



**FortiGate IPv6 support
FortiOS v3.0 MR7**



www.fortinet.com

FortiGate IPv6 support Technical Note

3 October 2008

01-30007-0081-20081003

© Copyright 2008 Fortinet, Inc. All rights reserved. No part of this publication including text, examples, diagrams or illustrations may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical or otherwise, for any purpose, without prior written permission of Fortinet, Inc.

Trademarks

Dynamic Threat Prevention System (DTPS), APSecure, FortiASIC, FortiBIOS, FortiBridge, FortiClient, FortiGate, FortiGate Unified Threat Management System, FortiGuard, FortiGuard-Antispam, FortiGuard-Antivirus, FortiGuard-Intrusion, FortiGuard-Web, FortiLog, FortiAnalyzer, FortiManager, Fortinet, FortiOS, FortiPartner, FortiProtect, FortiReporter, FortiResponse, FortiShield, FortiVoIP, and FortiWiFi are trademarks of Fortinet, Inc. in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

| | |
|---|-----------|
| IPv6 overview | 5 |
| Comparison of IPv4 to IPv6 | 5 |
| Overview of IPv6 address space | 6 |
| IPv6 address format | 6 |
| IP address notation | 7 |
| Netmasks | 8 |
| Address scopes | 8 |
| Address types | 8 |
| Unicast | 8 |
| Multicast | 8 |
| Anycast | 9 |
| Special addresses | 9 |
| IPv6 neighbor discovery | 10 |
| Transition from IPv4 to IPv6 | 11 |
| IPv4 addresses in IPv6 format | 12 |
| IPv4 - compatible IPv6 addresses | 12 |
| IPv4 - mapped IPv6 addresses | 12 |
| IPv6 tunneling | 13 |
| IPv6 ping6 command | 13 |
| IPv6 ping description | 14 |
| IPv6 ping options | 15 |
| Examples | 16 |
| Additional IPv6 resources | 17 |
| FortiGate IPv6 configuration | 19 |
| Configuring IPv6 interfaces | 20 |
| Adding an IPv6 address to an interface | 20 |
| Creating the prefix list for the interface | 21 |
| Configuring IPv6 routing | 22 |
| Configuring static routing | 22 |
| Configuring IPv6 router advertisements | 23 |
| Testing connections with ping6 | 24 |
| Configuring IPv6 over IPv4 tunneling | 24 |
| Creating the IPv6 tunnel | 25 |
| Defining the firewall policies | 25 |
| Defining routing | 26 |
| Configuring IPv6 IPsec VPNs | 27 |
| Certificates | 27 |
| Phase 1 configuration | 27 |
| Phase 2 configuration | 28 |
| Firewall policies | 28 |
| Routing | 28 |

Configuring IPv6 firewall policies 29

IPv6 overview

Internet Protocol version 6 (IPv6) is an Internet Layer protocol for packet-switched internetworks that has been designed to provide several advantages over Internet Protocol version 4 (IPv4). The Internet Engineering Task Force (IETF) has designated IPv6 as the successor of IPv4 for general use on the Internet. Both IPv6 and IPv4 define network layer protocol (how data is sent from one computer to another over packet-switched networks), but IPv6 has a much larger address space than IPv4 - it can provide billions more unique IP addresses.

This technical note describes IPv6 addressing support on all 01-30007-82573-20081003 units using firmware version 3.0 MR5 or later. This document has two chapters:

- [IPv6 overview](#), this chapter, provides some background information about IPv6 addressing and how networks are making the transition from IPv4 to IPv6 addressing.
- [FortiGate IPv6 configuration](#) provides information about the IPv6 features of FortiGate units and how to configure them.

Comparison of IPv4 to IPv6

The changes from IPv4 to IPv6 can be categorized as follows:

Larger address space

IPv4 addresses are 32 bits long while IPv6 addresses are 128 bits long. This increase supports 2^{128} addresses, or more than ten billion billion billion times as many addresses as IPv4 (2^{32}). IPv6 enables more levels of addressing hierarchy and simplifies auto-configuration of IP addresses. The IPv6 addressing scheme eliminates the need for Network Address Translation (NAT) that causes networking problems due to the end-to-end nature of the Internet, such as hiding multiple hosts behind a pool of IP addresses.

Simplified header formats

The IPv6 header format either drops or makes optional certain IPv4 header fields. This limits the bandwidth cost of the IPv6 header - even though the IPv6 addresses are four times longer than the IPv4 addresses, the IPv6 header is only twice the size of the IPv4 header.

Improved support for IP header options

Changes in the way IP header options are encoded allows for more efficient forwarding and less stringent limits on the length of options. The changes also provide greater flexibility for introducing new options in the future.

Prioritization of packet delivery using flow labeling

The IPv6 packet header contains a new "Flow Label" field that allows the sender to request special handling, such as "real-time service" or non-default quality of service. The "Flow Label" field replaces "Service Type" field in IPv4.

Supported authentication

IPv6 extensions support authentication, data integrity, and (optional) data confidentiality.

Overview of IPv6 address space

IPv6 addresses are assigned to interfaces rather than nodes, thereby recognizing that a node can have more than one interface, and you can assign more than one IPv6 address to an interface. In addition, the larger address space in IPv6 addresses allows flexibility in allocating addresses and routing traffic, and simplifies some aspects of address assignment and renumbering when changing Internet service providers.

With IPv4, complex Classless Inter-Domain Routing (CIDR) techniques were developed to make the best use of the small address space. CIDR facilitates routing by allowing blocks of addresses to be grouped together into a single routing table entry. With IPv4, renumbering an existing network for a new connectivity provider with different routing prefixes is a major effort (see RFC 2071, *Network Renumbering Overview: Why would I want it and what is it anyway?* and RFC 2072, *Router Renumbering Guide*). With IPv6, however, it is possible to renumber an entire network ad hoc by changing the prefix in a few routers, as the host identifiers are decoupled from the subnet identifiers and the network provider's routing prefix.

The size of each subnet in IPv6 is 264 addresses (64 bits), which is the square of the size of the entire IPv4 Internet. The actual address space utilized by IPv6 applications will most likely be small in IPv6, but both network management and routing will be more efficient.

Maximum Transmission Unit (MTU) refers to the size (in bytes) of the largest packet or frame that a given layer of a communications protocol can pass onwards. A higher MTU brings higher bandwidth efficiency. IPv6 requires an MTU of at least 1280 bytes. With encapsulations (for example, tunneling), an MTU of 1500 or more is recommended. For more information, see RFC-2640, *Internationalization of the File Transfer Protocol*.

IPv6 address format

The IPv6 address is 128 bits long and consists of eight, 16-bit fields. Each field is separated by a colon and must contain a hexadecimal number. In [Figure 1](#), an X represents each field.

