

Chapter 46

Firewall

| | |
|---|-------|
| Introduction | 46-4 |
| Firewall Technologies | 46-4 |
| Policies | 46-6 |
| Rules | 46-7 |
| Using Limit Rules to Limit Sessions | 46-12 |
| Access Lists | 46-13 |
| List Files | 46-13 |
| RADIUS Servers | 46-14 |
| Dynamic Interfaces | 46-16 |
| Dynamic Interface Templates | 46-16 |
| Configuring Dynamic Interfaces | 46-17 |
| Disabling the TCP Setup Proxy | 46-18 |
| Firewall UDP Session Timeouts | 46-18 |
| Multicast Packet Handling | 46-19 |
| Enhanced Packet Fragment Handling | 46-19 |
| Enabling the Secure Shell Server | 46-20 |
| Network Address Translation (NAT) | 46-20 |
| Enhanced NAT | 46-23 |
| Network Address and Port Translation (NAPT) | 46-24 |
| Enhanced Network Address and Port Translation (ENAPT) | 46-25 |
| Standard NAT | 46-27 |
| Other Advanced NAT Types | 46-28 |
| NAT on Ethernet Interfaces | 46-29 |
| FTP Session Handling | 46-30 |
| PPTP Pass Through | 46-31 |
| Universal Plug and Play (UPnP) | 46-32 |
| SMTP Proxy | 46-33 |
| Interaction Between the SMTP Proxy and Firewall Rules | 46-33 |
| Protecting the Email System | 46-33 |
| Email Relaying | 46-34 |
| HTTP Proxy | 46-34 |
| Firewall HTTP Proxies and Firewall Policies | 46-34 |
| HTTP Filters | 46-34 |
| Logging | 46-35 |
| SIP Application Layer Gateway: VoIP Phone Calls | 46-36 |
| Using Automatic Client Management Mode | 46-37 |
| Using Manual Client Management Mode | 46-40 |
| Displaying and Debugging the SIP ALG | 46-41 |
| Attacks Recognised by the Firewall | 46-42 |

| | |
|---|--------|
| Monitoring Firewall Activity | 46-45 |
| Notifications | 46-45 |
| Debugging | 46-46 |
| Event Triggers | 46-46 |
| Logging | 46-47 |
| Session Monitoring | 46-50 |
| Accounting | 46-53 |
| Configuration Examples | 46-53 |
| Minimum Configuration for a Small Office | 46-54 |
| Firewall with an ISP-Assigned Internet Address | 46-54 |
| Firewall with a Single Global Internet Address | 46-55 |
| Allowing Access to a WWW Server | 46-55 |
| Firewall with TCP Setup Proxy Disabled for File Sharing | 46-56 |
| Configuring a Firewall to Allow VoIP Phone Calls | 46-57 |
| Troubleshooting | 46-59 |
| Traffic Flow and Network Address Translation (NAT) | 46-59 |
| Traffic Allowed or Denied by Firewall | 46-59 |
| Traffic Logging and Firewall Alert Messages | 46-61 |
| SIP ALG and VoIP Phone Calls | 46-62 |
| Command Reference | 46-63 |
| add firewall monitor | 46-63 |
| add firewall policy apprule | 46-66 |
| add firewall policy dynamic | 46-68 |
| add firewall policy httpfilter | 46-69 |
| add firewall policy interface | 46-72 |
| add firewall policy limitrule | 46-74 |
| add firewall policy list | 46-76 |
| add firewall policy nat | 46-77 |
| add firewall policy proxy | 46-80 |
| add firewall policy rule | 46-82 |
| add firewall policy spamsources | 46-96 |
| add firewall policy udpporttimeout | 46-97 |
| create firewall policy | 46-98 |
| create firewall policy dynamic | 46-99 |
| delete firewall monitor | 46-99 |
| delete firewall policy apprule | 46-100 |
| delete firewall policy dynamic | 46-101 |
| delete firewall policy httpfilter | 46-102 |
| delete firewall policy interface | 46-103 |
| delete firewall policy limitrule | 46-104 |
| delete firewall policy list | 46-105 |
| delete firewall policy nat | 46-106 |
| delete firewall policy proxy | 46-108 |
| delete firewall policy rule | 46-109 |
| delete firewall policy spamsources | 46-110 |
| delete firewall policy udpporttimeout | 46-111 |
| delete firewall session | 46-111 |
| destroy firewall policy | 46-112 |
| destroy firewall policy dynamic | 46-112 |
| disable firewall | 46-113 |
| disable firewall monitor | 46-113 |
| disable firewall notify | 46-114 |
| disable firewall policy | 46-115 |
| disable firewall policy debug | 46-117 |
| disable firewall policy httpcookies | 46-119 |
| disable firewall policy identproxy | 46-120 |
| disable firewall policy smtprelay | 46-120 |
| disable firewall policy tcpsetupproxy | 46-121 |
| disable firewall sessionreport | 46-122 |

| | |
|--|--------|
| disable firewall sipalg | 46-122 |
| enable firewall | 46-123 |
| enable firewall monitor | 46-123 |
| enable firewall notify | 46-124 |
| enable firewall policy | 46-125 |
| enable firewall policy debug | 46-127 |
| enable firewall policy httpcookies | 46-129 |
| enable firewall policy identproxy | 46-129 |
| enable firewall policy smtprelay | 46-130 |
| enable firewall policy tcpsetupproxy | 46-131 |
| enable firewall sessionreport | 46-132 |
| enable firewall sipalg | 46-132 |
| reset firewall policy maccache | 46-133 |
| reset firewall sipalg autoclients | 46-134 |
| reset firewall sipalg counter | 46-134 |
| set firewall maxfragment | 46-135 |
| set firewall monitor | 46-136 |
| set firewall policy | 46-138 |
| set firewall policy attack | 46-140 |
| set firewall policy limitrule | 46-142 |
| set firewall policy rule | 46-144 |
| set firewall policy smtpdomain | 46-149 |
| set firewall policy udpporttimeout | 46-150 |
| set firewall sipalg | 46-151 |
| show firewall | 46-153 |
| show firewall accounting | 46-156 |
| show firewall arp | 46-158 |
| show firewall event | 46-160 |
| show firewall monitor | 46-163 |
| show firewall policy | 46-164 |
| show firewall policy attack | 46-182 |
| show firewall policy dynamic | 46-183 |
| show firewall policy limitrule | 46-184 |
| show firewall policy list | 46-186 |
| show firewall policy maccache | 46-187 |
| show firewall policy user | 46-188 |
| show firewall policy udpporttimeout | 46-189 |
| show firewall session | 46-190 |
| show firewall sipalg | 46-195 |
| show firewall sipalg autoclients | 46-199 |
| show firewall sipalg counter | 46-202 |

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.

You may need special feature licences to use the firewall, to increase the maximum number of firewall sessions, and to use some advanced features. For information, contact your 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 and stateful inspection.

Application gateway

Application gateway is the traditional approach to building a firewall. 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

Another 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.
- The firewall manages TCP request queues in a way that provides protection against SYN Flood Attacks. SYN Flood Attacks send TCP requests faster than a machine can process them. Depending on the number of suspicious hosts detected, the firewall aggressively ages any or, if necessary, all unestablished TCP sessions, until the number of suspicious hosts decreases again to an acceptable level.
- 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.

Commands 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
```

Default policies 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 51, Server Load Balancing](#).

Commands To display the current status and configuration of a policy or all policies, use the command:

```
show firewall policy=name [counter] [rule=rule-id[-rule-id]]
[summary]
```

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. To enable or disable processing ICMP packets, IP packets with options set, and ping packets on a per-policy basis, use the commands:

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

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

For example, to prevent the router 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 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 define exceptions to the default settings of firewall policies. If you do not configure rules, the default behaviour is for the firewall to:

- deny all publicly-initiated traffic flows, so that no-one can access the private network from outside
- allow all privately-initiated traffic flows, so that internal users have unlimited access to the public network
- translate addresses according to the policy NAT definition, if configured

This section describes how rules can allow some public hosts to access the private network, deny some private access to the public network, and perform or alter NAT.

Allowing or denying traffic

Policy rules define precisely when and how traffic can flow through the firewall based on such factors as IP addresses, UDP/TCP port numbers, protocol, day of the week, and time of the day. Typically, rules specify the access to or from a particular device. Common uses include:

- stopping private PCs from accessing certain websites. For example, you can prevent people from browsing illegal material.

- allowing a public PC to access a private server. For example, if you have a mail server on your private network, you can add a rule to allow TCP traffic to port 25 (the SMTP port) on the mail server host.
- stopping a particular private PC from accessing the Internet, or limiting access to certain times of the day. For example, an office could prevent a subset of employees from accessing the Internet.

If you need to control access to or from many IP addresses, you can use access lists instead of creating many individual rules. Access lists store IP or MAC addresses in files in the router's file subsystem or on a RADIUS server. Each rule can check up to four lists and one server. For more information, see ["Access Lists" on page 46-13](#).

Altering NAT

Firewall rules can specify or alter Network Address Translation (NAT). Use rules if you need to:

- apply NAT to particular traffic flows instead of all flows (rule action of **nat**)
- apply a different NAT to particular flows (rule action of **nat**)
- override NAT for particular flows (rule action of **nonat**)

For more information about NAT and rule-based NAT, see ["Network Address Translation \(NAT\)" on page 46-20](#).

Note that actions of **nat** and **nonat** implicitly allow traffic to pass through the firewall—make the rules as restrictive as possible.

Commands

To add a rule to a policy, use the **add firewall policy rule** command. Rules are processed from the lowest number to the highest number. If rules both deny and allow an activity, the rule with the lowest number takes precedence.

The router can dynamically renumber firewall rules, so that you can easily insert a new rule between two consecutive ones. For example, you can insert a new rule 2 on a policy with rules numbered 1, 2, 3, 7. The new rule takes position 2 in the rule list, while the existing rule 2, and the rest of the rules with numbers greater than 2, are renumbered and shuffled down the rule list until a gap in the numbering scheme is found. The new rule list is numbered 1, 2, 3, 4, 7.

Note that the second instance of a particular rule number keeps that number, not the first instance. This means that if you add a sequence of rules where two rules have the same number, the first of these rules may become significantly lower on the list. For example, if a configuration script has these rule numbers in this sequence:

```
add firewall policy=policy-name rule=1
add firewall policy=policy-name rule=3
add firewall policy=policy-name rule=3
add firewall policy=policy-name rule=4
add firewall policy=policy-name rule=5
```

then the first instance of rule 3 is eventually renumbered until it becomes rule 6. This occurs because the second rule 3 becomes rule 3 and renumbers the first rule 3 to rule 4. Then the second rule 4 renumbers it to rule 5, and the second rule 5 renumbers it to rule 6. The new list of rule numbers is 1, 3, 4, 5, 6.

To delete a rule from a policy, use the **delete firewall policy rule** command.

To modify an existing rule, use the **set firewall policy rule** command.

To display currently configured rules for a policy, use the **show firewall policy** command.

Rule processing sequence

The firewall processes rules in order, from the lowest to the highest numbered rule. It checks the rule parameters against relevant fields in each flow until either it finds the first rule that matches a flow or it has processed all rules. The following table describes how the firewall decides whether to continue processing a rule or to move to the next rule.

| When... | Then the firewall... |
|---|---|
| a rule does not include a parameter | processes the rule's next parameter. |
| <ul style="list-style-type: none"> a rule includes a parameter and the flow does not match the parameter value | <ul style="list-style-type: none"> does not apply that rule's action to the flow, and stops processing the rule, and starts checking the next rule. <p>The only exception is when the rule uses an IP address or RADIUS server-based access list, in which case the firewall does not check any more rules. See IP Address Lists and RADIUS.</p> |
| <ul style="list-style-type: none"> a rule includes a parameter and the flow matches the parameter value | <ul style="list-style-type: none"> notes that so far the rule applies to the flow, and processes the rule's next parameter. |
| <ul style="list-style-type: none"> the firewall gets to the end of the rule and all parameter values match the flow | <ul style="list-style-type: none"> applies that rule's action to the flow, and does not check any remaining rules. |

Parameter processing sequence

The firewall checks rule parameters in the order shown in the following table, and compares them against settings in the first IP packet in the flow.

| Step | For this setting in the rule... | The firewall checks... |
|------|--|---|
| 1 | protocol parameter | Layer 3 protocol (UDP, TCP etc.) in the packet's IP header. |
| 2 | port or gblport parameter | destination TCP/ UDP port in the packet's IP header. |
| 3 | srcport parameter | source TCP/UDP port in the packet's IP header. |
| 4 | encapsulation parameter | whether the packet has come out of an IPSec tunnel. |
| 5 | <ul style="list-style-type: none"> remoteip parameter—a remote IP address or range gblremoteip parameter—a remote IP address. Used with some types of NAT rules to match on flows that originate in the public network | <ul style="list-style-type: none"> for flows originating in the private network, the destination IP address in the packet's IP header. for flows originating in the public network, the source IP address in the packet's IP header. <p>For more information, see “IP and port parameters in policy rules” on page 46-89.</p> |

| Step | For this setting in the rule... | The firewall checks... |
|------|--|---|
| 6 | <ul style="list-style-type: none"> • ip parameter—an IP address or range. Used to match on flows that originate in the private network. Used to match on flows that originate in the public network when NAT does not apply. • gblip parameter—an IP address or range. Used to match on flows that originate in the public network when NAT applies. | <ul style="list-style-type: none"> • for flows originating in the private network, the source IP address in the packet's IP header. • for flows originating in the public network, the destination IP address in the packet's IP header. <p>For more information, see “IP and port parameters in policy rules” on page 46-89.</p> |
| 7 | after, before, and days parameters | time and date. |
| 8 | list parameter, when: <ul style="list-style-type: none"> • the list is a list of MAC addresses, and • the rule applies to an Eth or VLAN interface <p>For possible outcomes, see MAC Address Lists.</p> | source MAC address of the packet. |
| 9 | list parameter, when either: <ul style="list-style-type: none"> • the list is a list of IP addresses. For possible outcomes, see IP Address Lists. • the parameter is set to radius. For possible outcomes, see RADIUS. | <ul style="list-style-type: none"> • destination IP address, for flows originating in the private network in the packet's IP header. • the source IP address, for flows originating in the public network in the packet's IP header. • source MAC address of the packet. |

MAC Address Lists

A rule can query up to four lists of MAC addresses for information about matching traffic. See [“List Files” on page 46-13](#) for information about MAC address lists. The following table shows how the firewall handles rules that use lists of MAC addresses.

| When the list... | And the rule action is... | Then the firewall... |
|------------------------------|---------------------------|--|
| contains the address | allow | allows the flow. |
| | deny | denies the flow. |
| does not contain the address | allow or deny | checks the next rule, unless the current rule also has list=macradius , in which case it queries the RADIUS server. See RADIUS for possible outcomes. |

IP Address Lists

A rule can query up to four lists of IP addresses for information about matching traffic. See [“List Files” on page 46-13](#) for information about IP lists. The following table shows how the firewall handles rules that use lists of IP addresses.

| When the list... | And the rule action is... | Then the firewall... |
|------------------------------|--|---|
| contains the address | allow | allows the flow. |
| | deny | denies the flow. |
| does not contain the address | allow | denies the flow. Such entries are blocked because they are exceptions to a list of allowed addresses. The firewall does not process any more rules. |
| | <ul style="list-style-type: none"> deny and the new flow is from the private network | allows the flow, which is the firewall default action for private traffic. The firewall does not process any more rules. |
| | <ul style="list-style-type: none"> deny and the new flow is from the public network | checks the next rule. |
| | | |

RADIUS

A rule can query a RADIUS server for information about matching traffic. As described in [“RADIUS Servers” on page 46-14](#), the RADIUS server can store either of the following lists:

- IP addresses
- MAC addresses

The following table shows how the firewall handles rules that query RADIUS servers.

| When the server... | And the rule action is... | Then the firewall... |
|--|--|---|
| returns a valid address | allow or deny | allows the flow. |
| returns 0.0.0.0, which specifically denies that entry | allow or deny | denies the flow. |
| <ul style="list-style-type: none"> rejects the request, which means it does not have a record for the address, or does not respond to the request, which could mean the server is down or blocked | allow | denies the flow. Such entries are blocked because they are exceptions to a list of allowed addresses. The firewall does not process any more rules. |
| <ul style="list-style-type: none"> rejects the request or does not respond to the request | <ul style="list-style-type: none"> deny and the new flow is from the private network | allows the flow, which is the firewall default action for private traffic. The firewall does not process any more rules. |
| <ul style="list-style-type: none"> rejects the request or does not respond to the request | <ul style="list-style-type: none"> deny and the new flow is from the public network | denies the flow, which is the firewall default action for public traffic. The firewall does not process any more rules. |

Using Limit Rules to Limit Sessions

To limit the number of concurrent sessions a device can initiate, you can use the **limitrule** firewall commands. Limit rules apply to firewall sessions initiated by a device on either side of the firewall, and are attached to policies.

Each time a device initiates a session through the firewall, the router checks all the limit rules for the applicable firewall policy. If a session exceeds the limit in a matching rule, then the router does not allow the new session to start. The device can only start the new session once it has ended one or more of the current sessions. If a session does not match any limit rules, then no limit is applied. Each policy can have up to 100 limit rules.

All matching existing session numbers are included when the router checks the limit rules and more than one limit rule can apply to a session. However, if the firewall finds any matching rule that denies the session, then the session is denied, regardless of the other rules.

To add a limit rule to a policy, use the command:

```
add firewall policy=policy-name limitrule=rule-id
    srciplimit=0..10000 [interface=interface]
    [gblremoteip=ipadd[-ipadd]] [ip=ipadd[-ipadd]]
```

The **ip** and **gblremoteip** parameters specify the IP address range of the private (**ip**) and public (**gblremoteip**) devices that you are limiting the sessions for. The limit is set with the **srciplimit** parameter, and is applied to each device separately. That is, if a rule limits devices to 20 sessions, then any device can initiate a maximum of 20 sessions regardless of the other devices' activity.

Each limit rule applies to sessions initiated from both sides of the firewall. For example, consider the command:

```
add firewall policy=policy-name limitrule=1 srciplimit=3
    [interface=interface] gblremoteip=125.4.10.1-125.4.10.12
    ip=101.20.20.1
```

In the above example:

- the private device (101.20.20.1) can initiate a maximum of three sessions to all devices within the IP range 125.4.10.1 to 125.4.10.12
- each public device within the specified range can initiate up to three sessions each to the private device.

To modify a limit rule, use the command:

```
set firewall policy=policy-name limitrule=rule-id
    [interface=interface] [gblremoteip=ipadd[-ipadd]]
    [ip=ipadd[-ipadd]] [srciplimit=0..10000]
```

These commands limit sessions only as they are created; new or modified limit rules do not end any sessions already established by a device.

To delete a limit rule, use the command:

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

To display the limit rules set for a policy, use the command:

```
show firewall policy=policy-name limitrule[=rule-id[-rule-id]]
    [detail]
```

To display debugging related to limit rules, use the command:

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

You can use SNMP to monitor firewall session details. See [“Monitoring Firewall Sessions”](#) on page 46-45 for more information.

Access Lists

Access lists are lists of IP or MAC addresses to which access is controlled by one or more policy rules. You can store access lists on:

- the router in [List Files](#)
- [RADIUS Servers](#)

List Files

You can keep addresses of allowed or blocked addresses in list files on the router. A list file is an ASCII text file with a .txt file extension that is stored on the router's file system and contains a list of addresses. These files are suited to small lists of addresses that remain relatively static. Two types of list files can be used—IP address lists and hardware MAC address lists.

IP address lists An IP list file contains a list of IP host and network addresses. The firewall checks this list against the destination IP address for outbound (private to public) traffic and the source IP address for inbound (public to private) traffic.

Each line in an IP address file can have any one of the following:

- a single IP address in dotted decimal notation
- a single IP address in dotted decimal notation, followed a space or tab and the name of the host
- a range of IP addresses in dotted decimal notation separated by a hyphen, optionally followed by a text name.

Lines can also contain comments, which start with the “#” character. An entire line can be a comment, or the comment can be at the end of a line.

For example, the file listip.txt could 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 example network
```

To add an IP address list file to a policy, use the command:

```
add firewall policy=policy-name list=list-name
file=list-filename type=ip
```

where *list-name* is an arbitrary name for the list, and *list-filename* is the name of the file that contains the list on the router.

MAC address lists A hardware address list file contains a list of MAC addresses. The firewall checks this list against the source MAC address of the incoming or outgoing frame.

Each line in a MAC address file can have either of the following:

- a single MAC address in standard hexadecimal notation
- a single MAC address in standard hexadecimal notation, followed a space or tab and the name of the host

Lines can also contain comments, which start with the “#” character. An entire line can be a comment, or the comment can be at the end of a line.

For example, the file listmac.txt could contain the following:

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

To add a MAC address list file to a policy, use the command:

```
add firewall policy=policy-name list=list-name
file=list-filename type=address
```

where *list-name* is an arbitrary name for the list, and *list-filename* is the name of the file that contains the list on the router.

Using lists To create a rule to provide access control to or from 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...]
```

To add multiple (up to four) lists to a single rule, repeat the command:

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

RADIUS Servers

In some situations, you want to control access to or from a large number of addresses. For example:

- Your organisation may want to allow general access to the Internet but to restrict access to specific websites.
- You may want to block access to objectionable websites. Lists of such websites are commercially available and tend to be updated regularly by the provider.

Applications like these may result in long, frequently-updated lists of addresses. A RADIUS server is an ideal place to store this kind of list. You can configure the firewall to use one or more RADIUS servers to perform checks on user access rights. To specify a RADIUS server, use the command:

```
add radius server=ipadd secret=secret [other-options...]
```

To remove a RADIUS server, use the command:

```
delete radius server=ipadd
```

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

```
show radius
```

See [Chapter 41, User Authentication](#) for a detailed description of the **add radius server**, **delete radius server**, and **show radius server** commands.

IP address authentication using RADIUS

To configure the firewall to send queries about IP addresses to a RADIUS server, create a rule by using 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...]
```

The router makes RADIUS requests in 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.

To make a RADIUS server entry that specifically denies access, use the format:

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

To make a RADIUS server entry that specifically allows access, use the format:

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

MAC address authentication using RADIUS

To configure the firewall to send queries about MAC addresses to a RADIUS server, create a rule by using 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=macradius [other-options...]
```

The firewall uses the following process to determine whether to allow or deny a new flow:

1. A MAC address attempts to send a packet. The firewall looks for the MAC address in the MAC address cache.
2. If the firewall finds the MAC address in the cache, it allows or denies the flow, as specified in the cache entry.
3. If the firewall does not find the MAC address in the cache, it initiates a RADIUS query. In the query, it converts the MAC address to a formatted string, which functions as a username.
4. The firewall creates a session entry to store the packet, but does not permit traffic associated with the MAC address entry to pass. If the response from the RADIUS server indicates that the MAC address is acceptable, the firewall forwards the stored packet and keeps the session open. Otherwise it drops the packet and deletes the entry. For more information about whether the MAC address is acceptable, see [“RADIUS” on page 46-11](#).
5. The firewall creates a cache entry to record whether flows from the MAC address are allowed or denied.

To view the MAC cache, use the command:

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

To reset the MAC cache, use the command:

```
reset firewall policy=policy-name maccache
```

To make a RADIUS server entry that specifically denies access, use the format:

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

To make a RADIUS server entry that specifically allows access, use the format:

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

where:

- *macadd* is the MAC address of a host
- *ipadd* is the IP address of the device with the specified MAC address, if known, or 1.1.1.1

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 15-63 of Chapter 15, 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:

- is dropped when traffic is routed out a private interface
- bypasses 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.

Disabling the TCP Setup 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
```



Caution 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.

Firewall UDP Session Timeouts

You can configure a specific amount of time, per firewall policy, for which the firewall maintains inactive UDP sessions. This amount of time is called the UDP timeout.

To add a UDP timeout to the firewall policy, use the command:

```
set firewall policy udptimeout
```

As well as the firewall UDP timeout, you can also configure a UDP port timeout value per server port. For configured UDP ports only, the UDP port timeout value overrides the general UDP timeout configured for the firewall.

A list of UDP ports can be specified for each firewall policy, and each port can have a different UDP timeout value.

To add a UDP timeout to a specific UDP port or group of ports, use the command:

```
add firewall policy udpporttimeout=port-number
```

To modify a UDP timeout for a specific UDP port or group of ports, use the command:

```
set firewall policy udpporttimeout=port-number
```

To delete a UDP timeout from a specific UDP port or group of ports, use the command:

```
delete firewall policy udpporttimeout=port-number
```

To view the UDP port timeout settings that are configured for a firewall policy, use the command:

```
show firewall policy udpporttimeout
```

Multicast Packet Handling

Multicast packets must be carefully handled by the firewall because it does not know on which interface the packet must 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.

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}[,...]]
```

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
```

Network Address Translation (NAT)

Network Address Translation (NAT) allows a single device to act as an agent between the (public) Internet and a local (private) network. When you use NAT, you assign private IP addresses to devices on the private side of the firewall. When those devices send traffic, the firewall translates the private addresses to one or more publicly-valid addresses before routing the traffic. When the firewall receives traffic that is destined for those devices, it translates the public address back to the appropriate private address.

NAT terminology This Software Reference uses the following terms when discussing NAT:

| Term | Definition |
|----------------------------|--|
| Global IP address | An IP address that is only used by one device worldwide. |
| Public IP address | Therefore it uniquely identifies that device. |
| Globally-unique IP address | Global IP address ranges are distributed by Regional Internet Registries and are on sold or leased out by ISPs. These addresses are in short supply, so are generally only used by devices that need to contact the public Internet. |
| Private IP address | An IP address that is not globally-unique. As shown in the next table, several IP address ranges are reserved for private LANs. These addresses only identify devices within their local LAN, not worldwide. You can use these addresses freely within your LAN, but you cannot use them to connect to the public Internet. |

Private addresses The following IP address ranges are reserved as private ranges for use behind NAT devices:

| Private address range | Mask |
|-----------------------------|------------------------------|
| 10.0.0.0–10.255.255.255 | 255.0.0.0 |
| 172.16.0.0–172.31.255.255 | 255.240.0.0 or 255.255.0.0 |
| 192.168.0.0–192.168.255.255 | 255.255.0.0 or 255.255.255.0 |

Advantages of NAT

By allowing you to use these private addresses while still accessing the internet, NAT offers two advantages:

- it increases network security by hiding the actual IP addresses of devices in your network from public view
- it reduces the number of global IP addresses you require. You can configure the NAT device to translate many private addresses to and from a single public address.

NAT methodologies

The firewall offers two approaches for configuring NAT:

- **Interface-based NAT**

This provides a simple address translation for traffic passing between a pair of interfaces, one private and one public. Interface-based NAT translates all traffic between the two interfaces.

You can configure the following NAT types as interface-based NAT:

- [Enhanced NAT](#)
- [Enhanced Network Address and Port Translation \(ENAPT\)](#)
- [Standard NAT](#)

- **Rule-based NAT**

Rule-based NAT configures the firewall to translate a subset of the packets that are received on a particular firewall interface. The firewall can determine which packets are translated, and how, on the basis of such values as source address, destination address, protocol type, and port number (TCP/UDP).

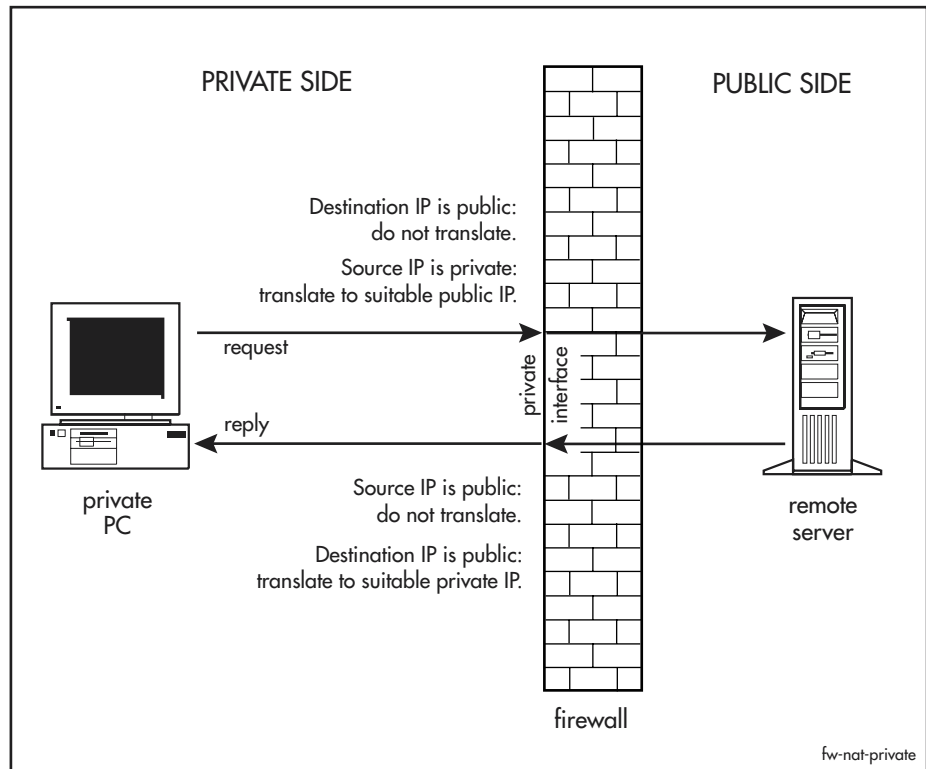
You can configure the following NAT types as rule-based NAT:

- [Enhanced NAT](#)
- [Network Address and Port Translation \(NAPT\)](#)
- [Enhanced Network Address and Port Translation \(ENAPT\)](#)
- [Standard NAT](#)
- [Other Advanced NAT Types](#)—Reverse NAT, Reverse Enhanced NAT, Double NAT, and Subnet translation.

