy.

Examples To delete a spam sources file named *spam.spa* from the firewall policy name *zone1*, use the command:

```
del fire poli=zone1 spam=spam.spa
```

Related Commands

add firewall policy rule
add firewall policy spamsources
delete firewall policy proxy
disable firewall
enable firewall
show firewall

delete firewall session

Syntax `DELete FIREwall SEssion={`*`session-number`*`|ALL}`

where *session-number* is the identifier for a currently active session

Description This command terminates a specific active session or flow or all of them.

The SESSION parameter specifies the identifier of the active session or flow to be terminated. If ALL is specified, all active sessions and flows are terminated. The session identifier is read from the output of the **show firewall session** command on page 41-101.

Examples To delete session number 1B32, use the command:

```
del fire se=1B32
```

Related Commands

show firewall session

destroy firewall policy

Syntax `DESTroy FIREwall POLIcy=`*`policy-name`*

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen (-), and the underscore character ("_")

Description This command destroys the specified policy. The POLICY parameter specifies the policy to be destroyed. The specified policy must already exist.

Examples To destroy a policy named "area1", use the command:

```
dest fire poli=area1
```

Related Commands
create firewall policy
disable firewall policy
enable firewall policy
show firewall policy

destroy firewall policy dynamic

Syntax `DESTroy FIREwall POLIcy=`*`policy-name`* `DYnamic=`*`template-name`*

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), hyphen ("-"), and the underscore character ("_").
- *template-name* is a character string 1 to 15 characters long. Valid characters are any printable character. If *template-name* includes spaces, it must be in double quotes.

Description This command removes the dynamic interface template from the specified policy.

Example To remove dynamic interface template *acc1* from the *management* policy, use the command:

```
dest fire poli=management dy=acc1
```

Related Commands
add firewall policy dynamic
create firewall policy dynamic
delete firewall policy dynamic
show firewall policy

disable firewall

Syntax `DISable FIREwall`

Description This command disables the firewall and generates a warning message, notification message, and log message.

Examples To disable the firewall, use the command:

```
dis fire
```

Related Commands
disable firewall notify
disable firewall policy
enable firewall
enable firewall notify
enable firewall policy
show firewall

disable firewall notify

Syntax `DISable FIREwall NOTify={ALL|MAIL|MANager|PORT|SNMP}`

Description This command disables the sending of notification messages about firewall events to the specified destinations. The destinations are assumed to belong to the firewall manager. Notifications can be sent to one or more destinations.

The NOTIFY parameter specifies where the notifications are no longer to be sent, and accepts either a single value or a comma-separated list of values. If ALL is specified, notifications are no longer sent to any destinations. If MAIL is specified, notifications are no longer sent to an email address. If MANAGER is specified, notifications are no longer sent to all users currently logged in with Manager privilege. If PORT is specified, notifications are no longer sent to an asynchronous port. If SNMP is specified, notifications are no longer sent as SNMP traps to a pre-configured SNMP trap host. The default is MANAGER.

Examples To disable the sending of notifications via SNMP and email, use the command:

```
dis fire not=mail,snmp
```

Related Commands
disable firewall
disable firewall policy
enable firewall
enable firewall notify
enable firewall policy
show firewall

disable firewall policy

Syntax

```
DISable FIREwall POLIcy=name [ACCcouting] [DEBug={ALL|ARP|
   HTTP|PACKET|PKT|PROCESS|PROXY|SMTP|UPNP}]
   [FRAgment={ICMP|UDP|OTHER}[,...]]
   [ICMP_Forwarding={ALL|PARAMETER|PING|SOURCEQUENCH|
   TIMEEXCEEDED|TImestamp|UNREACHABLE}] [LOG={ALLOW|DENY|
   DENYDUMP|EVERYDENY|INAICMP|INALLOW|INAOTHER|INATCP|
   INAUDP|INDDICMP|INDDOTHER|INDDTCP|INDDUDP|INDDUMP|
   INDENY|INDICMP|INDOTHER|INDTCP|INDUDP|OUTAICMP|
   OUTALLOW|OUTAOTHER|OUTATCP|OUTAUDP|OUTDDICMP|
   OUTDDOTHER|OUTDDTCP|OUTDDUDP|OUTDDUMP|OUTDENY|OUTDICMP|
   OUTDOTHER|OUTDTCP|OUTDUDP}] [OPtions={ALL|RECORD_ROUTE|
   SECURITY|SOURCEROUTING|TImestamp}] [PING]
```

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description

This command disables the processing of specific types of IP packets by the specified policy, and/or disables accounting, logging or debugging for the policy.

The POLICY parameter specifies the policy for which packet processing attributes, accounting, logging, or debugging are to be disabled. The specified policy must already exist.

The ACCOUNTING parameter disables the recording of accounting information for flows and sessions handled by the policy. The currently stored accounting records can be displayed using the **show firewall accounting** command on page 41-81. Accounting records are also written to the Logging Facility. The log can be displayed by using the **show log** command on page 33-34 of Chapter 33, Logging Facility.

The DEBUG parameter specifies the types of debugging information to be disabled. This parameter is not retained over a reboot. If ALL is specified, all debugging information is disabled. If PACKET or PKT is specified, the display of the first 56 bytes of each IP packet received is disabled. If PROCESS is specified, the display of information about the processing of a particular IP packet is disabled. If SMTP is specified, the display of information about sent and received SMTP commands in the SMTP proxy is disabled. If HTTP is specified, the display of information about request and response messages passing through the HTTP proxy is disabled. If PROXY is specified, the display of general information about firewall proxies is disabled. If UPnP is specified, UPnP debugging for the firewall policy is disabled. If ARP is specified, the display of information about ARP requests processed by the firewall is disabled.

The FRAGMENT parameter specifies that this policy does not permit the forwarding of IP packets of the specified protocol type that have been fragmented into more than 8 fragments, or have a total payload of more than 1780 bytes of data. Disabling this feature restores the default policy functionality where fragmented packets are only permitted by the policy if there are no more than 8 fragments and the combined payload consists of, at most, 1780 bytes. If OTHER is specified, the command applies to protocols other than ICMP and UDP, but not TCP. There is no default.

The ICMP_FORWARDING parameter disables the forwarding of the specified ICMP messages through the router. The value may be a single option or a

comma-separated list of options. The default is not to forward any ICMP messages because ICMP packets can be used as a method for denial of service attacks.

The LOG parameter disables the logging of the specified firewall events to the router's Logging Facility. The value may be a single option or a comma-separated list of options. Table 41-2 on page 41-23 lists the options and their meanings. If EVERYDENY is disabled, only the first instance of a deny for a given source IP, destination IP, and protocol combination is logged in a two minute period if a matching deny LOG option is enabled. The default for EVERYDENY is disabled.

The OPTIONS parameter disables the forwarding of packets with the specified IP option or options to the next level of firewall checking. The value may be a single option or a comma-separated list of options. The default is not to forward packets with IP options.

The PING parameter disables the handling of ping packets destined for the router itself. The default is to accept such ping packets.

Examples

To disable the forwarding of all ICMP messages to the next level of firewall checking defined in the policy named *zone1*, use the command:

```
dis fire poli=zone1 icmp_f=all
```

To disable the logging of all allowed sessions started from the public Internet, in the policy named *zone1*, use the command:

```
dis fire poli=zone1 log=inallow
```

To disable the processing of fragmented IP packets consisting of more than 8 fragments and/or more than 1780 bytes of protocol data, through the policy named *zone1*, use the command:

```
dis fire poli=zone1 fra=udp
```

To disable UPnP debugging on the firewall policy called *upnp*, use the command:

```
dis fire poli=upnp deb=upnp
```

Related Commands

disable firewall
disable firewall notify
enable firewall
enable firewall notify
enable firewall policy
show firewall

disable firewall policy httpcookies

Syntax `DISable FIREwall POLIcy=`*`policy-name`* `HTTPCookies`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character ("_").

Description This command disables the passing of HTTP cookie requests through HTTP proxies configured under the firewall policy. All requests by remote servers to set HTTP cookies are blocked by the HTTP proxy.

The POLICY parameter specifies name of the firewall policy for which cookie requests are to be disabled. The policy must already exist.

An HTTP proxy with direction set to OUT or BOTH must be configured for the specified policy in order for cookies to be blocked.

Examples To disable the passing of HTTP cookies through HTTP proxies configured for the policy *zone1*, use the command:

```
dis fire poli=zone1 httpc
```

Related Commands
add firewall policy proxy
create firewall policy
delete firewall policy proxy
destroy firewall policy
disable firewall policy httpcookies
enable firewall policy httpcookies

disable firewall policy identproxy

Syntax `DISable FIREwall POLIcy=`*`policy-name`* `IDentproxy`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command disables the firewall's IDENT proxy.

Certain protocols such as FTP and SMTP query the identity of the source of a new session using the IDENT protocol RFC 1413. If a firewall blocks these requests, then most FTP and SMTP servers timeout the request, which takes about 30 seconds, then continue. However, some FTP and SMTP servers reject the session. If IDENTPROXY is enabled, then the firewall overcomes this problem by proxying IDENT queries when necessary. If this is deemed to be inappropriate for the environment in which the firewall is working, then this feature should be disabled. In this case, the firewall immediately resets the IDENT request with a TCP reset. This overcomes the timeout wait.

IDENTPROXY is enabled by default.

Example To disable the **firewall policy identproxy**, use the command:

```
dis fire poli id
```

Related Commands **enable firewall policy identproxy**

disable firewall policy smtprelay

Syntax `DISable FIREwall POLIcy=`*`policy-name`* `SMTPRelay`

where *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").

Description This command disables emails that intend to use the third party relaying mechanism for delivery from passing through the SMTP proxy.

The POLICY parameter specifies the name of the firewall policy that SMTP third party relay is to be disabled on. There is no default.

Third party relaying is disabled by default. It should only be enabled for short periods if required for debugging because it exposes email servers on the private side of the firewall to abuse.