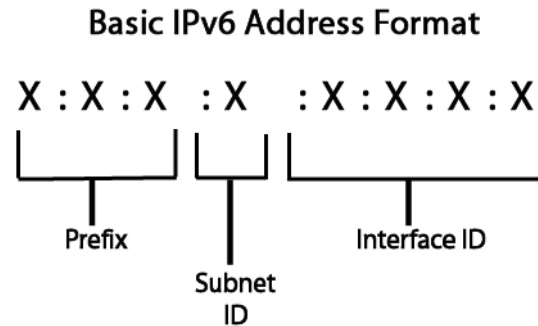
The IPv6 address is made up of two logical parts:

- 64-bit (sub)network prefix
- 64-bit host

The (sub)network prefix part contains the site prefix (first three fields, 48 bits) and the subnet ID (next two fields, 16-bits), for a total of 64-bits. The information contained in these fields is used for routing IPv6 packets. The (sub)network prefix defines the site topology to a router by specifying the specific link to which the subnet has been assigned. The site prefix details the public topology allocated (usually by an Internet Service Provider, ISP) to your site. The subnet ID details the private topology (or site topology) to a router that you assign to your site when you configure your IPv6 network.

The host part consists of the interface ID (or token) which is 64-bits in length and must be unique within the subnet. The length of the interface ID allows for the mapping of existing 48-bit MAC addresses currently used by many local area network (LAN) technologies such as Ethernet, and the mapping of 64-bit MAC addresses of IEEE 1394 (FireWire) and other future LAN technologies. The host is either configured automatically from the MAC address of the interface, or is manually configured.

Figure 1: IPv6 Address Format



IP address notation

IPv6 addresses are normally written as eight groups of 4 hexadecimal digits each, separated by a colon, for example:

2001:db8:3c4d:0d82:1725:6a2f:0370:6234

is a valid IPv6 address.

There are several ways to shorten the presentation of an IPv6 address. Most IPv6 addresses do not occupy all of the possible 128 bits. This results in fields that are “padded” with zeros or contain only zeros. If a 4-digit group is 0000, it may be replaced with two colons (::), for example:

2001:db8:3c4d:0000:1725:6a2f:0370:6234

is the same IPv6 address as:

2001:db8:3c4d::1725:6a2f:0370:6234

Leading zeroes in a group may be omitted, for example (in the address above):

2001:db8:3c4d::1725:6a2f:370:6234

The double colon (::) must only be used once in an IP address, as multiple occurrences lead to ambiguity in the address translation.

Examples of shortened IP address presentations:

19a4:0478:0000:0000:0000:0000:1a57:ac9e

19a4:0478:0000:0000:0000::1a57:ac9e

19a4:478:0:0:0:0:1a57:ac9e

19a4:478:0::0:1a57:ac9e

19a4:478::1a57:ac9e

All of these address presentations are valid and represent the same address.

For IPv4-compatible or IPv4-mapped IPv6 addresses (see [“Address types” on page 8](#)), you can enter the IPv4 portion using either hexadecimal or dotted decimal, but the FortiGate CLI always shows the IPv4 portion in dotted decimal format. For all other IPv6 addresses, the CLI accepts and displays only hexadecimal.

Netmasks

As with IP addresses, hexadecimal notation replaces the dotted decimal notation of IPv4. IPv4 Classless Inter-Domain Routing (CIDR) notation can also be used. This notation appends a slash (“/”) to the IP address, followed by the number of bits in the network portion of the address.

Table 1: IPv6 address notation

| | |
|------------------------|--|
| IP Address | 3ffe:ffff:1011:f101:0210:a4ff:fee3:9566 |
| Netmask | ffff:ffff:ffff:ffff:0000:0000:0000:0000 |
| Network | 3ffe:ffff:1011:f101:0000:0000:0000:0000 |
| CIDR IP/Netmask | 3ffe:ffff:1011:f101:0210:a4ff:fee3:9566/64 |

Address scopes

Address scopes define the region where an address may be defined as a unique identifier of an interface. The regions are: local link (link-local), site network (site-local), and global network. Each IPv6 address can only belong to one zone that corresponds to its scope.

Address types

IPv6 addresses are classified into three groups - [Unicast](#), [Multicast](#), and [Anycast](#).

Unicast

Identifies an interface of an individual node. Packets sent to a unicast address are sent to that specific interface. Unicast IPv6 addresses can have a scope reflected in more specific address names - global unicast address, link-local address, and unique local unicast address. For more information, see [“Global \(Unicast\)” on page 10](#), [“Link-local \(Unicast\)” on page 10](#), and [“Site-local \(Unicast\)” on page 10](#).

Multicast

Assigned to a group of interfaces that typically belong to different nodes. A packet that is sent to a multicast address is delivered to all interfaces identified by the address. Multicast addresses begin with the first octet one (1) bit. The four least significant bits of the second address octet identify the address scope or the span over which the multicast address is propagated. IPv6 multicast addresses have functionally replaced IPv4 broadcast addresses.

Anycast

Assigned to a group of interfaces usually belonging to different nodes. A packet sent to an anycast address is delivered to just one of the member interfaces, typically the 'nearest' according to the router protocols' choice of distance. They cannot be identified easily as their structure is the same as a normal unicast address, differ only by being injected into the routing protocol at multiple points in the network. When a unicast address is assigned to more than one interface (making it an anycast address), the address assigned to the nodes must be configured in such a way as to indicate that it is an anycast address.

Interfaces configured for IPv6 must have at least one link-local unicast address and additional ones for site-local or global addressing. Link-local addresses are often used in network address autoconfiguration where no external source of network addressing information is available.

Special addresses

The following are IPv6 special addresses:

- [Unspecified](#)
- [Loopback](#)

For more information about IPv6 addresses, see [RFC 3513, Internet Protocol version 6 \(IPv6\) Addressing Architecture](#).

Table 2: IPv6 addresses with prefix information

| Address Type | IPv6 notation Prefix/prefix length | Details |
|-----------------|---------------------------------------|--|
| Unspecified | ::/128 | Indicates the absence of an address, so must never be assigned to any node. Must not be used as a source address for IPv6 router, destination address of IPv6 packets, or in IPv6 routing headers. Equivalent to 0.0.0.0 in IPv4. |
| Loopback | ::1/128 | Used as a node to send an IPv6 packet to itself. Seen as link-local unicast address of a virtual interface (loopback interface) to an imaginary link that goes nowhere. Must never be assigned to a physical interface, or as the source address of IPv6 packets that are sent outside of the single node. IPv6 destination address of loopback should not be sent outside a single node, and never forwarded by an IPv6 router. Equivalent to 127.0.0.1 in IPv4. |
| IPv4-compatible | ::/96 | Lowest 32 bits can be in IPv6 hexadecimal or IPv4 dotted decimal format. |
| IPv4-mapped | ::FFFF/96 | Lowest 32 bits can be in IPv6 hexadecimal or IPv4 dotted decimal format. |