NAT on private and public interfaces

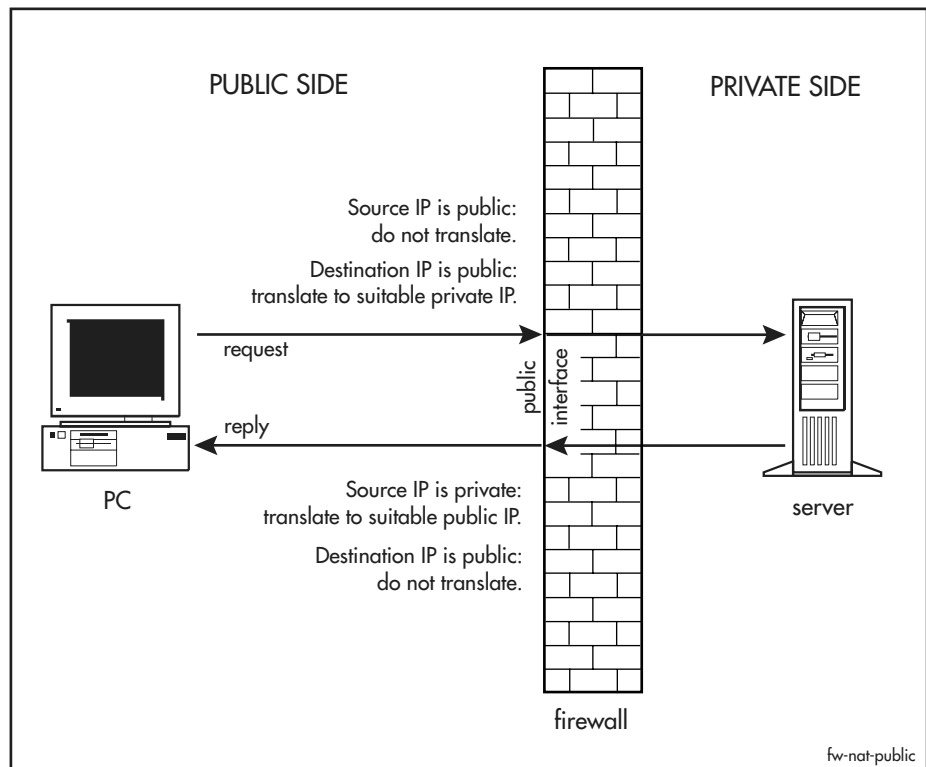
Depending on the translation you require, you can apply NAT on the private interface, the public interface, or both.

The private interface processes both outgoing traffic from sessions that initiate on the private side of the firewall, such as user web-browsing, and return traffic that is part of those sessions, such as web pages. The following figure shows this, and also shows an example of the kind of translation NAT may perform on the private interface.



Processing of sessions that initiate on the private side

The public interface processes both incoming traffic from sessions that initiate on the public side of the firewall, such as requests to a web server, and return traffic that is part of those sessions, such as the web pages. The following figure shows this, and also shows an example of the kind of translation NAT may perform on the public interface.



Processing of sessions that initiate on the public side

When NAT translates IP parameters of return traffic, it uses the reverse of the translation used for the initial traffic, as the previous two figures show. For example, if a user requests a web page and NAT translates the private IP address of the outgoing request to a global IP address, it will translate the global IP address of the incoming web page to the private IP address. The following table shows how this works, using the IP address translation parameters for NAPT, **ip** and **gblip** (“[Network Address and Port Translation \(NAPT\)](#)” on page 46-24).

| Interface | Traffic direction | Translation direction | Address translated | Example parameters | |
|-----------|--|-----------------------|--------------------|--------------------|-------|
| | | | | From | To |
| Private | Outgoing traffic | Private to public | Source | ip | gblip |
| | Incoming return traffic for sessions initiated on the private side | Public to private | Destination | gblip | ip |
| Public | Incoming traffic | Public to private | Destination | gblip | ip |
| | Outgoing return traffic for sessions initiated on the public side | Private to public | Source | ip | gblip |

Enhanced NAT

Enhanced NAT is the most common NAT implementation. It translates many private addresses into a single public address. The translation is performed on the private interface, on packets belonging to sessions that initiated inside the private LAN. Therefore, Enhanced NAT translates the source address of outbound packets and the destination address of replying inbound packets.

For example, when users on a LAN browse the Internet, Enhanced NAT makes all their web page requests appear to originate from the same public IP address. When web sites return the requested pages, they send them to this single public IP address. Enhanced NAT translates the public address to the correct private address, and forwards the pages.

Enhanced NAT uses TCP and UDP port numbers to track sessions. When the firewall receives a packet from a private device and opens a session, it translates the packet's source port to a randomly-chosen port number, and sends the packet to the destination device. The destination device replies by sending packets to that firewall-assigned port number. The firewall uses the incoming port number to determine which session the traffic belongs to, translates the port number back to the original source port, and forwards the packet appropriately.

Enhanced NAT allows an entire private network to access the internet through a single global internet address. It conserves IP addresses and improves security.

Configuring Enhanced NAT: interface-based

To add an interface-based Enhanced NAT to a policy, use the [add firewall policy nat](#) command:

```
add firewall policy=policy-name nat=enhanced
    interface=interface gblinterface=interface
    [gblip=ipadd[-ipadd]]
```

Enhanced NAT translates packets' private IP addresses to one of the following public addresses:

- the address specified by the **gblip** parameter, if you specify a single IP address
- the lowest address in the range of addresses specified by the **gblip** parameter, if you specify a range.
- the IP address of the public interface, if you do not specify **gblip**. This is useful if the address of the public interface is dynamically-assigned and therefore changes.

Configuring Enhanced NAT: rule-based

To add a rule-based Enhanced NAT to a policy, use the [add firewall policy rule](#) command:

```
add firewall policy=policy-name rule=rule-id action=nat
    nattype=enhanced interface=private-interface
    protocol={protocol|all|egp|gre|icmp|ospf|sa|tcp|udp}
    gblip=ipadd [ip=ipadd[-ipadd]]
    [other-options-to-match-packets]
```

For more information about which parameters are valid with Enhanced NAT rules, see [“IP and port parameters in policy rules”](#) on page 46-89.

Network Address and Port Translation (NAPT)

Network Address and Port Translation (NAPT) translates the IP address and TCP/UDP port of packets sent to and from devices on the private side. NAPT differs from Enhanced NAT by giving you control over the UDP or TCP port numbers that the firewall assigns to each user's sessions.

When to use NAPT

NAPT increases the reliability of VoIP phone calls through the SIP Application Layer Gateway (ALG) by avoiding changes to the UDP port number. The port number is important because public SIP proxy servers use it to locate users.

If you use Enhanced NAT instead of NAPT, the firewall randomly assigns a UDP port to each user's session and uses this port number to determine where to send incoming traffic. Once a session is established the firewall keeps it alive, so the port number is constant until—and only until—the session is closed. Sessions are closed, for example, if a user of a software phone logs off. When the user next logs on, the firewall gives the session a different UDP port number. The SIP proxy server only learns this port number when the user calls out, so cannot direct incoming phone calls to a user before the user has called out.

If you use NAPT, the firewall always gives the same UDP port number to each user. This unchanging port number ensures that the SIP proxy server can always connect to the user.

Like Enhanced NAT, NAPT also lets users on your LAN access the Internet when you have many private IP addresses on your LAN and one public IP address on the firewall.

Configuring NAPT To add an NAPT to a policy, use the `add firewall policy rule` command:

```
add firewall policy=name rule=id interface=interface
  action=nat nattype=napt protocol={udp|tcp}
  ip=private-ip-address gblip=public-ip-address
  gblport=public-port port=private-port [other-options-to-
  match-packets]
```

If you want to allow externally-initiated sessions, for example, so that the user can receive phone calls as well as make them, you need to create a rule to the public interface. For more information about which parameters are valid with NAPT rules, see [“IP and port parameters in policy rules”](#) on page 46-89.

Enhanced Network Address and Port Translation (ENAPT)

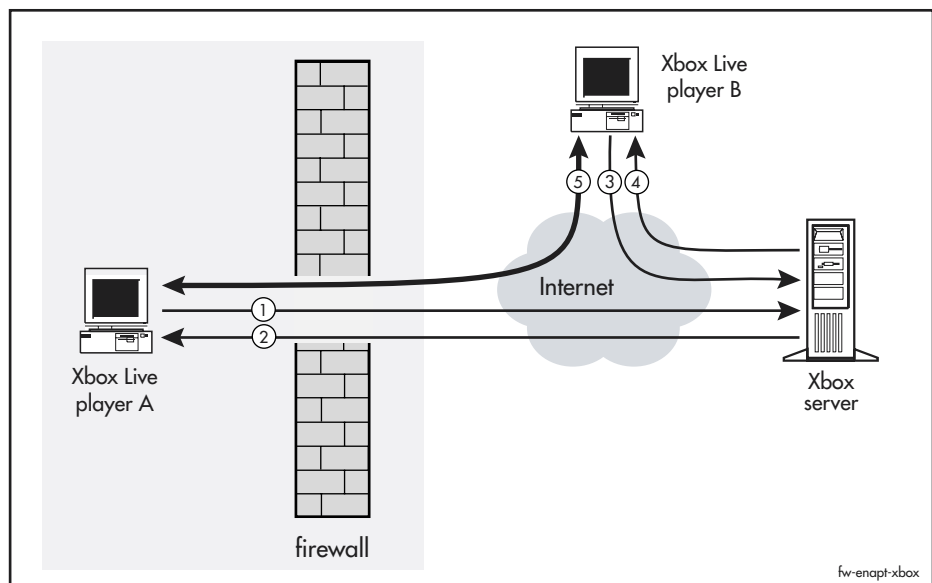
Enhanced Network Address and Port Translation (ENAPT) translates private IP addresses and ports to a public IP address and ports. It remembers the private to public mapping and applies the same mapping for all simultaneous sessions that involve the same private IP address and port.

ENAPT is a port restricted cone NAT, as defined in RFC 3489, *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*.

ENAPT combines:

- Enhanced NAT's ability to translate many private addresses to one public address
- NAPT's ability to avoid changes to the UDP or TCP port number

When to use ENAPT ENAPT enables the firewall to work with applications in which a private device may initiate sessions with multiple external servers or hosts. One such application is Xbox Live®, as shown in the following figure.



In the above figure, Xbox Live player A is behind the firewall which is performing ENAPT. Before playing, player A registers with the Xbox Live server (step 1) and the server replies (step 2). Likewise, player B registers with

the server (steps 3 and 4). When the players wish to start a game with each other, the server tells each Xbox the public IP address and port of the other Xbox, and they establish a session between them (step 5). Player A's Xbox must use the same public IP address and port when communicating with the server and with player B, or player B cannot connect to player A.

ENAPT deletes the private to public mapping when the last session that uses that mapping closes. This has no effect when using it with Xbox Live, because the first session is initiated by the private device, but makes ENAPT less suitable than NAPT for use with VoIP systems.

Creating an ENAPT: interface-based

To add an interface-based ENAPT to a policy, use the **nat=enapt** option in the **add firewall policy nat** command:

```
add firewall policy=policy-name nat=enapt interface=interface
gblip=interface [gblip=ipadd[-ipadd]]
```

ENAPT translates packets' private IP addresses to one of the following public addresses:

- the address specified by the **gblip** parameter, if you specify a single IP address
- the lowest address in the range of addresses specified by the **gblip** parameter, if you specify a range.
- the IP address of the public interface, if you do not specify **gblip**. This is useful if the address of the public interface is dynamically-assigned and therefore changes.

ENAPT also translates a private port (such as 3074 for Xbox gaming) to a public port. The firewall randomly allocates the public port and remembers the private to public mapping. If you want to apply ENAPT to a particular private port, create a rule-based ENAPT instead of an interface-based ENAPT. If you need to control the private and public port, create a rule-based NAPT instead of using ENAPT.

Creating an ENAPT: rule-based

To add a rule-based ENAPT to a policy, use the **nattype=enapt** option in the **add firewall policy rule** command:

```
add firewall policy=policy-name rule=rule-id action=nat
nattype=enapt interface=private-interface
protocol={protocol|all|egp|gre|icmp|ospf|sa|tcp|udp}
gblip=ipadd [ip=ipadd[-ipadd]] [port=port]
[sourceport=port]
[other-options-to-match-packets]
```

For more information about the IP address and port parameters that are valid with ENAPT rules, and the translations, see [“IP and port parameters in policy rules” on page 46-89](#).

You can create a rule that only applies to Xbox Live traffic by specifying the TCP/UDP port. All Xbox Live traffic has a source port of 3074. Traffic to the Xbox Live server also has a destination port of 3074, but the destination port of other Xboxes may vary. Therefore, to limit the rule to Xbox Live traffic, specify the source port by using **sourceport=3074**.

Increasing ICMP unreachable timeout

If you are configuring the firewall to allow Xbox Live sessions, also increase the ICMP unreachable message timeout. The timeout specifies the delay before the firewall deletes a session after it receives an ICMP unreachable message for that session. If you do not increase it, you may be unable to connect to remote Xboxes that are also behind a firewall. A suitable timeout is approximately 20 seconds. To set it, use the command:

```
set firewall policy=policy-name
    icmpunreachabletimeout=seconds [other-options]
```

Standard NAT

Standard NAT 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.

Standard NAT comes in two forms:

- Static, which is a one-to-one mapping of a single private IP address to a single global IP address. To use Static Standard NAT, you have to own a globally-unique IP address for each private address that you want to have access the Internet. Static Standard NAT is only a security feature—it hides your internal network structure—not an address-conservation feature.
- Dynamic, which maps more than one private IP address to a global IP address, but only one at a time. Different private devices can share your global IP address or addresses, but only one device can use each global address at a time.

Configuring Standard NAT: interface-based

To add an interface-based Standard NAT to a policy, use the [add firewall policy nat](#) command:

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

The **ip** parameter specifies the private address or addresses and the **gblip** parameter specifies the public address or addresses.

For the **ip** parameter:

- to configure Static Standard NAT, specify the IP address of the single private device that will access the Internet, by using **ip=ipadd**
- to configure Dynamic Standard NAT, when you want all devices on your LAN to be able to access the Internet, do not specify **ip**
- to configure Dynamic Standard NAT, when you only want some of the devices on your LAN to be able to access the Internet, specify the IP address range of those devices, by using **ip=ipadd-ipadd**

For the **gblip** parameter:

- when you configure Static Standard NAT, either use **gblip=ipadd** to specify your single global IP address, or do not specify **gblip**
- when you configure Dynamic Standard NAT, use **gblip=ipadd** to specify your single global IP address, use **gblip=ipadd-ipadd** to specify a range of global addresses if you own a range, or do not specify **gblip**

If you do not specify **gblip**, NAT uses the IP address of the public interface as the global IP Internet address. This is useful if the address of the public interface is dynamically-assigned and therefore changes.

Configuring Standard NAT: rule-based

To add a rule-based Standard NAT to a policy, use the command:

```
add firewall policy=policy-name rule=rule-id action=nat
    nattype=standard interface=interface
    protocol={protocol|all|egp|gre|icmp|ospf|sa|tcp|udp}
    [gblip=ipadd] [ip=ipadd[-ipadd]]
    [other-options-to-match-packets]
```

For more information about which parameters are valid with Standard NAT rules, see [“IP and port parameters in policy rules”](#) on page 46-89.

Other Advanced NAT Types

The firewall also offers the following NAT options:

- **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).
- **Reverse Enhanced NAT**
This is Enhanced NAT when applied to a public interface. It translates source address of inbound packets and destination address of outbound packets. This allows many public devices to share the same private IP address when accessing private devices.
- **Double NAT**
This translates both the public and private side source and destination addresses.
- **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.

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 traffic redirection.

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 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
```

NAT on 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]
```

FTP Session Handling

About FTP File Transfer Protocol (FTP) allows users to transfer files between two devices on a network. One device acts as an FTP server and accepts incoming FTP sessions, while the other device acts as an FTP client by initiating an FTP session.

FTP sessions use two different channel types: control channels, and data channels. An FTP control channel carries negotiation messages between the two devices. Each session only uses one control channel. An FTP data channel carries the data being transferred. Data channels are dynamically created and destroyed during an FTP session, and more than one data channel can occur per session. Both the control and data channels use TCP sessions.

Each FTP session operates in one of two modes: active and passive. The mode determines which device sets details about the data channel. In active mode, the client sets the IP address and TCP port number that each data channel is on. The server then initiates the TCP session on which the data is carried. In passive mode, the server sets the IP address and TCP port number, while the client initiates the TCP session.

Firewall support **The FTP Application Layer Gateway**

The firewall has an FTP Application Layer Gateway (ALG) to support the passage of FTP traffic between the private and public networks. The FTP ALG is automatically applied to any traffic using TCP port 21, the default port for FTP control channels. The firewall then applies the FTP ALG to any data channels negotiated over the control channel. To apply the FTP ALG onto a different port from port 21, you can use the command:

```
add firewall policy=policy-name apprule=rule-id action=allow  
interface=interface application=ftp port=port
```

FTP Data Channels

The firewall needs to know when FTP peers are creating or destroying a dynamic data channel, so that it can create or destroy a corresponding firewall session. To do this, it inspects the messages sent between the devices over the control channel. When the firewall sees a message that requests a new data channel, it takes the details from that message and creates a session for the data channel. However, it has to wait until the data session begins before it knows the source port details.

The FTP session mode that the devices use affects which source port is chosen for the data channel. The default port for data channels is 20. In passive mode, the client sets the source port, and can select a non-default port number. In active mode, the server sets the source port. RFC 959 states that FTP servers must set the port to 20, however, some FTP servers do not comply with the RFC. You can choose whether or not the firewall accepts data channels with a non-compliant source port set by the server, by using the **ftpdataport** parameter in the command:

```
set firewall policy=policy-name [ftpdataport={rfc|any}]  
[other-options]
```

FTP and session timeouts

The control channel is normally inactive while the devices are transferring the data. When the data transfer is large, the control channel may be idle for a long period of time. If the control channel is idle for longer than the timeout limit specified for TCP sessions, the firewall disconnects the control channel. This could cause the file transfer to fail. The default timeout limit for TCP channels on the firewall is 60 minutes.

To change the length of the timeout limit for TCP sessions on a policy, use the **tcptimeout** parameter in the command:

```
set firewall policy=policy-name [tcptimeout=0..43200]
[other-options]
```

Blocking FTP traffic

By default, the firewall does not allow FTP sessions initiated from the public network. To allow FTP traffic in from the public network to an FTP server on the private network, you need to create a rule. In the following example, the router is using NAT on the public interface (eth1). The FTP server on the private network has the private IP address 192.168.1.2, however because of NAT, the FTP clients on the public network access the server by connecting to the global IP address 202.49.72.1:

```
add firewall policy=policy-name rule=rule-id
action=allow interface=eth1 ip=192.168.1.2 protocol=tcp
port=ftp gblip=202.49.72.1 gblport=ftp
```

The firewall allows FTP sessions initiated from the private network by default. To prevent FTP sessions initiated on the private network (vlan1) from accessing the public network, use the command:

```
add firewall policy=policy-name rule=rule-id
action=deny interface=vlan1 protocol=tcp port=ftp
```

PPTP Pass Through

Multiple Point-To-Point Tunnelling Protocol (PPTP) tunnels can be initiated from the private side of the firewall by default. This allows private users to terminate their PPTP tunnels on their respective corporate or private networks. From the public side of the firewall, PPTP tunnels are blocked by default.

PPTP is a tunnelled mechanism to transfer Point-to-Point Protocol (PPP) frames across an intermediate network. PPTP connection is a method to create secure connections across public networks, such as the Internet, for both remote access and router-to-router virtual private network (VPN) connections, utilising the authentication, encryption, and protocol configuration mechanisms of PPP.

PPTP uses a TCP connection for tunnel management and Generic Routing Encapsulation (GRE) is used to encapsulate PPP frames for tunnelled data. In this implementation of PPTP, GRE tunnels are created and closed automatically when PPTP is enabled and disabled. The payloads of the encapsulated PPP frames can be encrypted or compressed, or both encrypted and compressed.

The list of pre-defined service names you can specify when adding or modifying a firewall policy rule includes PPTP:

Table 46-1: PPTP pre-defined IP protocol service name

| Service Name | Port Number | Standard Protocol |
|--------------|-------------|-------------------|
| PPTP | 1723 | TCP |

When PPTP clients are on the private side of the firewall, to add a rule **denying** PPTP tunnel traffic when adding or modifying a firewall policy rule, use the commands:

```
add firewall policy=policy-name rule=rule-id action=deny
    interface=interface protocol=tcp port=pptp
    [other-parameters]

set firewall policy=policy-name rule=rule-id action=deny
    interface=interface protocol=tcp port=pptp
    [other-parameters]
```

When PPTP clients are on the public side of the firewall, to add a rule **allowing** PPTP tunnel traffic when adding or modifying a firewall policy rule, use the commands:

```
add firewall policy=policy-name rule=rule-id action=allow
    interface=interface ip=ipadd[-ipadd] protocol=tcp
    port=pptp gblip=ipadd gblport=pptp [other-parameters]

set firewall policy=policy-name rule=rule-id action=allow
    interface=interface ip=ipadd[-ipadd] protocol=tcp
    port=pptp gblip=ipadd gblport=pptp [other-parameters]
```

Universal Plug and Play (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 47, UPnP](#).

SMTP Proxy

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 to deny SMTP sessions through the proxy, for example, 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 22-68 of Chapter 22, 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 46-96](#).

- 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 destined only 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 22-68 of Chapter 22, Internet Protocol \(IP\)](#).

HTTP 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} [other-options]

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 60, 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 46-45](#)).

SIP Application Layer Gateway: VoIP Phone Calls

To use internet telephony (VoIP) or video conferencing and still have your LAN protected by a firewall, you need these firewall features:

- the Session Initiation Protocol Application Layer Gateway (SIP ALG)
- **Network Address and Port Translation (NAPT)**

About the SIP ALG

VoIP and other multimedia applications create sessions over the Internet between users, for example between two people speaking on telephones. SIP establishes, maintains and terminates these sessions. People making phone calls use phone numbers or email-like addresses to “call” other users, and SIP proxy servers resolve these names into IP address and UDP port. This enables the SIP proxy servers to forward voice traffic appropriately.

If users are “hidden” from the Internet behind a firewall, they cannot receive SIP messages and so cannot use internet telephony. The SIP ALG enables the firewall to pass SIP messages to users behind the firewall. The SIP ALG inspects SIP packets and converts their IP addresses, UDP port numbers and other information as required.

Once SIP has established a session, the actual voice data in the phone call is carried by Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP). The SIP ALG dynamically controls the opening and closing of logical ports in order to establish, maintain, and terminate the RTP/RTCP sessions negotiated by the SIP protocol. It also modifies the RTP/RTCP packet IP addresses and port numbers to allow voice traffic across the firewall.

When NAT is configured on the router, the SIP ALG translates the private IP addresses embedded in SIP packets into globally routable IP addresses before sending the packets out onto the public network. This includes changing the IP address part in the Call-ID field of the SIP packets. The device that initiated the SIP session creates the Call-ID field by combining a random number and the device’s IP address. Changing the IP address part in the Call-ID field provides security by not revealing the private IP addresses in your network through the Call-ID.

An example of a Call-ID field with a private address is:

```
1874680886@192.168.1.2
```

The router only translates the Call-ID when the device that initiated the SIP session is a device within its private network.

For more information about SIP, see [Chapter 29, Voice over IP \(VoIP\)](#).

The SIP ALG requires a feature licence, which may be enabled by default for your router. For more information, contact your authorised distributor or reseller.

SIP server configuration

When using the SIP ALG, activate the Record-Route feature on your SIP server.

When Record-Route is not activated, SIP packets are initially sent via the SIP server, but later may be sent directly between the VoIP clients. This splits the call's SIP control packets between two or more different firewall sessions, which makes it difficult for the SIP ALG to correctly maintain the state of the SIP session. This may prevent it from correctly terminating calls.

When Record-Route is activated, it instructs VoIP clients to send all of their SIP packets via the SIP Server. Therefore, the same firewall session handles all SIP packets involved in a VoIP phone call. This ensures that the ALG performs the correct address and port translations on all SIP packets, including those that terminate a call.

Router configuration

Before configuring the SIP ALG, you must first decide the type of client management mode that you want it to operate in, either manual or automatic. In manual mode, you must individually configure each SIP client's settings using firewall rules. In automatic mode, the SIP ALG manages the clients automatically. Which method you use impacts how the SIP ALG is configured, and how the firewall handles SIP sessions. See these sections for further information:

- [Using Automatic Client Management Mode](#)
- [Using Manual Client Management Mode](#)

Using Automatic Client Management Mode

Automatic client management mode allows the SIP ALG to dynamically manage SIP clients and reserve firewall sessions for registered SIP clients. The SIP ALG does this by monitoring the messages sent by private SIP clients to SIP Registrars, and creating sessions that match the registration details. The SIP ALG also provides NAT when this is configured on the firewall.

For a VoIP phone to send and receive calls, it must register on the wider network with a SIP Registrar. When a SIP client registers, the SIP Registrar sends a response back to the SIP client informing the client of the expiry time limit for the registration. The SIP ALG looks for these messages and records the expiry time. It then makes sure that the firewall session created is retained until the registration expires. This means that the client is reachable through the firewall with the registered IP address and port for the entire duration of the registration. This is different to normal firewall session behaviour, where sessions are timed out and deleted if no traffic is seen for a certain time period.

Once registered, a SIP client can send and receive calls through a SIP Proxy Server. Often the proxy server is on the same device as the SIP Registrar, and uses the same firewall session created for the SIP Registrar. However, SIP clients can send and receive calls from proxy servers that are independent from the SIP Registrar.

When a proxy server is initiating a call to a SIP client, it uses the client's IP address and port details listed with the SIP Registrar. If the proxy server is on a different device from the Registrar, and you have configured the SIP ALG client management to allow calls from unknown proxy servers, then the SIP ALG creates a new firewall session for the proxy server. This new session uses the same global IP and port translation for the client that the firewall has assigned for the Registrar session. When the client initiates a session with any independent proxy server, the SIP ALG can also assign to the new session the same global IP and port that the Registrar session has. This gives the client a consistent identity on the public network.

Note that a private device may use the same global IP address and port number to send registration messages for more than one SIP URI. If this occurs, then the SIP ALG keeps the session open until all the registrations have expired.

Network address translation

In automatic mode, the SIP ALG uses NAT on the sessions when NAT has been configured on the firewall. We recommend that you select enhanced NAT. In automatic mode, the SIP ALG is designed to give each SIP client a consistent identity on the public network when NAT is in use.

It is possible to use the SIP ALG without NAT. This is an option for networks where the SIP clients have globally routable IP addresses, or the whole SIP network is restricted to a privately addressed network.

Configuring the SIP ALG in automatic mode

This section describes how to configure the firewall so that VoIP calls are managed using the SIP ALG in automatic mode. This includes configuring enhanced NAT on the firewall policy.

Before you start

This section describes the IP and firewall configuration. You also need to:

- configure the underlying connection to the Internet, such as PPP or ADSL
- create a security officer and enable system security, if required

Procedure

| Step | Action | Commands |
|------|--|--|
| 1 | Configure IP on the public and private interfaces: <ul style="list-style-type: none"> • assign IP addresses • create a default route on the public interface, if required | add ip interface = <i>interface</i> <i>ipaddress=ipadd</i> <i>[other-ip-parameters]</i> add ip route =0.0.0.0 mask=0.0.0.0 <i>interface=public-interface</i> <i>nexthop=ipadd</i> |
| 2 | Enable IP. | enable ip |
| 3 | Enable the SIP ALG. | enable firewall sipalg |
| 4 | Create a firewall policy. | create firewall policy = <i>name</i> <i>[other-policy-parameters]</i> |
| 5 | Use the policy on the router's public and private interfaces. | add firewall policy = <i>name</i> <i>interface=public-interface</i> <i>type=public</i> add firewall policy = <i>name</i> <i>interface=private-interface</i> <i>type=private</i> |
| 6 | Configure the NAT mode for the policy. | add firewall policy = <i>name</i> <i>nat=enhanced</i> |
| 7 | Configure the SIP ALG for the firewall: <ul style="list-style-type: none"> • assign the mode • specify the maximum number of automatic clients • specify how calls to and from Proxy Servers are dealt with | set firewall sipalg <i>mode=automatic</i> <i>[maxautoclients=1..1000]</i> <i>[multiservers={outonly off on yes}]</i> |
| 8 | Enable the firewall. | enable firewall |

Storing client information

In automatic mode, the SIP ALG stores the SIP client details in a client database. This database contains the registration expiry times as well as client information, and is stored both dynamically and statically. The dynamic version is stored on RAM, while a static copy is stored on flash. The static copy is designed to minimise any loss of service to SIP clients. If a router restart or reboot occurs, then the SIP ALG can immediately restore the firewall sessions using the information in this file.

To show details about the flash file and the current client sessions that the SIP ALG has, use the commands:

```
show firewall sipalg autoclients [=session-number] [summary]
show firewall sipalg autoclients ip=ipadd [-ipadd] [summary]
```

To delete the current details in the client database, use the command:

```
reset firewall sipalg autoclients
```

Resetting the database does not delete any established SIP sessions.

In a router where the flash autowrite feature is disabled, the SIP ALG cannot create a static copy of the client database. To enable this feature, use the command:

```
enable flash autowrite
```

If you disable flash autowrite while the SIP ALG is in automatic mode, then the SIP ALG can no longer write to the client database, but the file remains on the flash. To delete the file, use the command:

```
delete file=fwsipmap.sip
```

See [Chapter 6, Managing the File System](#) for further information about the flash autowrite feature.

Setting a trigger

You can set a firewall trigger to run a script when the SIP ALG reaches the limit for the number of SIP clients it can support in automatic mode. To do this, use the command:

```
create trigger=trigger-id firewall=sipautomax mode=start
[other-options]
```

See [Chapter 59, Trigger Facility](#) for more information about this trigger.

Using Manual Client Management Mode

In manual client management mode, the SIP ALG provides the normal services for SIP calls without providing any additional client management service, such as creating a database to manage the clients. Instead, clients are managed through firewall rules.

Please note that when changing from automatic mode to manual mode, the firewall deletes the dynamic and static versions of the client database. However, established SIP sessions are not affected.

When configuring the SIP ALG with NAT, up to two firewall policy rules are required for each SIP client that exists behind the firewall:

- A rule on the private interface is used to ensure the client has consistent NAT translation.
- A rule on the public interface is used to ensure that incoming calls are mapped to the correct SIP client. This rule is only required in case the firewall reboots or restarts, losing the current firewall session details, or in case the UDP timeout configured is less than the registration expiry time. To change the UDP timeout, use the **set firewall policy** command.