Examples To disable third party relay email through the SMTP proxy of the firewall policy named *zone1*, use the command:

```
dis fire poli=zone1 smtpr
```

Related Commands
add firewall policy rule
add firewall policy spamsources
delete firewall policy proxy
delete firewall policy spamsources
enable firewall policy smtprelay

disable firewall policy tcpsetupproxy

Syntax `DISable FIREwall POLIcy=`*`policy-name`* `TCPsetupproxy`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command disables the firewall's setup proxy to TCP connections initiated from the public side of the firewall for the specified firewall policy. This allows a permitted firewall TCP session initiated from a public host to connect directly to hosts on the private network. The firewall setup proxy is enabled by default. When the TCP proxy is disabled, the load balancer cannot be used.

Care should be taken when using this command because it can reduce the security of the firewall and leave the private network vulnerable to attack such as an SYN flood attack.

Example To disable the router's setup proxy to the firewall policy named *area1*, use the command:

```
dis fire poli=area1 tcp
```

Related Commands
disable firewall policy
enable firewall policy
enable firewall policy tcpsetupproxy
show firewall policy

enable firewall

Syntax

```
ENAble FIREwall
```

Description This command enables the firewall. A log message is generated when this command is issued.

Examples To enable the firewall software, use the command:

```
ena fire
```

Related Commands
disable firewall
disable firewall notify
disable firewall policy
enable firewall notify
enable firewall policy
show firewall

enable firewall notify

Syntax

```
ENAble FIREwall NOTify={ALL|MAIL|MANager|PORT|SNMP}[,...]
   [POrt=port-number] [TO=address]
```

where:

- *port-number* is the number of an asynchronous port on the router. Ports are numbered sequentially starting from 0.
- *address* is a character string 1 to 131 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command enables notification messages about firewall events to be sent to one or more destinations. The destinations are assumed to belong to the firewall manager.

The NOTIFY parameter specifies where the notifications are to be sent, and accepts either a single value or a comma-separated list of values. If ALL is specified, notifications are sent to all destinations. If MAIL is specified, notifications are sent via email to the email address specified by the TO parameter. The MAIL subsystem must also be configured. See Chapter 1, Operation for more information about configuring the mail subsystem. If MANAGER is specified, notifications are sent to all users currently logged in with Manager privilege. If PORT is specified, notifications are sent to the asynchronous port specified by the PORT parameter. The port must be configured to the correct baud rate and flow control for the terminal. If SNMP is specified, notifications are sent as SNMP traps to the pre-configured SNMP trap host. See Chapter 38, Simple Network Management Protocol (SNMP) for more information about configuring an SNMP trap host. The default is MANAGER.

The PORT parameter specifies the asynchronous port where notifications are sent. This parameter is required and is valid when NOTIFY is set to PORT or to a list of destinations including PORT.

The TO parameter specifies the email address where notifications are sent. This parameter is required and is valid when NOTIFY is set to MAIL or to a list of destinations including MAIL.

Example To send notifications via email to fireman@mycorp.com, use the command:

```
ena fire not=mail to="fireman@mycorp.com"
```

Related Commands

disable firewall
disable firewall notify
disable firewall policy
enable firewall
enable firewall policy
show firewall

enable firewall policy

Syntax

```
ENAble FIREwall POLIcy=policy-name [ACCounting]
   [DEBug={ALL|ARP|HTTP|PACKET|PKT|PROCESS|PROXY|SMTP|
   RADius|TCP|UPNP}] [FRAgment={ICMP|UDP|OTHER}[,...]]
   [ICMP_Forwarding={ALL|PARAMETER|PING|SOURCEQUENCH|
   TIMEEXCEEDED|TImestamp|UNREACHABLE}] [LOG={ALLOw|DENY|
   DENYDUMP|EVERYDENY|INAICMP|INALLOW|INAOTHER|INATCP|
   INAUDP|INDDICMP|INDDOTHER|INDDTCP|INDDUDP|INDDUMP|
   INDENY|INDICMP|INDOTHER|INDTCP|INDUDP|OUTAICMP|
   OUTALLOW|OUTAOTHER|OUTATCP|OUTAUDP|OUTDDICMP|
   OUTDDOTHER|OUTDDTCP|OUTDDUDP|OUTDDUMP|OUTDENY|OUTDICMP|
   OUTDOTHER|OUTDTCP|OUTDUDP}] [OPtions={ALL|RECORD_ROUTE|
   SECURITY|SOURCEROUTE|TImestamp}] [PING]
```

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9) and the underscore character ("_").

Description This command enables the processing of specific types of IP packets by the specified policy, and/or enables accounting, logging or debugging for the policy.

The POLICY parameter specifies the policy for which packet processing attributes, accounting, logging or debugging are to be enabled. The specified policy must already exist.

The ACCOUNTING parameter enables the recording of accounting information for flows and sessions handled by the policy. The currently stored accounting records can be displayed using the **show firewall accounting** command on page 41-81. Accounting records are also written to the Logging Facility. The log can be displayed using the **show log** command on page 33-34 of Chapter 33, Logging Facility.

The DEBUG parameter specifies the types of debugging information to be enabled. This parameter is not retained over a reboot. If ALL is specified, all debugging information is enabled. If PACKET or PKT is specified, the display of the first 56 bytes of each IP packet received is enabled. If PROCESS is specified, the display of information about the processing of a particular IP packet is enabled. If SMTP is specified, the display of information about sent and received SMTP commands in the SMTP proxy is enabled. If HTTP is specified, the display of information about request and response messages passing through the HTTP proxy is enabled. If PROXY is specified, the display of general information about firewall proxies is enabled. If RADIUS is specified, information about the firewall's RADIUS queries is displayed. If TCP is specified, information about TCP traffic traversing the firewall is displayed. If UPNP is specified, debugging for UPnP on the firewall policy is enabled. If ARP is specified, the display of information about ARP requests processed by the firewall is enabled.

The FRAGMENT parameter specifies that this policy permits the forwarding of IP packets of the specified protocol type that have been fragmented into more than 8 fragments. Fragmented packets are still subject to the rules and other constraints configured in the firewall policies.

There is no limit on total data within this number of fragments, other than the MTU restrictions of the interfaces involved in forwarding the packets. If ICMP is specified, the policy permits ICMP ping (echo) requests and replies that have been fragmented into more than 8 fragments. If OTHER is specified, protocols other than ICMP and UDP, but excluding TCP, can be specified. The default policy behaviour is that fragmented packets are permitted by the policy when there are no more than 8 fragments and the combined protocol data consists of a maximum of 1780 bytes. The number of fragments that can be handled is configured by the **set firewall maxfragment** command on page 41-72.

The ICMP_FORWARDING parameter forwards one or more ICMP messages through the router. Use a comma-separated list for multiples. The default is not to forward ICMP messages because ICMP packets can be used as a method for denial of service attacks.

The LOG parameter logs one or more firewall events to the router's Logging Facility. Table 41-2 on page 41-23 lists the possible options and their meanings. Logging some firewall events requires additional configuration. Table 41-3 on page 41-25 lists these options and the additional configuration that is required.

If EVERYDENY is enabled, every instance of a deny that matches one of the deny LOG options that are enabled is logged. This may result in a large number of log entries. If EVERYDENY is disabled, only the first instance of a deny for a given source IP, destination IP, and protocol combination is logged in a two minute period if a matching deny LOG option is enabled. The EVERYDENY option by itself does not cause any logging to occur. The default for EVERYDENY is disabled.

The OPTIONS parameter enables the forwarding of packets with the specified IP option or options to the next level of firewall checking. The value may be a single option or a comma-separated list of options. The default is not to forward packets with IP options.

The PING parameter enables the handling of ping packets destined for the router itself. An exception is when both firewall NAT and ICMP forwarding are enabled. In this case, the PING parameter has no effect and ping packets destined for the router itself are passed to the next level of firewall checking. The default is to accept such ping packets.

Examples

To enable all ICMP messages to pass to the next level of firewall checking defined in the policy named *zone1*, use the command:

```
ena fire poli=zone1 icmp_f=all
```

To enable the logging of all allowed sessions started from the public Internet and all denied sessions in both directions, in the policy named *zone1*, use the command:

```
ena fire poli=zone1 log=INALLOW,DENY
```

For example, to create a log entry for every outgoing TCP packet that is denied, use the command:

```
ena fire poli=name log=EVERYDENY,OUTDTCP
```

To enable the processing of fragmented UDP packets consisting of more than 8 fragments and/or more than 1780 bytes of protocol data, through the policy named *zone1*, use the command:

```
ena fire poli=zone1 FRAGMENT=udp
```

To enable NAT traversal for UPnP on the firewall policy called "upnp", use the command:

```
ena fire poli=upnp upnp=NATTRAVERSAL
```

Related Commands

disable firewall
disable firewall notify
disable firewall policy
enable firewall
enable firewall notify
show firewall

enable firewall policy httpcookies

Syntax `ENAble FIREwall POLIcy=`*`policy-name`* `HTTPCookies`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character ("_").

Description This command enables HTTP cookie requests to be passed through HTTP proxies configured under the firewall policy. However, it is possible that a cookie request may be blocked by an entry in the policy's HTTP filter. By default, HTTP cookie requests are allowed to pass through the HTTP proxy configured under the firewall policy.

The POLICY parameter specifies name of the firewall policy for which cookie requests are to be enabled. The policy must already exist.

Examples To enable the passing of HTTP cookies through HTTP proxies configured for the policy zone1, use the command:

```
ena fire poli=zone1 httpc
```

Related Commands **add firewall policy proxy**
create firewall policy
delete firewall policy proxy
disable firewall policy httpcookies

enable firewall policy identproxy

Syntax `ENAble FIREwall POLIcy=`*`policy-name`* `IDentproxy`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command enables the firewall's IDENT proxy.

Certain protocols such as FTP and SMTP query the identity of the source of a new session using the IDENT protocol RFC 1413. If a firewall blocks such requests, then most FTP and SMTP servers timeout the request, which takes about 30 seconds, then continue. However, some FTP and SMTP servers reject the session. If IDENTPROXY is enabled, then the firewall overcomes this problem by immediately returning a proxy IDENT reply for the appropriate FTP or SMTP session. If this is inappropriate for the environment in which the firewall is working, then this feature should be disabled. In this case, the firewall immediately resets the IDENT request with a TCP reset. This overcomes the timeout wait.