Table 2: IPv6 addresses with prefix information

| Address Type | IPv6 notation Prefix/prefix length | Details |
|----------------------|--|---|
| 6to4 | 2002::/16 | Used for communication between two nodes running both IPv4 and IPv6 over the Internet. Formed by combining the IPv6 prefix with the 32-bits of the public IPv4 address of the node, creating a 48-bit address prefix. |
| Multicast | ::FF00/8 | For more information, see “Multicast” on page 8 . |
| Anycast | All prefixes except those listed above | For more information, see “Anycast” on page 9 . |
| Link-local (Unicast) | FE80::/10 | Used for addressing on a single link for automatic address configuration, neighbor discovery, or when no routers are present. Routers must not forward packets with link-local source or destination addresses. |
| Site-local (Unicast) | FEC0::/10 | Used for addressing inside of a site without needing a global prefix. Routers must not forward packets with site-local source or destination addresses outside of the site. |
| Global (Unicast) | all other prefixes | Equivalent to public IPv4 addresses. Globally routable and reachable on the IPv6 internet. Addresses are designed to be summarized or aggregated to create an efficient router infrastructure. |

IPv6 neighbor discovery

IPv6 Neighbor Discovery (ND) is a set of messages and processes that determine relationships between neighboring nodes. Neighboring nodes are on the same link. The IPv6 ND protocol replaces the IPv4 protocols Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMPv4), Router Discovery (RDISC), and ICMP Redirect, and provides additional functionality. The IPv6 ND protocol facilitates the autoconfiguration of IPv6 addresses. Autoconfiguration is the ability of an IPv6 host to automatically generate its own IPv6 address, making address administration easier and less time-consuming.

Hosts use ND to:

- discover addresses, address prefixes, and other configuration parameters
- discover neighboring routers.

Routers use ND to:

- advertise their presence, host configuration parameters, and on-link prefixes
- inform hosts of ‘better’ next-hop address to forward packets for a specified destination.

Nodes use ND to:

- resolve link-layer address of a neighboring node to which an IPv6 packet is being forwarded and determine whether the link-layer address of a neighboring node has altered
- determine whether IPv6 packets can be sent to and received from a neighbor
- automatically configure IPv6 addresses for its interfaces.

To facilitate neighbor discovery, routers periodically send messages advertising their availability. This communication includes lists of the address prefixes for destinations available on each router's interfaces.

ND defines five different Internet Control Message Protocol (ICMP) packet types: a pair of Neighbor Solicitation and Neighbor Advertisement messages, a pair of Router Solicitation and Router Advertisement messages, and a Redirect message.

A Neighbor Solicitation is sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Also used for Duplicate Address Detection (how a node determines that an address it wants to use is not already in use by another node). The Neighbor Advertisement message is a response to a Neighbor Solicitation message. A node may also announce a link-layer address change by sending unsolicited Neighbor Advertisements.

A host may send a Router Solicitation when an interface becomes enabled, requesting routers to generate a Router Advertisement immediately rather than at their next scheduled time.

Routers advertise their presence together with various link and Internet parameters according to a specific schedule or in response to a Router Solicitation message. A Router Advertisement contains prefixes used for on-link determination and/or address configuration, a suggested hop limit value, etc.

The Redirect message is used by routers to inform hosts of a better first-hop for a destination.

For more information, see [RFC 2461, Neighbor Discovery for IP Version 6 \(IPv6\)](#).

Transition from IPv4 to IPv6

If the Internet is to take full advantage of the benefits of IPv6, there must be a period of transition to enable IPv6-only hosts to reach IPv4 services and to allow isolated IPv6 hosts and networks to reach the IPv6 Internet over the IPv4 infrastructure.

RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers* and RFC 2185, *Routing Aspects of IPv6 Transition* define several mechanisms to ensure that IPv6 hosts and routers maintain interoperability with the existing IPv4 infrastructure, and facilitate a gradual transition that does not impact the functionality of the Internet. The mechanisms, known collectively as Simple Internet Transition (SIT), include:

- dual-stack IP implementations for hosts and routers that must interoperate between IPv4 and IPv6
- embedding of IPv4 addresses in IPv6 addresses. IPv6 hosts are assigned addresses that are interoperable with IPv4, and IPv4 host addresses are mapped to IPv6
- IPv6-over-IPv4 tunneling mechanisms to encapsulate IPv6 packets within IPv4 headers to carry them over IPv4 infrastructure
- IPv4/IPv6 header translation, used when implementation of IPv6 is well-advanced and few IPv4 systems remain.

FortiGate units are dual IP layer IPv6/IPv4 nodes and they support IPv6 over IPv4 tunneling.

For more information, see [RFC 2893, Transition Mechanisms for IPv6 Hosts and Routers](#) and [RFC 2185, Routing Aspects of IPv6 Transition](#).

IPv4 addresses in IPv6 format

There are two ways that IPv4 addresses are represented in IPv6 format. In both cases, the first 80 bits are always zero, indicating an embedded IPv4 address of some kind. You can distinguish between the two types by the 16 bits that precede the IPv4 portion of the address:

Table 3: IPv6 formats for IPv4 addresses

| | | | |
|-------------------------------------|---------------------------------------|-------|---------------------------------|
| IPv4-compatible IPv6 address | 0000:0000:0000:0000:0000: or :: | 0000: | 874B:2B34 or 135.75.43.52 |
| IPv4-mapped IPv6 address | 0000:0000:0000:0000:0000: or :: | FFFF: | 874B:2B34 or 135.75.43.52 |

IPv4 - compatible IPv6 addresses

IPv4 - compatible addresses, derived from IPv4 public addresses, are used for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. Used in addition to the conventional IPv6 address for devices compatible with IPv4 and IPv6, the IPv4 - compatible IPv6 address begins with 96 zero bits, followed by 32 bits that represent the IPv4 address. These addresses help the migration process by enabling IPv6 features without the use of IPv6 routers.

IPv4 - mapped IPv6 addresses

IPv4 - mapped addresses map an IPv4 device that does not support IPv6 into the IPv6 address space. The address begins with 80 zeroes followed by 16 ones, and the 32-bits that represent the IPv4 address. The “FFFF” after the initial 80 indicates a conventional IPv4 device whose address has been mapped into the IPv6 format.

IPv6 tunneling

While the IPv6 routing infrastructure is being deployed, the existing IPv4 routing infrastructure must remain functional and also carry IPv6 traffic. Tunneling provides a method of utilizing the existing IPv4 routing infrastructure to carry IPv6 traffic.

Networks using IPv6 addressing can be linked through IPv4-addressed infrastructure using several tunneling techniques:

Table 4: Tunneling techniques

| | |
|-----------------------|---|
| IPv6-over-IPv4 | Encapsulates IPv6 packets within IPv4 so that they can be carried across IPv4 routing infrastructures. |
| Configured | The tunnel endpoint address is determined from configuration information on the encapsulating node. The tunnel endpoint address is used as the destination address for the IPv4 header. |
| Automatic | The IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet being tunneled. |
| IPv4 multicast | IPv4 tunnel endpoint address is determined using Neighbor Discovery. No address configuration is required, but the IPv4 infrastructure must support IPv4 multicast. |

FortiGate units support IPv6-over-IPv4 tunneling.