It is possible to use the SIP ALG without NAT. This is an option for networks where the SIP clients have globally routable IP addresses, or the whole SIP network is restricted to a privately addressed network. In these cases, a simple allow rule is needed on the public interface for each SIP client specifying their IP address and port. It is not necessary to configure a rule on the private interface.

For more information on configuring firewall rules, see the section [“Rules” on page 46-7](#) in this chapter.

Configuring the SIP ALG in manual mode

This section describes how to configure the SIP ALG on the firewall when the SIP ALG is in manual client management mode. For a detailed configuration example, see [“Configuring a Firewall to Allow VoIP Phone Calls” on page 46-57](#).

Before you start This section describes the IP and firewall configuration. You also need to:

- configure the underlying connection to the Internet, such as PPP or ADSL
- create a security officer and enable system security, if required

| Procedure | Step | Action | Commands |
|-----------|------|--|--|
| | 1 | Configure IP on the public and private interfaces: <ul style="list-style-type: none"> • assign IP addresses • create a default route on the public interface, if required • enable IP | add ip interface =interface ipaddress=ipadd [other-ip-parameters] add ip route =0.0.0.0 mask=0.0.0.0 interface=public-interface nexthop=ipadd enable ip |
| | 2 | Enable the firewall. | enable firewall |
| | 3 | Set the client mode to manual, if you have previously set it to automatic. | set firewall sipalg mode=automatic |
| | 4 | Enable the SIP ALG. | enable firewall sipalg |

| Procedure (cont.) | Step Action | Commands |
|-------------------|---|--|
| | 5 Create a firewall policy. | <code>create firewall policy=name</code> <code>[other-policy-parameters]</code> |
| | 6 Use the policy on the router's public and private interfaces. | <code>add firewall policy=name interface=public-interface type=public</code> <code>add firewall policy=name interface=private-interface type=private</code> |
| | 7 With NAT: Create policy rules to use NAPT for: <ul style="list-style-type: none"> • each user in the LAN, on • both the public and the private interfaces NAPT translates between public and private IP address and UDP port. Without NAT: Create policy rules on the public interface to allow SIP traffic through the firewall | <code>add firewall policy=name rule=id</code> <code>interface=interface protocol=udp</code> <code>action=nat nattytype=napt</code> <code>ip=user-private-ip gblip=public-ip</code> <code>port=private-sip-port gblport=user-global-sip-port</code> <code>add firewall policy=name rule=id</code> <code>interface=interface protocol=udp</code> <code>action=allow ip=user-ip port=sip-port</code> |

Displaying and Debugging the SIP ALG

To see whether the SIP ALG is enabled or disabled, and details about the SIP sessions using the SIP ALG, use the commands:

```
show firewall sipalg
show firewall sipalg ip=ipadd[-ipadd]
show firewall sipalg callid[=call-id]
show firewall sipalg summary
```

To show the counters for SIP sessions using the SIP ALG, use the command:

```
show firewall sipalg counter
```

To reset the counters for SIP sessions using the SIP ALG, use the command:

```
reset firewall sipalg counter
```

To see detailed information about how the firewall is processing and modifying SIP messages, use the command:

```
enable firewall policy[=name] debug=sipalg
debugmode={all|trace|message|parsing|errorcode}
[ip=ipadd[-ipadd]]
```

For a description of each of the debugging options, see the [enable firewall policy debug](#) command.

Attacks Recognised by the Firewall

The firewall recognises a wide range of attacks, originating from both the public and private sides of the firewall. When an attack occurs, the firewall responds appropriately to protect both the private and public network and provide the least amount of disruption possible to the private network.

The firewall defaults to using strict threshold values to recognise attacks. You can change these threshold values if desired. See the [set firewall policy attack command on page 46-140](#) for configuration details.

The firewall generates a notification whenever it detects an attack, and where possible responds by discarding the packets and sessions that are part of the attack. You can further enhance the firewall's response to any type of attack by using the Trigger Facility. See the section ["Monitoring Firewall Activity" on page 46-45](#) for further details.

DoS attacks The aim of a Denial of Service (DoS) attack is to bring a network to a standstill and limit communication between the target network and the wider network (Internet). Attacks on a target device vary. Some attacks try to use up as much of the device's bandwidth as possible, while others attempt to disable a device by causing high CPU usage or by crashing it. You can set threshold values and notifications for the following specific types of Denial of Service attacks:

DoS flood

An event where an attacker continually sends high volumes of unwanted traffic to the network, tying up bandwidth. DoS floods are indicated by an excessively high number of new session requests from the same source IP address to only a small group of destination IP addresses. IP spoofing of the source address is also used to hide the source address in a DoS flood. The router recognises a DoS flood by monitoring the number of new sessions occurring from the same source IP addresses. Once the firewall determines that a DoS flood is occurring, it generates a notification. Any further response to the DoS flood should be handled by a network administrator.

SYN attacks

An event where an attacker sends multiple opening TCP SYN packets to exhaust a device's available sessions or memory. These attacks send TCP requests faster than devices normally process them, which blocks legitimate sessions from being created.

To protect the private network from SYN attacks, the firewall uses a TCP setup proxy during the setup phase of a TCP session. The session is only allowed through to the private network after it has been established on the public side. There is a limit on the number of unestablished sessions from any one host. Depending on the number of suspicious hosts detected, the firewall aggressively ages some or, if necessary, all unestablished TCP sessions, until the number of suspicious hosts decreases again to an acceptable level.

Land attack

An event where an attacker sends IP packets with the same address in the source and destination address fields. This causes heavy CPU usage in the target device, and may cause it to crash. This is because the target device is trying to process an IP packet which has its own IP address as the source. The firewall stops land attacks by discarding packets from the public network where the source address is the same as the destination address. When NAT is in use, the firewall compares the source address against the destination address of the target device on the private network, not the translated NAT address.

Smurf

An event where the attacker uses the target device's IP address in ICMP echo requests with a broadcast destination address, sent to multiple networks. As each device on each network responds, a flood of responses are sent to the target device, consuming bandwidth. The firewall prevents smurf attacks by discarding ICMP messages with broadcast destination addresses.

Smurf Amp

An event similar to a smurf attack. During a smurf amp attack, the attacker sends TCP SYN packets with a broadcast destination address to multiple networks. As each device on each network responds, a flood of SYN responses are sent to the target device, consuming bandwidth. The firewall prevents smurf amp attacks by discarding TCP SYN messages with broadcast destination addresses.

Fragment attacks

An event where the attacker sends IP fragments that are either too large or can never be reassembled. These attacks aim to crash the target device. There are a variety of fragment attacks, including the teardrop attack, and overlapping fragment attack. The firewall protects the network from these attacks by attempting to reassemble fragmented packets before forwarding them on. If the fragmentation is suspicious, the firewall drops the packet.

Ping of death

An event where the attacker sends ping packets with illegal sizes, which can crash the target device, or sends an excessive number of ICMP messages that consume bandwidth. The firewall protects against ping of death attacks by discarding ICMP echo requests that are greater than 3000 bytes, or have suspicious fragmentation.

SMTP attacks

The aim of SMTP attacks is to misuse SMTP (email) services on your network. The firewall uses an SMTP proxy to protect the network from SMTP attacks. You can set threshold values and notifications for the following specific types of SMTP attacks:

SMTP relay

An event where the attacker attempts to get the private SMTP server to relay email to a non-local user. See [“SMTP Proxy” on page 46-33](#) for further details.

SPAM

An attempt to deliver email messages which are unsolicited to devices on the network. See [“SMTP Proxy” on page 46-33](#) for further details.

Penetration attacks

The aim of penetration attacks is to gain control or access to your network. These types of attacks either gather information about the network, or directly attempt to access devices. Both types threaten the security of individual devices on your network. You can set threshold values and notifications for the following specific types of penetration attacks:

Port scan

An event where an attacker checks for ports vulnerable to attack on the firewall or the private network. Ports in either the open or closed state can be vulnerable, depending on what application uses the port, and what operating system the target uses. Once an attacker finds a vulnerable port, they can use other attacks to compromise the target device, such as creating a buffer overflow. The firewall recognises port scans by monitoring how many ports on a target host a particular source IP address has sent packets to. Once the firewall detects a port scan, it drops any further packets from that source.

Host scan

An event where the attacker scans for hosts on a private network using a wide range of destination IP addresses. Host scans are indicated by a large number of sessions created by one source address to many devices on the network. For every new incoming or outgoing session that the firewall accepts, the source IP address is compared against an existing watch list of source addresses. If a source address has more sessions open than the threshold allows, the firewall considers traffic from that source as suspicious and generates a notification. Any further response should be handled by an administrator.

IP spoofing

An event where an attacker sends IP packets in which the addresses have been altered (spoofed). Attackers can spoof either the source IP address, or the destination IP address when attempting to penetrate a network. Packets sent with a changed source IP address, for example to the router's own IP address, are attempting to bypass filtering by appearing to be from a trusted device. Packets sent with a changed destination address are attempting to bypass a feature such as NAT. Attackers can attempt to bypass NAT and access a protected device by changing the destination IP address to the address of the device, which NAT is protecting. The firewall recognises both of these types of IP spoofing, and drops any packet with a spoofed IP address.

TCP tiny

An event where an attacker sends a target device very small TCP packets with fragmented headers. This attack attempts to circumvent a network device's filtering, by splitting up the TCP IP header across multiple packets. If it manages to do so, the attacker can then establish a TCP session, where previously a filter may have stopped them. The firewall stops TCP tiny attacks by reassembling the fragmented TCP packets before filtering them. Any packets sent to the router's own IP address with suspiciously fragmented TCP headers are dropped.

UDP attack

An event where an attacker sends UDP packets to probe for open UDP ports. A UDP attack is similar to a Portscan, with the difference that when a UDP packet doesn't receive a message back from a port, it assumes that the port is open. The firewall stops UDP attacks by monitoring the response to UDP packets sent to the network's devices. When a source of UDP packets is sent more ICMP unreachable messages than the firewall allows, the firewall determines that the source is attempting a UDP attack. Once the firewall detects a UDP attack, it drops any further UDP sessions from the source.

Other attack notifications

Some events may generate an "other" attack notification from the firewall. These are usually due to the firewall being misconfigured. These events include:

- packets received on a private interface that attempt to be forwarded to an interface that does not have a firewall policy
- packets received on a policy that attempt to be forwarded to an interface on a different policy
- packets received on interfaces that do not have a firewall policy that attempt to be forwarded to interfaces with firewall policies

Packets which fall within these category are dropped instead of being forwarded.

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 [“Emailing Alerts from the Router” on page 4-10 of Chapter 4, Configuring and Monitoring the System](#) 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 55, Simple Network Management Protocol \(SNMP\)](#) for information about configuring SNMP trap hosts.

You can set the event size (what constitutes a “significant event”) required to generate these notifications.

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

```
set firewall policy=policy-name
  attack={dosflood|fragment|hostscan|ipspooft|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 46-140](#) 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
```

Monitoring Firewall Sessions

You can use SNMP to monitor these session details:

- the total number of sessions through the firewall
- the number of current sessions that each private and public device has established through the firewall

To monitor the number of current sessions that individual devices are using, the firewall must generate a session report database. To enable the firewall to generate this database, use the command:

```
enable firewall sessionreport
```

Note that there is a resource cost for the router to maintain this database, so session reporting is disabled by default.

To disable session reporting, use the command:

```
enable firewall sessionreport
```

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.
- **SESSION**—The SESSION trigger activates when the first TCP session is created, and/or when the last active TCP session is closed. When all TCP sessions are closed, the router closes the link via which the TCP sessions were being transported. This avoids the cost of unused dial-up links.
- **SIPAUTOMAX**—The SIPAUTOMAX trigger activates with the SIP ALG reaches the limit for the number of SIP clients it can support in automatic mode.
- **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.

See [“Attacks Recognised by the Firewall” on page 46-42](#) for further descriptions of the attacks types and how the firewall responds to them.

You can configure the Trigger Facility to respond to these events by running management-defined scripts. Triggers can be activated by the start or end of an event. See [Chapter 59, 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|ipspooft|land|pingofdeath|portscan|smtpattack|smurfattack|synattack|tcptiny|udpatack} [intrigger=count] [outtrigger=count] [detail=count]
  [time=minutes]
```

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

```
show firewall policy attack
```

Logging

Table 46-2 describes events that the firewall can be configured to log to the router's Logging Facility.

Table 46-2: 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. |
| INAOOTHER | 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. |
| OUTAOOTHER | 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. |
| 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. |

Table 46-2: Log types and subtypes for firewall events (cont.)

| Option | Meaning |
|------------|--|
| 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. |
| SIPALG | Logs errors produced by the SIP application layer gateway. |

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|inddoother|inddtcp|inddudp|inddumpp|indeny|indicmp|indother|indtcp|indudp|outaicmp|outallow|outaother|outatcp|outaudp|outddicmp|outddother|outddtcp|outddudp|outddump|outdeny|outdicmp|outdoother|outdtcp|outdudp|sipalg}

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

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 60, Logging Facility](#)).

Further configuration is required to log firewall events whose log messages are assigned a severity of less than 3. [Table 46-3](#) lists log types that require additional configuration.

Table 46-3: Log types and subtypes requiring additional configuration

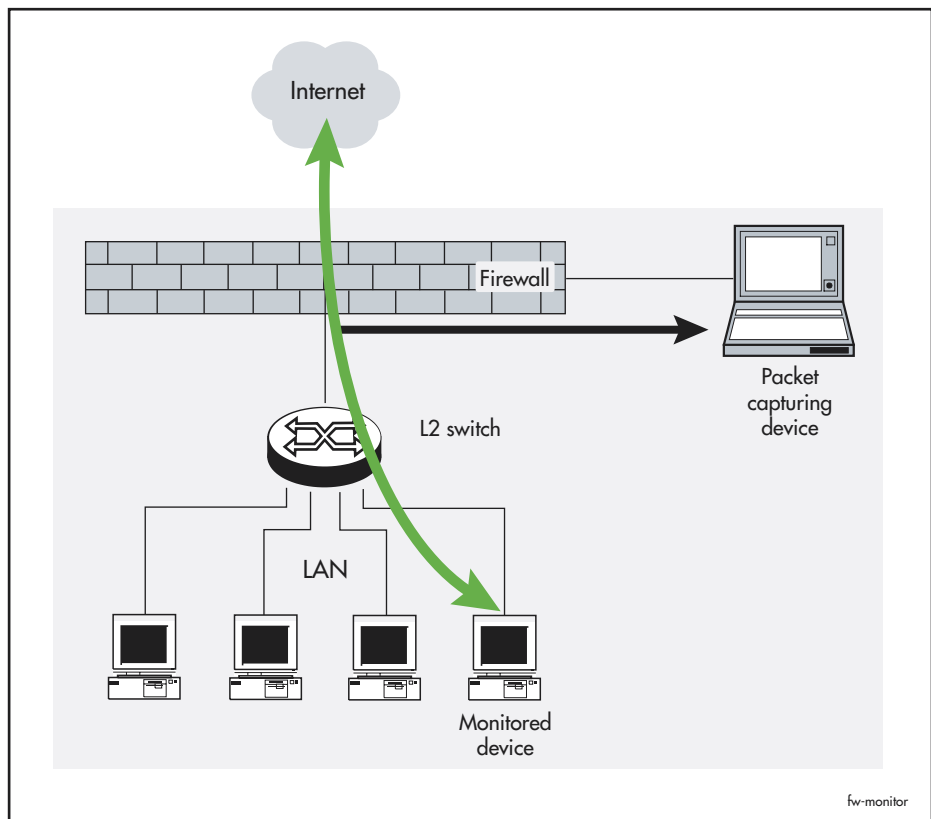
| Option | Additional Configuration |
|-----------|---|
| OUTATCP | ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTATCP |
| OUTAUDP | ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAUDP |
| OUTAICMP | ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAICMP |
| OUTAOTHER | ADD LOG OUTPUT=TEMPORARY TYPE=036 SUBTYPE=OUTAOTHER |
| 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 |

Session Monitoring

Firewall session monitoring enables the firewall to copy all traffic that goes to and from specified IP addresses and send the copies to a packet capturing device. You can choose whether to copy packets before or after the firewall has processed them, or both. Session monitoring is useful:

- as an advanced diagnostic tool to check firewall configuration. By capturing packets on both the public and private sides of the firewall, you can compare packets before and after firewall processing. This enables you to check the effect of features such as NAT.
- if you need to monitor the traffic that passes through the firewall to or from certain hosts.

The following figure demonstrates monitoring of traffic to and from a single host on a LAN.



Monitoring only copies packets that pass through the firewall. It does not copy packets that the firewall blocks.

Configuring Session Monitoring

Monitoring is disabled by default. To configure it, you need to set up a packet capturing device to collect the packet copies, create a monitor, and enable monitoring. The following table lists the commands to use on the router.

| Step | Command | Action |
|------|---|--|
| 1 | — | Connect a device to capture the copies, such as a PC running packet capturing software, to an Eth port or a switch port. |
| 2 | <pre>create vlan=vlan-name vid=vid add vlan=vlan-name port=port-number [other-options...] add ip interface={ethx vlanx} ip=ipadd [other-options...]</pre> | Configure the interface to which you connected the packet capturing device: <ul style="list-style-type: none"> • If you connected it to a switch port, put the port in a separate VLAN. • Give the Eth port or VLAN an IP address. |
| 3 | <pre>add firewall monitor=monitor-id ip=ipadd copyto=ip-interface [applyto={private public both}]</pre> | Create a monitor. Specify: <ul style="list-style-type: none"> • the IP address of the device you want to monitor • the interface to which you connected the capturing device, using the copyto parameter. • optionally, whether to monitor the private interface, the public interface, or both. The default is the private interface. |
| 4 | enable firewall monitor | Enable session monitoring. |
| 5 | show firewall monitor | Check the monitor configuration. |

Effect of deleting interfaces

If a monitor is configured to send duplicated packets to an interface (the **copyto** interface) and you delete that interface, then the firewall deactivates that monitor. If you add the interface again, the firewall automatically reactivates the monitor.

Effect on firewall throughput

The firewall's throughput is affected by on how much traffic it monitors at once. For example, if the firewall monitors all the traffic that passes through it at a given time, it processes packets approximately half as fast as if it monitors no traffic.

Multiple monitors

There is no limit on the number of devices you can monitor, although you should consider the performance impact of monitoring a high proportion of traffic.

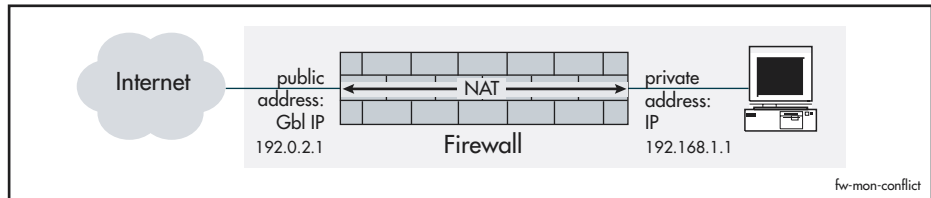
The firewall determines which monitor to use on traffic by checking the monitor's IP address against all IP address fields for the session. These session fields appear in the output of the **show firewall session** command, and are summarised in the following table.

| IP field name in session | Meaning |
|--------------------------|---|
| IP | 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 | 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. |

| IP field name in session (cont.) | Meaning (cont.) |
|----------------------------------|---|
| Gbl IP | 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. If NAT is not configured, this is the same as IP. |
| Gbl Remote IP | 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. If NAT is not configured, this is the same as Remote IP. |

Duplicate monitors

If two monitors monitor different addresses that are part of the same session, and both monitors apply on the same side of the firewall, then the firewall uses the last-created monitor. This avoids unnecessary packet duplication. For example, consider the scenario in the following diagram, in which NAT on the firewall translates between a private IP address (192.168.1.1, the IP entry in output from the **show firewall session** command) and a public IP address (192.0.2.1, the Gbl IP entry).



To monitor traffic in this scenario, you can apply a monitor to the private interface that specifies either the private address 192.168.1.1 or the public address 192.0.2.1. However, it is possible to create Monitor 1 that monitors the private address and then Monitor 2 that monitors the public address, by using the commands:

```
add firewall monitor=1 ip=192.168.1.1 copyto=vlan2
    applyto=private

add firewall monitor=2 ip=192.0.2.1 copyto=vlan3
    applyto=private
```

Both these monitors apply to sessions that match this scenario. The firewall uses Monitor 2, because it was the last monitor to be created. This means that copies of packets are sent to the **copyto** interface specified in Monitor 2, not the interface specified in Monitor 1.

If you delete the second monitor, the first monitor takes over. If the deleted monitor was monitoring a current session, monitoring may stop for a few seconds.

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 60-37 of Chapter 60, Logging Facility](#). This logging information can also be sent to a syslog daemon if required. See [Chapter 60, Logging Facility](#).

Configuration Examples

Examples in this section describe how to configure a firewall for the following:

- [Minimum Configuration for a Small Office](#)
- [Firewall with an ISP-Assigned Internet Address](#)
- [Firewall with a Single Global Internet Address](#)
- [Allowing Access to a WWW Server](#)
- [Firewall with TCP Setup Proxy Disabled for File Sharing](#)
- [Configuring a Firewall to Allow VoIP Phone Calls](#)

These configurations provide 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 shows how to configure a 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.

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.

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 15, Point-to-Point Protocol \(PPP\)](#).

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
```

Firewall with a Single Global Internet Address

This example is similar to the previous one 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.

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.

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
```

Firewall with TCP Setup Proxy Disabled for File Sharing

This example describes 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
```


Configuring a Firewall to Allow VoIP Phone Calls

This scenario shows you how to configure VoIP phone calls using the SIP ALG in manual mode. In this scenario (Figure 46-1):

- Three users need to receive and make phone calls through the firewall.
- The interface to the public Internet is eth0.
- The interface to the private LAN is vlan1. Each user is directly plugged into one of the LAN switch ports.

This example uses 10.10.10.10 instead of a globally-unique IP address on the firewall's public interface. Replace this address with a suitable global address for your network.

This example describes the configuration of the firewall to allow traffic to and from residential gateways and phones. You may also need to configure firewall rules for other devices in the LAN, such as servers and PCs.

Figure 46-1: Configuration to allow VoIP traffic through the firewall

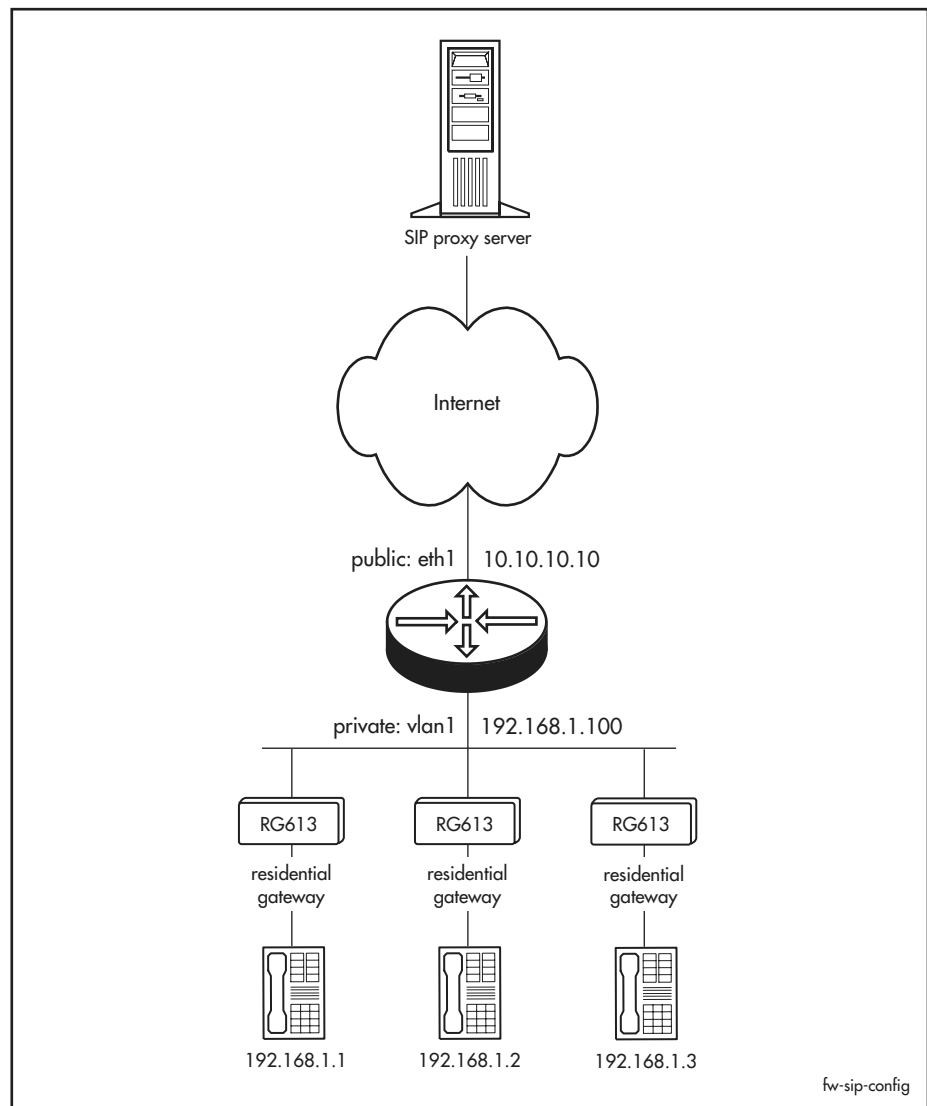


Figure 46-2: Configuration script to allow VoIP phone calls through the firewall

```
# Allowing VoIP phone calls through the firewall
# IP and firewall configuration

# Configure IP on eth1 public interface
# Note: Replace 10.10.10.10 in this example with your globally-unique IP address
enable ip
add ip interface=eth0 ip=10.10.10.10
add ip route=0.0.0.0 mask=0.0.0.0 interface=eth1 next=ip-address-of-your-isp

# Configure IP on vlan1 private interface
add ip interface=vlan1 ip=192.168.1.100 mask=255.255.255.0

# Enable the firewall and the SIP ALG
enable firewall
enable firewall sipalg

# Create a firewall policy and add the interfaces to it
create firewall policy=voip
add firewall policy=voip interface=eth0 type=public
add firewall policy=voip interface=vlan1 type=private

# Configure NAT by using firewall rules on public interface
# Note: Enter each command into the router on a single line
add firewall policy=voip rule=11 interface=eth0 protocol=udp action=nat nattype=napt
    ip=192.168.1.1 gblip=10.10.10.10 port=5060 gblport=61001
add firewall policy=voip rule=12 interface=eth0 protocol=udp action=nat nattype=napt
    ip=192.168.1.2 gblip=10.10.10.10 port=5060 gblport=61002
add firewall policy=voip rule=13 interface=eth0 protocol=udp action=nat nattype=napt
    ip=192.168.1.3 gblip=10.10.10.10 port=5060 gblport=61003

# Configure NAT by using firewall rules on private interface
# Note: Enter each command into the router on a single line
add firewall policy=voip rule=1 interface=vlan1 protocol=udp action=nat nattype=napt
    ip=192.168.1.1 gblip=10.10.10.10 port=5060 gblport=61001
add firewall policy=voip rule=2 interface=vlan1 protocol=udp action=nat nattype=napt
    ip=192.168.1.2 gblip=10.10.10.10 port=5060 gblport=61002
add firewall policy=voip rule=3 interface=vlan1 protocol=udp action=nat nattype=napt
    ip=192.168.1.3 gblip=10.10.10.10 port=5060 gblport=61003
```

Troubleshooting

Traffic Flow and Network Address Translation (NAT)

Problem A device on the LAN or DMZ can send some traffic out, but cannot receive traffic.

Solution If you are using a static Standard NAT, this problem may indicate that NAT is mapping to an invalid IP address. To use the GUI to check this, select Configuration > Firewall > NAT.

Problem Incoming traffic is sent to the wrong host.

Solution If you are using a static Standard NAT, this problem may indicate that NAT is mapping to a valid IP address, but which belongs to the wrong host. To use the GUI to correct the IP address, select Configuration > Firewall > NAT.

Problem Only one device on the LAN or DMZ can access the Internet.

Solution

- If you are using a static Standard NAT, only one device from the LAN can access the Internet. If you wish to have more than one device access the Internet, use Enhanced NAT instead. To use the GUI to select enhanced NAT, see the Configuration > Firewall > NAT page.
- It is also possible that no other device has been configured with the correct gateway.

Traffic Allowed or Denied by Firewall

Diagnosis To see information about the traffic that the firewall has denied, use the command:

```
show firewall event=deny
```

To see information about the traffic that the firewall has allowed, use the command:

```
show firewall event=allow
```

Problem Legitimate traffic is not reaching your LAN or DMZ.

Solutions

- Check that a rule exists to allow the traffic. To use the GUI to check this, select Firewall > Configuration > Traffic Rules.

Activating a DMZ does not provide access to servers on it. Rules must be created for each server on the DMZ. Likewise, by default there is no access to any devices on the private LAN.

- If the rule exists, it may be incorrect or insufficient. Check that:
 - Rules intended to allow traffic have an action of “Allow”.
 - The firewall is processing the rules in the order you expected, and that specific rules (e.g. allow IP address x access to FTP on the server) have lower numbers than general rules (e.g. deny all FTP access).
 - The ports, services and protocols are correct.
 - The IP addresses the rules apply to are entered correctly, and belong to the specified devices.
 - The rules apply to the correct days and time.
- Check the NAT configuration. See [“Traffic Flow and Network Address Translation \(NAT\)”](#) on page 46-59.

Problem [Illegitimate traffic is reaching your LAN or DMZ.](#)

- Solutions**
- The most likely cause of this problem is an incorrect rule. Check that:
 - “Allow” rules are tight enough that only the intended traffic types are allowed through.
 - The firewall is processing the rules in the order you expected, and that specific rules (e.g. deny IP address x access to FTP on the server) have lower numbers than general rules (e.g. allow all FTP access).
 - Rules intended to block traffic have an action of “Deny”.
 - The ports, services and protocols are correct.
 - The IP addresses the rules apply to are entered correctly, and actually belong to the specified devices.
 - The rules apply to the correct days and time.
 - Some traffic is allowed through the firewall so that protocols work correctly. You can specify on the Firewall Policy Options page of the GUI (Configuration > Firewall > Interfaces > Policy options tab) which ICMP traffic you want. For example, if **Ping** is checked on this page, ping packets addressed to the private LAN will be allowed.