IDENTPROXY is enabled by default.

Example To enable the **firewall policy identproxy**, use the command:

```
ena fire poli id
```

Related Commands **disable firewall policy identproxy**

enable firewall policy smtprelay

Syntax `ENAble FIREwall POLIcy=`*`policy-name`* `SMTPRelay`

where *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").

Description This command enables emails which intend to use the third party relaying mechanism to pass through the SMTP proxy.

The POLICY parameter specifies the name of the firewall policy that allows third party relay email through the SMTP proxy. There is no default.

Enabling third party relay email to pass through the firewall makes email servers on the private network susceptible to "Third Party Relay Attack", where an attacker uses private email servers to distribute large quantities of spam mail without permission and hide their own identity. Such relaying can often lead to "black-listing" relay servers, which may result in blocking email from legitimate users of email servers. Third party relaying is DISABLED by default and should only be enabled for short periods if required for diagnostic purposes.

Examples To enable third party relay email through the SMTP proxy of the firewall policy named *zone1*, use the command:

```
ena fire poli=zone1 smtpr
```

Related Commands
add firewall policy rule
add firewall policy spamsources
delete firewall policy proxy
delete firewall policy spamsources

enable firewall policy tcpsetupproxy

Syntax `ENAble FIREwall POLIcy=`*`policy-name`* `TCPsetupproxy`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command enables the firewall's setup proxy to TCP connections initiated from the public side of the firewall for the specified policy. When the firewall's setup proxy is enabled, any new connections are firstly established with the firewall before they are established with the private host. This is the default action of the firewall setup proxy. To disable the firewall setup proxy, use the **disable firewall policy tcpsetupproxy** command on page 41-65.

Example To enable the router's setup proxy to the firewall policy named *area1*, use the command:

```
ena fire poli=area1 tcp
```

Related Commands
disable firewall policy
disable firewall policy tcpsetupproxy
enable firewall policy
show firewall policy

set firewall maxfragment

Syntax

```
SET FIREwall MAXFragments=8..50
```

Description

This command sets the maximum number of fragments that a fragmented IP packet may consist of when enhanced fragment handling is enabled for a firewall policy.

The MAXFRAGMENTS parameter specifies the maximum number of fragments that an IP packet may consist of. The default is 20. The specified value applies to all firewall policies that have enhanced fragment handling enabled.

Enhanced fragment handling for the firewall is disabled by default. When disabled, fragmented IP packets can only be processed by the firewall when the packet consists of no more than 8 fragments and the total data contained in all the fragments is 1780 bytes or less. Enhanced fragment handling for a firewall policy is enabled with the **enable firewall policy** command on page 41-67.

Examples

To set the maximum number of fragments in a packet to be processed by any firewall policy with enhanced fragment handling enabled to 25, use the command:

```
set fire maxf=25
```

Related Commands
disable firewall policy
enable firewall policy
show firewall
show firewall policy

set firewall policy

Syntax

```
SET FIREwall POLIcy=policy-name [MAXUPNPPORTMAPS={0-1000}]
   [OTHERTimeout=minutes] [TCPTimeout=minutes]
   [UDPTimeout=minutes] [UPNP={ON|OFF|YES|NO|ENABLED|
   DISABLED}]
```

where:

- *minutes* is a time period from 0 to 43200 minutes.
- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description

This command sets timeouts for TCP, UDP, and other protocols. The firewall times out inactive sessions after the set period. This command also optionally enables the specified policy for UPnP. You can also use this command to set a limit to the number of port maps for UPnP.

The MAXUPNPPORTMAPS parameter sets an explicit limit to the number of dynamic NAT entries (or port maps) available for UPnP on the firewall policy. The default is 250.

The OTHERTIMEOUT parameter specifies the timeout period in minutes for sessions other than TCP or UDP. The default is 20 minutes, except for ICMP sessions, which defaults to 10 minutes. If the value of OTHERTIMEOUT is set to more than 10 minutes, ICMP sessions timeout after 10 minutes.

The TCPTIMEOUT parameter specifies the timeout period in minutes for a TCP session. The default is 60 minutes.

The UDPTIMEOUT parameter specifies the timeout period in minutes for a UDP session. The default is 20 minutes.

The UPNP parameter specifies whether this firewall policy is enabled for UPnP sessions. If set to ON, YES, or ENABLED, this policy is enabled. If set to OFF, NO, or DISABLED, this policy is disabled.

If you enable the load balancer, the value configured for the load balancer's ORPHANTIMEOUT parameter in the **set loadbalancer** command on page 50-37 of Chapter 50, Load Balancer, overwrites values set for the TCPTIMEOUT, UDPTIMEOUT, and OTHERTIMEOUT parameters.

Specifying a value of 0 minutes for any timeout parameters sets the timeout period to 30 seconds (0.5 minutes).

Examples

To timeout TCP sessions for zone1 after 15 seconds of inactivity, use the command:

```
set fire poli=zone1 tcpt=15
```

To set the number of port maps to the maximum of 1000 on the firewall policy called *upnp*, use the command:

```
set fire poli=upnp maxupnpportmaps=1000
```

Related Commands

delete firewall session
show firewall policy

set firewall policy attack

Syntax

```
SET FIREwall POLIcy=policy-name ATTack={DOSFlood|FRAGment|
   HOSTScan|IPSPoof|LAND|PINGOfdeath|PORTScan|SMTPrelay|
   SMURF|SMURFAmp|SPAM|SYNAttack|TCPTIny|UDPAttack}
   [INTrigger=count] [OUTTrigger=count] [DETail=count]
   [TIme=minutes]
```

where:

- *count* is a number from 0 to 4294967295.
- *minutes* is a time period from 5 to 4294967295 minutes.
- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description

This command sets the threshold levels at which notifications and triggers are generated for attack events. One of the parameters INTRIGGER, OUTTRIGGER, DETAIL or TIME must be specified.

The ATTACK parameter specifies the type of attack for which thresholds are being set, and is one of the following:

- DOSFLOOD—A denial of service attack when a remote user continually sends unwanted traffic.
- FRAGMENT—An attack using TCP fragments that are either too large or can never be reassembled.
- HOSTSCAN—A scan of the hosts of the private network.
- IPSPOOF—An attack using IP packets in which the source address has been spoofed (altered).
- LAND—A denial of service attack where a remote user sends IP packets with the same address in the source and destination address fields.
- PINGOFDEATH—A denial of service attack where a remote user sends ping packets with illegal sizes, or an excessive number of ICMP messages.
- PORTSCAN—A portscan of the firewall or private network.
- SMTPRELAY—An attempt by a remote SMTP server to get the private SMTP server to relay email to a non-local user.
- SMURF—An *Internet Control Message Protocol* (ICMP) echo request with a broadcast destination address.
- SMURFAMP—A TCP SYN packet with a broadcast destination address.
- SPAM—An attempt to deliver an email message, which has a source address that has been identified as a source of SPAM (unsolicited email), to the private SMTP server.
- SYNATTACK—An attack on a host using multiple opening TCP SYN packets to exhaust a host's available sessions or memory.
- TCPTINY—An attack on a host using TCP tiny fragments.
- UDPATTACK—An attack using UDP packets to probe for open UDP ports.

The INTRIGGER parameter specifies the number of events that must occur in traffic from a public interface before a notify event is generated.

The OUTTRIGGER parameter specifies the number of events that must occur in traffic from a private interface before a notify event is generated.

The DETAIL parameter specifies the number of packets recorded in the deny event queue for a notify event. This can be useful for tracking port scan and host scan attacks. The deny event queue can be displayed with the **show firewall event** command.

The TIME parameter specifies the time period in minutes within which event counters must reach the defined levels to trigger a notify event.

Default settings for INTRIGGER, OUTTRIGGER, DETAIL, and TIME depend on the type of attack (Table 41-7).

Table 41-7: Defaults for **set firewall policy attack** command parameters.

ATTACK	INTRIGGER	OUTTRIGGER	TIME	DETAIL	Trigger
DOSFLOOD	80	160	2	5	DOSATTACK
FRAGMENT	1	1	2	0	FRAGMENT
HOSTSCAN	64	128	2	5	HOSTSCAN
IPSPOOF	1	1	2	0	DOSATTACK
LAND	1	1	2	0	DOSATTACK
OTHER	64	128	2	5	DOSATTACK
PINGOFDEATH	1	1	2	0	DOSATTACK
PORTSCAN	64	128	2	5	PORTSCAN
SMTPRELAY	1	1	2	5	SMTPATTACK
SMURF	1	1	2	0	SMURFATTACK
SMURFAMP	1	1	2	5	SMTPATTACK
SPAM	1	1	2	5	SMTPATTACK
SYNATTACK	32	128	2	5	SYNATTACK
TCPTINY	1	1	2	0	TCPATTACK
UDPATTACK	32	128	2	5	DOSATTACK

When the number of attacks recorded by the firewall exceeds the threshold for that type of attack within the time period, the firewall generates a *start of attack* notification event and a trigger. For each time period that the attacks continue to exceed the threshold, the firewall generates an *attack in progress* notification event. When the number of attacks falls below the threshold for the time period, the firewall generates an *end of attack* notification event and a trigger.