For the period while IPv6 hosts and routers co-exist with IPv4, a number of transition mechanisms are needed to enable IPv6-only hosts to reach IPv4 services and to allow isolated IPv6 hosts and networks to reach the IPv6 Internet over the IPv4 infrastructure.

These techniques, collectively called Simple Internet Transition (SIT), include:

- dual-stack IP implementations for interoperating hosts and routers
- embedding IPv4 addresses in IPv6 addresses
- IPv6-over-IPv4 tunneling mechanisms
- IPv4/IPv6 header translation

IPv6 ping6 command

You can use the IPv6 ping command to:

- Send an ICMP echo request packet to the IPv6 address that you specify.
- Specify a source interface other than the one from which the probe originates by using the source interface keywords.
- Specify a source IP address other than the one from which the probe originates by using the source address keywords

You can specify the following options:

| | |
|-----------------------------------|--|
| packetCount | Number of packets to send to the destination IPv6 address. If you specify a zero, echo requests packets are sent indefinitely. |
| data-pattern | Sets the type of bits contained in the packet to all ones, all zeros, a random mixture of ones and zeros, or a specific hexadecimal data pattern that can range from 0x0 to 0xFFFFFFFF. The default is all zeros. |
| extended header attributes | Set the interface type and specifier of a destination address on the system that is configured for external loopback; the command succeeds only if the specified interface is configured for external loopback. |
| sweep interval | Specifies the change in the size of subsequent ping packets while sweeping across a range of sizes. For example, you can configure the sweep interval to sweep across the range of packets from 100 bytes to 1000 bytes in increments specified by the sweep interval. By default, the system increments packets by one byte; for example, it sends 100, 101, 102, 103, ... 1000. If the sweep interval is 5, the system sends 100, 105, 110, 115, ... 1000. |
| sweep sizes | Enables you to vary the sizes of the echo packets being sent. Used to determine the minimum sizes of the MTUs configured on the nodes along the path to the destination address. This reduces packet fragmentation, which contributes to performance problems. The default is to not sweep (all packets are the same size). |
| timeout | Sets the number of seconds to wait for an ICMP echo reply packet before the connection attempt times out. |
| hop limit | Sets the time-to-live hop count in the range 1-255; the default is 255. |

The following characters may appear in the display after the ping command is issued:

- ! - reply received
- . - timed out while waiting for a reply
- ? - unknown packet type
- A - admin unreachable
- b - packet too big
- H - host unreachable
- N - network unreachable
- P - port unreachable
- p - parameter problem
- S - source beyond scope
- t - hop limit expired (TTL expired)

IPv6 ping description

Ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of "pad" bytes used to fill out the packet.

IPv6 ping options

| | |
|-----------------------------|---|
| -a | Audible ping. |
| -A | Adaptive ping. Interpacket interval adapts to round-trip time, so effectively no more than one (or more, if preload is set) unanswered probe is present in the network. Minimal interval is 200msec for any user other than administrator. On networks with low rtt this mode is essentially equivalent to flood mode. |
| -b | Allow pinging of a broadcast address. |
| -B | Do not allow ping to change source address of probes. The address is bound to one selected when the ping starts. |
| -c count | Stop after sending count ECHO_REQUEST packets. With deadline option, ping waits for count ECHO_REPLY packets, until the timeout expires. |
| -d | Set the SO_DEBUG option on the socket being used. This socket option is not used by a Linux kernel. |
| -F flow label | Allocate and set 20 bit flow label on echo request packets (only ping6). If value is zero, kernel allocates random flow label. |
| -f | Flood ping. For every ECHO_REQUEST sent a period "." is displayed, while for every ECHO_REPLY received a backspace is displayed. This provides a rapid display of how many packets are being dropped. If interval is not specified, it is set to zero and packets are output as fast as they come back or one hundred times per second, whichever is faster. Only the administrator may use this option with zero interval. |
| -i interval | Wait a specified interval of seconds between sending each packet. The default is 1 second between each packet, or no wait in flood mode. Only an administrator can set the interval to a value of less than 0.2 seconds. |
| -I interface address | Set source address to specified interface address. Argument may be numeric IP address or name of device. This option is required when you ping an IPv6 link-local address. |
| -l preload | If preload is specified, ping sends this number of packets that are not waiting for a reply. Only the administrator may select a preload of more than 3. |
| -L | Suppress loopback of multicast packets. This flag only applies if the ping destination is a multicast address. |
| -n | Numeric output only. No attempt will be made to look up symbolic names for host addresses. |
| -p pattern | You may specify up to 16 "pad" bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, -p ff will cause the sent packet to be filled with all ones. |
| -Q tos | Set Quality of Service -related bits in ICMP datagrams. tos can be either decimal or hex number. Traditionally (RFC1349), these have been interpreted as: 0 for reserved (currently being redefined as congestion control), 1-4 for Type of Service and 5-7 for Precedence. Possible settings for Type of Service are: minimal cost: 0x02, reliability: 0x04, throughput: 0x08, low delay: 0x10. Multiple TOS bits should not be set simultaneously. Possible settings for special Precedence range from priority (0x20) to net control (0xe0). You must be root (CAP_NET_ADMIN capability) to use Critical or higher precedence value. You cannot set bit 0x01 (reserved) unless ECN has been enabled in the kernel. In RFC2474, these fields has been redefined as 8-bit Differentiated Services (DS), consisting of: bits 0-1 of separate data (ECN will be used, here), and bits 2-7 of Differentiated Services Codepoint (DSCP). |
| -q | Quiet output. Nothing is displayed except the summary lines at startup time and when finished |

| | |
|-----------------------------------|---|
| -R | Record route. (IPv4 only) Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note that the IP header is only large enough for nine such routes. Many hosts ignore or discard this option. |
| -r | Bypass the normal routing tables and send directly to a host on an attached interface. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it provided the option -I is also used. |
| -s <i>packetsize</i> | Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data. |
| -S <i>sndbuf</i> | Set socket sndbuf (send buffer). If not specified, it is selected to buffer not more than one packet. |
| -t <i>tll</i> | Set the IP Time to Live. |
| -T <i>timestamp option</i> | Set special IP timestamp options. May be either tsonly (only timestamps), tsandaddr (timestamps and addresses) or tsprespec host1 [host2 [host3 [host4]]] (timestamp prespecified hops). |
| -M <i>hint</i> | Select Path MTU Discovery strategy. hint may be either do (prohibit fragmentation, even local one), want (do PMTU discovery, fragment locally when packet size is large), or don't (do not set DF flag). |
| -U | Print full user-to-user latency (the old behaviour). Normally ping prints network round trip time, which can be different f.e. due to DNS failures. |
| -v | Verbose output. |
| -V | Show version and exit. |
| -w <i>deadline</i> | Specify a timeout, in seconds, before ping exits regardless of how many packets have been sent or received. In this case ping does not stop after count packet are sent, it waits either for deadline expire or until count probes are answered or for some error notification from network. |
| -W <i>timeout</i> | Time to wait for a response, in seconds. The option affects only timeout in absence of any responses, otherwise ping waits for two RTTs. |