Problem [A device on your LAN or DMZ cannot access the Internet.](#)

- Solutions**
- The most likely cause of this problem is an incorrect outgoing rule. Check that:
 - “Deny” rules are not too tight and therefore blocking more traffic than intended.
 - The firewall is processing the rules in the order you expected, and that specific rules (e.g. allow IP address x to use FTP) have lower numbers than general rules (e.g. deny all outgoing FTP requests).
 - Rules intended to allow traffic have an action of “Allow”.
 - The rules apply to the correct IP services (by name or port number).
 - The IP addresses the rules apply to are entered correctly, and actually belong to the specified devices.
 - The rules apply to the correct days and time.
 - Check that the device’s gateway address is correct.
 - Check the NAT configuration. See [“Traffic Flow and Network Address Translation \(NAT\)”](#) on page 46-59.

- If an IP address-based rule exists to allow traffic from this particular device, check that the device has a permanently-assigned IP address. If the router is assigning IP addresses as a DHCP server, you can give the required device a permanent IP address by making it a static entry. To use the GUI to do this, select Configuration > DHCP Server.

Problem A device on your LAN or DMZ can access a service on the Internet even though it should be blocked.

- Solutions**
- The most likely cause of this problem is an incorrect outgoing rule. Check that:
 - Rules intended to block traffic have an action of “Deny”.
 - The firewall is processing the rules in the order you expected, and that specific rules (e.g. block IP address x from using FTP) have lower numbers than general rules (e.g. allow all outgoing FTP requests).
 - The rules apply to the correct IP services (by name or port number).
 - The IP addresses the rules apply to are entered correctly, and actually belong to the specified devices.
 - The rules apply to the correct days and time.
 - If an IP address-based rule exists to block traffic from this particular device, check that the device has a permanently-assigned IP address. If the router is assigning IP addresses as a DHCP server, you can give the required device a permanent IP address by making it a static entry. To use the GUI to do this, select Configuration > DHCP Server.

Traffic Logging and Firewall Alert Messages

Problem Firewall Alert messages are not being emailed.

- Solution**
- Check that sending of firewall alerts is enabled. To enable it, use the command:

```
enable firewall notify
```

To use the GUI to do this, select Enable Email Firewall Alerts on the Configuration > Firewall > Events > Alarms tab.

- Check that alerts are being sent to the correct email address.
- Check that the router is set to use the correct DNS server and that the server’s IP address is correct. To use the GUI to do this, select Configuration > Internet Protocol > General.
- Check that a hostname is correctly specified. To use the GUI to do this, select Configuration > System > General.
- Make sure that the mail server has an account set up for the router.

Problem You are not receiving email notifications of all attacks that the firewall intercepts.

- Solution**
- Your alarm thresholds may be set too high. To use the GUI to check this, select Configuration > Firewall > Events > Alarms tab. Be careful when reducing the thresholds, because if the threshold is too low, your mail service may be flooded.

Problem You are receiving email notifications for “attacks” that actually are not attacks.

Solution Your alarm thresholds may be set too low. To use the GUI to check this, select Configuration > Firewall > Events > Alarms tab. Be careful when increasing the thresholds, because if the threshold is too high, you may not be warned about actual attack attempts.

Problem The time in log packets is incorrect.

Solution See “Time and Date” on page 4-4 of Chapter 4, Configuring and Monitoring the System.

SIP ALG and VoIP Phone Calls

Diagnosis To see information about SIP ALG sessions, for example, to which IP addresses a session applies, and the number of packets exchanged in a session, use the commands:

```
show firewall sipalg
show firewall sipalg counter
```

To see detailed information about how the SIP ALG processes SIP packets, use the command:

```
enable firewall policy debug=sipalg
```

For an example of debug output, see the How To note [How To Configure the Firewall VoIP Support Service \(SIP ALG\)](#). This How To note is also available in the Resource Center on your Documentation and Tools CD-ROM.

Problem Calls do not connect, or the SIP endpoint fails to register.

Solution Enable SIP ALG debugging and check the UDP port numbers that the ALG is applying to the SIP packets. The SIP proxy or the peer endpoint may not be configured to accept connections from the particular UDP port numbers that the SIP ALG is applying to the packets. If so, correct the proxy or peer endpoint.

Problem Calls establish themselves, but you cannot hear the voice at one or both ends of the call.

Solution Compare the UDP ports that the SIP ALG chooses and the UDP ports on the actual packets that are being exchanged. If they are different, this indicates an error in the configuration of the remote SIP end point. Correct that error, or contact the administrator of that VoIP network.

Problem Calls do not seem to terminate—when the person you are talking to hangs up, you do not hear a disconnect tone.

Solution

- To end that call, just hang up.
- To ensure that calls terminate correctly, make sure that the SIP server uses the Record-Route feature (see “SIP server configuration” on page 46-37).

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 22, 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 lxv of About this Software Reference](#) 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 monitor

Syntax `ADD FIREwall MOnitor=monitor-id IP=ipadd
COpyto=ip-interface [APPlyto={PRIVate|PUBLIC|BOTH}]`

where:

- *monitor-id* is an integer from 1 to 65535
- *ipadd* is an IPv4 address in dotted decimal notation
- *ip-interface* is a VLAN or Eth interface such as `vlan2` or `eth0`. The interface can be a logical interface such as `vlan2-1` or `eth0-1`

Description This command specifies an IP address for the firewall to monitor. The firewall makes a copy of every packet that comes from and goes to that address. It sends the copy over the Eth interface or VLAN that you specify.

There is no limit on the number of IP addresses you can monitor, although you should consider the speed impact of monitoring a high proportion of traffic.



Caution: If you create two or more monitors that monitor a given firewall session on the same firewall **applyto** interface, the firewall only uses the last-created monitor.

The **monitor** parameter specifies an identification number for the monitor.

The **ip** parameter specifies the IP address of the monitored device. The firewall monitors any firewall sessions that have this IP address in any of the session fields. These session fields display in output from the **show firewall session** command, and are summarised in the following table.

| IP field name in session | Meaning |
|--------------------------|---|
| IP | 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 | 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 | 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. If NAT is not configured, this is the same as IP. |
| Gbl Remote IP | 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. If NAT is not configured, this is the same as Remote IP. |

Therefore, sessions are monitored whether the device:

- sends the packets
- receives the packets
- initiates the session
- responds to a session initiated by another device

The **copyto** parameter specifies the Eth interface or VLAN to which the firewall sends the copies of monitored packets. Packets are sent as Layer 2 broadcasts to this interface. You should connect a device directly to this interface that can correctly capture the broadcast packets, such as a PC running packet capturing software. In particular, the device should not forward or reply to the packets. Duplicated packets use the router's MAC address as their source MAC address, and have a broadcast destination MAC address (ff:ff:ff:ff:ff:ff).

The **applyto** parameter specifies where the monitoring for this device applies. If you specify **private**, the firewall copies packets at the private interface. This is before firewall processing for outgoing packets and after firewall processing for incoming packets. If you specify **public**, the firewall copies packets at the public interface. This is before firewall processing for incoming packets and after firewall processing for outgoing packets. If you specify **both**, the firewall copies packets at both the public interface and the private interface. The default is **private**.

The combination of **ip** and **applyto** uniquely identifies a monitor. For example, you can create different monitors to monitor the same IP address on the private and the public interfaces.

Example To monitor traffic to and from the host whose IP address is 192.168.1.1, when the monitor is plugged into the port in vlan2, use the command:

```
add fire mo=1 ip=192.168.1.1 cop=vlan2
```

To monitor traffic to and from the host whose IP address is 192.168.1.1 so that you can check the firewall's NAT configuration, make a monitor by using the command:

```
add fire mo=1 ip=192.168.1.1 cop=vlan2 app=both
```

Use filtering within your packet capturing software to separate the private and public traffic. Alternatively, you can make two monitors by using the commands:

```
add fire mo=1 ip=192.168.1.1 cop=vlan2 app=priv
```

```
add fire mo=2 ip=192.168.1.1 cop=vlan3 app=pub
```

Using two monitors may make it easier to see which traffic came from the private interface and which came from the public interface.

Related Commands

- [delete firewall monitor](#)
- [disable firewall monitor](#)
- [enable firewall monitor](#)
- [set firewall monitor](#)
- [show firewall monitor](#)
- [show firewall session](#)

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 4294967295.
- *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 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.

The total number of **rules** (see [add firewall policy rule](#)) and **apprules** that a policy can associate with an interface is 2099. These policy rules are cumulative. That is, a policy cannot assign more than 2099 **rules** and **apprules** combined to an interface.

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 (such as eth0, eth0-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)
- X.25T (such as x25t0, x25t0-1)
- PPP (such as ppp0, ppp1-1)

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). The interface must already exist. To see a list of all currently available interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 46-10 on page 46-165](#), [Table 46-11 on page 46-170](#)).

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](#)
[add firewall policy rule](#)
[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 string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character.
- *template* is a 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 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 dynamic](#)

add firewall policy httpfilter

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

where:

- *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, 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.

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.

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 an 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.

The following figure contains an example of a URL filter file.

Figure 46-3: 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**.

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=YES|NO|True|False] [UPNptype={Lan|Wan}]`

where:

- *policy-name* is a 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 (such as eth0, eth0-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)
- X.25T (such as x25t0, x25t0-1)
- PPP (such as ppp0, ppp1-1)

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). To see a list of all currently available interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 upnp=w
```

Related Commands

- [create firewall policy](#)
- [create firewall policy dynamic](#)
- [delete firewall policy interface](#)
- [show firewall policy](#)

add firewall policy limitrule

Syntax `ADD FIREwall POLIcy=policy-name LIMIt rule=rule-id
[INTerface=interface] [IP=ipadd[-ipadd]]
[GBLRemoteip=ipadd[-ipadd]] [SRCIp limit=0..10000]`

Description This command adds a limit rule to a firewall policy. Limit rules apply a limit to the number of concurrent sessions that a device can initiate through the firewall. Each firewall policy can have up to 100 limit rules. The details for a session must match all values set for the **interface**, **ip**, and **gblremote** parameters for the limit rule to apply.

Each time a device initiates a session across the firewall, the router checks all the limit rules attached to a policy. If a session exceeds the limit in a matching rule, then the router does not allow the new session to start. The device can only start the new session once it has ended one or more of the current sessions.

This command only applies the limit as sessions are created; it does not end any sessions established by the device before this rule was added. However, all matching existing session numbers are included when the router checks the limit rules.

| Parameter | Description |
|-------------|---|
| POLIcy | The policy that the rule is added to. The <i>policy-name</i> is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character. The specified policy must already exist. |
| LIMIt rule | A numerical identifier for the rule for this policy. The <i>rule-id</i> is a decimal number from 1 to 4294967295. |
| INTerface | The interface that the rule is applied to. The interface must already exist and belong to the policy. Valid interfaces are: <ul style="list-style-type: none"> • eth (such as eth0, eth0-1) • VLAN (such as vlan1, vlan1-1) • FR (such as fr0, fr0-1) • X.25T (such as x25t0, x25t0-1) • PPP (such as ppp0, ppp1-1) Alternatively, this may be a dynamic interface, formed by concatenating the string "dyn-" with the name of a dynamic interface template (e.g. dyn-remote). See "Dynamic Interfaces" on page 46-16 . To see a list of all currently available interfaces, use the show interface command on page 9-72 of Chapter 9, Interfaces. Default: all interfaces attached to the policy |
| IP | IP address of the private device or range of devices you are limiting the sessions for. Devices must be on the private side of the firewall. The IP address is specified using dotted decimal notation. Default: all private devices |
| GBLRemoteip | IP address of the public device or range of devices you are limiting the sessions for. Devices must be on the public side of the firewall. The IP address is specified using dotted decimal notation. Default: all public devices |
| SRCIp limit | Number of sessions matching this rule that each device is allowed. Default: 0 (no limit set) |

Examples To limit all devices on the interface vlan2 to a maximum of 12 active sessions per device, using the policy named “AT_Field”, use the command:

```
add fire poli=AT_Field lim=1 int=vlan2 srci=12
```

Related Commands [delete firewall policy limitrule](#)
[set firewall policy limitrule](#)
[show firewall policy limitrule](#)

add firewall policy list

Syntax `ADD FIREwall POLIcy=policy-name LISt=list-name
File=filename TYpe={IP|Address}`

where:

- *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character.
- *list-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, 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 46-13](#) 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={ENAPt | ENHanced | STAndard}`
`GBLInterface=interface INTerface=interface`
`[GBLIP=ipadd[-ipadd]] [IP=ipadd[-ipadd]]`

Description This command adds a NAT translation to the specified policy.

If you want to allow access to a private server behind the firewall, you must also create firewall rules to allow the desired traffic. See [“IP and port parameters in policy rules” on page 46-89](#) for information about the IP address and port parameters to specify in such rules.

| Parameter | Description |
|-----------|--|
| POLIcy | An existing policy to add the NAT translation to. The <i>policy-name</i> is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character. Default: no default |
| NAT | The type of NAT that the firewall performs on traffic. Default: no default |
| ENAPt | Enhanced NAPT translates all private IP addresses to one global IP address, and also translates TCP or UDP ports. It remembers the private to public mapping and applies the same mapping for all simultaneous sessions that involve the same private IP address and port. You need ENAPT for applications like Xbox Live. For more information, see “Enhanced Network Address and Port Translation (ENAPT)” on page 46-25 . ENAPT translates the private IP address of traffic to the IP address of the public interface, or to the public address specified by the gblip parameter. |
| ENHanced | Enhanced NAT translates many private addresses into a single public address, and is the most common NAT implementation. The benefit of enhanced NAT is that it maps an entire private network to one public IP address. For more information, see “Enhanced NAT” on page 46-23 . Enhanced NAT translates the private IP address of traffic to the IP address of the public interface, or to the public address specified by the gblip parameter. As well as addresses, enhanced NAT also translates port numbers because it uses these to track sessions. You cannot control the port translation. If you need to do this, create firewall rules that use NAPT. |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| NAT (cont.) | <p>STANDARD Standard NAT translates one or more private addresses into one or more public addresses. With standard NAT, only one private host can use each public IP address at once. For more information, see “Standard NAT” on page 46-27.</p> <p>If you also specify the ip parameter, standard NAT translates the private IP address specified with ip into the public address specified with gblip.</p> <p>If you do not specify the ip parameter, standard NAT translates the private IP address of traffic to a public address, which you can specify with the gblip parameter. If you specify a range of public addresses, standard NAT chooses an address from the range and uses it for that session.</p> <p>If you do not specify gblip, standard NAT uses the IP address of the public interface.</p> |
| GBLInterface | <p>A public interface. The firewall translates all traffic it receives on this interface, then passes it to the private interface that you specify with the interface parameter.</p> <p>The interface must already exist and belong to the policy. Valid interfaces are:</p> <ul style="list-style-type: none"> • eth (such as eth0, eth0-1) • VLAN (such as vlan1, vlan1-1) • FR (such as fr0, fr0-1) • X.25T (such as x25t0, x25t0-1) • PPP (such as ppp0, ppp1-1) <p>To see a list of all currently available interfaces, use the show interface command on page 9-72 of Chapter 9, Interfaces.</p> <p>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.</p> <p>Default: no default</p> |
| INterface | <p>A private interface. The firewall translates all traffic it receives on this interface, then passes it to the public interface that you specify with the gblinterface parameter.</p> <p>The interface must already exist and belong to the policy. Valid interfaces are:</p> <ul style="list-style-type: none"> • eth (such as eth0, eth0-1) • VLAN (such as vlan1, vlan1-1) • FR (such as fr0, fr0-1) • X.25T (such as x25t0, x25t0-1) • PPP (such as ppp0, ppp1-1) <p>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). See “Dynamic Interfaces” on page 46-16.</p> <p>To see a list of all currently available interfaces, use the show interface command on page 9-72 of Chapter 9, Interfaces.</p> <p>Default: no default</p> |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| GBLIP | <p>The public IP address to which NAT translates the private address, in dotted decimal notation, or a range of public addresses.</p> <p>If nat is set to enhanced or enapt, then you just need to specify a single public IP address. You only need to specify a range of public addresses to allow sessions to be initiated from the public side to private hosts via multiple public addresses. For example, if you have two private servers offering the same service and each server corresponds to a different public IP address, you must specify a range that includes both public IP addresses. However, NAT only uses the first address of the range as a source address for packets in outgoing sessions. You need to specify all the public addresses so that you can configure rules to pass the traffic through to the correct private host.</p> <p>If nat is set to standard, then you need to specify at least one public address for each private host that you want to access the public network at the same time. Standard NAT chooses an address from this range. When the last simultaneous session closes for a particular private to public address mapping, then NAT returns that public IP address to the pool for reuse.</p> <p>Default: the IP address of the public interface. 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.</p> |
| IP | <p>The private IP address that NAT translates, in dotted decimal notation, or a range of private addresses.</p> <p>If you specify a single IP address, this is the private IP address used when a single public IP address is mapped to that private IP address.</p> <p>If you specify a range, this is the private IP addresses that can access the public Internet. See “Standard NAT” on page 46-27 for more detail.</p> <p>Only valid when nat is set to standard.</p> <p>Default: no default</p> |

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
```

To translate IP addresses and ports for all traffic between the private interface vlan2 and the public interface vlan3, which are attached to the policy named “xbox”, use the command:

```
add fire poli=xbox nat=enap int=vlan2 gblin=vlan3
```

Related Commands

- [add firewall policy rule](#)
- [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 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.

Important 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 46-4](#).

Table 46-4: 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 (such as eth0, eth0-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)
- X.25T (such as x25t0, x25t0-1)
- PPP (such as ppp0, ppp1-1)

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). The interface must already exist. To see a list of all currently available interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 gblin=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={ALL|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|MACRADIus}]`
`[NATType={DOuble|ENAPt|ENHanced|NAPt|REVerse|STAndard}]`
`[NATMask=ipadd] [PORT={ALL|port[-port] |service-name}]`
`[REMoteip=ipadd[-ipadd]] [SOurceport={ALL|port[-port] }]`
`[TTL=hh:mm]`

Description This command adds a rule to a firewall policy. Policy rules are filters that act on the packets processed by the specified policy on the specified public or private interface. Policy rules give you fine control over access and NAT. You can:

- block or allow packets on the basis of their IP address, protocol and port settings
- block or allow packets at specified times and on specified days
- apply NAT to selected packets
- prevent NAT from being applied to selected packets
- any combination of the above

The total number of **rules** and **apprules** (see [add firewall policy apprule](#)) that a policy can associate with an interface is 2099. These policy rules are cumulative. That is, a policy cannot assign more than 2099 **rules** and **apprules** combined to an interface.

By default, no rules exist and therefore:

- all access from public interfaces (outside the firewall) is denied and all access from private interfaces (inside the firewall) is allowed
- NAT settings are determined by using the [add firewall policy nat](#) command

The firewall processes each rule for a specific interface in a policy in numerical order, starting with the lowest numbered rule. It continues until it finds a match or reaches the highest numbered rule. Therefore, assign low numbers to specific rules and higher numbers to more general rules.

Firewall rules can determine whether to process ICMP packets, and can use IP address and other selectors to determine which packets to process.

Alternatively, you can use the [enable firewall policy](#) and [disable firewall policy](#) commands to choose whether the policy processes ICMP packets, IP packets that have specific options set, and ping requests.

| Parameter | Description |
|-----------|--|
| POLICY | An existing policy to add the rule to. Default: no default |
| RULE | A number from 1 to 4294967295 that is both an identifier for the rule and the rule position in this policy's list of rules. Rules are processed in order, from the lowest to the highest numbered rule. If you add a rule which has the same number as an existing rule, then the existing rule, and any higher numbered rules, are renumbered and shuffled down the rule list until a gap in the numbering scheme is found. Default: no default |
| ACTION | What the firewall does with traffic that matches the selectors defined for this rule. Default: no default |
| | ALLOW Matching traffic passes through the firewall. |
| | DENY Matching traffic is dropped by the firewall. |
| | NAT Matching traffic is NATed: the firewall translates between public and private addresses and sometimes ports. You must also specify the type of NAT, using nattype . See “Network Address Translation (NAT)” on page 46-20 . NAT rules take precedence over NAT relationships specified by using the add firewall policy nat command. Action=nat implicitly allows traffic to pass through the firewall; plan your rules carefully. |
| | NONAT Matching traffic is not NATed, even if you have configured the firewall to use NAT on this interface by using the add firewall policy nat command. Action=nonat implicitly allows traffic to pass through the firewall; plan your rules carefully. |
| INTERFACE | The interface on which the rule applies. The interface must already exist and belong to the policy. Valid interfaces are: <ul style="list-style-type: none"> • eth (such as eth0, eth0-1) • VLAN (such as vlan1, vlan1-1) • FR (such as fr0, fr0-1) • X.25T (such as x25t0, x25t0-1) • PPP (such as ppp0, ppp1-1) 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). See “Dynamic Interfaces” on page 46-16 . To see a list of all currently available interfaces, use the show interface command on page 9-72 of Chapter 9, Interfaces. Default: no default |
| PROTOCOL | The IP protocol number or the name of a predefined protocol type (“Predefined IP protocol service names” on page 46-93). The rule applies to packets from that protocol. If you specify tcp or udp , also specify a port to match. Default: no default |

| Parameter (cont.) | Description (cont.) | | | | | | | | | | | | | | | | | | | | |
|-------------------|--|-------|---|------|---|-----|----------|-----|------------|-----|-----------|-----|---------|-----|-----------|-----|---------|---------|--|---------|-------------------------------------|
| AFTer | <p>The time of day when the rule applies: from this time until either midnight or the time specified with before. Enter times in 24-hour format.</p> <p>Default: no default (the rule always applies)</p> | | | | | | | | | | | | | | | | | | | | |
| BEFore | <p>The time of day when the rule applies: from either midnight or the time specified with after, until this time. Enter times in 24-hour format.</p> <p>Default: no default (the rule always applies)</p> | | | | | | | | | | | | | | | | | | | | |
| DAYS | <p>The days on which the rule applies, in a comma-separated list.</p> <p>Default: all</p> <table> <tr> <td>ALL</td><td>All days of the week.</td></tr> <tr> <td>MON</td><td>Monday.</td></tr> <tr> <td>TUE</td><td>Tuesday.</td></tr> <tr> <td>WED</td><td>Wednesday.</td></tr> <tr> <td>THU</td><td>Thursday.</td></tr> <tr> <td>FRI</td><td>Friday.</td></tr> <tr> <td>SAT</td><td>Saturday.</td></tr> <tr> <td>SUN</td><td>Sunday.</td></tr> <tr> <td>WEEKDay</td><td>A shortcut for all days from Monday to Friday.</td></tr> <tr> <td>WEEKEnd</td><td>A shortcut for Saturday and Sunday.</td></tr> </table> | ALL | All days of the week. | MON | Monday. | TUE | Tuesday. | WED | Wednesday. | THU | Thursday. | FRI | Friday. | SAT | Saturday. | SUN | Sunday. | WEEKDay | A shortcut for all days from Monday to Friday. | WEEKEnd | A shortcut for Saturday and Sunday. |
| ALL | All days of the week. | | | | | | | | | | | | | | | | | | | | |
| MON | Monday. | | | | | | | | | | | | | | | | | | | | |
| TUE | Tuesday. | | | | | | | | | | | | | | | | | | | | |
| WED | Wednesday. | | | | | | | | | | | | | | | | | | | | |
| THU | Thursday. | | | | | | | | | | | | | | | | | | | | |
| FRI | Friday. | | | | | | | | | | | | | | | | | | | | |
| SAT | Saturday. | | | | | | | | | | | | | | | | | | | | |
| SUN | Sunday. | | | | | | | | | | | | | | | | | | | | |
| WEEKDay | A shortcut for all days from Monday to Friday. | | | | | | | | | | | | | | | | | | | | |
| WEEKEnd | A shortcut for Saturday and Sunday. | | | | | | | | | | | | | | | | | | | | |
| ENCapsulation | <p>The way in which traffic must be encapsulated, in order to match the rule.</p> <p>Default: none</p> <table> <tr> <td>IPSec</td><td>The rule applies to traffic that 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.</td></tr> <tr> <td>NONE</td><td>The rule does not match on encapsulation.</td></tr> </table> | IPSec | The rule applies to traffic that 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. | NONE | The rule does not match on encapsulation. | | | | | | | | | | | | | | | | |
| IPSec | The rule applies to traffic that 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. | | | | | | | | | | | | | | | | | | | | |
| NONE | The rule does not match on encapsulation. | | | | | | | | | | | | | | | | | | | | |
| GBLIP | <p>A public IP address. Only valid when the interface is using NAT or when action=nat.</p> <ul style="list-style-type: none"> On a public interface, the rule applies to incoming packets with this destination IP address. If action=nat, the firewall translates this address to ip. On a private interface, the firewall translates private IP addresses to this IP address in outgoing packets. <p>If nattype=enhanced and the interface is a public interface, this parameter is invalid. Use ip instead.</p> <p>If the interface uses NAT and the global IP address is not specified because it is dynamically assigned, you can match this IP address by gblip=0.0.0.0.</p> <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> | | | | | | | | | | | | | | | | | | | | |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| GBLPort | <p>A TCP or UDP port, a range of ports, or a predefined service name (such as FTP—see “Predefined IP protocol service names” on page 46-93).</p> <ul style="list-style-type: none"> On a public interface that uses NAT (as configured with the add firewall policy nat command), the rule applies to incoming packets that are destined for this port. If the interface uses enhanced NAT, the rule then translates the port to the value specified by port. On a private interface, if action=nat and nattype=napt, then the firewall translates private port numbers to this port number. On a public interface, if action=nat and nattype=napt, then the firewall translates this port number to the private port number specified by port. If action=nat and nattype is not napt, then this parameter is invalid. Use the sourceport or port parameters if you want the rule to match against ports. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |
| GBLRemoteip | <p>A public IP address or range of addresses. Only valid when action=nat and nattype=enhanced, double or reverse.</p> <ul style="list-style-type: none"> On a public interface, the rule applies to packets from this IP address. The firewall translates this address to remoteip. If nattype=enhanced, gblremoteip can be a single IP address or a range of addresses. If nattype=reverse or double, then gblremoteip is a single IP address. On a private interface when nattype=reverse or double, the firewall translates the destination address of packets to this address. Gblremoteip is not valid with enhanced NAT on a private interface. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |
| IP | <p>An IP address or range of addresses.</p> <p>On a private interface, the rule applies to packets that come from this IP address. If NAT is active on the interface or if action=nat, then the firewall translates this address to gblip.</p> <p>On a public interface:</p> <ul style="list-style-type: none"> If neither the interface nor the rule use NAT, or if nattype=reverse or enhanced, then the rule applies to packets sent to this IP address. If NAT is active on the interface, or if action=nat and nattype=standard, enhanced or double, then the firewall translates the destination address of incoming packets to this address. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| LIST | <p>A list of addresses. The rule applies to packets whose source or destination address is in this list.</p> <p>You can add up to four lists to a rule by repeating this command. You cannot specify both radius and macradius for a single rule.</p> <p>Default: no default</p> |
| <i>list-name</i> | The name of a predefined list of IP or MAC addresses. |
| RADius | <p>A RADIUS lookup is performed to check the source or destination address of the new flow. If the rule is applied to a public interface, the source address is checked. If the rule is applied to a private interface, the destination address is checked. The information received from the RADIUS server indicates whether the IP address is allowed or denied.</p> <p>You must also define the RADIUS server, using the add radius server command on page 41-30 of Chapter 41, User Authentication.</p> |
| MACRADius | <p>A RADIUS lookup is performed on the source Ethernet MAC address of the new flow. The information received from the RADIUS server indicates whether the MAC address is allowed or denied.</p> <p>You must also define the RADIUS server, using the add radius server command.</p> <p>Note that is only possible to do a RADIUS lookup on a packet's source MAC address when the packet is received on either an Ethernet or a VLAN interface. When macradius is specified for a rule that is not applied to an Ethernet or VLAN interface, no RADIUS lookup is performed.</p> |
| NATType | <p>The type of NAT that the firewall performs on traffic that matches the rule. For more information about each NAT type, see "Network Address Translation (NAT)" on page 46-20. For more information about which parameters are valid with each NAT type, see "IP and port parameters in policy rules" on page 46-89.</p> <p>Only valid when action=nat.</p> |
| DOuble | <p>On a private interface, this translates both:</p> <ul style="list-style-type: none"> the private side source address (the ip parameter) to a source address suitable for the public side of the firewall (the gblip parameter) the destination address (the remoteip parameter) to a destination address suitable for the public side of the firewall (the gblremoteip parameter) <p>On a public interface, this translates both:</p> <ul style="list-style-type: none"> the public side source address (the gblremoteip parameter) to a source address suitable for the private side of the firewall (the remoteip parameter) the destination address (the gblip parameter) to a destination address suitable for the private side of the firewall (the ip parameter) |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| NATType (cont.) | <p>ENAPT</p> <p>Enhanced Network Address and Port Translation, which translates private IP addresses and ports to a public IP address and ports. It remembers the private to public mapping and applies the same mapping for all simultaneous sessions that involve the same private IP address and port.</p> <p>On a private interface, this translates the private side source address to a source address suitable for the public side of the firewall (the gblip parameter). You can specify a specific source address or range of source addresses, using the ip parameter.</p> <p>Not valid on public interfaces.</p> |
| | <p>ENHanced</p> <p>On a private interface, this translates the private side source address to a source address suitable for the public side of the firewall (the gblip parameter). You can specify a specific source address or range of source addresses, using the ip parameter.</p> <p>On a public interface, this translates the public side source address to a source address suitable for the private side of the firewall (the remoteip parameter). You can specify a specific source address or range of source addresses, using the gblremoteip parameter.</p> <p>Enhanced NAT also translates port numbers because it uses these to track sessions. You cannot control the port translation. If you need to do this, use NATP.</p> |
| | <p>NApt</p> <p>Network Address and Port Translation, which translates the address and port of packets sent to and from a private side device to an address and port suitable for the public side of the firewall. Therefore, on a private interface it translates the source address and port, and on a public interface it translates the destination address and port. To specify:</p> <ul style="list-style-type: none"> • the private address, use the ip parameter • the public address, use the gblip parameter • the public port, use the gblport parameter <p>You do not normally need to specify the private port.</p> |
| | <p>REVerse</p> <p>This translates the address of a public side device (the gblremoteip parameter) to an address suitable for the private side of the firewall (the remoteip parameter). Therefore, on a private interface it translates the destination address, and on a public interface it translates the source address.</p> |
| | <p>STANDARD</p> <p>This translates the address of a private side device (the ip parameter) to an address suitable for the public side of the firewall (the gblip parameter). Therefore, on a private interface it translates the source address, and on a public interface it translates the destination address.</p> |

| Parameter (cont.) | Description (cont.) | | | | | | |
|---------------------|---|-----|---|--------------------|---|---------------------|---|
| NATMask | <p>An IP address mask that translates IP addresses from one subnet to another. For example, this would allow you to translate all 192.168.x.x addresses to 202.202.x.x addresses. Only valid when action=nat and nattype=double, reverse or standard.</p> <p>If nattype=double, the NAT mask is applied to the ip, gblip, remoteip and gblremoteip parameters.</p> <p>If nattype=reverse, the NAT mask is applied to both the remoteip and gblremoteip parameters.</p> <p>If nattype=standard, the NAT mask is applied to both the ip and gblip parameters.</p> <p>The parameters to which the NAT mask applies must specify a single IP address if the natmask parameter is used.</p> | | | | | | |
| Port | <p>A TCP or UDP port or a range of port numbers.</p> <p>On a private interface, the rule applies to packets that are destined for this port on the public side of the firewall.</p> <p>On a public interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets that are destined for this port on the public side of the firewall. • If the interface uses Enhanced NAT, then the firewall translates the destination port to this port number. If action=nat and nattype=napt, then gblport specifies the destination port, so the firewall translates gblport to port. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: all</p> <table> <tr> <td>ALL</td><td>Any port number. The rule does not use destination port number to select packets.</td></tr> <tr> <td><i>port[-port]</i></td><td>A single port number, from 0 to 65535, or a range of numbers.</td></tr> <tr> <td><i>service-name</i></td><td>A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93.</td></tr> </table> | ALL | Any port number. The rule does not use destination port number to select packets. | <i>port[-port]</i> | A single port number, from 0 to 65535, or a range of numbers. | <i>service-name</i> | A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93 . |
| ALL | Any port number. The rule does not use destination port number to select packets. | | | | | | |
| <i>port[-port]</i> | A single port number, from 0 to 65535, or a range of numbers. | | | | | | |
| <i>service-name</i> | A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93 . | | | | | | |
| REMoteip | <p>A public IP address or a range of public IP addresses.</p> <p>On a private interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets that are destined for this address. • If action=nat and nattype=reverse or double, then the firewall translates this address to gblremoteip. <p>On a public interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets from this address. • If action=nat and nattype=enhanced, reverse or double, then the firewall translates gblremoteip to this address. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> | | | | | | |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| SSourceport | <p>A TCP or UDP port, or a range of port numbers. The rule applies to packets that come from this port.</p> <p>On a private interface, if action=nat and nattype=enhanced or enapt, then the firewall translates this port to a global port.</p> <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: all</p> |
| ALL | Any port number. |
| port[-port] | A single port number, from 0 to 65535, or a range of numbers. |
| TTL | <p>The length of time for which the rule exists, in hours and minutes. The firewall starts applying the rule as soon as you create it and deletes it when the specified time expires. When the time expires, the firewall also destroys all entries that it created from this rule.</p> <p>Note that rules defined with a TTL value do not appear in router-generated configuration scripts because they are dynamic.</p> |

IP and port parameters in policy rules

The tables in the following sections show which IP address and port parameters you can use in rules, depending on the NAT configuration:

- [Without NAT](#)
- [With rule-based NAT](#)
- [With interface-based NAT](#)

Each table also indicates whether the rule matches source or destination IP address or port, for traffic over the indicated interface in the indicated direction. For example, when the private interface processes an outgoing packet for a session that the private side initiated, **ip** is the packet’s source address and **remoteip** is its destination address.