Example To set a notification threshold for a denial of service attack (on zone1) of 150 events within 5 minutes from a public interface, use the command:

```
set fire poli=zone1 att=dosf int=150 ti=5
```

Related Commands **show firewall policy attack**

set firewall policy rule

Syntax

```
SET FIREwall POLIcy=policy-name RUle=rule-id
   [PROTocol={protocol|ALL|EGP|GRE|ICmp|OSPF|SA|TCP|UDP}]
   [AFTer=hh:mm] [BEFore=hh:mm] [DAYs={MON|TUE|WED|THU|
   FRI|SAT|SUN|WEEKDAY|WEEKEND}[,...]]
   [ENCapsulation={NONE|IPSec}] [GBLIP=ipadd]
   [GBLPort={ALL|port[-port]|service-name}]
   [GBLRemoteip=ipadd[-ipadd]] [IP=ipadd[-ipadd]]
   [NATMask=ipadd] [POrt={ALL|port[-port]|service-name}]
   [REMoteip=ipadd[-ipadd]] [SOurceport={ALL|port[-port]}]
   [TTL=hh:mm]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *rule-id* is a number from 1 to 299.
- *protocol* is an Internet IP protocol number.
- *hh:mm* is a time in hours and minutes.
- *ipadd* is an IP addresses in dotted decimal notation.
- *port* is an Internet service port number or name.
- *list-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *service-name* is a pre-defined name for an IP service (Table 41-5 on page 41-46).

Description

This command modifies a rule defining the access allowed between private and public interfaces of the specified policy. By default, access from public interfaces (outside the firewall) is denied and access from private interfaces (inside the firewall) is allowed. To refine the security policy, additional rules can be added to allow or deny access based on IP addresses, port numbers, day of the week, or time of day. Each rule for a specific interface in a policy is processed in order, starting with the lowest numbered rule and proceeding to the highest numbered rule, or until a match is found.

When a firewall policy rule is modified, new firewall sessions apply the modified rule. Firewall sessions that existed before the modification continue according to the unmodified rule until:

- they timeout due to a lack of traffic
- the firewall session is deleted with the **delete firewall session** command
- the firewall is reset with the **disable firewall** command then the **enable firewall** command.

In addition to rules based on IP address, port, protocol, date and time, the processing of ICMP packets, IP packets with options set and ping packets can be enabled or disabled on a per-policy basis using the **enable firewall policy** command on page 41-67 and the **disable firewall policy** command on page 41-62.

The POLICY parameter specifies the policy containing the rule to be modified. The specified policy must already exist.

The RULE parameter specifies the rule to be modified.

The AFTER and BEFORE parameters specify the time period when the rule is active. Time periods begin and end at midnight, and should be entered in 24-hour format. The AFTER and BEFORE parameters can be used together, but the time specified for AFTER must be earlier than the time specified for BEFORE. If AFTER is specified in a rule without specifying BEFORE, the rule becomes active from AFTER until midnight. If BEFORE is specified without AFTER, the rule becomes active from midnight until BEFORE.

When set to IPSEC, the ENCAPSULATION parameter specifies that traffic must match the rule when it has been decapsulated from an IPsec tunnel. This is sometimes useful for selecting traffic arriving from an IPsec tunnel that is using a dynamically assigned IP source address. If NONE is specified, encapsulation does not take place.

The DAYS parameter specifies the days in a comma-separated list that the rule applies. This lets rules be active on certain days of the week. The value WEEKDAY covers Monday to Friday. The value WEEKEND covers Saturday and Sunday.

The GBLIP parameter specifies a single IP address that is matched to the destination address of packets received on a public interface. The GBLIP parameter also specifies the global IP address to be used as the public IP address for private side devices if NAT is active on the interface, or if the value specified for the ACTION parameter is NAT.

The GBLPORT parameter specifies the port number, service name, or range of port numbers that apply to the rule if NAT is active on an interface. The GBLPORT parameter cannot be used when ACTION=NAT is specified. In this situation, the PORT parameter should be used instead.

The application of the GBLREMOTEIP parameter changes depending on the type of interface it is applied to. If the INTERFACE parameter specifies a public interface and the NATTYPE is REVERSE or DOUBLE, the GBLREMOTEIP parameter specifies a single IP address that is matched to the source IP address of packets received on that interface. If the INTERFACE parameter specifies a public interface, and the NATTYPE parameter is ENHANCED, then the GBLREMOTEIP parameter specifies a single IP address or a range of IP addresses that is matched to the source IP address of packets received on that interface. If the INTERFACE parameter specifies a private interface, the GBLREMOTEIP parameter is substituted as the destination address for packets received on the interface. This parameter should be specified only when the ACTION parameter is NAT and the NATTYPE is ENHANCED, REVERSE, or DOUBLE.

The GBLREMOTEIP parameter specifies a single IP address or a range of IP addresses that is matched to the source address of the packets received on a public interface. The GBLREMOTEIP parameter also specifies the global IP address to be used as the public IP address for public side devices if the value specified for the ACTION parameter is NAT.

The IP parameter specifies a single IP address or a range of IP addresses that match the source address of packets received on a private interface. The IP parameter also specifies the IP address to be used as the private IP address for

private side devices if NAT is active on the interface, or if the value specified for the ACTION parameter is NAT.

The NATMASK parameter specifies an IP address mask to be used to translate IP addresses from one subnet to another. The MASK parameter must be specified when the rule action is NAT and the NATTYPE is specified as DOUBLE, REVERSE, or STANDARD. The NATMASK parameter can be used when translating entire subnets from one address to another. If DOUBLE NAT is specified, the NATMASK is applied to the IP, GBLIP, REMOTEIP and GBLREMOTEIP parameters. If REVERSE NAT is specified, the NATMASK is applied to both the REMOTEIP and GBLREMOTEIP parameters. If STANDARD NAT is specified, the NATMASK is applied to both the IP and GBLIP parameters. The IP, GBLIP, REMOTEIP and GBLREMOTEIP parameters must specify a single IP address if the NATMASK parameter is used.

The required parameters for the firewall NAT Rules explained in the IP, REMOTEIP, GBLIP, GBLREMOTEIP, and NATMASK paragraphs above are summarised in Table 41-6 on page 41-47.

The PORT parameter specifies a port number, a range of port numbers, or a pre-defined service name (Table 41-5 on page 41-46) to match. If ALL is specified, the rule matches any port number. If dynamic NAT is active on the interface, it is possible to re-map a global port number to a different internal port number. For rules applied to a private interface, PORT is the destination port on the public network. For rules applied to a public interface, PORT is either the destination port on the private network or, in the case of NAT being applied, the destination port on the private network where traffic is mapped. When ACTION=NAT is specified, the PORT parameter specifies the port number or range of port numbers to match.

The PROTOCOL parameter specifies the IP protocol number or the name of a predefined protocol type to apply to the rule. If TCP or UDP is specified, then the PORT parameter must also be specified.

The REMOTEIP parameter specifies a single IP address or a range of IP addresses that match the destination address of packets received on a private interface. If the value specified for the ACTION parameter is not NAT, the REMOTEIP parameter also specifies a single IP address or range of IP addresses that match the source address of packets received on a public interface. If the value specified for the ACTION parameter is NAT, the REMOTEIP parameter also specifies the IP address to be used as the private IP address for public side devices.

The SOURCEPORT parameter specifies a source port for a TCP or UDP flow. This allows rules to be made based on the source port of the IP flow.

The TTL (time to live) parameter specifies the time in hours and minutes that the rule exists. The rule is active from the creation of the rule and is deleted after the specified time expires. All entries created from this rule are destroyed when the rule expires.

Examples To modify rule number 1 in the policy named *zone1* to match IP address 202.36.163.114, use the command:

```
set fire poli=zone1 ru=1 ip=202.36.163.114
```

To modify rule number 12 in the policy named *zone3* to change the TTL value, use the command:

```
set fire poli=zone3 ru=12 ttl=1:23
```

Related Commands

add firewall policy rule
delete firewall policy rule
show firewall policy

set firewall policy smtpdomain

Syntax

```
SET FIREwall POLIcy=policy-name SMTPdomain={domain-name|
   NONE}
```

where:

- *policy-name* is a character string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits (0-9), and the underscore character ("_").
- *domain-name* is a character string 1 to 131 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character ("_").

Description

This command sets a domain name for the SMTP proxy. The domain name is normally the same as the SMTP server that is located on the private side of the firewall. The specified domain name is used to compare with either:

- the domains of the destination addresses of all SMTP sessions that originate from the public side of the firewall, or
- the domains of the source addresses of all SMTP sessions that originate from the private side of the firewall.

If the domain name does not match, the firewall concludes that the email is trying to use the third party relay mechanism for delivery. If SMTP relaying is disabled then the session is terminated.

The POLICY parameter specifies the firewall policy with which the SMTP domain name is to be associated. The specified policy must already exit.

The SMTPDOMAIN parameter specifies the domain name of the SMTP server located on the private side of the firewall that needs to receive email from the public side of the firewall via an SMTP proxy. If NONE is specified, no domain matching is performed by the SMTP proxy.

Setting an SMTP domain name for a policy has effect only if the policy uses an SMTP proxy. If you do not set an SMTP domain name for a policy, the proxy rejects all inbound SMTP sessions.

Examples

To set *alliedtelesyn.co.nz* as the domain name for use by firewall policy *zone1*, use the command:

```
set fire poli=zone1 smtp=alliedtelesyn.co.nz
```

Related Commands

add firewall policy rule
add firewall policy spamsources
delete firewall policy proxy
delete firewall policy spamsources
disable firewall policy smtprelay
enable firewall policy smtprelay

show firewall

Syntax `SHow FIREwall`

Description This command displays a summary of all security policies that have been created and the interfaces assigned to each policy (Figure 41-4 on page 41-80, Table 41-8 on page 41-80).

Figure 41-4: Example output from the **show firewall** command

```
Firewall Configuration

Status ..................... enabled
Enabled Notify Options .... all
Notify Port ............... 1
Notify Mail To ............ root@netman.company.com
Maximum Packet Fragments .. 20

Policy : test
  UPNP ............................. enabled
    WAN interfaces .................. eth0
    LAN interfaces .................. vlan1
    Maximum port maps ............... 250
  Private Interface : eth0
  Public Interface  : eth1
    Method ..................... dynamic
    NAT ........................ enhanced
      Method .................. enhanced dynamic
      Private Interface ...... eth0
      Global IP .............. 192.168.72.89
```

Table 41-8: Parameters in the output of the **show firewall** command .

Parameter	Meaning
Status	Whether the firewall is enabled or disabled.
Enabled Notify Options	A list of the notification destinations currently enabled. One or more of: all mail manager port snmp none
Notify Port	The asynchronous port to which notifications are sent. Displayed when *Enable Notify Options* includes "port".
Notify Mail To	The email address to which notifications are sent. Displayed when *Enable Notify Options* includes "mail".
Maximum Packet Fragments	The maximum number of fragments that a packet may consist of when enhanced fragment handling is enabled.
Policy	The name of a policy.
UPnP	Whether UPnP is enabled or disabled for the policy.
WAN Interfaces	The WAN interface that UPnP uses on this policy.
LAN Interfaces	The LAN interfaces that UPnP uses on this policy.
Maximum port maps	The maximum number of port maps for UPnP allowed on this policy.

Table 41-8: Parameters in the output of the **show firewall** command

Parameter	Meaning
Private Interface	The name of a private interface assigned to the policy.
Public Interface	The name of a public interface assigned to the policy.
Method	Whether the method that passes packets to or from the public interface is dynamic or pass all.
NAT	Whether the type of NAT translation enabled is standard or enhanced. Displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation: None Static Static interface Dynamic Dynamic interface Enhanced static Enhanced dynamic Enhanced interface This field depends on the combination of options configured in the **add firewall policy nat** command on page 41-39, and is displayed when NAT is enabled on the policy.
NAT/Private Interface	The private interface to which NAT translations apply. Displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations. Displayed when NAT is enabled on the policy.

Related Commands

add firewall policy interface
create firewall policy
delete firewall policy interface
destroy firewall policy
disable firewall
enable firewall

show firewall accounting

Syntax

```
SHow FIREwall ACCounting [POLIcy=policy-name]
   [REVerse=number] [TAil=number]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *number* is a decimal number from 1 to 60.