Examples

How to ping a global V6 address with a 1400 byte packet from FortiGate CLI:

```
Exec ping6 -s 1400 2001:480:332::10
```

How to ping Multicast group from Ping6 command on FortiGate CLI (-I and port name must be specified for CLI ping6 command to ping v6 multicast group):

```
Exec ping6 -I port1 ff02::1
```

How to ping localnet v6 address from FortiGate CLI:

```
Exec ping6 FE80:0:0:0:213:e8ff:fe9e:ccf7
```

This address would normally be written as FE80::213:e8ff:fe9e:ccf7.

Additional IPv6 resources

There are many RFCs available regarding IPv6. The following table lists the major IPv6 articles and their Internet Engineering Task Force (IETF) web locations.

Table 5: Additional IPv6 resources

| RFC | Subject | Location |
|---|---|---|
| RFC 1933, <i>Transition Mechanisms for IPv6 Hosts and Routers</i> | Describes IPv4 compatibility mechanisms that can be implemented by IPv6 hosts and routers | http://www.ietf.org/rfc/rfc1933 |
| RFC 2185, <i>Routing Aspects of IPv6 Transition</i> | Provides an overview of the routing aspects of the IPv6 transition | http://www.ietf.org/rfc/rfc2185 |
| RFC 2373, <i>IP Version 6 Addressing Architecture</i> | Defines the addressing architecture of the IP Version 6 protocol [IPv6] | http://www.ietf.org/rfc/rfc2373 |
| RFC 2402, <i>IP Authentication Header</i> | Describes functionality and implementation of IP Authentication Headers (AH) | http://www.ietf.org/rfc/rfc2402 |
| RFC 2460, <i>Internet Protocol, Version 6 (IPv6) Specification</i> | Describes functionality, configuration of IP version 6 (IPv6) and differences from IPv4. | http://www.ietf.org/rfc/rfc2460 |
| RFC 2461, <i>Neighbor Discovery for IP Version 6 (IPv6)</i> | Describes the features and functions of IPv6 Neighbor Discovery protocol | http://www.ietf.org/rfc/rfc2461 |
| RFC 2462, <i>IPv6 Stateless Address Autoconfiguration</i> | Specifies the steps a host takes in deciding how to autoconfigure its interfaces in IPv6 | http://www.ietf.org/rfc/rfc2462 |
| RFC 2893, <i>Transition Mechanisms for IPv6 Hosts and Routers</i> | Specifies IPv4 compatibility mechanisms that can be implemented by IPv6 hosts and routers | http://www.ietf.org/rfc/rfc2893 |
| RFC 3306, <i>Unicast-Prefix-Based IPv6 Multicast Addresses</i> | Describes the format and types of IPv6 multicast addresses | http://www.ietf.org/rfc/rfc3306 |
| RFC 3484, <i>Default Address Selection for Internet protocol version 6 (IPv6)</i> | Describes the algorithms used in IPv6 default address selection | http://www.ietf.org/rfc/rfc3484 |
| RFC 3513, <i>Internet Protocol version 6 (IPv6) Addressing Architecture</i> | Contains details about the types of IPv6 addresses and includes examples | http://www.ietf.org/rfc/rfc3513 |
| RFC 3587, <i>IPv6 Global Unicast Address Format</i> | Defines the standard format for IPv6 unicast addresses | http://www.ietf.org/rfc/rfc3587 |

FortiGate IPv6 configuration

This chapter describes how to configure your FortiGate unit's IPv6 functionality. Currently, the FortiGate unit supports IPv6 routing, tunneling, firewall policies and IPsec VPN.

You can configure your FortiGate unit for IPv6 operation through the Command Line Interface (CLI) and the web-based manager. Before you can work with IPv6 on your FortiGate unit, you must enable IPv6 support. Once IPv6 support is enabled, you can configure the IPv6 options using the web-based manager or the CLI. For more information, see the [FortiGate CLI Reference Guide](#) and the [FortiGate Administration Guide](#).

This technical note contains the following sections:

- [Configuring IPv6 interfaces](#)
- [Configuring IPv6 routing](#)
- [Configuring IPv6 over IPv4 tunneling](#)
- [Configuring IPv6 IPsec VPNs](#)
- [Configuring IPv6 firewall policies](#)

Configuring IPv6 interfaces

You can assign both an IPv4 and an IPv6 address to any interface on a FortiGate unit. Assigning an IPv6 address to the interface does not affect its IPv4 functionality. The IPv6 address you assign to the interface receives only IPv6-addressed packets. (Note: IPv6 is not supported over PPPoE or modem).

Adding an IPv6 address to an interface

To enable configuration of IPv6 in the FortiGate web-based manager, go to **System > Admin > Settings** and select IPv6 Support on GUI. For more information, see the [FortiGate Administration Guide](#). To enable configuration of IPv6 in the web-based manager using the CLI, use the following command:

```
config system global
  set gui-ipv6 enable
end
```

The following CLI commands are used to create an IPv6 address on an interface and to set administrative access to the interface:

```
config system interface
  edit <interface_name>
    config ipv6
      set ip6-address <if_ipv6mask>
      set ip6-allowaccess
    end
  end
```

| Variable | Description | Default |
|---------------------------------|--|---------|
| edit <interface_name> | Edit an existing interface or create a new VLAN interface. | None. |
| ip6-address <if_ipv6mask> | The interface IPv6 address and netmask. The format for IPv6 addresses and netmasks is described in RFC 3513. This is available in NAT/Route mode only. | ::/0 |
| ip6-allowaccess {ping any} | Type of management access permitted on this IPv6 interface. Enter ping or any (both settings indicate ping management access). | ping |

The `config ipv6` subcommand also contains keywords for defining the prefix list and configuring router advertisements. See [“Creating the prefix list for the interface”](#) next and [“Configuring IPv6 router advertisements”](#) on page 13.

The following example sets 3f30:0000:0000:0000:0000:2348:9abc as the IPv6 address for the internal interface with ping administrative access:

```
config system interface
  edit internal
    config ipv6
      set ip6-address 3f30::2348:9abc/60
      set ip6-allowaccess ping
    end
  end
```

Creating the prefix list for the interface

In IPv4-addressed networks the subnet mask alone determines which addresses are available on an interface. IPv6-addressing is more flexible. Routers exchange lists of address prefixes considered to be “on-link”, meaning that they are reachable on the interface (the address is assigned to the interface on a specified link). This is part of the IPv6 Neighbor Discovery (ND) process. Hosts and routers use Neighbor Discovery to determine the link-layer addresses for neighbors known to reside on attached links and to purge cached values that become invalid. Hosts also use ND to find neighboring routers willing to forward packets on their behalf. Nodes will actively keep track of which neighbors are reachable and which are not, and detect changed link-layer addresses. For more information, see [“IPv6 neighbor discovery” on page 10](#).