Without NAT

The following table shows the IP address and port parameters that you can use in rules when the firewall is not performing NAT (neither interface nor rule-based NAT). It indicates which parameters the rule can match against to select applicable packets.

| Interface | Type of address or port | Match |
|------------------------------|-------------------------|-------------------|
| Private: outgoing traffic | Source IP | ip |
| | Destination IP | remoteip |
| | Source port | sourceport |
| | Destination port | port |
| Public: incoming traffic | Source IP | remoteip |
| | Destination IP | ip |
| | Source port | sourceport |
| | Destination port | port |

With rule-based NAT

The following table shows the IP address and port parameters that you can use when you create a rule to apply NAT to matching traffic. It indicates which parameters the rule can match against to select packets, and which parameters specify translations.

Note that NAPT translates the packet's source port without you specifying it, so very few NAPT configurations require you to specify **sourceport**. However, if you have multiple sessions from one IP address to the same service, specifying source port may help you to distinguish them.

| Rule-based NAT type | Interface | Type of address or port | Match | Translate to |
|---|---------------------------|-------------------------|--------------------------------------|------------------------------|
| Enhanced NAT (nattype=enhanced) | Private: outgoing traffic | Source IP | ip | glbip (required) |
| | | Destination IP | remoteip | Not translated |
| | | Source port | sourceport | Translated; no user control |
| | | Destination port | port | Not translated |
| ENAPT (nattype=enapt) | Private: outgoing traffic | Source IP | ip | glbip (required) |
| | | Destination IP | remoteip | Not translated |
| | | Source port | sourceport | Translated; no user control |
| | | Destination port | port | Not translated |
| NAPT (nattype=napt) | Private: outgoing traffic | Source IP | ip (required) | glbip (required) |
| | | Destination IP | remoteip | Not translated |
| | | Source port | sourceport (rarely needed) | gblport (required) |
| | | Destination port | port | Not translated |
| | Public: incoming traffic | Source IP | remoteip | Not translated |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport (rarely needed) | Not translated |
| | | Destination port | gblport (required) | port (required) |

| Rule-based NAT type (cont.) | Interface | Type of address or port | Match | Translate to |
|--|---------------------------|-------------------------|----------------------------------|----------------------------------|
| Standard NAT (nattype=standard) | Private: outgoing traffic | Source IP | ip (required) | glbip (required) |
| | | Destination IP | remoteip | Not translated |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |
| | Public: incoming traffic | Source IP | remoteip | Not translated |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |
| Reverse Enhanced NAT (nattype=enhanced) | Public: incoming traffic | Source IP | glbremoteip | remoteip (required) |
| | | Destination IP | ip | Not translated |
| | | Source port | sourceport | Translated; no user control |
| | | Destination port | port | Not translated |
| Reverse NAT (nattype=reverse) | Private: outgoing traffic | Source IP | ip | Not translated |
| | | Destination IP | remoteip (required) | glbremoteip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |
| | Public: incoming traffic | Source IP | glbremoteip (required) | remoteip (required) |
| | | Destination IP | ip | Not translated |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |
| Double NAT (nattype=double) | Private: outgoing traffic | Source IP | ip (required) | glbip (required) |
| | | Destination IP | remoteip (required) | glbremoteip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |
| | Public: incoming traffic | Source IP | glbremoteip (required) | remoteip (required) |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | port | Not translated |

With interface-based NAT

The following table shows the IP address and port parameters that you can use when you create a rule on a policy that uses interface-based NAT. It indicates which parameters the rule can match against to select packets, and which parameters specify translations. In this situation, the rule specifies whether to allow or deny the traffic, and what the IP address and port are translated to. The NAT is defined by using the [add firewall policy nat](#) command, but the rule translations override the interface-based translations.

| Interface-based NAT type | Interface | Type of address or port | Match | Translate to |
|--------------------------|---|-------------------------|--|---------------------------|
| Enhanced NAT | Public: incoming traffic destined for private server etc. | Source IP | remoteip | Not translated |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | gblport (required) | port (required) |
| ENAPT | Public: incoming traffic destined for a private server etc. | Source IP | remoteip | Not translated |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | gblport (required) | port (required) |
| Standard NAT | Public: incoming traffic destined for private server etc. | Source IP | remoteip | Not translated |
| | | Destination IP | glbip (required) | ip (required) |
| | | Source port | sourceport | Not translated |
| | | Destination port | port (required if prot=tcp or udp) | Not translated |

**Predefined 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 |
| 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 |

Examples In this example, you want to allow WWW access to an internal server at IP address 202.36.163.12. The server is attached to a private interface via the public interface PPP0. The interface is attached to the firewall policy named "zone1". To configure this, use the command:

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

This example extends the configuration of the previous example, by only permitting external access during the company's business hours of 8 a.m. to 5 p.m. To configure this, use the command:

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

This example further extends the configuration of the previous example, by denying staff WWW access during the company's business hours of 8 a.m. to 5 p.m. To configure this, use the command:

```
add fire pol=zone1 ru=3 ac=deny int=eth0 prot=tcp po=www
aft=08:00 bef=17:00
```

In this example, you want to allow DNS queries from a server at 192.168.12.2 to a private DNS server at IP address 192.168.34.1. The queries use UDP port 53. To configure this, 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 po=53
```

In this example, the router uses NAT on the public interface, which is ppp0. You want to allow Telnet access to a UNIX server on a private network, which has the (private) IP address 192.168.1.1. Telnet clients on the public network access the server by connecting to the global IP address 202.49.72.1. The traffic is translated through to the server's private IP address. To configure this, use the command:

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

In this example, you want users on the private network to only access certain destinations. You have specified some of the valid destinations in a file called listip.txt and some via a RADIUS server. To configure this, use the commands:

```
add fire poli=zone1 lis=listallow ty=IP fi=listip.txt
add fire poli=zone1 ru=7 ac=allo int=eth0 lis=listallow
prot=all
add fire poli=zone1 ru=7 lis=rad
```

In this example, when a subset of private users send packets to any address in the range 192.168.1.1 to 192.168.1.100, you want the packets to appear to come from a single global IP address (192.168.1.53). The users have a source address in the range 192.168.2.1 to 192.168.2.100. To configure this, apply an Enhanced NAT rule to the private interface, by using the command:

```
add fire poli=zone1 ru=7 ac=nat natt=enh int=eth0 prot=all
ip=192.168.2.1-192.168.2.100
rem=192.168.1.1-192.168.1.100 gblip=192.168.1.53
```

In this example, when a private user from the 10.1.2.0 subnet sends packets to any address in the range 204.22.3.1 to 204.22.3.99, you want the packets to appear to come from the global subnet 210.25.4.0. To configure this, apply a Standard NAT rule to the private interface and specify a subnet, by using the command:

```
add fire poli=zone1 ru=10 ac=nat natt=sta int=eth0 prot=all
gblip=210.25.4.0 ip=10.1.2.0 natm=255.255.255.0
rem=204.22.3.1-204.22.3.99
```

In this example, you want to perform the same translation as the previous example but for sessions initiated on the public side of the firewall. To configure this, apply a Standard NAT rule to the public interface, to translate to the private subnet 10.1.2.0, by using 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=204.22.3.1-204.22.3.99
```

In this example, when a user with IP address 192.168.0.74 sends traffic to the IP address 192.168.2.27, you want to translate the user's source address to 210.25.4.1 **and** translate the destination address to 210.25.7.1. To configure this, apply a Double NAT rule to the private interface, by using 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 rem=192.168.2.27
gblr=210.25.7.1
```

In this example, you want to redirect all traffic received on the private interface to a destination of 210.25.7.1, without changing the source address. To configure this, apply a Reverse NAT rule to the private interface, by using the command:

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

In this example, the host with private IP address 192.168.1.1 wants to play Xbox Live, over the private interface vlan1. The router's public IP address is 192.0.2.1. You want to limit the rule so that it only translates Xbox Live traffic, which has a source port of 3074. To configure this, use the commands:

```
add fire poli=zone1 ru=1 ac=nat natt=enap int=vlan1 prot=udp
ip=192.168.1.1 gblip=192.0.2.1 so=3074

add fire poli=zone1 ru=2 ac=nat natt=enap int=vlan1 prot=tcp
ip=192.168.1.1 gblip=192.0.2.1 so=3074
```

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 string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, 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 46-4](#) shows an example of an SMTP spam source file.

Figure 46-4: 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), 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 “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](#)

add firewall policy udpporttimeout

Syntax `ADD FIREwall POLIcy=policy-name UDPPorttimeout=port
[TIMEout={0..43200|DEFAULT}]`

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, and the underscore character.
- *port* is a UDP port number or a list of comma-separated UDP port numbers from 1 to 65535.

Description This command assigns a UDP port timeout value to a UDP server port, or group of ports, on the specified policy. For the specified port only, the UDP port timeout value overrides the UDP timeout that is defined with the **set firewall policy** command.

The UDP port timeout is applied to all UDP sessions that use the specified server port. The router ends any inactive sessions on the port when the defined UDP port timeout period expires.

The **udpporttimeout** parameter specifies the port to assign the UDP port timeout value to.

The **timeout** parameter specifies the timeout period for the port in minutes. If you specify **0**, the timeout period is set to 30 seconds. If you specify no value or **default**, then the policy's UDP timeout configured is used. To set the policy's UDP timeout, use the **set firewall policy udptimeout** command.

Example To add a timeout of 25 minutes for all UDP sessions using UDP port 5060, to the policy "zone1", use the command:

```
add fire poli=zone1 udpp=5060 tim=25
```

Related commands [delete firewall policy udpporttimeout](#)
[set firewall policy udpporttimeout](#)
[show firewall policy udpporttimeout](#)

create firewall policy

Syntax `CREate FIREwall POLIcy=policy-name`

where *policy-name* is a 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. Use the following 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 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 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 monitor

Syntax `DELeTe FIREwall MOnitor=monitor-id`

where:

- *monitor-id* is an integer from 1 to 65535

Description This command deletes a monitor. The firewall stops copying packets that come to and from the IP address specified in that monitor.

Example To stop monitoring the host with IP address 192.168.1.1, which is monitored by Monitor 1, use the command:

```
del fire mo=1
```

Related Commands [add firewall monitor](#)
[disable firewall monitor](#)
[enable firewall monitor](#)
[set firewall monitor](#)
[show firewall monitor](#)
[show firewall session](#)

delete firewall policy apprule

Syntax `DELEte FIREWall POLIcy=policy-name APPRule=app-rule-id`

where:

- *app-rule-id* is a number from 1 to 4294967295.
- *policy-name* is a 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 string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character.
- *template* is a 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 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 string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, 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 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 (such as eth0, eth0-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)
- X.25T (such as x25t0, x25t0-1)
- PPP (such as ppp0, ppp1-1)

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). To see a list of current valid interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 limitrule

Syntax `DELEte FIREWall POLIcy=policy-name LIMitrule=rule-id`

Description This command deletes a limit rule from the specified policy.

| Parameter | Description |
|-----------|---|
| POLIcy | The policy you are deleting the rule from. The <i>policy-name</i> is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character. The specified policy must already exist. |
| LIMitrule | The numerical identifier for the rule you are deleting. The <i>rule-id</i> is a decimal number from 1 to 4294967295. |

Examples To delete limit rule 1 from the policy named “EVA”, use the command:

```
del fire poli=EVA lim=1
```

Related Commands [add firewall policy limitrule](#)
[set firewall policy limitrule](#)
[show firewall policy limitrule](#)

delete firewall policy list

Syntax `DELEte FIREwall POLIcy=policy-name LISt=list-name`

where:

- *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character.
- *list-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, 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={ENAPt|ENHanced|STAndard} INTErface=interface`
`GBLInterface=interface [IP=ipadd]`

where:

- *policy-name* is a 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 from an 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.

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

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

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). To see a list of current valid interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 an enhanced 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 string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, 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 (such as eth0, eth0-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)
- X.25T (such as x25t0, x25t0-1)
- PPP (such as ppp0, ppp1-1)

For dynamic interfaces, see [“Dynamic Interfaces” on page 46-16](#). The interface must already exist. To see a list of all currently available interfaces, use the [show interface command on page 9-72 of Chapter 9, 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 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 4294967295.

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 string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, 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. 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 "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 policy udpporttimeout

Syntax `DELEte FIREwall POLIcy=policy-name UDPPorttimeout=port`

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, and the underscore character.
- *port* is a UDP port number or a list of comma-separated UDP port numbers from 1 to 65535.

Description This command deletes a previously defined UDP port timeout from the specified port. The UDP timeout defined with the **set firewall policy** command is once again used for the port.

The **udpporttimeout** parameter specifies the port to delete the previously defined UDP port timeout from.

Example To delete the UDP port timeout for port 5060 on the policy “zone 1”, use the command:

```
del fire poli=zone1 udpp=5060
```

Related commands [add firewall policy udpporttimeout](#)
[set firewall policy udpporttimeout](#)
[show firewall policy udpporttimeout](#)

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.

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 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 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 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 monitor

Syntax DISable FIREwall MOnitor

Description This command stops the firewall from monitoring traffic. Monitoring is disabled by default.

Example To stop the firewall from monitoring any hosts, use the command:

```
dis fire mo
```

Related Commands [add firewall monitor](#)
[delete firewall monitor](#)
[enable firewall monitor](#)
[set firewall monitor](#)
[show firewall monitor](#)
[show firewall session](#)

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* [ACCcounting]
 [FRAGment={ICMP|UDP|OTHER}[,...]]
 [ICMP_Forwarding={ALL|PARAMeter|PING|SOURCEquench|
 TIMEExceeded|TIMESstamp|UNREachable}]
 [LOG={ALLOW|DENY|DENYDump|EVERYDeny|INAIcmp|INALlow|
 INAOther|INATcp|INAUdp|INDDIcmp|INDDOther|INDDTtcp|
 INDDUdp|INDDump|INDENy|INDIcmp|INDOther|INDTtcp|INDUdp|
 OUTAIcmp|OUTAllow|OUTAOther|OUTATcp|OUTAUdp|OUTDDIcmp|
 OUTDDOther|OUTDDTtcp|OUTDDUdp|OUTDDump|OUTDENy|OUTDIcmp|
 OUTDOther|OUTDTtcp|OUTDUdp|SIPAlg}]
 [Options={ALL|RECOOrd_route|SECURity|SOURcerouting|
 TIMESstamp}] [PING]

where *policy-name* is a 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 or logging 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 46-156](#). Accounting records are also written to the Logging Facility. The log can be displayed by using the [show log command on page 60-37 of Chapter 60, Logging Facility](#).

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 46-2 on page 46-47](#) 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 "zone2", use the command:

```
dis fire poli=zone2 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 "zone3", use the command:

```
dis fire poli=zone3 fra=udp
```

Related Commands

- [disable firewall](#)
- [disable firewall notify](#)
- [enable firewall](#)
- [enable firewall notify](#)
- [enable firewall policy](#)
- [show firewall](#)

disable firewall policy debug

Syntax `DISable FIREwall POLIcy[=policy-name]
DEBug={ALL|ARP|CHecksum|HTTP|IDentproxy|LIMitrule|
PACKet|PKT|PRocess|PROXy|RADius|SIPAlg|SMTP|TCP|UPNP}
[DEBUGMode={ALL|ERRORcode|MESSAge|PARSIng|TRAcE}]`

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

Description This command disables debugging of the specified policy or all policies. Debugging is disabled by default.

| Parameter | Description |
|-------------|---|
| POLIcy | The policy that you are disabling debugging on. Any policy specified must already exist. Default: debugging is disabled on all policies |
| DEBug | Specifies one or more types of debugging to be disabled. You can specify a single type or a comma-separated list of types. Default: no default |
| ALL | Disables all debugging options. |
| ARP | Disables the display of information about ARP requests processed by the firewall. |
| CHecksum | Disables the display of information about errors detected in TCP checksums, and TCP sequence and acknowledgement numbers. |
| HTTP | Disables the display of information about request and response messages passing through the HTTP proxy. |
| IDentproxy | Disables the display of information about the IDENT proxy's request processing. |
| LIMitrule | Disables the display of information related to limit rules. |
| PACKet, PKT | Disables the display of the first 56 bytes of each IP packet received. |
| PRocess | Disables the display of information about the processing of a particular IP packet. |
| PROXy | Disables the display of general information about firewall proxies. |
| RADius | Disables the display of information about the firewall's RADIUS queries. |
| SIPAlg | Disables the display of information about the SIP application layer gateway and packets it processes. |
| SMTP | Disables the display of information about sent and received SMTP commands in the SMTP proxy. |
| TCP | Disables the display of information about TCP traffic traversing the firewall. |
| UPNP | Disables the display of UPNP debugging information. |

| Parameter | Description (cont.) |
|-----------|---|
| DEBUGMode | Specifies the types of debugging information to be disabled. You can specify a single mode or a comma-separated list of modes. This parameter is only valid when debug=sipalg . Default: all |
| ALL | Disables all SIP ALG debugging mode options. |
| ERRORcode | Disables the display of translated internal SIP ALG error codes. |
| MESSage | Disables the display of line by line, translated SIP messages. |
| PARSing | Disables the display of the steps the firewall takes during the parsing of a SIP message. |
| TRAcE | Disables the display of the names of all the functions that the SIP ALG calls when it processes a SIP message. |

Examples To stop displaying packet debugging for all policies, use the command:

```
dis fire poli deb=pack
```

To stop displaying how the firewall modifies SIP messages processed by the policy “voip”, use the command:

```
dis fire poli=voip deb=sipa debugm=pars
```

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

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

Related Commands [enable firewall policy debug](#)
[show firewall policy](#)

disable firewall policy httpcookies

Syntax `DISable FIREwall POLIcy=policy-name HTTPCookies`

where *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, 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 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.

The IDENT proxy on the router is enabled by default.

Example To disable the firewall's IDENT proxy, 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 string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, 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 smtp
```

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 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 a 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](#)

disable firewall sessionreport

Syntax DISable FIREwall SESSionreport

Description This command stops the firewall from maintaining a database of the number of firewall sessions that private and public devices use. SNMP reporting will no longer have access to this database. Note that there is a resource cost for the router to maintain this database, so session reporting is disabled by default.

Examples To stop the firewall from maintaining the session report for SNMP, use the command:

```
dis fire ses
```

Related Commands [enable firewall sessionreport](#)
[show firewall](#)

disable firewall sipalg

Syntax DISable FIREwall SIPAlg

Description This command disables the Session Initiation Protocol (SIP) Application Layer Gateway (ALG). The SIP ALG is disabled by default.

Example To stop SIP from establishing multimedia sessions through firewall, use the command:

```
dis fire sipa
```

Related Commands [enable firewall sipalg](#)
[set firewall sipalg](#)
[show firewall sipalg](#)
[show firewall sipalg autoclients](#)

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 monitor

Syntax `ENAbLe FIREWall MOnitor`

Description This command enables the firewall to monitor traffic. When you enable monitoring and specify the IP addresses of devices to monitor, the router makes a copy of all packets that go to or from those devices. To specify devices to monitor, use the [add firewall monitor command on page 46-63](#).

Monitoring is disabled by default.

Example To allow the firewall to start monitoring devices, use the command:

```
ena fire mo
```

Related Commands [add firewall monitor](#)
[delete firewall monitor](#)
[disable firewall monitor](#)
[set firewall monitor](#)
[show firewall monitor](#)
[show firewall session](#)

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 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 4, Configuring and Monitoring the System](#) 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 55, 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] [FRAGment={ICMP|UDP|OTHER}[,...]] [ICMP_Forwarding={ALL|PARAMeter|PING|SOURcequench|TIMEExceeded|TIMESstamp|UNREachable}] [LOG={ALLOW|DENY|DENYDump|EVERYDeny|INAIcmp|INALlow|INAOther|INATcp|INAUdp|INDDIcmp|INDDOther|INDDTtcp|INDDUdp|INDDump|INDENy|INDIcmp|INDOther|INDTtcp|INDUdp|OUTAIcmp|OUTAllow|OUTAOther|OUTATcp|OUTAUdp|OUTDDIcmp|OUTDDOther|OUTDDTtcp|OUTDDUdp|OUTDDump|OUTDENy|OUTDIcmp|OUTDOther|OUTDTtcp|OUTDUdp|SIPAlg}] [Options={ALL|RECORD_route|SECURITY|SOURcerouting|TIMESstamp}] [PING]`

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

Description This command enables one or more of:

- the processing of specific types of IP packets by the specified policy
- accounting for the policy
- logging for the policy.

The **policy** parameter specifies the policy for which packet processing attributes, accounting or logging 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 46-156](#). Accounting records are also written to the Logging Facility. The log can be displayed using the [show log command on page 60-37 of Chapter 60, Logging Facility](#).

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 46-135](#).

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 46-2 on page 46-47](#) lists the possible options and their meanings. Logging some firewall events requires additional configuration. [Table 46-3 on page 46-49](#) 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=inal,deny
```

To create a log entry for every outgoing TCP packet that is denied by the policy named "zone1", use the command:

```
ena fire poli=zone1 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 fra=udp
```

Related Commands

- [disable firewall](#)
- [disable firewall notify](#)
- [disable firewall policy](#)
- [enable firewall](#)
- [enable firewall notify](#)
- [show firewall](#)

enable firewall policy debug

Syntax `ENable FIREwall POLIcy[=policy-name]
DEBUg={ALL|ARP|CHECKsum|HTTP|IDEntproxy|LIMitrule|
PACKet|PKT|PROcess|PROXY|RADius|SIPAlg|SMTP|TCP|UPNP}
[DEBUGMode={ALL|ERRORcode|MESSAge|PARSing|TRAcE}]
[IP=ipadd[-ipadd]]`

where:

- *policy-name* is a 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

Description This command enables debugging for the specified firewall policy or for all policies. Debugging is disabled by default.

| Parameter | Description |
|-------------|--|
| POLICY | The policy that you are enabling debugging on. Any policy specified must already exist. Default: debugging is enabled on all policies |
| DEBUg | Specifies one or more types of debugging to be enabled. You can specify a single type or a comma-separated list of types. This parameter is not retained over a reboot. Default: no default |
| ALL | Enables all debugging options. |
| ARP | Displays information about ARP requests processed by the firewall. |
| CHECKsum | Displays information about errors detected in TCP checksums, and TCP sequence and acknowledgement numbers. |
| HTTP | Displays information about request and response messages passing through the HTTP proxy. |
| IDEntproxy | Displays information about the IDENT proxy's request processing. |
| LIMitrule | Displays information related to limit rules. |
| PACKet, PKT | Displays the first 56 bytes of each IP packet received. |
| PROcess | Displays information about the processing of a particular IP packet. |
| PROXY | Displays general information about firewall proxies. |
| RADius | Displays information about the firewall's RADIUS queries. |
| SIPAlg | Displays information about the SIP application layer gateway and packets it processes. |
| SMTP | Displays information about sent and received SMTP commands in the SMTP proxy. |
| TCP | Displays information about TCP traffic traversing the firewall. |
| UPNP | Enables UPNP debugging on the firewall policy. |

| Parameter | Description (cont.) |
|-----------|--|
| DEBUGMode | Specifies the types of debugging information to be enabled. You can specify a single mode or a comma-separated list of modes. This parameter is only valid when debug=sipalg . Default: errorcode, message, parsing |
| ALL | Enables all SIP ALG debugging mode options. |
| ERRORcode | Translates internal SIP ALG error codes into meaningful messages, displaying any errors encountered during processing. |
| MESSage | Translates each SIP message that is passed to the SIP ALG and displays its contents line by line. The contents of a SIP message include a SIP header and may include a Session Description Protocol (SDP) message body. Each message is displayed first in its unmodified state as it arrives for processing by the SIP ALG, then in its modified state after processing. |
| PARSing | Displays the steps the firewall takes during the parsing of a SIP message (header and body) while they are occurring. This includes showing how the message is modified to facilitate communication across the firewall. |
| TRAcE | Displays the names of all the functions that the SIP ALG calls when it processes a SIP message. |
| IP | Specifies an IP address or a range of addresses, and is valid only when debug=sipalg . If you specify ip , the firewall only displays debugging messages for packets whose IP address matches the specified address. The firewall matches the specified IP address against the source and destination addresses of packets on both the private and public interfaces. Default: displays debugging messages for all IP addresses |

Examples To see details about TCP traffic traversing the firewall for all policies, use the command:

```
ena fire poli deb=tcp
```

To see how the firewall modifies SIP messages processed by the “voip” policy, use the command:

```
ena fire poli=voip deb=sipa debugm=pars
```

Related Commands [disable firewall policy debug](#)
[show firewall policy](#)

enable firewall policy httpcookies

Syntax `ENABle FIREwall POLIcy=policy-name HTTPCookies`

where *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, 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 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 for a firewall, 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.

The IDENT proxy on the router is enabled by default.