Description

This command displays the currently stored accounting records for the specified or all policies (Figure 41-5 on page 41-82, Table 41-9 on page 41-82). Alternatively, accounting records can be displayed using the **show log** command on page 33-34 of Chapter 33, Logging Facility.

The POLICY parameter specifies the policy for which accounting records are to be displayed. The specified policy must already exist. If a value is not specified, accounting records for all policies are displayed.

The REVERSE parameter specifies that the accounting records are to be displayed in reverse order. If a value is specified, output is limited to the specified number of records.

The TAIL parameter specifies that the only the most recent accounting records are to be displayed. If a value is specified, output is limited to the specified number of records.

Figure 41-5: Example output from the **show firewall accounting** command

```
Policy : test
Date/Time   Event  Dir Prot  IP:Port <-> Dest IP:Port /Traffic statistics
------------------------------------------------------------------------------
20 10:10:00 START  OUT TCP   202.36.163.10:1113 192.168.72.50:80
20 10:10:01 END    OUT TCP   202.36.163.10:1112 192.168.72.50:80
                             Traffic out 5:695 in 5:367
20 10:10:15 START  OUT TCP   202.36.163.6:1025 192.168.72.50:23
20 10:10:15 START  IN  TCP   192.168.72.50:10778 192.168.72.89:113
20 10:11:01 END    OUT TCP   202.36.163.10:1069 192.168.72.50:80
                             Traffic out 5:692 in 5:366
20 10:11:01 END    OUT TCP   202.36.163.10:1070 192.168.72.50:80
                             Traffic out 5:696 in 5:365
20 10:11:02 END    OUT TCP   202.36.163.10:1071 192.168.72.50:80
                             Traffic out 5:696 in 5:365
20 10:12:01 END    OUT TCP   202.36.163.10:1113 192.168.72.50:80
                             Traffic out 5:695 in 5:367
20 10:12:15 END    IN  TCP   192.168.72.50:10778 192.168.72.89:113
                             Traffic out 3:164 in 6:264
------------------------------------------------------------------------------
```

Table 41-9: Parameters in the output of the **show firewall accounting** command .

Parameter	Meaning
Policy	The name of the policy.
Date/Time	The date and time of the entry.
Event	The event recorded by the entry; either START or END.
Dir	The direction of the flow; either IN or OUT.
Prot	The protocol for the flow; either ICMP, TCP, UDP, or the IP protocol number.
IP:Port	The source IP address and port for the flow.
Dest IP:Port	The destination IP address and port for the flow.
Traffic statistics	The number of packets and octets processed for the outgoing or incoming traffic flows, expressed in the format "*direction packets*:*octets*".

ICMP pings display end records only to reduce the number of records stored.

Related Commands

disable firewall policy
enable firewall policy
show firewall policy

show firewall arp

Syntax `SHow FIREwall ARP [POLIcy=`*`policy-name`*`]`

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description This command displays information about IP addresses specified in Firewall NAT configurations associated with that policy for which ARP responses from the router may be required (Figure 41-6 and Table 41-10).

The POLICY parameter specifies a firewall policy and displays IP addresses for NAT configurations with that policy. If this parameter is not specified, IP addresses are displayed for all policies.

Figure 41-6: Example output from the **show firewall arp** command

```
IP                ARP Interfaces  NAT Type    Int        Gbl Int    Rule
(range)           Policy
-------------------------------------------------------------------------------
172.20.8.50       Public          Int based   eth0-0     eth1-0     -
                  Office
172.20.8.57       All Public      Rule        eth0-1     -          1
 -172.20.8.62     LAN
-------------------------------------------------------------------------------
```

Table 41-10: Parameters in the output of the **show firewall arp** command .

Parameter	Meaning
IP (range)	An IP address or range for which the router may be required to send ARP responses.
Policy	The name of the policy whose NAT configuration the IP address (range) belongs to.
ARP Interfaces	Interfaces in the policy on which ARP requests are permitted: Public - ARP requests are permitted on the public interface specified by the Gbl Int parameter All Public - ARP requests are permitted on all of the policy's public interfaces Private - ARP requests are permitted on the private interface specified by the Int parameter All Private - ARP requests are permitted on all of the policy's private interfaces An address in an ARP request must match the subnet of the interface on which the ARP request is received.
NAT Type	The type of NAT configuration associated with the IP address: Int Based - The address (range) was specified by an interface-based NAT configured with the **add firewall policy nat** command Rule - The address (range) was specified by a NAT rule configured by the **add firewall policy rule** command, where the ACTION parameter was specified as NAT

Table 41-10: Parameters in the output of the **show firewall arp** command (continued).

Parameter	Meaning
Int	The private interface associated with the NAT configuration. If the NAT Type is Int based, this is the private interface specified by the INTERFACE parameter in the **add firewall policy nat** command. If the NAT Type is Rule, this is the interface to which the rule is attached. If this is a private interface, a dash indicates that the rule is attached to a public interface (see the Gbl Int parameter).
Gbl Int	The public interface associated with the NAT configuration. If the NAT Type is Int based, this is the public interface specified by the GBLINTERFACE parameter in the **add firewall policy nat** command. If the NAT Type is Rule, this is the interface to which the rule is attached. if this is a public interface, a dash indicates that the rule is attached to a private interface (see the Int parameter).
Rule	The number of the rule associated with this entry. When the NAT Type is Int based, no value is displayed.

Examples

To display ARP information for the firewall policy named "Office", use the command:

```
sh fire arp poli=Office
```

Related Commands

add firewall policy nat
add firewall policy rule
delete firewall policy nat
delete firewall policy rule
set firewall policy rule

show firewall event

Syntax

```
SHow FIREwall EVent={ALLOw|DENY|NOTify}
   [POLIcy=policy-name] [REVerse=number] [TAil=number]
```

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *number* is a decimal number from 1 to 60.

Description

This command displays information about recent firewall events (Figure 41-7 on page 41-85, Table 41-11 on page 41-86).

The EVENT parameter specifies which category of events to display. If a value is not specified, all events are displayed. If ALLOW is specified, events for flows that have been allowed are displayed. If DENY is specified, events for flows that have been denied are displayed. If NOTIFY is specified, notification events are displayed.

The POLICY parameter specifies the policy for which events are to be displayed. The specified policy must already exist. If a value is not specified, events for all policies are displayed.

The REVERSE parameter specifies that the events are to be displayed in reverse order. If a value is specified, output is limited to the specified number of events.

The TAIL parameter specifies that the only the most recent events are to be displayed. If a value is specified, output is limited to the specified number of events.

Figure 41-7: Example output from the **show firewall event** command

```
Policy : test - Notify Events:
Date/Time   Dir Prot Number IP:Port <map> Dest IP:Port /Reason /IP header
-------------------------------------------------------------------------
15 15:21:58 IN  TCP       2 203.97.191.217:1046 192.168.72.33:20
                SYN attack underway
15 15:22:00 IN  TCP       2 203.97.191.217:0 192.168.72.33:0
                Port scan underway
                45000044 8d8f4000 3f061097 cb61bfd9 ca314821 04160014 9610e710
                00000000 c0024000
15 15:25:55 IN  TCP       1 203.97.191.217:0 192.168.72.33:0
                Port scan finished
                45000044 8d8f4000 3f061097 cb61bfd9 ca314821 04160014 9610e710
                00000000 c0024000
15 15:28:55 IN  TCP       1 203.97.191.217:1046 192.168.72.33:20
                SYN attack finished
-------------------------------------------------------------------------

Policy : test - Deny Events:
Date/Time   Dir Prot Number IP:Port <map> Dest IP:Port /Reason /IP header
-------------------------------------------------------------------------
19 18:32:43 OUT TCP      10 192.168.72.33:23366 192.12.33.2:113
                Policy rejected
                45000033 c83d4000 40067f26 ca314821 c00c2102 5b460071 a207ca65
                04fb64e5 50187c00
19 20:32:35 OUT TCP       1 192.168.72.33:26973 210.55.162.101:25
                TCP open failed
19 21:34:54 OUT TCP      10 192.168.72.33:28897 12.7.242.94:113
                Policy rejected
                45000034 d9994000 40065072 ca314821 0c07f25e 70e10071 3d6a5027
                05014535 50187c00
20 01:59:51 OUT TCP       1 192.168.72.33:6595 210.55.162.101:25
                TCP open failed
20 09:53:37 OUT TCP       1 192.168.72.33:19610 207.46.131.137:80
                Policy rejected
                45000222 203e4000 4006b38d ca314821 cf2e8389 4c9a0050 644c1cf8
                0520df4d 50187c00
-------------------------------------------------------------------------

Policy : test - Allow Events:
Date/Time   Dir Prot Number IP:Port <map> Dest IP:Port /Reason /IP header
-------------------------------------------------------------------------
20 09:51:11 OUT TCP       1 192.168.72.33:17972 207.46.131.137:80
                TCP session started
20 09:51:39 IN  UDP       1 192.168.72.41:53 192.168.72.33:53
                UDP flow started
20 09:51:44 IN  TCP       1 128.230.18.29:2013 192.168.72.33:25
                TCP session started
20 09:51:44 IN  TCP       1 137.103.210.2:1345 192.168.72.33:25
                TCP session started
-------------------------------------------------------------------------
```

Table 41-11: Parameters in the output of the **show firewall event** command .