If a router or a path to a router fails, the host searches for alternatives. The node will consider an address to be on-link if:

- it is covered by one of the link prefixes (found in the on-link flag of the Prefix Information option)
- a neighboring router specifies the address as the target of a Redirect message
- a Neighbor Advertisement message is received for the target address
- any Neighbor Discovery message is received from the address.

In a message sent by a requesting router to a delegating router, the values in the fields may indicate the requesting router's preference for those values. The requesting router may set the IPv6 prefix field to zero and have a given value in the prefix-length field to indicate a preference for the size of the prefix to be delegated, or send a value of zero to indicate no preference.

The following commands configure prefix lists:

```
config system interface
  edit <interface_name>
    config ipv6
      config ip6-prefix-list
        edit <ipv6_prefix>
          set autonomous-flag {enable | disable}
          set onlink-flag {enable | disable}
          set preferred-life-time <seconds>
          set valid-life-time <seconds>
        end
      end
    end
  end
```

| Variable | Description | Default |
|---------------------------------------|---|---------|
| edit <interface_name> | Edit an existing interface or create a new VLAN interface. | None. |
| edit <ipv6_prefix> | Edit an existing prefix or create a new one. | |
| autonomous-flag {enable disable} | Set the state of the autonomous flag for the IPv6 prefix. | disable |
| onlink-flag {enable disable} | Set the state of the on-link flag (“L-bit”) in the IPv6 prefix. | disable |

| Variable | Description | Default |
|----------------------------------|--|---------|
| preferred-life-time <seconds> | Enter the preferred lifetime, in seconds, for this IPv6 prefix. Indicates time that the IPv6 prefix will stay preferred on the Requesting Router. | 604800 |
| valid-life-time <seconds> | Enter the valid lifetime, in seconds, for this IPv6 prefix. Indicates time that the IPv6 prefix is allowed to stay valid for a Requesting Router to use. | 2592000 |

Example

```

config system interface
  edit internal
    config ipv6
      config ip6-prefix-list
        edit 5f00::/64
          set autonomous-flag enable
          set preferred-life-time 432000
        end
      end
    end
  end
end

```

Configuring IPv6 routing

You can configure static routes and the router advertisements that the FortiGate unit sends on each interface.

Configuring static routing

FortiGate units support static routing for IPv6-addressed packets. To configure static routing for IPv6 using the web-based manager, go to **Router > Static > Static Route** and select **Create New**. For more information, see the [FortiGate Administration Guide](#). The following command specifies static IPv6 routes in the CLI:

```

config router static6
  edit <sequence_number>
    set device <interface_name>
    set dst <destination-address_ipv6mask>
    set gateway <gateway-address_ipv6>
  end
end

```

| Keywords and variables | Description | Default |
|------------------------------------|---|-------------|
| edit <sequence_number> | Enter a sequence number for the route. | No default. |
| device <interface_name> | The name of the FortiGate interface through which to route traffic. | Null. |
| dst <destination-address_ipv6mask> | The destination IPv6 address and netmask for this route. You can enter ::/0 to create a new static default route for IPv6 traffic. | ::/0 |
| gateway <gateway-address_ipv6> | The IPv6 address of the next-hop router to which traffic is forwarded. | :: |

Example

```

config router static6
  edit 2
    set dev internal
    set dst 12AB:0:0:CD30::/60
    set gateway 12AB:0:0:CD30:123:4567:89AB:CDEF
  end

```

Configuring IPv6 router advertisements

The FortiGate CLI provides the following commands to configure router advertisements for the interface.

```

config system interface
  edit <interface_name>
    config ipv6
      set ip6-address <if_ipv6mask>
      set ip6-allowaccess <access_types>
      set ip6-default-life <ipv6_life_seconds>
      set ip6-hop-limit <ipv6_hops_limit>
      set ip6-link-mtu <ipv6_mtu>
      set ip6-manage-flag {disable | enable}
      set ip6-max-interval <advert_max_seconds>
      set ip6-min-interval <advert_min_seconds>
      set ip6-other-flag {disable | enable}
      set ip6-reachable-time <reachable_msecs>
      set ip6-retrans-time <retrans_msecs>
      set ip6-send-adv {enable | disable}
      config ip6-prefix-list
        ...
      end
    end
  end
end

```

| Keywords and variables | Description | Default |
|---|---|---------|
| edit <interface_name> | Edit an existing interface or create a new VLAN interface. | None. |
| ip6-address ip6-allowaccess | See “Adding an IPv6 address to an interface” on page 10. | |
| ip6-default-life <ipv6_life_seconds> | Enter the number, in seconds, to add to the Router Lifetime field of router advertisements sent from the interface. The valid range is 0 to 9000. | 1800 |
| ip6-hop-limit <ipv6_hops_limit> | Enter the number to be added to the Cur Hop Limit field in the router advertisements sent out this interface. Entering 0 means no hop limit is specified. This is available in NAT/Route mode only. | 0 |
| ip6-link-mtu <ipv6_mtu> | Enter the MTU number to add to the router advertisements options field. Entering 0 means that no MTU options are sent. | 0 |
| ip6-manage-flag {disable enable} | Enable or disable the managed address configuration flag in router advertisements. | disable |

| Keywords and variables | Description | Default |
|--|---|---------|
| ip6-max-interval <advert_max_seconds> | Enter the maximum time interval, in seconds, between sending unsolicited multicast router advertisements from the interface. The valid range is 4 to 1800. | 600 |
| ip6-min-interval <advert_min_seconds> | Enter the minimum time interval, in seconds, between sending unsolicited multicast router advertisements from the interface. The valid range is 4 to 1800. | 198 |
| ip6-other-flag {disable enable} | Enable or disable the other stateful configuration flag in router advertisements. | disable |
| ip6-reachable-time <reachable_msecs> | Enter the number to be added to the reachable time field in the router advertisements. The valid range is 0 to 3600. Entering 0 means no reachable time is specified. | 0 |
| ip6-retrans-time <retrans_msecs> | Enter the number to be added to the Retrans Timer field in the router advertisements. Entering 0 means that the Retrans Timer is not specified. | 0 |
| ip6-send-adv {enable disable} | Enable or disable the flag indicating whether or not to send periodic router advertisements and to respond to router solicitations. | disable |
| config ip6-prefix-list | See “Creating the prefix list for the interface” on page 11. | |

For information about configuring prefix lists, see [“Creating the prefix list for the interface” on page 11.](#)