Example To enable the firewall's IDENT proxy for the "office" policy, use the command:

```
ena fire poli=office id
```

Related Commands [disable firewall policy identproxy](#)

enable firewall policy smtprelay

Syntax `ENABle FIREwall POLIcy=policy-name SMTPRelay`

where *policy-name* is a string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, 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 smtp
```

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 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.

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](#)

enable firewall sessionreport

Syntax ENable FIREwall SESSionreport

Description This command enables the firewall to create a database that records individual firewall sessions for SNMP reporting. The database monitors the number of sessions created by private and public devices. Note that there is a resource cost for the router to maintain this database, so session reporting is disabled by default.

Examples To enable the firewall to create a session report that SNMP can access, use the command:

```
ena fire ses
```

Related Commands [disable firewall sessionreport](#)
[show firewall](#)

enable firewall sipalg

Syntax ENable FIREwall SIPAlg

Description This command enables the Session Initiation Protocol Application Layer Gateway (SIP ALG). The SIP ALG allows SIP to set up sessions through the firewall when used in combination with NAPT firewall policy rules to modify SIP packets. The SIP ALG is disabled by default.

Example To enable the SIP ALG, use the command:

```
ena fire sipa
```

Related Commands [add firewall policy rule](#)
[disable firewall sipalg](#)
[set firewall sipalg](#)
[show firewall policy](#)
[show firewall sipalg](#)
[show firewall sipalg autoclients](#)

reset firewall policy maccache

Syntax `RESET FIREwall POLIcy=policy-name MACCache`

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

Description This command resets the firewall MAC address cache. Entries are added to the MAC address cache when information is received from a RADIUS sever in response to MAC address queries. MAC address RADIUS queries are generated by firewall policy rules with **macradius** specified in the **list** parameter.

All entries in the cache are deleted. New queries are sent to the RADIUS server to determine if a particular MAC address is allowed or denied, as required.

The **policy** parameter specifies the policy for which MAC cache entries will be deleted. If no policy is specified, all policies' cache entries are deleted.

Examples To reset the MAC address cache for the “office” policy, use the command:

```
reset firewall poli=office macc
```

Related Commands

- [set firewall policy](#)
- [show firewall policy](#)
- [show firewall policy maccache](#)
- [add firewall policy rule](#)

reset firewall sipalg autoclients

Syntax RESET FIREwall SIPAlg AUTOclients

Description This command deletes the SIP ALG's current client database. The SIP ALG generates this database when it is in automatic client management mode. This command deletes both the dynamic version in RAM and the static version in flash memory. Established SIP sessions are not affected.

Note that you can delete single automatic sessions using the **delete firewall session** command. Use the **show firewall sipalg autoclients** command to determine the session number.

Example To reset the client details created by the SIP ALG in automatic mode, use the command:

```
reset fire sipa auto
```

Related Commands

- [delete file](#) in Chapter 6, Managing the File System
- [delete firewall session](#)
- [disable firewall sipalg](#)
- [disable flash autowrite](#) in Chapter 6, Managing the File System
- [enable firewall sipalg](#)
- [enable flash autowrite](#) in Chapter 6, Managing the File System
- [set firewall sipalg](#)
- [show firewall sipalg autoclients](#)

reset firewall sipalg counter

Syntax RESET FIREwall SIPAlg COUnTer

Description This command resets the counters for the SIP ALG, which are displayed by using the **show firewall sipalg counter** command.

Examples To reset the counters for the SIP ALG, use the command:

```
reset fire sipa cou
```

Related Commands

- [disable firewall sipalg](#)
- [enable firewall sipalg](#)
- [show firewall sipalg counter](#)

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.

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 monitor

Syntax SET FIREwall MOnitor=*monitor-id* [IP=*ipadd*]
[COpyto=*ip-interface*] [APPLYto={PRIVate|PUBLIC|BOTH}]

where:

- *monitor-id* is an integer from 1 to 65535
- *ipadd* is an IPv4 address in dotted decimal notation
- *ip-interface* is a VLAN or Eth interface such as vlan2 or eth0. The interface can be a logical interface such as vlan2-1 or eth0-1

Description This command modifies a session monitor.

Note that modifying the monitor does not reset its counters.

The **monitor** parameter specifies the identification number for the monitor.

The **ip** parameter specifies the IP address of the monitored device. The firewall monitors any firewall sessions that have this IP address in any of the session fields. These session fields display in output from the **show firewall session** command, and are summarised in the following table.

| IP field name in session | Meaning |
|--------------------------|---|
| IP | 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 | 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 | 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. If NAT is not configured, this is the same as IP. |
| Gbl Remote IP | 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. If NAT is not configured, this is the same as Remote IP. |

Therefore, sessions are monitored whether the device:

- sends the packets
- receives the packets
- initiates the session
- responds to a session initiated by another device

The **copyto** parameter specifies the Eth interface or VLAN to which the firewall sends the copies of monitored packets. Packets are sent as Layer 2 broadcasts to this interface. You should connect a device directly to this interface that can correctly capture the broadcast packets, such as a PC running packet capturing software. In particular, the device should not forward or reply to the packets. Duplicated packets use the router's MAC address as their source MAC address, and have a broadcast destination MAC address (ff:ff:ff:ff:ff:ff).

The **applyto** parameter specifies where the monitoring for this device applies. If you specify **private**, the firewall copies packets at the private interface. This is before firewall processing for outgoing packets and after firewall processing for incoming packets. If you specify **public**, the firewall copies packets at the public interface. This is before firewall processing for incoming packets and after firewall processing for outgoing packets. If you specify **both**, the firewall copies packets at both the public interface and the private interface. The default is **private**.

The combination of **ip** and **applyto** uniquely identifies a monitor. For example, you can create different monitors to monitor the same IP address on the private and the public interfaces.

Example To change Monitor 1 so that it sends copied packets out over vlan3, use the command:

```
add fire mo=1 cop=vlan3
```

Related Commands

- [add firewall monitor](#)
- [delete firewall monitor](#)
- [disable firewall monitor](#)
- [enable firewall monitor](#)
- [show firewall monitor](#)
- [show firewall session](#)

set firewall policy

Syntax SET FIREwall POLIcy=*policy-name* [FTPDataport={RFC|ANY}]
 [ICMPUnreachabletimeout=0..65535]
 [MACCachetimeout=1..43200] [MAXupnpportmaps={0..1000}]
 [OTHERTimeout=0..43200] [RADIUSlimit=1..500]
 [TCPTimeout=0..43200] [UDPTimeout=0..43200]
 [UPNP={ON|OFF|YES|NO|ENabled|Disabled}]

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

Description This command sets various timeout periods and limits for a firewall policy. 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.

If you enable the load balancer, the value configured for the load balancer's **orphantimeout** parameter (see the [set loadbalancer command on page 51-41 of Chapter 51, Server Load Balancing](#)) overwrites values set for the **tcptimeout**, **udptimeout**, and **othertimeout** parameters.

| Parameter | Description |
|------------------------|---|
| POLlcy | The policy that you are altering the timeout or limit for. The specified policy must already exist. Default: no default |
| FTPDataport | Whether or not the firewall allows only FTP data channels with source ports that conform to RFC 959. This states that the source port for FTP data channels must be 20 when the FTP session is using active mode. See " FTP Session Handling " on page 46-30 for further details. Default: rfc |
| | RFC When the FTP session is in active mode, the firewall only allows data channels using port 20 to traverse it. |
| | ANY The firewall allows data channels using any port to traverse it. |
| ICMPUnreachabletimeout | The delay before the firewall deletes a session after it receives an ICMP unreachable message for that session. If you are configuring the firewall to allow Xbox Live sessions, increase this timeout to a few seconds, for example, 20. Otherwise you may be unable to connect to remote Xboxes that are also behind a firewall. Default: 0 |
| | 0 The firewall deletes the associated session immediately. |
| | 1..65535 Delay time in seconds. The firewall deletes the associated session once it reaches this time delay. |
| MACCachetimeout | The maximum amount of time in minutes for which MAC address entries created by this policy may remain in the MAC address cache. Entries are added to the MAC address cache when information is received from a RADIUS server in response to MAC address queries. MAC address RADIUS queries are generated by firewall policy rules with macradius specified in the list parameter. Default: 1440 (24 hours) |

| Parameter | Description (cont.) | | | | |
|------------------|--|------------------|---|------------------|--|
| MAXupnpport maps | An explicit limit to the number of dynamic NAT entries (or port maps) available for UPnP on the firewall policy. Default: 250 | | | | |
| OTHERTimeout | The timeout period in minutes for sessions other than TCP or UDP. The default is 20 minutes. Note the following exceptions to the timeout specified by this parameter: <ul style="list-style-type: none"> • ICMP sessions have a default and maximum timeout of 10 minutes. If othertimeout is set to more than 10 minutes, ICMP sessions timeout after 10 minutes. • When a firewall session establishes, its timeout is initially set to 5 minutes. Once the session processes two or more packets, its timeout changes to the value specified by this parameter. Default: 20 , except as described above <table> <tr> <td>0</td><td>The session times out after 30 seconds.</td></tr> <tr> <td>1..43200</td><td>Time in minutes after which the session times out.</td></tr> </table> | 0 | The session times out after 30 seconds. | 1..43200 | Time in minutes after which the session times out. |
| 0 | The session times out after 30 seconds. | | | | |
| 1..43200 | Time in minutes after which the session times out. | | | | |
| RADiuslimit | The maximum number of outstanding RADIUS queries that this policy can have at any one time. RADIUS queries are generated by rules that specify the radius or macradius value for the list parameter. Default: 100 | | | | |
| TCPTimeout | The timeout period in minutes for a TCP session. Default: 60 <table> <tr> <td>0</td><td>The TCP session times out after 30 seconds.</td></tr> <tr> <td>1..43200</td><td>Time in minutes after which the session times out.</td></tr> </table> | 0 | The TCP session times out after 30 seconds. | 1..43200 | Time in minutes after which the session times out. |
| 0 | The TCP session times out after 30 seconds. | | | | |
| 1..43200 | Time in minutes after which the session times out. | | | | |
| UDPTimeout | The timeout period in minutes for a UDP session. Default: 20 <table> <tr> <td>0</td><td>The UDP session times out after 30 seconds.</td></tr> <tr> <td>1..43200</td><td>Time in minutes after which the session times out.</td></tr> </table> | 0 | The UDP session times out after 30 seconds. | 1..43200 | Time in minutes after which the session times out. |
| 0 | The UDP session times out after 30 seconds. | | | | |
| 1..43200 | Time in minutes after which the session times out. | | | | |
| UPNP | Whether this firewall policy is enabled for UPnP sessions. Default: no <table> <tr> <td>YES, ON, ENabled</td><td>UPnP sessions are allowed on this policy.</td></tr> <tr> <td>NO, OFF DIsabled</td><td>UPnP sessions are not allowed on this policy.</td></tr> </table> | YES, ON, ENabled | UPnP sessions are allowed on this policy. | NO, OFF DIsabled | UPnP sessions are not allowed on this policy. |
| YES, ON, ENabled | UPnP sessions are allowed on this policy. | | | | |
| NO, OFF DIsabled | UPnP sessions are not allowed on this policy. | | | | |

Examples To time out TCP sessions for the policy “zone1” after 15 minutes 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 max=1000
```

Related Commands

- [delete firewall session](#)
- [show firewall policy](#)
- [set firewall policy](#)
- [set firewall policy attack](#)
- [set firewall policy rule](#)
- [set firewall policy smtpdomain](#)

set firewall policy attack

Syntax SET FIREwall POLIcy=*policy-name*
 ATtack={DOSFlood|FRAGment|HOSTScan|IPSPoof|LAND|OTHER|
 PINGOfdeath|PORTScan|SMTPrelay|SMURF|SMURFamp|SPAM|
 SYNAttack|TCPTiny|UDPAAttack} [INTrigger=*count*]
 [OUTTrigger=*count*] [DETail=*count*] [Time=*minutes*]

where:

- *count* is a number from 0 to 4294967295.
- *minutes* is a time period from 1 to 4294967295 minutes.
- *policy-name* is a 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. For a full description of each attack type, see [“Attacks Recognised by the Firewall” on page 46-42](#). The following table summarises the attack types:

| Type | Description |
|-------------|--|
| 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. |
| OTHER | Miscellaneous unexpected events that may constitute an attack. These events typically occur when the firewall is misconfigured. |
| 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 <i>Internet Control Message Protocol</i> (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. |
| UDPAATTACK | 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 46-5](#)).

Table 46-5: Defaults for **set firewall policy attack** command parameters

| attack | intrigger | outtrigger | time | detail | trigger name |
|-------------|-----------|------------|------|--------|--------------|
| 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 the notification threshold on the “zone1” policy for Dos flood attacks to 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 limitrule

Syntax SET FIREwall POLIcy=*policy-name* LIMIt rule=*rule-id*
 [INTErface=*interface*] [IP=*ipadd*[-*ipadd*]]
 [GBLRemoteip=*ipadd*[-*ipadd*]] [SRCIp limit=0..10000]

Description This command modifies a limit rule attached to a firewall policy. Limit rules apply a limit to the number of concurrent sessions that a device can initiate through the firewall. Each firewall policy can have up to 100 limit rules. The details for a session must match all values set for the **interface**, **ip**, and **gblremote** parameters for the limit rule to apply.

Each time a device initiates a session across the firewall, the router checks all the limit rules attached to a policy. If a session exceeds the limit in a matching rule, then the router does not allow the new session to start. The device can only start the new session once it has ended one or more of the current sessions.

This command only applies the limit as sessions are created; it does not end any sessions established by a device before this rule was modified. However, all matching existing session numbers are included when the router checks the limit rules.

| Parameter | Description |
|------------|---|
| POLIcy | The policy that the rule is added to. The <i>policy-name</i> is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character. The specified policy must already exist. |
| LIMIt rule | A numerical identifier for the rule for this policy. The <i>rule-id</i> is a decimal number from 1 to 4294967295. The specified rule must already exist. |
| INTErface | The interface that the rule is attached to. The interface must already exist and belong to the policy. Valid interfaces are: <ul style="list-style-type: none"> • eth (such as eth0, eth0-1) • VLAN (such as vlan1, vlan1-1) • FR (such as fr0, fr0-1) • X.25T (such as x25t0, x25t0-1) • PPP (such as ppp0, ppp1-1) 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). See "Dynamic Interfaces" on page 46-16 . To see a list of all currently available interfaces, use the show interface command on page 9-72 of Chapter 9, Interfaces. Default: all interfaces attached to the policy |
| IP | IP address of the private device or range of devices you are limiting the sessions for. Devices must be on the private side of the firewall. The IP address is specified using dotted decimal notation. Default: all private devices |

| Parameter | Description |
|-------------|--|
| GBLRemoteip | IP address of a public device or range of devices you are limiting the sessions for. Devices must be on the public side of the firewall. The IP address is specified using dotted decimal notation. Default: all public devices |
| SCRlimit | Number of sessions matching this rule that each device is allowed. Default: 0 (no limit set) |

Examples To modify limit rule 1 attached to vlan2 for the “Nerv_office” policy to match IP address 202.36.164.113, use the command:

```
set fire poli=Nerv_office lim=1 int=vlan2 ip=202.36.164.113
```

Related Commands [add firewall policy limitrule](#)
[delete firewall policy limitrule](#)
[show firewall policy limitrule](#)

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 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 4294967295.
- *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.
- *service-name* is a pre-defined name for an IP service ("[Predefined IP protocol service names](#)" on page 46-93).

Description This command modifies a policy rule. Policy rules are filters that act on the packets processed by the specified policy on the specified public or private interface. Policy rules give you fine control over access and NAT.

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

- 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.

| Parameter | Description |
|-----------|--|
| POLICY | The policy to add the rule to. The specified policy must already exist. Default: no default |
| RULE | A number from 1 to 4294967295 that is both an identifier for the rule and the rule position in this policy's list of rules. Rules are processed in order, from the lowest to the highest numbered rule. Default: no default |
| PROTOcol | The IP protocol number or the name of a predefined protocol type (" Predefined IP protocol service names " on page 46-93). The rule applies to packets from that protocol. If you specify tcp or udp , also specify a port to match. Default: no default |

| Parameter (cont.) | Description (cont.) | | | | | | | | | | | | | | | | | | | | |
|-------------------|--|-------|---|------|---|-----|----------|-----|------------|-----|-----------|-----|---------|-----|-----------|-----|---------|---------|--|---------|-------------------------------------|
| AFTer | <p>The time of day when the rule applies: from this time until either midnight or the time specified with before. Enter times in 24-hour format.</p> <p>Default: no default (the rule always applies)</p> | | | | | | | | | | | | | | | | | | | | |
| BEFore | <p>The time of day when the rule applies: from either midnight or the time specified with after, until this time. Enter times in 24-hour format.</p> <p>Default: no default (the rule always applies)</p> | | | | | | | | | | | | | | | | | | | | |
| DAYS | <p>The days on which the rule applies, in a comma-separated list.</p> <p>Default: all</p> <table> <tr> <td>ALL</td><td>All days of the week.</td></tr> <tr> <td>MON</td><td>Monday.</td></tr> <tr> <td>TUE</td><td>Tuesday.</td></tr> <tr> <td>WED</td><td>Wednesday.</td></tr> <tr> <td>THU</td><td>Thursday.</td></tr> <tr> <td>FRI</td><td>Friday.</td></tr> <tr> <td>SAT</td><td>Saturday.</td></tr> <tr> <td>SUN</td><td>Sunday.</td></tr> <tr> <td>WEEKDay</td><td>A shortcut for all days from Monday to Friday.</td></tr> <tr> <td>WEEKEnd</td><td>A shortcut for Saturday and Sunday.</td></tr> </table> | ALL | All days of the week. | MON | Monday. | TUE | Tuesday. | WED | Wednesday. | THU | Thursday. | FRI | Friday. | SAT | Saturday. | SUN | Sunday. | WEEKDay | A shortcut for all days from Monday to Friday. | WEEKEnd | A shortcut for Saturday and Sunday. |
| ALL | All days of the week. | | | | | | | | | | | | | | | | | | | | |
| MON | Monday. | | | | | | | | | | | | | | | | | | | | |
| TUE | Tuesday. | | | | | | | | | | | | | | | | | | | | |
| WED | Wednesday. | | | | | | | | | | | | | | | | | | | | |
| THU | Thursday. | | | | | | | | | | | | | | | | | | | | |
| FRI | Friday. | | | | | | | | | | | | | | | | | | | | |
| SAT | Saturday. | | | | | | | | | | | | | | | | | | | | |
| SUN | Sunday. | | | | | | | | | | | | | | | | | | | | |
| WEEKDay | A shortcut for all days from Monday to Friday. | | | | | | | | | | | | | | | | | | | | |
| WEEKEnd | A shortcut for Saturday and Sunday. | | | | | | | | | | | | | | | | | | | | |
| ENCapsulation | <p>The way in which traffic must be encapsulated, in order to match the rule.</p> <p>Default: none</p> <table> <tr> <td>IPSec</td><td>The rule applies to traffic that 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.</td></tr> <tr> <td>NONE</td><td>The rule does not match on encapsulation.</td></tr> </table> | IPSec | The rule applies to traffic that 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. | NONE | The rule does not match on encapsulation. | | | | | | | | | | | | | | | | |
| IPSec | The rule applies to traffic that 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. | | | | | | | | | | | | | | | | | | | | |
| NONE | The rule does not match on encapsulation. | | | | | | | | | | | | | | | | | | | | |
| GBLIP | <p>A public IP address. Only valid when the interface is using NAT or when action=nat.</p> <ul style="list-style-type: none"> On a public interface, the rule applies to incoming packets with this destination IP address. If action=nat, the firewall translates this address to ip. On a private interface, the firewall translates private IP addresses to this IP address in outgoing packets. <p>If nattype=enhanced and the interface is a public interface, this parameter is invalid. Use ip instead.</p> <p>If the interface uses NAT and the global IP address is dynamically assigned, you can match this IP address by gblip=0.0.0.0.</p> <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> | | | | | | | | | | | | | | | | | | | | |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| GBLPort | <p>A TCP or UDP port, a range of ports, or a predefined service name (such as FTP—see “Predefined IP protocol service names” on page 46-93).</p> <ul style="list-style-type: none"> On a public interface that uses NAT (as configured with the add firewall policy nat command), the rule applies to incoming packets that are destined for this port. If the interface uses enhanced NAT, the rule then translates the port to the value specified by port. On a private interface, if action=nat and nattype=napt, then the firewall translates private port numbers to this port number. On a public interface, if action=nat and nattype=napt, then the firewall translates this port number to the private port number specified by port. If action=nat and nattype is not napt, then this parameter is invalid. Use the sourceport or port parameters if you want the rule to match against ports. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |
| GBLRemoteip | <p>A public IP address or range of addresses. Only valid when action=nat and nattype=enhanced, double or reverse.</p> <ul style="list-style-type: none"> On a public interface, the rule applies to packets from this IP address. The firewall translates this address to remoteip. If nattype=enhanced, gbremoteip can be a single IP address or a range of addresses. If nattype=reverse or double, then gbremoteip is a single IP address. On a private interface when nattype=reverse or double, the firewall translates the destination address of packets to this address. Gblremoteip is not valid with enhanced NAT. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |
| IP | <p>An IP address or range of addresses.</p> <p>On a private interface, the rule applies to packets that come from this IP address. If NAT is active on the interface or if action=nat, then the firewall translates this address to gblip.</p> <p>On a public interface:</p> <ul style="list-style-type: none"> If neither the interface nor the rule use NAT, or if nattype=reverse or enhanced, then the rule applies to packets sent to this IP address. If NAT is active on the interface, or if action=nat and nattype=standard, enhanced or double, then the firewall translates the destination address of incoming packets to this address. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> |

| Parameter (cont.) | Description (cont.) | | | | | | |
|---------------------|---|-----|---|--------------------|---|---------------------|---|
| NATMask | <p>An IP address mask that translates IP addresses from one subnet to another. For example, this would allow you to translate all 192.168.x.x addresses to 202.202.x.x addresses. Only valid when action=nat and nattype=double, reverse or standard.</p> <p>If nattype=double, the NAT mask is applied to the ip, gblip, remoteip and gblremoteip parameters.</p> <p>If nattype=reverse, the NAT mask is applied to both the remoteip and gblremoteip parameters.</p> <p>If nattype=standard, the NAT mask is applied to both the ip and gblip parameters.</p> <p>The parameters to which the NAT mask applies must specify a single IP address if the natmask parameter is used.</p> | | | | | | |
| Port | <p>A TCP or UDP port or a range of port numbers.</p> <p>On a private interface, the rule applies to packets that are destined for this port on the public side of the firewall.</p> <p>On a public interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets that are destined for this port on the public side of the firewall. • If the interface uses Enhanced NAT, then the firewall translates the destination port to this port number. If action=nat and nattype=napt, then gblport specifies the destination port, so the firewall translates gblport to port. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: all</p> <table> <tr> <td>ALL</td><td>Any port number. The rule does not use destination port number to select packets.</td></tr> <tr> <td><i>port[-port]</i></td><td>A single port number, from 0 to 65535, or a range of numbers.</td></tr> <tr> <td><i>service-name</i></td><td>A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93.</td></tr> </table> | ALL | Any port number. The rule does not use destination port number to select packets. | <i>port[-port]</i> | A single port number, from 0 to 65535, or a range of numbers. | <i>service-name</i> | A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93 . |
| ALL | Any port number. The rule does not use destination port number to select packets. | | | | | | |
| <i>port[-port]</i> | A single port number, from 0 to 65535, or a range of numbers. | | | | | | |
| <i>service-name</i> | A named service, which is equivalent to a port number. For a list of named services, see “Predefined IP protocol service names” on page 46-93 . | | | | | | |
| REMOteip | <p>A public IP address or a range of public IP addresses.</p> <p>On a private interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets that are destined for this address. • If action=nat and nattype=reverse or double, then the firewall translates this address to gblremoteip. <p>On a public interface:</p> <ul style="list-style-type: none"> • If neither the interface nor the rule use NAT, then the rule applies to packets from this address. • If action=nat and nattype=enhanced, reverse or double, then the firewall translates gblremoteip to this address. <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: no default</p> | | | | | | |

| Parameter (cont.) | Description (cont.) |
|--------------------|---|
| SOurceport | <p>A TCP or UDP port, or a range of port numbers. The rule applies to packets that come from this port.</p> <p>See “IP and port parameters in policy rules” on page 46-89 for a summary of valid parameter combinations.</p> <p>Default: all</p> |
| ALL | Any port number |
| <i>port[-port]</i> | A single port number, from 0 to 65535, or a range of numbers. |
| TTL | <p>The length of time for which the rule exists, in hours and minutes. The firewall starts applying the rule as soon as you create it and deletes it when the specified time expires. When the time expires, the firewall also destroys all entries that it created from this rule.</p> <p>Note that rules defined with a TTL value do not appear in router-generated configuration scripts because they are dynamic.</p> |

Examples To modify rule number 1 in the policy “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 “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 string 1 to 15 characters long. It may contain uppercase and lowercase letters, digits, and the underscore character.
- *domain-name* is a string 1 to 131 characters long. Valid characters are uppercase and lowercase letters, digits, 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 compared with either of the following:

- 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 exist.

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 "alliedtelesis.com" as the domain name for use by firewall policy "zone1", use the command:

```
set fire poli=zone1 smtp=alliedtelesis.com
```

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](#)

set firewall policy udpporttimeout

Syntax SET FIREwall POLIcy=*policy-name* UDPPorttimeout=*port*
TIMEout={0..43200|DEFAULT}

where:

- *policy-name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits, and the underscore character.
- *port* is a UDP port number or a list of comma-separated UDP port numbers from 1 to 65535.

Description This command sets a UDP port timeout for the specified server port. You must first add a UDP port timeout to the port with the **add firewall policy udpporttimeout** command.

The UDP port timeout is applied to all UDP sessions that use the specified port. Any inactive sessions on the port are ended when the defined UDP port timeout period expires.

The **udpporttimeout** parameter specifies the port to assign the UDP port timeout to.

The **timeout** parameter specifies the timeout period for the port in minutes. If 0 is specified, the timeout period is set to 30 seconds. If **default** is specified, then the UDP timeout configured for the policy with the **set firewall policy** command is used.

Example To set a udp port timeout of 30 minutes for UDP port 5060, for the policy "zone1", use the command:

```
set fire poli=zone1 udpp=5060 tim=30
```

Related commands [add firewall policy udpporttimeout](#)
[delete firewall policy udpporttimeout](#)
[show firewall policy udpporttimeout](#)

set firewall sipalg

Syntax SET FIREwall SIPAlg
[CALLIdtranslation={False|NO|OFF|ON|True|YES}]
[MODE={MANual|AUTOMatic}] [MAXAutoclients=1..1000]
[MULTIservers={OUTOnly|False|NO|OFF|ON|True|YES}]

Description This command modifies how the SIP ALG operates on the router.

| Parameter | Description |
|-------------------|--|
| CALLIdtranslation | Whether the Call-ID field of a SIP message sent from the private side of the firewall is translated. The firewall only translates the Call-ID when a device within its private network has initiated the SIP session. Default: on |
| | False, NO, OFF The SIP ALG replaces the IP address part of the Call-ID with a globally routable IP address. |
| | ON, True, YES The SIP ALG sends SIP packets with the Call-ID field unchanged. |
| MODE | Whether the clients are managed automatically by the SIP ALG, or manually using policy rules. For more information about using these modes, see these sections: <ul style="list-style-type: none">• “Using Automatic Client Management Mode” on page 46-37• “Using Manual Client Management Mode” on page 46-40 Default: manual |
| | MANual You must configure policy rules for each VoIP client to control their SIP sessions and provide NAT. |
| | AUTOMatic The SIP ALG automatically manages firewall sessions for VoIP clients, and the firewall does not need policy rules configured for SIP traffic. The SIP ALG provides NAT for the clients by using the settings configured by the add firewall policy nat command. The recommended NAT setting is enhanced NAT. |
| MAXAutoclients | The maximum number of SIP clients that the SIP ALG will support when in automatic mode. Once the number of clients registered with firewall sessions reaches this maximum, registrations by other SIP clients are only permitted according to normal firewall behaviour or any configured firewall rules. These excess client's session details are not stored in flash memory, and will age out based on the configured udptimeout or udpporttimeout for the policy. This may interrupt SIP sessions for these clients. Take care not to set maxautoclients to a lower value than the current number of active clients registered, because this may interrupt the SIP sessions. This parameter is only valid when mode is set to automatic . Default: 100 |

| Parameter (cont.) | Description (cont.) |
|-------------------|---|
| MULTIservers | <p>How the SIP ALG interacts with sessions initiated to and from SIP Proxy Servers that are independent of the SIP Registrar. An independent proxy server does not have the same IP or port details as the SIP Registrar.</p> <p>This parameter is only valid when mode is set to automatic.</p> <p>Default: no</p> |
| OUTOnly | <p>Sessions sent by a SIP client to independent proxy servers are matched to the original session created between the SIP client and the SIP Registrar. If NAT is configured, then the translation is the same as the original session between the SIP client and the registrar. However, calls sent by proxy servers that do not have an existing session with the SIP client and do not have a matching allow rule are dropped.</p> |
| False, NO, OFF | <p>A new firewall session is created for any calls to an independent proxy server that the SIP client makes. If NAT is configured, the translation does not match the original session created between the SIP client and the SIP Registrar. If no session currently exists between the proxy server and the SIP client and there is no matching allow rule for the server, any calls sent by the proxy server are dropped.</p> |
| ON, True, YES | <p>Sessions sent and received from independent proxy servers are matched to the original session created between the SIP client and the SIP Registrar. If NAT is configured, then the translation is the same as the original session between the SIP client and the registrar. Calls from proxy servers that match the IP and port details the client has registered with the SIP Registrar are allowed through the firewall.</p> |

Example To enable the SIP ALG to automatically manage SIP clients on the private network, allowing a maximum of 50 SIP clients, use the command:

```
set fire sipa mod=auto maxa=50
```

To disable SIP Call-ID translation, use the command:

```
set fire sipa calli=off
```

Related commands

- [add firewall policy nat](#)
- [disable firewall sipalg](#)
- [enable firewall sipalg](#)
- [reset firewall sipalg autoclients](#)
- [show firewall sipalg counter](#)
- [show firewall sipalg](#)
- [show firewall sipalg autoclients](#)
- [show firewall sipalg counter](#)

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 46-5, Table 46-6).