Parameter	Meaning
Policy	The name of the policy to which the following events apply.
Date/Time	The date and time of the event.
Dir	The direction of the flow; either IN or OUT.
Prot	The protocol for the flow; either ICMP, TCP, UDP, or the IP protocol number.
Number	The number of times the event has occurred.
IP:Port	The source IP address and port for the flow.
Dest IP:Port	The destination IP address and port for the flow.
Reason	The reason for the event record.
IP Header	A dump of the first nine octets of the IP header of the packet causing the event.

Related Commands

disable firewall notify
enable firewall notify
show firewall accounting
show firewall policy
show firewall session

show firewall policy

Syntax

```
SHow FIREwall POLIcy=policy-name [COUnter] [DYnamic]
   [LISt] [SUMmary] [USer]
```

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description

This command displays detailed information about the specified or all policies (Figure 41-8 on page 41-87, Table 41-12 on page 41-88).

The POLICY parameter specifies the policy to be displayed. The specified policy must already exist. If a value is not specified then information for all policies is displayed.

The COUNTER parameter displays counters for the specified policy or all policies (Figure 41-9 on page 41-93, Table 41-13 on page 41-94).

The DYNAMIC parameter displays a list of the usernames assigned to the specified dynamic interface template or all dynamic interface templates in a policy (Figure 41-10 on page 41-98, Table 41-14 on page 41-98).

The LIST parameter displays information about address lists assigned to the specified policy or all policies (Figure 41-11 on page 41-99, Table 41-15 on page 41-99).

The SUMMARY parameter displays a summary of the information for each policy.

The USER parameter displays the specified username or all usernames and the dynamic interface template(s) to which the username(s) are assigned (Figure 41-12 on page 41-99, Table 41-16 on page 41-99).

Figure 41-8: Example output from the **show firewall policy** command

```
Policy : admin
  TCP Timeout(s)..................... 3600
  UDP Timeout(s)..................... 1200
  Other Timeout(s)................... 1200
  Accounting ........................ enabled
  Enabled Logging Options ........... allow denydump
  Enabled Debug Options ............. checksum
  Enhanced Fragment Handling ........ udp
  Enabled IP options ................ none
  Enabled ICMP forwarding ........... ping timeexceeded
  Receive of ICMP PINGS ............. enabled
  Number of Notifications ........... 0
  Number of Deny Events ............. 20
  Number of Allow Events ............ 8987
  Number of Active TCP Opens ........ 0
  Number of Active Sessions ......... 1
  Cache Hits ........................ 429073
  Discarded ICMP Packets ............ 74
  Spam Source Files ................. spam.spa
  SMTP Domain ....................... alliedtelesyn.co.nz
  HTTP Proxy Filter File ............ urlfilt.txt
  Cookies ........................... enabled
  TCP Setup Proxy ................... enabled
  UPNP .............................. enabled
    WAN interfaces .................. eth0
    LAN interfaces .................. vlan1
    Maximum port maps ............... 250
  Private Interface: eth0
    Trust Private ................... yes
  Public Interface:  eth1
    Method ............................ dynamic
    Proxy ............................. http
      Private Interface ............... eth0
      IP .............................. 192.168.1.10
      Direction ....................... both
      Days ............................ all
      Apprule ......................... 1
         Application .................. MMS
         Action ....................... Allow
      Apprule ......................... 2
         Application .................. BBMS
         Action ....................... Allow
```

Table 41-12: Parameters in the output of the **show firewall policy** command .

Parameter	Meaning
Policy	The name of a policy.
TCP Timeout	The timeout period in seconds for inactive TCP flows.
UDP Timeout	The timeout period in seconds for inactive UDP flows.
Other Timeout	The timeout period in seconds for inactive flows for protocols other than TCP or UDP.
Accounting	Whether accounting is enabled or disabled for the policy.
Enabled Logging Options	One or more of the logging options currently enabled: Allow Deny Denydump Everydeny Inaicmp Inallow Inaother Inatcp Inaudp Inddicmp Inddother Inddtcp Inddudp Inddump Indeny Indicmp Indother Indtcp Indudp Outaicmp Outallow Outaother Outatcp Outaudp Outddicmp Outddother Outddtcp Outddudp Outddump Outdeny Outdicmp Outdother Outdtcp Outdudp If no options are specified, "none" is displayed.

Table 41-12: Parameters in the output of the **show firewall policy** command (continued).

Parameter	Meaning
Enabled Debug Options	One or more of the debug options currently enabled: All ARP HTTP Packet Process Proxy SMTP UPnP If no options are specified, "none" is displayed.
Enhanced Fragment Handling	A list of the protocol types for which the policy can handle large fragmented packets. The list may contain one or more of "ICMP", "UDP", "other" or "none". If "other" is listed, protocols that are not ICMP or UDP (or TCP) are permitted to send large fragmented packets. If "none" is listed, no protocols are permitted to send large fragmented packets. If a protocol is not listed, then the default fragment constraints apply, which means that the IP packet must consist of no more than 8 fragments with a total of 1780 bytes of data.
Enabled IP options	One or more of the IP options allowed in IP packets forwarded by this policy: All Record_route Security Sourcerouting Timestamp If no options have been specified, "none" is displayed.
Enabled ICMP forwarding	One or more of the ICMP packet types forwarded by this policy: All Parameter Ping Sourcequench Timeexceeded Timestamp Unreachable If no packet types have been specified, "none" is displayed.
Receive of ICMP PINGS	Whether the reception of ICMP PING packets is enabled or disabled for this policy.
Number of Notifications	The number of notifications generated.
Number of Deny Events	The number of deny events for this policy.
Number of Allow Events	The number of allow events for this policy.
Number of Active TCP Opens	The number of currently active TCP connections for this policy.
Number of Active Sessions	The number of currently active sessions for this policy.

Table 41-12: Parameters in the output of the **show firewall policy** command (continued).

Parameter	Meaning
Cache Hits	The number of flow lookups found from the cache.
Discarded ICMP Packets	The number of ICMP packets discarded by this policy.
Spam Source Files	A list of files each containing a list of email addresses and domain names that are not permitted to send email through the firewall SMTP proxy.
SMTP Domain	The domain name of the email server found on the private side of the firewall.
HTTP Proxy Filter File	Name of a text file containing a list of domain names, keywords and cookie options that are not allowed to pass through HTTP proxies configured under this policy. This parameter is only shown if a URL filter file has been specified for this policy.
Cookies	Whether cookies are allowed to pass through HTTP proxies configured under this policy. If enabled, all cookies are permitted unless specifically denied by an entry in the HTTP proxy filter file. If disabled, no cookies are permitted. This parameter is shown when an HTTP proxy has been configured for this policy with Direction set to Out or Both.
UPnP	Whether UPnP is enabled or disabled for the policy.
WAN Interfaces	The WAN interface that UPnP uses on this policy.
LAN Interfaces	The LAN interfaces that UPnP uses on this policy.
Maximum port maps	The maximum number of port maps for UPnP allowed on this policy.
TCP Setup Proxy	Indicates whether the TCP setup proxy is disabled or enabled. The default is enabled.
Proxy	The type of proxy in use.
IP	The IP address of the server on the private intranet. Incoming traffic received by the proxy is sent to this address.
Direction	The direction in which traffic is permitted to flow through the proxy.
Dynamic Template	The name of a dynamic interface template associated with the policy.
IP List	The name of an IP list assigned to this policy.
Hardware List	The name of a hardware address list assigned to this policy.
File name	The name of the file containing the list.
Number IP hosts	The number of IP hosts in the list.
Number Networks	The number of IP networks in the list.
Number MAC addresses	The number of MAC addresses in the list.
Private Interface	The name of a private interface assigned to the policy.
Trust Private	Whether devices connected to the interface have unrestricted access to the router. This parameter is displayed for private interfaces, not public ones.
Public Interface	The name of a public interface assigned to the policy.
Method	The method used to packets to or from the public interface; either Dynamic or Passall.

Table 41-12: Parameters in the output of the **show firewall policy** command (continued).

Parameter	Meaning
NAT	Whether the type of NAT translation enabled is Standard or Enhanced. Only displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation: None Static Static interface Dynamic Dynamic interface Enhanced static Enhanced dynamic Enhanced interface This field depends on the combination of options configured in the **add firewall policy nat** command on page 41-39, and is only displayed when NAT is enabled on the policy.
NAT/Private Interface	The private interface to which NAT translations apply. Displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations.Displayed when NAT is enabled on the policy.
Rule	The identifier for a rule associated with the private or public interface.
Action	The action to perform when a flow matches this rule; either Allow or Deny.
IP List	The name (and file) of an IP list referenced by this rule.
Hardware List	The name (and file) of a hardware address list referenced by this rule.
Protocol	The IP protocol type to apply to this rule.
Port	The port number, service name (Table 41-5 on page 41-46) or range of port numbers to apply to this rule.
Global IP	The IP address to apply to this rule when NAT is active on the interface.
Global Port	The port number, service name (Table 41-5 on page 41-46) or range of port numbers to apply to this rule when NAT is active on the interface.
Remote IP	The remote IP address to match for this rule.
Source Port	The source port to match for this rule.

Table 41-12: Parameters in the output of the **show firewall policy** command (continued).

Parameter	Meaning
Days	One or more of the days on which this rule is active: Mon Tue Wed Thu Fri Sat Sun All
Apprule	The identifier for an application rule associated with the private or public interface.
Application	The name of the application protocol to which this rule applies.
Action	The action to perform when a flow matches this rule; either Allow or Deny.
After	The time of day after which this rule is active.
Before	The time of day before which this rule is active.