Testing connections with ping6

The ping command is a much-used tool in networking. FortiGate units provide an IPv6-specific version of the ping command.

```
execute ping6 {<address_ipv6> | <host-name_str>}
execute ping6 12AB:0:0:CD30:123:4567:89AB:CDEF
```

For more information, see [“IPv6 ping6 command” on page 13.](#)

Configuring IPv6 over IPv4 tunneling

FortiGate units support the transmission of IPv6-addressed traffic over an IPv4-addressed network. This technique is called IPv6 tunneling. You need to:

- create the tunnel (a virtual interface)
- create firewall policies
- define at least one route

Creating the IPv6 tunnel

To create an IPv6 tunnel in the web-based manager, go to **VPN > IPSEC > Auto Key (IKE)** and select **Create Phase 1**. For more information, see the [FortiGate Administration Guide](#) and the [FortiGate IPsec VPN User Guide](#). You configure a tunnel using the following FortiGate CLI command:

```
config system ipv6-tunnel
  edit <tunnel_name>
    set destination <tunnel_address>
    set interface <name>
    set ip6 <address_ipv6mask>
    set source <address_ipv4>
  end
```

| Variables | Description | Default |
|---------------------------------|---|-------------|
| edit <tunnel_name> | Enter a name for the IPv6 tunnel. | No default. |
| destination <tunnel_address> | The destination IPv4 address for this tunnel. | 0.0.0.0 |
| interface <name> | The interface used to send and receive traffic for this tunnel. | No default. |
| ip6 <address_ipv6mask> | The network prefix (IPv6 address and netmask) assigned to the interface to enable IPv6 processing on the interface. | ::/0 |
| source <address_ipv4> | The source IPv4 address for this tunnel. | 0.0.0.0 |

Example

The following example creates a tunnel, 6tunnel, that uses port3 to send and receive traffic. 6tunnel is a virtual interface that you use in firewall policies and routes.

```
config system ipv6-tunnel
  edit 6tunnel
    set destination 10.10.10.1
    set interface port3
    set ip6 12AB:0:0:CD30::/60
    set source 192.168.50.1
  end
```

Defining the firewall policies

You need to define firewall policies to permit traffic to flow between the local subnet and the IPv6 tunnel interface. A policy is required for each direction.



Note: As of MR7, you must configure an “ALL” address6 manually via the CLI prior to creating an IPv6 firewall policy:

```
config firewall address6
  edit ALL
    set ip6 ::/0
  end
end

config firewall policy6
  edit 1
    set srcintf port2
```

```
        set dstintf 6tunnel
        set action accept
        ...
    next
edit 2
    set srcintf 6tunnel
    set dstintf port2
    set action accept
    ...
next
end
```

For more information about IPv6 firewall policies, see [“Configuring IPv6 firewall policies” on page 19](#).

Defining routing

You need at least one route so that the IPv6 traffic is sent through the tunnel virtual interface. For example, 6tunnel, created in an earlier example, is the route to a particular IPv6 subnet:

```
config router static6
    edit 1
        set dst 1200:2345::3450/64
        set device 6tunnel
    next
end
```

Configuring IPv6 IPsec VPNs

The FortiGate unit supports interface-based IPv6 IPsec, but not policy-based. This section describes only how IPv6 IPsec support differs from IPv4 IPsec support.

FortiOS 3.0 supports IPv6 VPNs in the CLI and the web-based manager, however there are configuration options only accessible using the CLI.

Where both the gateways and the protected networks use IPv6 addresses, sometimes called IPv6 over IPv6, you can create either an auto-keyed or manually-keyed VPN. You can combine IPv6 and IPv4 addressing in an auto-keyed VPN in the following ways:

- IPv4 over IPv6 The VPN gateways have IPv6 addresses.
The protected networks have IPv4 addresses. The phase 2 configurations at either end use IPv4 selectors.
- IPv6 over IPv4 The VPN gateways have IPv4 addresses.
The protected networks use IPv6 addresses. The phase 2 configurations at either end use IPv6 selectors.

Compared with IPv4 IPsec VPN functionality, there are some limitations:

- Except for IPv6 over IPv4, remote gateways with Dynamic DNS are not supported. This is because FortiOS 3.0 does not support IPv6 DNS.
- You cannot use RSA certificates in which the common name (cn) is a domain name that resolves to an IPv6 address. This is because FortiOS 3.0 does not support IPv6 DNS.
- DHCP over IPsec is not supported, because FortiOS 3.0 does not support IPv6 DHCP.
- Selectors cannot be firewall address names. Only IP address, address range and subnet are supported.
- Redundant IPv6 tunnels are not supported.

Certificates

On a VPN with IPv6 phase 1 configuration, you can authenticate using VPN certificates in which the common name (cn) is an IPv6 address. The `cn-type` keyword of the `user peer` command has a new option, `ipv6`, to support this.

Phase 1 configuration

You define an IPsec phase 1 configuration as IPv6 by setting `ip-version` to 6. Its default value is 4. Then, the `local-gw` and `remote-gw` keywords are hidden and the corresponding `local-gw6` and `remote-gw6` keywords are available. The values for `local-gw6` and `remote-gw6` must be IPv6 addresses.

To configure IPv6 IPsec VPN phase 1

```
config vpn ipsec phase1-interface
edit tunnel6
set ip-version 6
set remote-gw6 0:123:4567::1234
set interface port3
set proposal 3des-md5
end
```

Phase 2 configuration

An IPv6 IPsec phase 2 configuration has IPv6 address selectors. The `src-addr-type` and `dst-addr-type` options `ip6`, `range6` and `subnet6` require IPv6 addresses, but are otherwise the same as the similarly-named IPv4 options. The `name` option, referring to a firewall address or address group name, applies only to IPv4 configurations.

To configure IPv6 IPsec VPN phase 2

```
config vpn ipsec phase2-interface
edit tunnel6_p2
    set src-addr-type subnet6
    set dst-addr-type subnet6
    set dst-subnet6 1200:2345::3456/64
    set interface port3
    set proposal 3des-md5
end
```

Firewall policies

To complete the VPN configuration, you need a firewall policy in each direction to permit traffic between the protected network's port and the IPsec interface. You need IPv6 policies unless the VPN is IPv4 over IPv6.

Routing

Appropriate routing is needed for both the IPsec packets and the encapsulated traffic within them. You need a route, which could be the default route, to the remote VPN gateway via the appropriate interface. You also need a route to the remote protected network via the IPsec interface. For example, where the remote network is `fec0:0000:0000:0004::/64` and the IPsec interface is `toB`:

```
config router static6
edit 1
    set device port2
    set dst 0::/0
next
edit 2
    set device toB
    set dst fec0:0000:0000:0004::/64
next
end
```

If the VPN is IPv4 over IPv6, the route to the remote protected network is an IPv4 route. If the VPN is IPv6 over IPv4, the route to the remote VPN gateway is an IPv4 route.