Figure 46-5: 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
SNMP Session Report ..... disabled
Maximum Packet Fragments .. 20
Sessions:
    Maximum ..... 4000
    Peak ..... 2589
    Active ..... 400

Policy : test
    TCP Timeout (s) ..... 3600
    UDP Timeout (s) ..... 1200
    Other Timeout (s) ..... 1200
    ICMP Unreachable Timeout (s) ..... 0
    TCP Handshake Timeout Mode ..... Normal
    SMTP Domain ..... not set
    TCP Setup Proxy ..... enabled
    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

Policy : voip
    TCP Timeout (s) ..... 3600
    UDP Timeout (s) ..... 1200
    Other Timeout (s) ..... 1200
    TCP Handshake Timeout Mode ..... Normal
    SMTP Domain ..... not set
    TCP Setup Proxy ..... enabled
    UPNP ..... disabled
        WAN interfaces ..... none
        LAN interfaces ..... none
        Maximum port maps ..... 250
    SIP ALG ..... enabled
    Private Interface : eth1-1
    Private Interface : eth1-2
    Public Interface  : eth0-0
        Method ..... dynamic

```

Table 46-6: Parameters in 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 <i>Enable Notify Options</i> includes "port". |
| Notify Mail To | The email address to which notifications are sent. Displayed when <i>Enable Notify Options</i> includes "mail". |
| SNMP Session Report | Status of SNMP session reporting; either enabled or disabled. |
| SIP ALG enabled | Whether the SIP application layer gateway is enabled on the firewall. |
| Maximum Packet Fragments | The maximum number of fragments that a packet may consist of when enhanced fragment handling is enabled. |
| Sessions | Information about the firewall sessions. |
| Maximum | The maximum number of sessions that will be permitted though the firewall. |
| Peak | Peak usage: the maximum number of active sessions that have been opened at one time. |
| Active | The number of sessions currently in use. |
| Policy | The name of a policy. |
| TCP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive TCP sessions. |
| UDP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive UDP sessions. |
| Other Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive non-TCP/UDP sessions. |
| ICMP Unreachable Timeout (s) | The number of seconds before the firewall deletes a session after it receives an ICMP unreachable message for that session. |
| TCP Handshake Timeout Mode | The mode for an automated mechanism for aging unestablished TCP sessions. Depending on the number of suspicious hosts detected, any, or if necessary, all unestablished TCP sessions are aggressively aged, until the number of suspicious hosts decreases again to an acceptable level. If this parameter shows: Normal - All unestablished TCP sessions are aged normally Semi Aggressive - Any unestablished TCP sessions that involve suspicious hosts are being aged aggressively Aggressive - All unestablished TCP sessions are being aged aggressively. |

Table 46-6: Parameters in output of the **show firewall** command (cont.)

| Parameter | Meaning |
|-----------------------|--|
| SMTP Domain | The domain name of the email server found on the private side of the firewall. |
| TCP Setup Proxy | Whether the TCP setup proxy is disabled or enabled. The default is enabled. |
| 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. |
| SIP ALG enabled | Whether the SIP application layer gateway is enabled on the firewall. You cannot enable SIP ALG on individual policies. |
| 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 enhanced, ENAPT, or standard. 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 46-77 , 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. |

Examples To display a summary of all security policies that have been created and the interfaces assigned to each policy, use the command:

```
sh fire
```

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 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 a specified policy, or all policies (Figure 46-6, Table 46-7 on page 46-157). Alternatively, accounting records can be displayed using the [show log command on page 60-37 of Chapter 60, Logging Facility](#).

The **policy** parameter specifies the policy for which accounting records are to be displayed. The specified policy must already exist. If this parameter is not specified, then 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 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 46-6: 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 46-7: Parameters in 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 " <i>direction packets:octets</i> ". |

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

Examples To display the 50 most recent accounting records for the firewall policy named "office", use the command:

```
sh fire acc poli=office ta=50
```

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 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 46-7, Table 46-8).

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 46-7: Example output from the **show firewall arp** command

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

Table 46-8: Parameters in 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 | <p>Interfaces in the policy on which ARP requests are permitted:</p> <p>Public - ARP requests are permitted on the public interface specified by the Gbl Int parameter</p> <p>All Public - ARP requests are permitted on all of the policy's public interfaces</p> <p>Private - ARP requests are permitted on the private interface specified by the Int parameter</p> <p>All Private - ARP requests are permitted on all of the policy's private interfaces</p> <p>An address in an ARP request must match the subnet of the interface on which the ARP request is received.</p> |
| NAT Type | <p>The type of NAT configuration associated with the IP address:</p> <p>Int Based - The address (range) was specified by an interface-based NAT configured with the add firewall policy nat command</p> <p>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</p> |

Table 46-8: Parameters in output of the **show firewall arp** command (cont.)

| Parameter | Meaning |
|-----------|--|
| Int | <p>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.</p> <p>If the NAT Type is Rule, this is the interface to which the rule is attached.</p> <p>If this is a private interface, a dash indicates that the rule is attached to a public interface (see the Gbl Int parameter).</p> |
| Gbl Int | <p>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.</p> <p>If the NAT Type is Rule, this is the interface to which the rule is attached.</p> <p>if this is a public interface, a dash indicates that the rule is attached to a private interface (see the Int parameter).</p> |
| Rule | <p>The number of the rule associated with this entry. When the NAT Type is Int based, no value is displayed.</p> |

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 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 for a specified policy, or for all policies (Figure 46-8, Table 46-9 on page 46-161). The event log stores up to 60 events for each type of event, per policy.

The **event** parameter specifies which type of event 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 this parameter 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 only the most recent events are to be displayed. If a value is specified, output is limited to the specified number of events.

Figure 46-8: 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
-----
```


Figure 46-8: Example output from the **show firewall event** command (cont.)

```

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 46-9: Parameters in 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. |

Examples To display the ten most recent notification events generated for each firewall policy, use the command:

```
sh fire ev=not ta=10
```

Related Commands [disable firewall notify](#)
[enable firewall notify](#)
[show firewall accounting](#)
[show firewall policy](#)
[show firewall session](#)

show firewall monitor

Syntax SHow FIREwall MOnitor

Description This command displays information about session monitoring (Figure 46-9, Table 46-10).

Figure 46-9: Example output from the **show firewall monitor** command

| | | | | | |
|----------------------|-------------|----------|---------|-----------|------------|
| Firewall Monitoring | | | | | |
| Status enabled | | | | | |
| Monitor | IP | Apply to | Copy to | In (pkts) | Out (pkts) |
| 1 | 192.168.1.1 | PRIVATE | VLAN2 | 0 | 0 |
| 2 | 192.168.1.2 | PRIVATE | VLAN2 | 24 | 26 |

Table 46-10: Parameters in output of the **show firewall monitor** command

| Parameter | Meaning |
|-----------|--|
| Status | Whether firewall session monitoring is enabled or disabled. |
| Monitor | The identification number of each monitor. This number uniquely identifies the monitored device. |
| IP | The IP address of the monitored device. The firewall copies all traffic that comes to or from this address. |
| Copy to | The interface to which the firewall transmits copies of packets; one of a VLAN, an Eth interface, or "deleted" if the interface has been deleted. Deleting the interface deactivates the monitor. Adding the interface back again reactivates the monitor. |
| Apply to | The firewall interface on which the firewall captures packets; one of PRIVATE, PUBLIC, or BOTH. PRIVATE means that packets are copied before firewall processing for outgoing packets and after firewall processing for incoming packets. PUBLIC means that packets are copied before firewall processing for incoming packets and after firewall processing for outgoing packets. |
| In | The number of incoming packets that the firewall has captured using this monitor. The counter resets when the router restarts. |
| Out | The number of outgoing packets that the firewall has captured using this monitor. The counter resets when the router restarts. |

Example To display the number of packets that the firewall has copied, use the command:

```
sh fire mo
```

Related Commands

- [add firewall monitor](#)
- [delete firewall monitor](#)
- [disable firewall monitor](#)
- [enable firewall monitor](#)
- [set firewall monitor](#)
- [show firewall session](#)

show firewall policy

Syntax SHow FIREwall POLIcy[=*policy-name*] [COUnter]
[RUle=*rule-id*[-*rule-id*]] [SUMmary]

where

- *policy-name* is a 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 or range from 1 to 4294967295

Description This command displays detailed information about the specified policy or all policies. You can also limit the rules displayed for each policy using the **rule** parameter. [Table 46-11 on page 46-170](#) lists all the parameters that the **show firewall policy** command may display, and gives their meanings. The information actually displayed depends on the policy type, and on whether you specify any of the optional **counter** and **summary** parameters.

The following figures show some of the possible example outputs:

- a policy that uses policy-based rules and interface-based NAT ([Figure 46-10 on page 46-165](#))
- a policy that uses an HTTP proxy and a filter list ([Figure 46-11 on page 46-166](#))
- a policy that uses application rules ([Figure 46-12 on page 46-167](#))
- counters for a policy that uses policy-based rules and interface-based NAT ([Figure 46-13 on page 46-168](#)). To see counters, specify the **counter** parameter.
- summary information for a policy that uses policy-based rules and interface-based NAT ([Figure 46-14 on page 46-178](#), [Table 46-12 on page 46-179](#)). To see a summary, specify the **summary** parameter.

Figure 46-10: Example output from the **show firewall policy** command for a policy that uses rules and interface-based NAT.

```

Policy : Office
  TCP Timeout (s) ..... 3600
  UDP Timeout (s) ..... 1200
  Other Timeout (s) ..... 1200
  ICMP Unreachable Timeout (s) ..... 0
  TCP Handshake Timeout Mode ..... Normal
  MAC Cache Timeout (m) ..... 1440
  RADIUS Limit ..... 100
  Accounting ..... disabled
  Enabled Logging Options ..... none
  Enabled Debug Options ..... none
  Enabled Debug Modes ..... none
  Enabled Debug IP Address ..... none
  Identification Protocol Proxy ..... enabled
  Enabled IP options ..... none
  Enhanced Fragment Handling ..... none
  Enabled ICMP forwarding ..... none
  Receive of ICMP PINGS ..... enabled
  Number of Notifications ..... 1
  Number of Deny Events ..... 28
  Number of Allow Events ..... 172
  Number of Active TCP Opens ..... 3
  Number of Active Sessions ..... 31
  Cache Hits ..... 812
  Discarded ICMP Packets ..... 19
  SMTP Domain ..... not set
  FTP Data Port ..... RFC enforced
  TCP Setup Proxy ..... enabled
  UPNP ..... disabled
    WAN interfaces ..... none
    LAN interfaces ..... none
    Maximum port maps ..... 250
  SIP ALG ..... disabled
  Private Interface : eth0
    Trust Private ..... yes
    Rule ..... 2
      Action ..... deny
      Protocol ..... UDP
      Port ..... 5060
      Global Port ..... all
      Source Port ..... all
      Days ..... all
  Public Interface : eth1
    Method ..... dynamic
    NAT ..... enhanced
      Method ..... enhanced interface
      Private Interface ..... eth0
      Global IP ..... 172.20.8.8
    Rule ..... 1
      Action ..... allow
      IP ..... 192.168.1.14
      Protocol ..... TCP
      Port ..... 80
      Global IP ..... 0.0.0.0
      Global Port ..... 80
      Source Port ..... all
      Days ..... all

```

For a description of the above output parameters, see [Table 46-11 on page 46-170](#).

Figure 46-11: Example output from the **show firewall policy** command for a policy that uses an HTTP proxy and a filter file.

```

Policy : Office
TCP Timeout (s) ..... 3600
UDP Timeout (s) ..... 1200
Other Timeout (s) ..... 1200
ICMP Unreachable Timeout (s) ..... 0
TCP Handshake Timeout Mode ..... Normal
MAC Cache Timeout (m) ..... 1440
RADIUS Limit ..... 100
Accounting ..... disabled
Enabled Logging Options ..... none
Enabled Debug Options ..... none
Enabled Debug Modes ..... none
Enabled Debug IP Address ..... none
Identification Protocol Proxy ..... enabled
Enabled IP options ..... none
Enhanced Fragment Handling ..... none
Enabled ICMP forwarding ..... none
Receive of ICMP PINGS ..... enabled
Number of Notifications ..... 1
Number of Deny Events ..... 28
Number of Allow Events ..... 172
Number of Active TCP Opens ..... 3
Number of Active Sessions ..... 31
Cache Hits ..... 812
Discarded ICMP Packets ..... 19
SMTP Domain ..... not set
FTP Data Port ..... RFC enforced
TCP Setup Proxy ..... enabled
HTTP Proxy Filter File ..... url.txt
HTTP Cookies ..... enabled
UPNP ..... disabled
  WAN interfaces ..... none
  LAN interfaces ..... none
  Maximum port maps ..... 250
SIP ALG ..... disabled
Number of Limitrules ..... 2
Private Interface : eth0
  Trust Private ..... yes
Public Interface : eth1
  Method ..... dynamic
  Proxy ..... http
    Private Interface ..... eth0
    Global IP ..... 172.20.8.8
    Direction ..... out
    Days ..... all

```

For a description of the above output parameters, see [Table 46-11 on page 46-170](#).

Figure 46-12: Example output from the **show firewall policy** command for a policy that uses application rules.

```

Policy : Office
TCP Timeout (s) ..... 3600
UDP Timeout (s) ..... 1200
Other Timeout (s) ..... 1200
ICMP Unreachable Timeout (s) ..... 0
TCP Handshake Timeout Mode ..... Normal
MAC Cache Timeout (m) ..... 1440
RADIUS Limit ..... 100
Accounting ..... disabled
Enabled Logging Options ..... none
Enabled Debug Options ..... none
Enabled Debug Modes ..... none
Enabled Debug IP Address ..... none
Identification Protocol Proxy ..... enabled
Enabled IP options ..... none
Enhanced Fragment Handling ..... none
Enabled ICMP forwarding ..... none
Receive of ICMP PINGS ..... enabled
Number of Notifications ..... 1
Number of Deny Events ..... 28
Number of Allow Events ..... 172
Number of Active TCP Opens ..... 3
Number of Active Sessions ..... 31
Cache Hits ..... 812
Discarded ICMP Packets ..... 19
SMTP Domain ..... not set
FTP Data Port ..... RFC enforced
TCP Setup Proxy ..... enabled
UPNP ..... disabled
    WAN interfaces ..... none
    LAN interfaces ..... none
    Maximum port maps ..... 250
SIP ALG ..... disabled
Number of Limitrules ..... 2
Private Interface : eth0
    Trust Private ..... yes
    Apprule ..... 2
        Application ..... mms
        Action ..... allow
        Use Port ..... 1947
    Apprule ..... 3
        Application ..... rtsp
        Action ..... allow
        Use Port ..... 1812
Public Interface : eth1
    Method ..... dynamic

```

For a description of the above output parameters, see [Table 46-11 on page 46-170](#).

Figure 46-13: Example output from the **show firewall policy counter** command for a policy that uses rules and interface-based NAT

```

Policy : Office
  TCP Timeout (s) ..... 3600
  UDP Timeout (s) ..... 1200
  Other Timeout (s) ..... 1200
  ICMP Unreachable Timeout (s) ..... 0
  TCP Handshake Timeout Mode ..... Normal
  MAC Cache Timeout (m) ..... 1440
  RADIUS Limit ..... 100
  Accounting ..... disabled
  Enabled Logging Options ..... none
  Enabled Debug Options ..... none
  Enabled Debug Modes ..... none
  Enabled Debug IP Address ..... none
  Identification Protocol Proxy ..... enabled
  Enabled IP options ..... none
  Enhanced Fragment Handling ..... none
  Enabled ICMP forwarding ..... none
  Receive of ICMP PINGS ..... enabled
  Number of Notifications ..... 0
  Number of Deny Events ..... 0
  Number of Allow Events ..... 0
  Number of Active TCP Opens ..... 0
  Number of Active Sessions ..... 0
  Cache Hits ..... 0
  Discarded ICMP Packets ..... 0
  SMTP Domain ..... not set
  FTP Data Port ..... RFC enforced
  TCP Setup Proxy ..... enabled
  UPNP ..... disabled
    WAN interfaces ..... none
    LAN interfaces ..... none
    Maximum port maps ..... 250
    Number Port Mappings ..... 0
    Spawned Sessions ..... 0
  SIP ALG ..... disabled
  Number of Limitrules ..... 2
  Private Interface : eth0
    Total Packets Received ..... 18076
    Number Flows Started ..... 817
    Number Cache Hits ..... 17820
    Number Dropped Packets ..... 256
    Number Unknown IP Protocols ..... 0
    Number Bad ICMP Packets ..... 0
    Number Dumped ICMP Packets ..... 81
    Number Spoofing Packets ..... 0
    Number Dropped GBLIP is Zero ..... 0
    Number No Spare Entries ..... 0
    Number FTP Port Commands ..... 12
    Number Bad FTP Port Commands ..... 0
  Rule ..... 2
    Action ..... deny
    Protocol ..... UDP
    Port ..... 5060
    Global Port ..... all
    Source Port ..... all
    Number Hits ..... 0
    Days ..... all

```


Figure 46-13: Example output from the **show firewall policy counter** command for a policy that uses rules and interface-based NAT (cont.)

```

Public Interface : eth1
  Method ..... dynamic
  Total Packets Received ..... 20917
  Number Flows Started ..... 512
  Number Cache Hits ..... 18092
  Number Dropped Packets ..... 2825
  Number Unknown IP Protocols ..... 0
  Number Bad ICMP Packets ..... 0
  Number Dumped ICMP Packets ..... 28
  Number Spoofing Packets ..... 0
  Number Dropped GBLIP is Zero .... 0
  Number No Spare Entries ..... 0
  Number FTP Port Commands ..... 0
  Number Bad FTP Port Commands .... 0
  NAT ..... enhanced
    Method ..... enhanced interface
    Private Interface ..... eth0
    Global IP ..... 172.20.8.8
  Rule ..... 1
    Action ..... allow
    IP ..... 192.168.1.14
    Protocol ..... TCP
    Port ..... 80
    Global IP ..... 0.0.0.0
    Global Port ..... 80
    Source Port ..... all
    Number Hits ..... 512
    Days ..... all

```

For a description of the above output parameters, see [Table 46-11](#).

Table 46-11: Parameters in output of the **show firewall policy [counter]** command

| Parameter | Meaning |
|---|--|
| General information about the policy | |
| Policy | The name of a policy. |
| TCP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive TCP sessions. |
| UDP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive UDP sessions. |
| Other Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive non-TCP/UDP sessions. |
| ICMP Unreachable Timeout (s) | The number of seconds before the firewall deletes a session after it receives an ICMP unreachable message for that session. |
| TCP Handshake Timeout Mode | <p>The mode for an automated mechanism for aging unestablished TCP sessions. Depending on the number of suspicious hosts detected, any, or if necessary, all unestablished TCP sessions are aggressively aged, until the number of suspicious hosts decreases again to an acceptable level.</p> <p>If this parameter shows:</p> <p>Normal - All unestablished TCP sessions are aged normally</p> <p>Semi Aggressive - Any unestablished TCP sessions that involve suspicious hosts are being aged aggressively</p> <p>Aggressive - All unestablished TCP sessions are being aged aggressively.</p> |
| MAC Cache Timeout | <p>The maximum amount of time in minutes that MAC address entries created by this policy may remain in the MAC address cache.</p> <p>Entries are added to the MAC address cache as a result of rules that use RADIUS to authenticate the source MAC address of new flows.</p> |
| RADIUS Limit | The maximum number of outstanding RADIUS queries that this policy can have at any one time. RADIUS queries are generated by rules that use RADIUS to authenticate either the source or destination IP address, or the source MAC address of new flows. |
| RADIUS MAC Lookup | Indicates that this rule uses RADIUS to authenticate the source MAC address of new flows that match this rule. |
| RADIUS IP Lookup | Indicates that this rule uses RADIUS to authenticate the source or destination IP address of new flows that match this rule. |
| Accounting | Whether accounting is enabled or disabled for the policy. |
| Enabled Logging Options | The logging options that are currently enabled. For a description of the available logging options, see Table 46-2 on page 46-47 . If no options are enabled, "none" is displayed. |
| Enabled Debug Options | The debugging options that are currently enabled. For a description of the available debugging options, see the enable firewall policy debug command. If no options are enabled, "none" is displayed. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|----------------------------|--|
| Enabled Debug Modes | The debug modes that are currently enabled, if SIP ALG debugging is enabled; one or more of ALL, ERRORCODE, MESSAGE, PARSING and TRACE. For a description of the available debugging options, see the enable firewall policy debug command. If SIP ALG debugging is disabled, "none" is displayed. SIP ALG is not available on some models. |
| Enabled Debug IP Address | A single IP address or IP address range. If SIP ALG debugging is enabled, the firewall only displays debugging messages for packets whose IP address matches this address. If the firewall displays debugging messages for all IP addresses, "all" is displayed. If SIP ALG debugging is disabled, "none" is displayed. SIP ALG is not available on some models. |
| Enabled IP options | The IP options that are allowed in IP packets forwarded by this policy. One or more of: All Record_route Security Sourcerouting Timestamp If no options have been 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 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. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|----------------------------|---|
| Number of Active TCP Opens | The number of TCP connections for this policy that are currently trying to reach the established state. |
| 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. |
| FTP Data Port | Whether the policy allows only FTP data channels with source ports which conform to RFC 959. This states that the source port for FTP data channels must be 20 when the FTP session is using active mode. "RFC enforced" means that the firewall will only allow data channels if the source port is 20. "RFC not enforced" means that the firewall will allow data channels using any source port. |
| TCP Setup Proxy | Whether the TCP setup proxy is disabled or enabled. The default is enabled. |
| HTTP Proxy Filter File | Name of a text file containing a list of domain name, keyword and cookie permissions that the HTTP proxy enforces under this policy. This parameter is only shown if a URL filter file has been specified for this policy. |
| HTTP 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. |
| Number Port Mappings | [displayed when you specify the counter parameter] The number of current port mappings added by the <i>AddPortMapping</i> 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 | [displayed when you specify the counter parameter] The number of UPnP port mappings created based on UPnP template port mappings. |
| SIP ALG | Whether the SIP application layer gateway is enabled. |
| Number of Limitrules | The number of limit rules configured for the policy. |
| IP List | The name of an IP list used by the policy. |
| File name | The name of the file containing the IP list. |
| Number IP hosts | The number of IP hosts in the IP list. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|--|---|
| Number Networks | The number of IP networks in the IP list. |
| Hardware List | The name of a hardware address list used by the policy. |
| File name | The name of the file containing the hardware list. |
| Number MAC addresses | The number of MAC addresses in the hardware list. |
| Dynamic Template | The name of a dynamic interface template associated with the policy. |
| Information about private and public interfaces | |
| Private Interface | The name of a private interface that is attached to the policy. If the interface is dynamic, the template name and username follow this entry. |
| Trust Private | Whether devices connected to the private interface have unrestricted access to the router. |
| Total Packets Received | [displayed when you specify the counter parameter] The total number of packets received on the private interface. |
| Number Flows Started | [displayed when you specify the counter parameter] The number of flows started on the private interface. |
| Number Cache Hits | [displayed when you specify the counter parameter] The number of flow lookups for the private interface found from the cache. |
| Number Dropped Packets | [displayed when you specify the counter parameter] The number of packets received on the private interface that were dropped. |
| Number Unknown IP Protocols | [displayed when you specify the counter parameter] The number of packets received on the private interface with an unknown IP protocol. |
| Number Bad ICMP Packets | [displayed when you specify the counter parameter] The number of badly formatted ICMP packets received on the private interface. |
| Number Dumped ICMP Packets | [displayed when you specify the counter parameter] The number of ICMP packets received on the private interface that were dumped. |
| Number Spoofing Packets | [displayed when you specify the counter parameter] The number of packets received on the private interface with a spoofed address. |
| Number Dropped GBLIP Zero | [displayed when you specify the counter parameter] The number of packets received on the private interface that were dumped because a valid global IP address was not configured. |
| Number No Spare Entries | [displayed when you specify the counter parameter] The number of packets received on the private interface that were dumped because the system had insufficient memory. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|------------------------------|---|
| Number FTP Port Commands | [displayed when you specify the counter parameter] The number of valid FTP port commands received on the private interface. |
| Number Bad FTP Port Commands | [displayed when you specify the counter parameter] The number of invalid FTP port commands received on the private interface. |
| Public Interface | The name of a public interface that is attached to the policy. If the interface is dynamic, the template name and username follow this entry. |
| Method | The method used to send packets to or from the public interface; either Dynamic (the firewall uses dynamic packet filtering) or Passall (the firewall does not interfere with packet flow). |
| Total Packets Received | [displayed when you specify the counter parameter] The total number of packets received on the public interface. |
| Number Flows Started | [displayed when you specify the counter parameter] The number of flows started on the public interface. |
| Number Cache Hits | [displayed when you specify the counter parameter] The number of flow lookups for the public interface found from the cache. |
| Number Dropped Packets | [displayed when you specify the counter parameter] The number of packets received on the public interface that were dropped. |
| Number Unknown IP Protocols | [displayed when you specify the counter parameter] The number of packets received on the public interface with an unknown IP protocol. |
| Number Bad ICMP Packets | [displayed when you specify the counter parameter] The number of badly formatted ICMP packets received on the public interface. |
| Number Dumped ICMP Packets | [displayed when you specify the counter parameter] The number of ICMP packets received on the public interface that were dumped. |
| Number Spoofing Packets | [displayed when you specify the counter parameter] The number of packets received on the public interface with a spoofed address. |
| Number Dropped GBLIP Zero | [displayed when you specify the counter parameter] The number of packets received on the public interface that were dumped because a valid global IP address was not configured. |
| Number No Spare Entries | [displayed when you specify the counter parameter] The number of packets received on the public interface that were dumped because the system had insufficient memory. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|---|---|
| Number FTP Port Commands | [displayed when you specify the counter parameter] The number of valid FTP port commands received on the public interface. |
| Number Bad FTP Port Commands | [displayed when you specify the counter parameter] The number of invalid FTP port commands received on the public interface. |
| NAT | Whether the type of NAT translation is enhanced, ENAPT, or standard. Only displayed when NAT is enabled on the policy. |
| Method | The method used to perform NAT: 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 46-77 , and is only displayed when NAT has been enabled on the policy by using this command. |
| Private Interface | The private interface to which NATs apply. Displayed when NAT is enabled on the policy. |
| Global IP | The global IP address used by NATs. Displayed when NAT is enabled on the policy. |
| IP | The single private IP address for which the NAT applies. Displayed when the NAT method is Static or Static interface. |
| Interface information for policies that use policy rules (Figure 46-10 on page 46-165) | |
| Rule | The identification number of a rule associated with the public or private interface. |
| Action | The action to perform when a flow matches this rule; one of Allow, Deny, Nat or NoNat. |
| NAT type | Whether the rule applies NAPT or Standard, Enhanced, Reverse or Double NAT to matching traffic. |
| NAT mask | The IP subnet mask used by the rule. Displayed by subnet NAT rules. |
| IP | The IP address or range of the private side devices. For more information see "IP and port parameters in policy rules" on page 46-89 and the ip parameter of the add firewall policy rule command. |
| 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 that the rule applies to. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|---------------|---|
| Port | A port number, service name (" Predefined IP protocol service names " on page 46-93) or range of port numbers, used on the private side of the firewall. For more information see " IP and port parameters in policy rules " on page 46-89 and the port parameter of the add firewall policy rule command. |
| Global IP | A public IP address. For more information see " IP and port parameters in policy rules " on page 46-89 and the gblip parameter of the add firewall policy rule command. |
| Global Port | A port number, service name (" Predefined IP protocol service names " on page 46-93) or range of port numbers, used on the public side of the firewall. For more information see " IP and port parameters in policy rules " on page 46-89 and the gblport parameter of the add firewall policy rule command. |
| Remote IP | An IP address or range of addresses. For more information see " IP and port parameters in policy rules " on page 46-89 and the remoteip parameter of the add firewall policy rule command. |
| Gbl Remote IP | A public IP address or range of addresses. For more information see " IP and port parameters in policy rules " on page 46-89 and the gblremoteip parameter of the add firewall policy rule command. |
| Source Port | The source port to match for this rule. |
| Number Hits | [displayed when you specify the counter parameter] The number of sessions that matched the rule. |
| After | The time of day after which this rule applies. |
| Before | The time of day before which this rule applies. |
| Days | The day or days on which this rule applies. |
| Encapsulation | The encapsulation type of the packets to which the rule applies. |
| TTL | The time until this rule expires and is deleted. |

Interface information for policies that use a proxy (Figure 46-11 on page 46-166)

| | |
|-------------------|--|
| Proxy | The type of proxy in use on the public interface. |
| Private Interface | The interface that receives packets from the private LAN. The proxy processes packets that arrive at this interface and sends them to the public interface. |
| Global IP | The public interface IP address that the proxy uses as the source address for traffic it sends to the public side. The destination address for sessions initiated from the public side, when the direction is "in" or "both". |
| IP | The IP address of the server on the private intranet. Incoming traffic received by the proxy is sent to this address. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|----------------------------|---|
| Direction | The direction in which traffic is permitted to flow through the proxy. |
| Sessions Handled | [displayed when you specify the counter parameter and have an HTTP proxy] The number of TCP sessions that have been handled by the HTTP proxy. |
| URL Denies | [displayed when you specify the counter parameter and have an HTTP proxy] 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 | [displayed when you specify the counter parameter and have an HTTP proxy] 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 | [displayed when you specify the counter parameter and have an HTTP proxy] 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. |
| Number Hits | [displayed when you specify the counter parameter and have an SMTP proxy] The number of sessions that have used the proxy. |
| Rejected Spam Messages | [displayed when you specify the counter parameter and have an SMTP proxy] The number of mail messages with source addresses matching an entry in the SMTP proxy configuration file that have been rejected. |
| Rejected SMTP Relays | [displayed when you specify the counter parameter and have an SMTP proxy] The number of messages requesting third party relay that have been rejected. |
| Rejected Smurf Amp Attacks | [displayed when you specify the counter parameter and have an SMTP proxy] The number of mail messages with a broadcast source or destination address that have been rejected. |
| After | The time of day after which the proxy is active. |
| Before | The time of day before which the proxy is active. |
| Days | The day or days on which the proxy is active. |

Table 46-11: Parameters in output of the **show firewall policy [counter]** command (cont.)

| Parameter | Meaning |
|--|--|
| Interface information for policies that use application rules (Figure 46-12 on page 46-167) | |
| Apprule | The identification number of an application rule associated with the public or private interface. |
| Application | The protocol that traffic is treated as, if the traffic matches this rule. |
| Action | The action to perform when a flow matches this application rule; either Allow or Deny. |
| Use Port | The port that the router uses for the application. Flows to this port are treated as flows for the application. |
| Command | When the application is FTP, whether the rule applies to the Get or Put command. |
| Number Hits | [displayed when you specify the counter parameter] The number of sessions that matched the application rule. |

Figure 46-14: Example output from the **show firewall policy summary** command for a policy that uses rules and interface-based NAT.