Figure 41-9: Example output from the **show firewall policy counter** command

```
Policy : test
Accounting ..................... enabled
Enabled Logging Options ........ allow denydump
Enabled Debug Options .......... none
Enhanced Fragment Handling ..... udp
Enabled IP options ............. none
Enabled ICMP forwarding ........ ping timeexceeded
Receive of ICMP PINGS .......... enabled
Number of Notifications ........ 0
Number of Deny Events .......... 20
Number of Allow Events ......... 9101
Number of Active TCP Opens ..... 0
Number of Active Sessions ...... 1
Cache Hits ..................... 430160
Discarded ICMP Packets ......... 74
Spam Source Files: ............. spam.spa
SMTP Domain .................... alliedtelesyn.co.nz
HTTP Proxy Filter File ......... urlfilt.txt
Cookies ........................  enabled
UPNP ........................... enabled
   WAN interfaces ................ eth0
   LAN interfaces ................ vlan1
   Maximum port maps ............. 250
   Number Port Mappings .......... 0
   Spawned Sessions .............. 2
Private Interface ............ eth0
  Total Packets Received ........ 186331
  Number Flows Started .......... 9083
  Number Cache Hits ............. 173174
  Number Dropped Packets ........ 0
  Number Unknown IP Protocols ... 0
  Number Bad ICMP Packets ....... 0
  Number Dumped ICMP Packets .... 0
  Number Spoofing Packets ....... 0
  Number Dropped GBLIP is Zero .. 0
  Number No Spare Entries ....... 0
Public Interface ............  eth1
  Method ........................ dynamic
  Total Packets Received ........ 0
  Number Flows Started .......... 0
  Number Cache Hits ............. 0
  Number Dropped Packets ........ 0
  Number Unknown IP Protocols ... 0
  Number Bad ICMP Packets ....... 0
  Number Dumped ICMP Packets .... 0
  Number Spoofing Packets ....... 0
  Number Dropped GBLIP is Zero .. 0
  Number No Spare Entries ....... 0
  Proxy ......................... http
    Private Interface ........... eth0
    IP .......................... 172.22.199.4
    Direction ................... both
    Sessions Handled ............ 10
    URL Denies .................. 2
    URL Allows .................. 5
    Cookie denies ............... 3
    Days ........................ all
```

Table 41-13: Parameters in the output of the **show firewall policy counter** command .

Parameter	Meaning
Policy	The name of a policy.
Accounting	Whether accounting is enabled or disabled for the policy.
Enabled Logging Options	One or more of the logging options currently enabled: Allow Deny Denydump Inaicmp Inallow Inaother Inatcp Inaudp Inddicmp Inddother Inddtcp Inddudp Inddump Indeny Indicmp Indother Indtcp Indudp Outaicmp Outallow Outaother Outatcp Outaudp Outddicmp Outddother Outddtcp Outddudp Outddump Outdeny Outdicmp Outdother Outdtcp Outdudp If no options are selected, "none" is displayed.
Enabled Debug Options	One or more of the debug options currently enabled: All Packet Process If no options are selected, "none" is displayed.

Table 41-13: Parameters in the output of the **show firewall policy counter** command

Parameter	Meaning
Enabled IP options	One or more of the IP options allowed in IP packets to be forwarded by this policy: All Record_route Security Sourcerouting Timestamp If no options are selected, "none" is displayed.
Enhanced Fragment Handling	A list of the protocol types for which the policy can handle large fragmented packets. The list may contain one or more of ICMP, UDP, other or none. If other is listed, protocols that are not ICMP or UDP (or TCP) are permitted to send large fragmented packets. If none is listed, no protocols are permitted to send large fragmented packets. If a protocol is not listed then the default fragment constraints apply, meaning that the IP packet must consist of no more than 8 fragments with a total of 1780 bytes of data.
Enabled ICMP forwarding	One or more of the ICMP packet types forwarded by this policy: All Parameter Ping Sourcequench Timeexceeded Timestamp Unreachable If no packet types are selected, "none" is displayed.
Receive of ICMP PINGS	Whether the reception of ICMP PING packets is enabled or disabled for this policy.
Number of Notifications	The number of notifications generated.
Number of Deny Events	The number of deny events for this policy.
Number of Allow Events	The number of allow events for this policy.
Number of Active TCP Opens	The number of currently active TCP connections for this policy.
Number of Active Sessions	The number of currently active sessions for this policy.
Cache Hits	The number of flow lookups found from the cache.
Discarded ICMP Packets	The number of ICMP packets discarded by this policy.
Spam Source Files	A list of files each containing a list of email addresses and domain names that are not permitted to send email through the firewall SMTP proxy.
SMTP Domain	The domain name of the email server found on the private side of the firewall.
HTTP Proxy Filter File	Name of a text file containing a list of domain names, URL's, keywords and cookie domain filters that are not allowed to pass through HTTP proxies configured under this policy. This parameter is only shown if a URL filter file has been specified for this policy.

Table 41-13: Parameters in the output of the **show firewall policy counter** command

Parameter	Meaning
Cookies	Indicates whether cookies are allowed to pass through HTTP proxies configured under this policy. If "enabled" is shown all cookies are permitted unless specifically denied by an entry in the HTTP proxy filter file. If "disabled" is shown no cookies are permitted. This parameter is only shown if an HTTP proxy has been configured for this policy with direction set to "out" or "both".
UPnP	Whether UPnP is enabled for the policy; one of Enabled or Disabled.
WAN Interfaces	The interfaces that this policy uses as public interfaces for UPnP.
LAN Interfaces	The interfaces that this policy uses as private interfaces for UPnP.
Maximum port maps	The maximum number of port maps for UPnP allowed on this policy.
Number Port Mappings	The number of current port mappings added by the *AddPortMapping* UPnP action. This includes port mappings that specify the remote host and those that have a wild card for the remote host (templates).
Spawned sessions	The number of UPnP port mappings created based on UPnP template port mappings.
Sessions Handled	The number of TCP sessions that have been handled by the proxy.
URL Denies	The number of times a match to a requested URL has been found in the HTTP proxy filter file resulting in the request being denied.
URL Allows	The number of times a match to a requested URL has been found in the HTTP proxy filter file resulting in the request being allowed.
Cookie Denies	The number of times a match to a domain or URL requesting the setting of a cookie has been found in the HTTP proxy filter file resulting in the request being denied.
Proxy	The type of proxy in use.
IP	The IP address of the server on the private intranet. Incoming traffic intercepted by the proxy is sent to this address.
Direction	The direction in which traffic flows through the proxy.
Number Hits	The number of sessions to use the proxy.
Rejected Spam Messages	The number of mail messages with source addresses matching an entry in the SMTP proxy configuration file that have been rejected. This parameter is only present with an SMTP proxy.
Rejected SMTP Relays	The number of messages requesting third party relay that have been rejected. This parameter is only present with an SMTP proxy.
Rejected Smurf Amp Attacks	The number of mail messages with a broadcast source of destination address that have been rejected. This parameter is only present with an SMTP proxy.
IP List	The name of an IP list assigned to this policy.

Table 41-13: Parameters in the output of the **show firewall policy counter** command

Parameter	Meaning
Hardware List	The name of a hardware address list assigned to this policy.
File name	The name of the file containing the list.
Number IP hosts	The number of IP hosts in the list.
Number Networks	The number of IP networks in the list.
Number MAC addresses	The number of MAC addresses in the list.
Private Interface	The name of a private interface assigned to the policy.
Public Interface	The name of a public interface assigned to the policy.
Total Packets Received	The total number of packets received on the interface.
Number Flows Started	The number of flows started on the interface.
Number Cache Hits	The number of flow lookups for the interface found from the cache.
Number Dropped Packets	The number of packets received on the interface that were dropped.
Number Unknown IP Protocols	The number of packets received on the interface with an unknown IP protocol.
Number Bad ICMP Packets	The number of badly formatted ICMP packets received on the interface.
Number Dumped ICMP Packets	The number of ICMP packets received on the interface that were dumped.
Number Spoofing Packets	The number of packets received on the interface with a spoofed address.
Number Dropped GBLIP Zero	The number of packets received on the interface that were dumped because the global IP address was zero.
Number No Spare Entries	The number of packets received on the interface that were dumped because the system had insufficient memory.
Number FTP Port Commands	The number of valid FTP port commands received on the interface.
Number Bad FTP Port Commands	The number of invalid FTP port commands received on the interface.
Method	The method used to packets to or from the public interface; either Dynamic or Passall.
NAT	The type of NAT translation enabled; either Standard or Enhanced. Displayed when NAT is enabled on the policy.
NAT/Method	The method used to perform NAT translation: None Static Dynamic Enhanced static Enhanced dynamic Enhanced interface This field depends on the combination of options configured in the **add firewall policy nat** command on page 41-39, and is displayed when NAT is enabled on the policy.

Table 41-13: Parameters in the output of the **show firewall policy counter** command

Parameter	Meaning
NAT/Private Interface	The private interface to which NAT translations apply. Displayed when NAT is enabled on the policy.
NAT Global IP	The global IP address used by NAT translations. Displayed when NAT is enabled on the policy.
Rule	The identifier for a rule associated with the private or public interface.
Action	The action to perform when a flow matches this rule; either Allow or Deny.
IP List	The name (and file) of an IP list referenced by this rule.
Hardware List	The name (and file) of a hardware address list referenced by this rule.
Protocol	The IP protocol type to apply to this rule.
Port	The port number, service name (Table 41-5 on page 41-46) or range of port numbers to apply to this rule.
Global IP	The IP address to apply to this rule when NAT is active on the interface.
Global Port	The port number, service name (Table 41-5 on page 41-46) or range of port numbers to apply to this rule, if NAT is active on the interface.
Remote IP	The remote IP address to match for this rule.
Source Port	The source port to match for this rule.
Days	The days on which this rule is active; a list of one or more of "mon", "tue", "wed", "thu", "fri", "sat", "sun" or "all".
After	The time of day after which this rule is active.
Before	The time of day before which this rule is active.

Figure 41-10: Example output from the **show firewall policy dynamic** command

```
Policy : test

Dynamic template : acc1
  Filename : fire.txt
     Users : user$qwerty user-jim user-very-long-name usera1
             usera10 usera2

  Users : graeme tony
```

Table 41-14: Parameters in the output of the **show firewall policy dynamic** command .