For more information, see the [FortiGate CLI Reference Guide](#) and the [FortiGate Administration Guide](#).

Configuring IPv6 firewall policies

The FortiGate unit supports IPv6 firewall policies. To configure an IPv6 firewall policy using the web-based manager, go to **Firewall > Policy > IPv6 Policy** and select **Create New**. For more information, see the [FortiGate Administration Guide](#).



Note: As of MR7, you must configure an “ALL” address6 manually via the CLI prior to creating an IPv6 firewall policy using the web-based manager, or the CLI:

```
config firewall address6
  edit ALL
    set ip6 ::/0
  end
end
```

To configure an IPv6 firewall policy using the CLI, use the following commands. For more information, see the [FortiGate CLI Reference Guide](#).

To create an IPv6 firewall address:

```
config firewall address6
  edit <address_name>
    set ip6 <ipv6_address_prefix>
  end
```

| Keywords and variables | Description | Default |
|---------------------------|----------------------------|-------------|
| <address_name> | The name of the address. | No default. |
| ip6 <ipv6_address_prefix> | Enter the IPv6 IP address. | No default. |

To create an IPv6 firewall policy:

```
config firewall policy6
  edit <id_integer>
    set action {accept | deny}
    set comments <comment_str>
    set diffserv-forward {enable | disable}
    set diffserv-reverse {enable | disable}
    set diffservcode-forward <outbound_binary>
    set diffservcode-rev <reply_binary>
    set dstaddr <name_str>
    set dstintf <name_str>
    set fsae {enable | disable}
    set fixedport {enable | disable}
    set gbandwidth <bandwidth_integer>
    set groups <name_str>
    set logtraffic {enable | disable}
    set maxbandwidth <bandwidth_integer>
    set nat {enable | disable}
    set priority {high | low | medium}
    set profile <name_str>
    set profile-status {enable | disable}
    set schedule <name_str>
    set service <name_str>
    set srcaddr [all | <name_str>]
```

```

set srcintf <name_str>
set status {enable | disable}
set trafficshaping {enable | disable}
end

```

| Keywords and variables | Description | Default |
|--|--|-------------|
| id_integer | The unique ID number of this policy. | No default |
| action {accept deny} | Enter <code>accept</code> to accept packets that match the firewall policy. Enter <code>deny</code> to deny packets that match the firewall policy. | deny |
| comments <comment_str> | Optionally add a description or other information about the policy. <code>comment_str</code> is limited to 63 characters. Enclose the string in single quotes to enter special characters or spaces. | No default. |
| diffserv-forward {enable disable} | Enable or disable forward (original) Differentiated Services traffic for this policy. | disable |
| diffserv-reverse {enable disable} | Enable or disable reverse (reply) Differentiated Services traffic for this policy. | disable |
| diffservcode-forward <outbound_binary> | Set the Differentiated Services Code Point (DSCP) value in the Diffserv field of outbound packets. The value is 6 bits binary. The valid range is 000000-111111. | 000000 |
| diffservcode-rev <reply_binary> | Set the Differentiated Services Code Point (DSCP) value in the Diffserv field of reply packets. The value is 6 bits binary. The valid range is 000000-111111. | 000000 |
| dstaddr <name_str> | Enter the destination address for the policy. For a NAT policy a virtual IP can be added. <code>name_str</code> is case-sensitive. | null |
| dstintf <name_str> | Enter the destination interface for the policy. The interface can be a physical interface, a VLAN subinterface or a zone. If the interface or VLAN subinterface has been added to a zone, the interface or VLAN subinterface cannot be used for <code>dstintf</code> . | null |
| fsae {enable disable} | Enable or disable ActiveDirectory authentication. | disable |
| fixedport {enable disable} | When the action is set to <code>accept</code> , prevent a NAT policy from translating the source port. Some applications do not function correctly if the source port is changed. If <code>fixedport</code> is entered, also enable IP pools. Not enabling IP pools means a policy with <code>fixedport</code> can only allow one connection at a time for this port or service. | disable |
| gbandwidth <bandwidth_integer> | When traffic shaping is enabled, guarantee the amount of bandwidth available for traffic controlled by the policy. <code>bandwidth_integer</code> can be 0 to 100000 Kbytes/second. | 0 |
| groups <name_str> | To require authentication, enter the names of the user groups that are allowed to use this policy. This is available when <code>action</code> is set to <code>accept</code> . When user groups are created, they are paired with protection profiles. The user group name is case sensitive. | No Default. |

| Keywords and variables | Description | Default |
|--------------------------------------|---|-------------|
| logtraffic {enable disable} | Enable or disable recording traffic log messages for this policy. | disable |
| maxbandwidth <bandwidth_integer> | When traffic shaping is enabled, limit the maximum amount of bandwidth available for traffic controlled by the policy. <code>bandwidth_integer</code> can be 0 to 100000 Kbytes/second. If maximum bandwidth is set to 0 no traffic is allowed by the policy. | 100 |
| nat {enable disable} | When the action is set to accept, configure the policy for network address translation (NAT). NAT translates the source address and the source port of packets accepted by the policy. When NAT is enabled, <code>ippool</code> and <code>fixedport</code> can also be enabled or disabled. | disable |
| priority {high low medium} | When traffic shaping is enabled, set the priority for traffic controlled by the policy. The available settings are <code>high</code> for high priority traffic, <code>medium</code> for medium priority traffic, and <code>low</code> for low priority traffic. | high |
| profile <name_str> | When a protection profile is being used, enter the name of a profile to add the protection profile to the policy. The <code>name_str</code> variable is case-sensitive. This is automatically disabled if a user group with a protection profile has been selected for authentication. | No Default. |
| profile-status {enable disable} | Enable or disable using a protection profile for the policy. This is automatically disabled if a user group has been selected for authentication. | disable |
| schedule <name_str> | Enter the name of the one-time or recurring schedule to use for the policy. The <code>name_str</code> variable is case-sensitive. | No default. |
| service <name_str> | Enter the name of the service to use for the policy. The <code>name_str</code> variable is case-sensitive. | No default. |
| srcaddr [all <name_str>] | Enter the source address for the policy. The <code>name_str</code> variable is case-sensitive. | null |
| srcintf <name_str> | Enter the source interface for the policy. The interface can be a physical interface, a VLAN subinterface or a zone. If the interface or VLAN subinterface has been added to a zone, interface or VLAN subinterface cannot be used for <code>srcintf</code> . | null |
| status {enable disable} | Enable or disable the policy. | enable |
| trafficshaping {enable disable} | Enable or disable traffic shaping. Also set <code>gbandwidth</code> , <code>maxbandwidth</code> , and <code>priority</code> . | disable |

For more information, see the [FortiGate CLI Reference Guide](#) and the [FortiGate Administration Guide](#).

FORTINET™

www.fortinet.com

FORTINET™

www.fortinet.com