```

Policy : Office
TCP Timeout (s) ..... 3600
UDP Timeout (s) ..... 1200
Other Timeout (s) ..... 1200
ICMP Unreachable Timeout (s) ..... 0
TCP Handshake Timeout Mode ..... Normal
SMTP Domain ..... not set
TCP Setup Proxy ..... enabled
UPNP ..... disabled
  WAN interfaces ..... none
  LAN interfaces ..... none
  Maximum port maps ..... 250
SIP ALG ..... disabled
Number of Limitrules ..... 2
Private Interface : eth0
Public Interface : eth1
  Method ..... dynamic
  NAT ..... enhanced
    Method ..... enhanced interface
    Private Interface ..... eth0
    Global IP ..... 172.20.8.8

```

For a description of the above output parameters, see [Table 46-12](#).

Table 46-12: Parameters in output of the **show firewall policy summary** command

| Parameter | Meaning |
|---|--|
| General information about the policy | |
| Policy | The name of a policy. |
| TCP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive TCP sessions. |
| UDP Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive UDP sessions. |
| Other Timeout (s) | The length of time, in seconds, for which the firewall maintains inactive non-TCP/UDP sessions. |
| ICMP Unreachable Timeout (s) | The number of seconds before the firewall deletes a session after it receives an ICMP unreachable message for that session. |
| TCP Handshake Timeout Mode | <p>The mode for an automated mechanism for aging unestablished TCP sessions. Depending on the number of suspicious hosts detected, any, or if necessary, all unestablished TCP sessions are aggressively aged, until the number of suspicious hosts decreases again to an acceptable level.</p> <p>If this parameter shows:</p> <p>Normal - All unestablished TCP sessions are aged normally</p> <p>Semi Aggressive - Any unestablished TCP sessions that involve suspicious hosts are being aged aggressively</p> <p>Aggressive - All unestablished TCP sessions are being aged aggressively.</p> |
| SMTP Domain | The domain name of the email server found on the private side of the firewall. |
| TCP Setup Proxy | Whether the TCP setup proxy is disabled or enabled. The default is enabled. |
| HTTP Proxy Filter File | Name of a text file containing a list of domain name, keyword and cookie permissions that the HTTP proxy enforces under this policy. This parameter is only shown if a URL filter file has been specified for this policy. |
| HTTP 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. |
| SIP ALG | Whether the SIP application layer gateway is enabled. |
| Number of Limitrules | The number of limit rules configured for the policy. |
| IP List | The name of an IP list used by the policy. |
| File name | The name of the file containing the IP list. |

Table 46-12: Parameters in output of the **show firewall policy summary** command (cont.)

| Parameter | Meaning |
|--|---|
| Number IP hosts | The number of IP hosts in the IP list. |
| Number Networks | The number of IP networks in the IP list. |
| Hardware List | The name of a hardware address list used by the policy. |
| File name | The name of the file containing the hardware list. |
| Number MAC addresses | The number of MAC addresses in the hardware list. |
| Dynamic Template | The name of a dynamic interface template associated with the policy. |
| Information about private and public interfaces | |
| Private Interface | The name of a private interface that is attached to the policy. If the interface is dynamic, the template name and username follow this entry. |
| Public Interface | The name of a public interface that is attached to the policy. If the interface is dynamic, the template name and username follow this entry. |
| Method | The method used to send packets to or from the public interface; either Dynamic (the firewall uses dynamic packet filtering) or Passall (the firewall does not interfere with packet flow). |
| NAT | Whether the type of NAT translation enabled is Standard or Enhanced. Only displayed when interface NAT is enabled on the policy. |
| 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 46-77 , and is only displayed when NAT has been enabled on the policy by using this command. |
| Private Interface | The private interface to which NAT translations apply. Displayed when NAT is enabled on the policy. |
| Global IP | The global IP address used by NAT translations. Displayed when NAT is enabled on the policy. |
| Interface information for policies that use a proxy | |
| Proxy | The type of proxy in use on the public interface. |
| Private Interface | The interface that receives packets from the private LAN. The proxy processes packets that arrive at this interface and sends them to the public interface. |
| Global IP | The public interface IP address that the proxy uses as the source address for traffic it sends to the public side. The destination address for sessions initiated from the public side, when the direction is "in" or "both". |

Table 46-12: Parameters in output of the **show firewall policy summary** command (cont.)

| Parameter | Meaning |
|-----------|---|
| 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. |
| After | The time of day after which the proxy is active. |
| Before | The time of day before which the proxy is active. |
| Days | The day or days on which the proxy is active. |

Examples To display counters and information about all firewall policies, use the command:

```
sh fire poli cou
```

Related Commands

- [add firewall policy rule](#)
- [create firewall policy](#)
- [delete firewall policy rule](#)
- [destroy firewall policy](#)
- [disable firewall policy](#)
- [enable firewall policy](#)
- [set firewall policy](#)
- [show firewall policy attack](#)
- [show firewall policy dynamic](#)
- [show firewall policy list](#)
- [show firewall policy maccache](#)
- [show firewall policy user](#)

show firewall policy attack

Syntax `SHoW FIREwaLL POLIcy[=policy-name] ATTack`

where *policy-name* is a 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 46-15](#), [Table 46-13](#)).

Figure 46-15: Example output from the **show firewall policy attack** command

| Policy : test | | | | |
|----------------------|------------|-------------|--------------------|----------|
| Current Attack Setup | | | | |
| Attack | In Trigger | Out Trigger | Time Period (mins) | Detailed |
| Attack Logged | | | | |
| ----- | | | | |
| dosflood | 80 | 160 | 2 | 5 |
| fragment | 1 | 1 | 2 | 0 |
| hostscan | 64 | 128 | 2 | 5 |
| ipspooft | 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 46-13: Parameters in 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 policy dynamic

Syntax `SHoW FIREwaLL POLIcy[=policy-name] DYnamic[=template]`

where:

- *policy-name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), and the underscore character.
- *template* is a 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 displays a list of the usernames assigned to the specified dynamic interface template or all dynamic interface templates in a policy (Figure 46-16, Table 46-14).

Figure 46-16: 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 46-14: Parameters in 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. |

Examples To display the users assigned to dynamic templates for a policy called “zone1”, use the command:

```
sh fire poli=zone1 dyn
```

Related Commands [add firewall policy dynamic](#)
[create firewall policy dynamic](#)
[delete firewall policy dynamic](#)
[show firewall policy](#)

show firewall policy limitrule

Syntax `SHoW FIREwaLL POLIcy=policy-name
LIMitrule[=rule-id[-rule-id]] [DETail]`

Description This command displays detailed information about the specified or all limit rules under the specified or all policies ([Figure 46-25 on page 46-195](#), [Table 46-22 on page 46-196](#)).

| Parameter | Description |
|-----------|--|
| POLICY | Name of the policy you wish to see the limit rule information for. |
| LIMitrule | Limit rule or range of limit rules to display. In no <i>rule-id</i> is specified, all limit rules for the policy are shown. |
| DETail | Displays a list of the devices that have active sessions matching the limit rule, and the number of sessions the device has active (Figure 46-18 on page 46-185 , Table 46-22 on page 46-196). |

Figure 46-17: Example output from the **show firewall policy limitrule** command

```
Policy=AT_Field
-----

Limitrule 1
-----
Interface ..... vlan2
IP ..... all
GBL Remote IP ..... all
Source IP Limit ..... 12

Limitrule 2
-----
Interface ..... all
IP ..... all
GBL Remote IP ..... all
Source IP Limit ..... 30
```


Figure 46-18: Example output from the **show firewall policy limitrule detail** command

```

Policy=Nerv_office
-----

Limitrule 1
-----
Interface ..... vlan1
IP ..... 202.36.164.113
GBL Remote IP ..... all
Source IP Limit ..... 1
-----

Per Source IP Count
Source IP Address      Active Sessions
202.36.164.113 ..... 1
-----

Limitrule 2
-----
Interface ..... all
IP ..... all
GBL Remote IP ..... all
Source IP Limit ..... 12
-----

Per Source IP Count
Source IP Address      Active Sessions
101.111.12.13 ..... 5
101.111.12.1 ..... 12
202.36.164.113 ..... 1

```

Table 46-15: Parameters in output of the **show firewall limitrule detail** command

| Parameter | Meaning |
|---------------------|--|
| Policy | Name of the policy that the limit rules apply to. |
| Limitrule | Rule identification number for the limit rule. |
| Interface | Interface that the rule applies to. |
| IP | IP address or address range of the private devices that sessions are limited for. |
| GBL Remote IP | IP address or address range of the public devices that sessions are limited for. |
| Source IP Limit | Maximum number of active sessions matching this limit rule that a device can have. |
| Per Source IP Count | Summary of any current matching sessions a device has for the limit rule. |
| Source IP Address | IP address of the device that initiated the session. |
| Active Sessions | Current number of active session initiated by the device. |

Examples To display the configuration of limit rule 1 of firewall policy “Nerv_office”, use the command:

```
sh fire poli=Nerv_office lim=1
```

Related Commands

- [add firewall policy limitrule](#)
- [delete firewall policy limitrule](#)
- [set firewall policy limitrule](#)

show firewall policy list

Syntax `SHoW FIREwaLL POLIcy[=policy-name] LISt`

where *policy-name* is a 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 address lists assigned to the specified policy or all policies (Figure 46-19, Table 46-16).

Figure 46-19: 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 46-16: Parameters in 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. |

Examples To display the lists used by the policy called “zone1”, use the command:

```
sh fire poli=zone1 lis
```

Related Commands [add firewall policy list](#)
[delete firewall policy list](#)
[show firewall policy](#)

show firewall policy maccache

Syntax `SHOW FIREwall POLIcy=policy-name MACCache`

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

Description This command displays information about the contents of the firewall MAC address cache (Figure 46-20, Table 46-17). Entries are added to the MAC address cache when information is received from a RADIUS server in response to MAC address queries. MAC address RADIUS queries are generated by firewall policy rules with **macradius** specified in the **list** parameter.

If a policy name is specified, only cache entries for queries generated by that policy are displayed. If no policy name is specified, cache entries for queries generated by all policies will be shown

Figure 46-20: Example output from the **show firewall policy maccache** command

| Policy : test - Cached MAC addresses | | | | |
|--------------------------------------|-----------|---------------|--------------|------------|
| MAC Address | Rule Type | RADIUS Result | Expiry (min) | Cache Hits |
| 00-00-cd-0b-8c-84 | Deny | Deny | 205 | 16 |
| 00-00-cd-00-ab-dc | Deny | Allow | 996 | 400 |
| 00-0a-17-29-11-91 | Allow | Allow | 360 | 98 |

Table 46-17: Parameters in output of the **show firewall policy maccache** command

| Parameter | Meaning |
|---|--|
| Policy | The name of the policy that created the entries in the MAC address cache |
| MAC Address | The Ethernet MAC address to which the cache entry applies |
| Rule Type | The action parameter of the rule that created this entry. "Deny" indicates that the entry was created by a rule with an action of deny . "Allow" indicates that the entry was created by a rule with an action of allow , nonat or nat . |
| RADIUS Result | The result of the RADIUS query; one of "Deny" or "Allow". This is the result that will be applied to new flows with a source MAC address that match this entry. |
| Expiry | The time, in minutes, for which this cache entry will remain valid. After this time the entry is deleted from the cache. |
| Note If the cache is full, entries may be expired before this time in order to make space for new entries. The oldest entries are deleted first. | |
| Cache Hits | The number of times this cache entry has been used since it was added to the MAC address cache. |

Examples To display the MAC address cache entries that were created by the “Lab” policy, use the command:

```
sh fire poli=lab macc
```

Related Commands [reset firewall policy maccache](#)
[set firewall policy](#)
[show firewall policy](#)

show firewall policy user

Syntax `SHoW FIREwall POLiCy[=policy-name] USer[=username]`

where *policy-name* is a 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 specified username or all usernames and the dynamic interface template(s) to which the username(s) are assigned (Figure 46-21, Table 46-18).

Figure 46-21: Example output from the **show firewall policy user** command

```
Policy : test

Dynamic template : acc1
  User : graeme
```

Table 46-18: Parameters in 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. |

Examples To display the dynamic templates that users are assigned to, use the command:

```
sh fire poli us
```

Related Commands [add firewall policy dynamic](#)
[delete firewall policy dynamic](#)
[show firewall policy](#)
[show firewall policy dynamic](#)

show firewall policy udpporttimeout

Syntax `SHoW FIREwaLL POLIcy[=policy-name] UDPPortttimeout`

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

Description This command displays information about any UDP ports on the firewall that are explicitly set with a UDP port timeout.

Figure 46-22: Example output from the **show firewall policy udpporttimeout** command

| Policy : test | |
|--|-------------|
| Default UDP Timeout (s) : 1200 | |
| Number of Configured UDP Port Timeouts : 5 | |
| UDP Port | Timeout (s) |
| ----- | |
| 5000 | 1800 |
| 5060 | 1800 |
| 6000 | 300 |
| 7000 | 2400 |
| 8000 | default |
| ----- | |

Table 46-19: Parameters in output of the **show firewall policy udpporttimeout** command

| Parameter | Meaning |
|--|---|
| Policy | The name of a policy. |
| Default UDP Timeout (s) | The length of time, in seconds, for which the firewall policy maintains inactive UDP sessions. This is also the amount of time, in seconds, for which UDP ports with a configured UDP port timeout of default remain inactive before the session times out. |
| Number of Configured UDP Port Timeouts | The number of ports in this policy that have a specific UDP port timeout value configured. |
| UDP Port | The UDP port numbers for which a specific UDP port timeout value is configured. |
| Timeout (s) | The amount of time, in seconds, for which UDP ports with a specific UDP port timeout may remain inactive before the session times out. If default is displayed, the port is using the Default UDP Timeout. |

Examples To display the UDP timeout information for the policy “area2”, use the command:

```
sh fire poli=area2 udpp
```

Related commands

- [add firewall policy udpporttimeout](#)
- [delete firewall policy udpporttimeout](#)
- [set firewall policy udpporttimeout](#)

show firewall session

Syntax `SHoW FIREwaLL SESSion[=session-number]
 [POLIcy=policy-name] [COUnTer] [IP=ipadd[-ipadd]]
 [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 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
- *port* is an Internet service port number or name.
- *service-name* is a pre-defined name for an IP service ([“Predefined IP protocol service names” on page 46-93](#)).
- *protocol* is an Internet IP protocol number.

Description This command displays information about the sessions and flows currently active for a specified policy, or for all policies ([Figure 46-23 on page 46-191](#)). 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 this parameter is not specified, session information for all policies is displayed.

If **counter** is specified, session counters for the specified policy are displayed.

The **ip** parameter specifies an IP address or a range of addresses. If you specify **ip**, sessions that involve that IP address are displayed. The firewall matches the specified IP address against the source and destination addresses of packets on both the private and public interfaces.

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 ([“Predefined IP protocol service names” on page 46-93](#)).

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 46-24 on page 46-193](#), [Table 46-20 on page 46-191](#)). 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 46-23: 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 46-20: Parameters in output of the **show firewall session** command

| Parameter | Meaning |
|-------------------------|--|
| Policy | The name of a policy. |
| <i>hex-num</i> | The session identifier |
| TCP/UDP/ <i>number</i> | 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. |
| 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 46-20: Parameters in output of the **show firewall session** command (cont.)

| Parameter | Meaning |
|-------------------------|---|
| TCP state | The status 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. |
| 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 46-24: 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 46-21: Parameters in output of the **show firewall session upnp** command

| Parameter | Meaning |
|------------------------|---|
| Policy | The name of a policy. |
| <i>hex-num</i> | The session identifier |
| TCP/UDP/ <i>number</i> | 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. |
| TCP state | The status of the TCP session: Free Closed Listen SynSent SynReceived Established FinWait1 FinWait2 CloseWait LastAck Closing TimeWait DeleteTCB SynSent SynReceived RADIUS query |

Table 46-21: Parameters in output of the **show firewall session upnp** command (cont.)

| Parameter | Meaning |
|---------------------|--|
| 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. |

Examples To display details about the active TCP sessions for the policy “area3”, use the command:

```
sh fire se poli=area3 prot=tcp
```

Related Commands

- [delete firewall session](#)
- [show firewall event](#)
- [show firewall monitor](#)
- [show firewall policy](#)

show firewall sipalg

Syntax SHow FIREwall SIPAlg

SHow FIREwall SIPAlg IP=*ipadd*[-*ipadd*]

SHow FIREwall SIPAlg CALLId=*call-id*

SHow FIREwall SIPAlg SUMmary

Description This command displays summary or detailed information for active SIP sessions using the SIP ALG (Figure 46-25, Table 46-22 on page 46-196).

| Parameter | Description |
|-----------|---|
| IP | Displays only the active sessions related to a specified IP address or range (Figure 46-25, Table 46-22 on page 46-196). This matches to both source and destination IP addresses. You can specify either a single IP address, or an IP address range. Use dotted decimal notation to specify each IP address. Default: no default |
| CALLId | Displays only the active session with the specified Call-ID (Figure 46-25, Table 46-22 on page 46-196). The Call-ID is a unique call identifier assigned to the SIP session by the device that initiated the session. Default: no default |
| SUMmary | Displays summary information for all the active sessions on the router (Figure 46-26 on page 46-197, Table 46-23 on page 46-197). |

Figure 46-25: Example output from the **show firewall sipalg** command

```
SIP ALG Configuration
  Status ..... Enabled
  Mode ..... Automatic
  Maximum automatic clients .... 50
  Multiple servers ..... No
  Call-ID translation ..... Enabled

Active SIP Sessions
-----
Call-ID .... 1536371071@198.18.1.2
TO ..... <sip:1234@20.20.20.1>
TO tag ..... 860468594
FROM ..... <sip:6789@20.20.20.1>
FROM tag ... 836088012
Direction .. Private to public
Audio Session[1]:
  (RTP)
    IP: 198.18.1.2:5010           Remote IP: 20.20.20.88:22984
    Gbl IP: 20.20.20.89:7280      Gbl Remote IP: 20.20.20.88:22984
    Start time ..... 10:04:24 22-Feb-2006
    Seconds to deletion ..... 1200
  (RTCP)
    IP: 198.18.1.2:5011           Remote IP: 20.20.20.88:22985
    Gbl IP: 20.20.20.89:7281      Gbl Remote IP: 20.20.20.88:22985
    Start time ..... 10:04:24 22-Feb-2006
    Seconds to deletion ..... 576
-----
```

Table 46-22: Parameters in output of the **show firewall sipalg** command

| Parameter | Meaning |
|---------------------------|---|
| SIP ALG Configuration | The current SIP ALG settings on the router. |
| Status | Whether the SIP ALG is "enabled" or "disabled" on the router. |
| Mode | Whether the SIP ALG is in "automatic" or "manual" client management mode. |
| Maximum automatic clients | Maximum number of clients that the SIP ALG is configured to support when the SIP ALG is in automatic client management mode. |
| Multiple servers | How the SIP ALG interacts with sessions initiated to and from SIP Proxy Servers that are independent of the SIP Registrar when the SIP ALG is in automatic client management mode. One of "Yes", "No" and "Outonly". |
| CALL-ID translation | Whether Call-ID translation is "enabled" or "disabled" on router. When enabled, the IP address portion of the Call-ID field is translated from a private IP address to the global, routable IP address of router. The router only translates this when the session is initiated by a device within the private network protected by the firewall. |
| Active SIP Sessions | Details about current SIP sessions using the SIP ALG, including information about the current audio sessions for each SIP session. |
| CALL-ID | Unique call identifier assigned to the SIP session by the device that initiated the session. The Call-ID includes the IP address of the device that initiated the SIP session. |
| TO | SIP URI address of the device that received the SIP session request. |
| TO tag | Tag number assigned to the SIP session by the device that received the SIP session request. The router uses this, along with the FROM tag and the Call-ID, to identify a current SIP session. |
| FROM | SIP URI address of the device that initiated the SIP session request. |
| FROM tag | Tag number assigned to the SIP session by the device that initiated the SIP session request. The router uses this, along with the TO tag and Call-ID, to identify a current SIP session. |
| Direction | Location of the devices using the SIP session, and who initiated the call. "Private" indicates a device located behind the firewall, "public" indicates the device located outside of the firewall. The device that initiated the call is listed first. For example, "Private to public" indicates that a device from behind the firewall initiated a SIP session to a device on the public side of the firewall. |
| Audio Session | Details about the current audio sessions using the SIP session. The number in brackets indicates the direction of the call; [1] is private to public, and [2] is public to private. |
| RTP | Details about the Real-time Transport Protocol (RTP). RTP carries the audio data. |
| RTCP | Details about the Real-time Transport Control Protocol (RTCP). RTCP provides feedback to applications about RTP's quality of service. |

Table 46-22: Parameters in output of the **show firewall sipalg** command (cont.)

| 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. |
| Start time | Date and time that the session was started. |
| Seconds to deletion | Number of seconds remaining before the session is automatically deleted. |

Figure 46-26: Example output from the **show firewall sipalg summary** command

```
SIP ALG Configuration
Status ..... Enabled
  Mode ..... Automatic
  Maximum automatic clients .... 50
  Multiple servers ..... No
  Call-ID translation ..... Enabled

Active SIP Sessions
-----
Index  Start time                From
      Call-ID                  To
      Direction
-----
  1  12:12:37 22-Feb-2006      <sip:6789@20.20.20.1>
      1874680886@198.18.1.2    <sip:1234@20.20.20.1>
      private to public
  2  12:15:11 22-Feb-2006      <sip:3456@20.20.20.1>
      1721829112@202.12.9.172  <sip:1982@20.20.20.1>
      public to private
-----
```

Table 46-23: Parameters in output of the **show firewall sipalg summary** command

| Parameter | Meaning |
|---------------------------|--|
| SIP ALG Configuration | The current SIP ALG settings on the router. |
| Status | Whether the SIP ALG is "enabled" or "disabled" on the router. |
| Mode | Whether the SIP ALG is in "automatic" or "manual" client management mode. |
| Maximum automatic clients | Maximum number of clients that the SIP ALG is configured to support. Valid only when the SIP ALG is in automatic client management mode. |

Table 46-23: Parameters in output of the **show firewall sipalg summary** command

| Parameter | Meaning |
|---------------------|---|
| Multiple servers | How the SIP ALG interacts with sessions initiated to and from SIP Proxy Servers that are independent of the SIP Registrar, one of "Yes", "No" and "Outonly". Valid only when the SIP ALG is in automatic client management mode. |
| CALL-ID translation | Whether the IP address portion of the Call-ID is translated from a private IP address to the global, routable IP address of router. The router only translates IP addresses originating from the private network protected by the firewall. |
| Active SIP Sessions | Summary output of all SIP sessions that are active through the firewall. |
| Index | List number assigned to each SIP session. Used for this list only. |
| Start time | Date and time that the session was started. |
| Call-ID | The unique call identifier assigned to the SIP session by the device that initiated the session. The Call-ID includes the IP address of the device that initiated the SIP session. |
| Direction | The location of the devices using the SIP session, and who initiated the call. "Private" indicates a device located within the firewall, "public" indicates the device located outside of the firewall. The device that initiated the call is listed first. For example, "Private to public" indicates that a device from within the firewall initiated a SIP session with a device on the public side of the firewall. |
| From | The SIP URI address of the device that initiated the SIP session request. |
| To | The SIP URI address of the device that received the SIP session request. |

Examples To display any SIP sessions using the SIP ALG within the IP range 192.168.1.2 to 192.168.1.8, use the command:

```
show fire sipa ip=192.168.1.2-192.168.1.8
```

Related Commands

- [disable firewall sipalg](#)
- [enable firewall sipalg](#)
- [reset firewall sipalg counter](#)
- [set firewall sipalg](#)
- [show firewall sipalg counter](#)

show firewall sipalg autoclients

Syntax `SHoW FIREwaLL SIPAlg AUTOclients [=session-number]
[SUMmary]`

`SHoW FIREwaLL SIPAlg AUTOclients IP=ipadd [-ipadd]
[SUMmary]`

Description This command displays the client database details collected by the SIP ALG when in automatic client management mode ([Figure 46-27](#), [Table 46-24 on page 46-200](#)).

| Parameter | Description |
|-------------|--|
| AUTOclients | Displays the client database details. The <i>session-number</i> is an identifier assigned to an active SIP session, and if specified, only the details for that session are displayed. Specifying a session number is not valid when the ip parameter is specified. |
| IP | Displays only the active sessions related to a specified IP address or range. This matches to the source address for private devices only. You can specify either a single IP address, or an IP address range, in dotted decimal notation. Default: no default |
| SUMmary | Displays summary information for all the active sessions on the firewall. If a session number or the ip parameter is specified, then the summary details are filtered according to those parameters (Figure 46-28 on page 46-200 , Table 46-24 on page 46-200). |

Figure 46-27: Example output from the **show firewall sipalg autoclients** command

```
SIP ALG Automatic Clients
-----
Automatic client file ..... fwsipalg.sip
Number of clients ..... 2
Last updated ..... 10:11:55 4-Jul-2006
Update pending ..... No
Active clients
Number of active clients ... 2
Last updated ..... 12:38:23 4-Jul-2006

Active Automatic Clients
-----
Session number ..... 2131
SIP client IP:Port ..... 192.168.1.2:5060
Gbl IP:Gbl port ..... 20.20.20.89:22984
SIP registrar IP:Port ..... 20.20.20.88:5060
First registration time ..... 10:04:24 4-Jul-2006
Seconds to expiry ..... 2436
Session number ..... 2fbc
SIP client IP:Port ..... 192.168.1.3:5060
Gbl IP:Gbl port ..... 20.20.20.89:4132
SIP registrar IP:Port ..... 20.20.20.88:5060
First registration time ..... 10:11:44 4-Jul-2006
Seconds to expiry ..... 3214
-----
```

Figure 46-28: Example output from the **show firewall sipalg autoclients summary** command

| | | | |
|---------------------------|---------------------|---------------------|------------------|
| SIP ALG Automatic Clients | | | |
| ----- | | | |
| Automatic client file | | fwsipalg.sip | |
| Number of clients | | 2 | |
| Last updated | | 10:11:55 4-Jul-2006 | |
| Update pending | | No | |
| Active clients | | | |
| Number of active clients | ... | 2 | |
| Last updated | | 12:38:23 4-Jul-2006 | |
| Active Automatic Clients | | | |
| ----- | | | |
| Session | SIP client IP:Port. | Gbl IP:Gbl port. | SIP registrar |
| ----- | | | |
| 2131 | 192.168.1.2:5060 | 20.20.20.89:22984 | 20.20.20.88:5060 |
| 9fbc | 192.168.1.3:5060 | 20.20.20.89:4132 | 20.20.20.88:5060 |
| ----- | | | |

Table 46-24: Parameters in the output of the **show firewall sipalg autoclients** command

| Parameter | Meaning |
|--------------------------------------|--|
| Automatic client file | Name of the client database file saved on flash memory. This static version of the database allows the SIP ALG to recover client sessions in case of a router restart or reboot. |
| Number of clients | Number of clients stored in the file on flash memory. |
| Last updated | Time and date of the last update to the file on flash memory. This file is updated regularly from the dynamic client database stored on RAM. |
| Update pending | Whether the file on flash needs updating. "Yes" indicates that the dynamic version has changed, and the static version is scheduled to be updated. "No" indicates that the dynamic and static versions of the database are identical and there is no need for the router to update the static version. |
| Active clients | Details about the clients the SIP ALG has listed in its client database. This includes any SIP client currently registered with a SIP Registrar as well as clients with calls in progress. |
| Number of active clients | Number of active SIP clients. This number is obtained from the dynamic version of the client database, so if an update is pending this number may be different from the number of clients listed for the static version of the database. |
| Last updated | Time and date of the last event that changed the client database details. |
| Session number, Session | An identifier for the session assigned by the firewall. |
| SIP client IP:Port | Private IP address and UDP source port used by the client. |
| Gbl IP:Gbl port | Public IP address and UDP port that the SIP ALG has assigned to the client, when NAT is configured. |
| SIP registrar IP:Port, SIP registrar | IP address of the SIP Registrar that the client has registered with. The port is the UDP port for SIP. |

Table 46-24: Parameters in the output of the **show firewall sipalg autoclients** command

| Parameter | Meaning |
|-------------------------|---|
| First registration time | Time and date that the client first registered with the SIP Registrar using this session. The same session is used each time the SIP client re-registers with the SIP Registrar, unless the session expires. The session should not expire unless the client does not re-register with the SIP Registrar within the expiry time limit set by the registrar. |
| Seconds to expiry | Time remaining until the client's existing registration expires with the SIP Registrar. The session is deleted if the client does not re-register before this time runs out. |

Example To display information about every SIP client currently managed by the SIP ALG, use the command:

```
sh fire sipa auto
```

Related Commands

- [disable firewall sipalg](#)
- [enable firewall sipalg](#)
- [reset firewall sipalg autoclients](#)
- [set firewall sipalg](#)
- [show firewall sipalg](#)

show firewall sipalg counter

Syntax SHow FIREwall SIPAlg COUnter

Description This command displays counters related to SIP sessions that have used or are using the SIP ALG on the router.

Figure 46-29: Example output from the **show firewall sipalg counter** command

```
SIP ALG Session Counters
-----
Current SIP sessions ..... 1
Current audio sessions ..... 2
SIP sessions created since start up or reset ..... 6
Audio sessions created since start up or reset ..... 10
SIP messages received since start up or reset ..... 102
SIP messages ignored since start up or reset ..... 0
-----
```

Table 46-25: Parameters in output of the **show firewall sipalg counter** command

| Parameter | Meaning |
|--|---|
| Current SIP sessions | Number of active SIP sessions using the SIP ALG. |
| Current audio sessions | Number of active audio sessions travelling through the firewall. |
| SIP sessions created since start up or reset | Total number of SIP sessions created, including both past and current sessions. |
| Audio sessions created since start up or reset | Total number of audio sessions created, including both past and current sessions. |
| SIP messages received since start up or reset | Total number of SIP messages received, including those from past sessions. |
| SIP messages ignored since start up or reset | Total number of SIP messages received that the SIP ALG ignored because the message was an unsupported type. These messages are forwarded without the SIP ALG altering them. |

Examples To display counters for the SIP ALG's activity on the router, use the command:

```
show fire sipa cou
```

Related Commands

- [disable firewall sipalg](#)
- [enable firewall sipalg](#)
- [reset firewall sipalg counter](#)
- [set firewall sipalg](#)
- [show firewall sipalg](#)