Parameter	Meaning
Policy	The name of a policy.
Dynamic Template	The name of a dynamic interface template associated with the policy.
File name/Users	The name of a file containing a list of usernames added with the **add firewall policy dynamic file** command, and the usernames read from the file.
Users	A list of usernames added with the **add firewall policy dynamic user** command.

Figure 41-11: Example output from the **show firewall policy list** command

```
Policy : office

Hardware List : devices ( listmac.txt )
MAC Address        Label
-------------------------------------------------
00-00-cd-02-03-01
00-00-cd-02-03-05  John's PC
00-00-ef-39-08-01  access server
-------------------------------------------------

IP List : iphosts ( listip.txt )
IP              - IP               Label
-------------------------------------------------
192.168.163.6                      FTP host
192.168.16.0      192.168.16.255   Test network
-------------------------------------------------
```

Table 41-15: Parameters in the output of the **show firewall policy list** command .

Parameter	Meaning
Policy	The name of a policy.
Hardware List	The name (and filename) of a hardware address list assigned to this policy.
IP List	The name (and filename) of an IP list assigned to this policy.
MAC address	A hardware address in the hardware address list.
IP	A IP address or network in the IP address list
Label	The name of the host associated with the address.

Figure 41-12: Example output from the **show firewall policy user** command

```
Policy : test

Dynamic template : acc1
  User : graeme
```

Table 41-16: Parameters in the output of the **show firewall policy user** command

Parameter	Meaning
Policy	The name of a policy.
Dynamic Template	The name of a dynamic interface template associated with the policy.
Users	A list of usernames added using the **add firewall policy dynamic user** command.

Related Commands

add firewall policy interface
add firewall policy list
add firewall policy nat
add firewall policy rule
create firewall policy
delete firewall policy interface
delete firewall policy list
delete firewall policy nat
delete firewall policy proxy
delete firewall policy rule
destroy firewall policy
disable firewall notify
disable firewall policy
enable firewall notify
enable firewall policy
set firewall policy rule
show firewall
show firewall event

show firewall policy attack

Syntax

```
SHow FIREwall POLIcy[=policy-name] ATTack
```

where *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").

Description

This command displays the trigger settings for a policy (Figure 41-13 on page 41-100 and Table 41-17 on page 41-101).

Figure 41-13: Example output from the **show firewall policy attack** command

```
Policy : test
Current Attack Setup

Attack        In Trigger   Out Trigger   Time Period (mins)   Detailed
Logged
-------------------------------------------------------------------------
dosflood         80           160          2                    5
fragment         1            1            2                    0
hostscan         64           128          2                    5
ipspoof          1            1            2                    0
land             1            1            2                    0
other            64           128          2                    5
pingofdeath      1            1            2                    0
portscan         32           64           2                    5
smtprelay        1            1            2                    5
smurf            1            1            2                    0
smurfamp         1            1            2                    5
spam             1            1            2                    5
synattack        32           64           2                    5
tcptiny          1            1            2                    0
udpattack        32           64           2                    5
-------------------------------------------------------------------------
```

Table 41-17: Parameters in the output of the **show firewall policy attack** command

Parameter	Meaning
Policy	The name of the firewall policy.
Attack Logged	The type of attack being logged.
In Trigger	The number of events that must occur in traffic from a public interface, within the time period, before an event notification is generated.
Out Trigger	The number of events that must occur in traffic from a private interface, within the time period, before an event notification is generated.
Time Period (mins)	The time period, in minutes, within which the specified number of events must occur before an event notification is generated.
Detailed	The number of packets recorded in the deny event queue (displayed using the **show firewall event** command) for each notification event.

Examples To display the trigger settings for a policy called "zone1", use the command:

```
sh fire poli=zone1 att
```

Related Commands **set firewall policy attack**

show firewall session

Syntax

```
SHow FIREwall SEssion[=session-number]
   [POLIcy=policy-name] [COUnter] [POrt={port-port|
   service-name}] [PROTocol={protocol|ALL|EGP|GRE|ICmp|
   OSPF|TCP|UDP}] [SUMmary] [UPNP]
```

where:

- *session-number* is the identifier for a currently active session.
- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character ("_").
- *port* is an Internet service port number or name.
- *service-name* is a pre-defined name for an IP service (Table 41-5 on page 41-46).
- *protocol* is an Internet IP protocol number.

Description This command displays information about the sessions and flows currently active for the specified policy (Figure 41-14 on page 41-102). If SESSION is specified, only information about the specified session is displayed. Otherwise, information about all sessions is displayed.

The POLICY parameter specifies the policy for which session information is to be displayed. The specified policy must already exist. If a value is not specified, session information for all policies is displayed.

If COUNTER is specified, session counters for the specified policy are displayed.

If SUMMARY is specified, only summary information for the specified policy is displayed.

If PROTOCOL is specified, the display is limited to sessions based on the specified IP protocol type.

If PORT is specified, the display is limited to sessions between ports in the specified range of ports or using the specified service (Table 41-5 on page 41-46).

If UPNP is specified, this command displays information about the sessions and flows through the firewall that are associated with UPnP. The sessions shown are the result of UPnP listen ports being used from the access point (a client PC) (Figure 41-15 on page 41-104, Table 41-18 on page 41-102). A session with both the Remote IP and Global Remote IP addresses showing 0.0.0.0 indicates that they are active listen ports. Listen ports are a result of an access point using the *AddPortMapping* action in the WANIPConnection service within UPnP.

Figure 41-14: Example output from the **show firewall session** command

```
Policy : test
Current Sessions
-------------------------------------------------------------------------------
2131 TCP      IP: 10.8.0.172:23           Remote IP: 192.168.1.10;31729
          Gbl IP: 192.168.1.1:23        Gbl Remote IP: 192.168.1.10:31729
     TCP state ............................ established
     Start time ........................... 00:15:02 06-Sep-2001
     Seconds to deletion .................. 3594
-------------------------------------------------------------------------------
```

Table 41-18: Parameters in the output of the **show firewall session** command .

Parameter	Meaning
Policy	The name of a policy.
hex-num	The session identifier
TCP/UDP/*number*	The IP protocol (either TCP, UDP, or an IP protocol number), followed by the source address:port, the global IP address:mapped port, and the destination IP address:port
Packets from private IP	The number of packets forwarded from the private network to the public network.
Octets from private IP	The number of octets forwarded from the private network to the public network.
Packets to private IP	The number of packets forwarded from the public network to the private network.
Octets to private IP	The number of octets forwarded from the public network to the private network.

Table 41-18: Parameters in the output of the **show firewall session** command

Parameter	Meaning
IP	This IP address is the source address of outbound packets and the destination address of inbound packets in this session, as seen on the private side of the firewall.
Remote IP	This IP address is the destination address of outbound packets and the source address of inbound packets in this session, as seen on the private side of the firewall.
Gbl IP	This IP address is the source address of outbound packets and the destination address of inbound packets in this session, as seen on the public side of the firewall.
Gbl Remote IP	This IP address is the destination address of outbound packets and the source address of inbound packets in this session, as seen on the public side of the firewall.
TCP state	The state of the TCP session: Free Closed Listen SynSent SynReceived Established FinWait1 FinWait2 CloseWait LastAck Closing TimeWait DeleteTCB SynSent SynReceived RADIUS query
Private SEQ number	The current sequence number for the TCP connection to the private IP address.
Private ACK number	The current acknowledgement number for the TCP connection to the private IP address.
Private max window size	The current maximum window size for the TCP connection to the private IP address.
Public SEQ number	The current sequence number for the TCP connection to the public IP address.
Public ACK number	The current acknowledgement number for the TCP connection to the public IP address.
Public max window size	The current maximum window size for the TCP connection to the public IP address.
Sequence Delta	The difference between the current sequence numbers for the private and public connections.

Table 41-18: Parameters in the output of the **show firewall session** command

Parameter	Meaning
ICMP type	The type of ICMP request, for ICMP sessions; either "Echo request", "Time request", "Name request" or "Unknown ICMP type".
Start time	The date and time that the session was started.
Seconds to deletion	The number of seconds remaining before the session is automatically deleted.

Figure 41-15: Example output from the **show firewall session upnp** command.

```
Current Sessions
-------------------------------------------------------------------------------
4fed UDP      IP: 192.168.0.16:11767        Remote IP: 0.0.0.0:0
         Gbl IP: 202.49.72.23:20461    Gbl Remote IP: 0.0.0.0:0
     Start time ........................... 12:45:27 21-Oct-2003
     Seconds to deletion .................. 31535910
39f6 TCP      IP: 192.168.0.16:13953        Remote IP: 0.0.0.0:0
         Gbl IP: 202.49.72.23:14838    Gbl Remote IP: 0.0.0.0:0
     TCP state ............................ closed
     Start time ........................... 12:45:27 21-Oct-2003
     Seconds to deletion .................. 31535910
-------------------------------------------------------------------------------
```

Table 41-19: Parameters in the output of the **show firewall session upnp** command .

Parameter	Meaning
Policy	The name of a policy.
hex-num	The session identifier
TCP/UDP/*number*	The IP protocol (one of "TCP", "UDP" or an IP protocol number), followed by the source address:port, the global IP address:mapped port, and the destination IP address:port
IP	This IP address is the source address of outbound packets and the destination address of inbound packets in this session, as seen on the private side of the firewall.
Remote IP	This IP address is the destination address of outbound packets and the source address of inbound packets in this session, as seen on the private side of the firewall.
Gbl IP	This IP address is the source address of outbound packets and the destination address of inbound packets in this session, as seen on the public side of the firewall.
Gbl Remote IP	This IP address is the destination address of outbound packets and the source address of inbound packets in this session, as seen on the public side of the firewall.

Table 41-19: Parameters in the output of the **show firewall session upnp** command

Parameter	Meaning
TCP state	The state of the TCP session: free closed listen synSent synReceived established finWait1 finWait2 closeWait lastAck closing timeWait deleteTCB synSent synReceived RADIUS query
Start time	The date and time that the session was started.
Seconds to deletion	The number of seconds remaining before the session is automatically deleted.

Related Commands

delete firewall session
show firewall event
show firewall policy