Policy and Objects

This section contains topics on configuring policies and traffic shaping:

Policies

The firewall policy is the axis around which most features of the FortiGate revolve. Many firewall settings end up relating to or being associated with the firewall policies and the traffic they govern. Any traffic going through a FortiGate has to be associated with a policy. These policies are essentially discrete compartmentalized sets of instructions that control the traffic flow going through the firewall. These instructions control where the traffic goes, how it is processed, if it is processed, and whether or not it is allowed to pass through the FortiGate.

When the firewall receives a connection packet, it analyzes the source address, destination address, and service (by port number). It also registers the incoming interface, the outgoing interface it needs to use, and the time of day. Using this information, the FortiGate firewall attempts to locate a security policy that matches the packet. If a policy matches the parameters, then the FortiGate takes the required action for that policy. If it is *Accept*, the traffic is allowed to proceed to the next step. If the action is *Deny* or a match cannot be found, the traffic is not allowed to proceed.

The two basic actions at the initial connection are either *Accept* or *Deny*:

- If the action is *Accept*, the policy permits communication sessions. There may be other packet processing instructions, such as requiring authentication to use the policy or restrictions on the source and destination of the traffic.
- If the action is *Deny*, the policy blocks communication sessions, and you can optionally log the denied traffic. If no security policy matches the traffic, the packets are dropped. A *Deny* security policy is needed when it is required to log the denied traffic, also called *violation traffic*.

One other action can be associated with the policy:

- *IPsec*: this is an *Accept* action that is specifically for IPsec VPNs.

Each field in a firewall policy that accepts multiple inputs, such as srcaddr and dstaddr, can accept as many inputs as there are unique objects created. The maximum number of objects depends on the model. See the Maximum Values Table for more details.

The following topics provide information on the available types of policies and configuration instructions:

- DoS policy on page 1362
- Access control lists on page 1369
- Interface policies on page 1370

The following topics provide instructions on configuring policies:

- Source NAT on page 1371
- Destination NAT on page 1394
- Examples and policy actions on page 1418

Firewall policy

The firewall policy is the axis around which most of the other features of the FortiGate firewall revolve. A large portion of the settings in the firewall at some point will end up relating to or being associated with the firewall policies and the traffic that they govern. Any traffic going through a FortiGate unit has to be associated with a policy. These policies are essentially discrete compartmentalized sets of instructions that control the traffic flow going through the firewall. These instructions control where the traffic goes, how it's processed, if it's processed, and even whether or not it's allowed to pass through the FortiGate.

The following topics provide information on the firewall policy and configuration:

- Firewall policy parameters on page 1319
- Configurations in the GUI on page 1320
- Configurations in the CLI on page 1325
- Policy views on page 1330
- Policy lookup on page 1336
- Services on page 1337

Firewall policy parameters

For traffic to flow through the FortiGate firewall, there must be a policy that matches its parameters:

- Incoming interface(s)
- Outgoing interface(s)
- Source address(es)
- User(s) identity
- Destination address(es)
- Internet service(s)
- Schedule
- Service

Traffic parameters are checked against the configured policies for a match. If the parameters do not match any configured policies, the traffic is denied.

Traffic flow initiated from each direction requires a policy, that is, if sessions can be initiated from both directions, each direction requires a policy.

Just because packets can go from point A to point B on port X does not mean that the traffic can flow from point B to point A on port X. A policy must be configured for each direction.

When designing a policy, there is often reference to the traffic flow, but most communication is two-way so trying to determine the direction of the flow might be confusing. If traffic is HTTP web traffic, the user sends a request to the website, but most of the traffic flow will be coming from the website to the user or in both directions? For the purposes of determining the direction for a policy, the important factor is the direction of the initiating communication. The user is sending a request to the website, so this is the initial communication; the website is responding so the traffic is from the user's network to the Internet.

FortiOS does not perform a reverse-path check on reply traffic that matches an allowed session based on the IP tuple. The request traffic can be sent on one interface and the reply traffic could return on another interface.

Configurations in the GUI

Firewall policies can be created in the GUI by configuring the necessary parameters.

Incoming interface(s)	This is the interface or interfaces by which the traffic is first connected to the FortiGate unit. The exception being traffic that the FortiGate generates itself. This is not limited to the physical Ethernet ports found on the device. The incoming interface can also be a logical or virtual interface such as a VPN tunnel, a Virtual WAN link, or a wireless interface.
Outgoing interface(s)	This is the interface or interfaces used by traffic leaving a port once it has been processed by the firewall. Similar to incoming interfaces, it is not limited to only physical interfaces.
Source address(es)	The addresses that a policy can receive traffic from can be wide open or tightly controlled. For a public web server that the world at large should be able to access, the best choice will be *all*. If the destination is a private web server that only the branch offices of a company should be able to access, or a list of internal computers that are the only ones allowed to access an external resource, then a group of preconfigured addresses is the better strategy
User(s) identity	This parameter is based on a user identity that can be from a number of authentication authorities. It will be an account or group that has been set up in advance that can be selected from the drop down menu. The exception to this is the feature that allows the importing of LDAP Users. When the feature is used, a small wizard window will appear to guide the user through the setup. The caveat is that the LDAP server object in the *User & Authentication > LDAP Servers* section has to be already configured to allow the use of this import feature.
Destination address(es)	In the same way that the source address may need to be limited, the destination address can be used as a traffic filter. When the traffic is destined for internal resources, the specific address of the resource can be defined to better protect the other resources on the network. One of the specialized destination address options is to use a Virtual IP address. If the destination address doesn't need to be internal, you can define policies that are only for connecting to specific addresses on the Internet.

Internet service(s)	In this context, an Internet service is a combination of one or more addresses and one or more services associated with a service found on the Internet such as an update service for software.
Schedule	The time frame that is applied to the policy. This can be something as simple as a time range that the sessions are allowed to start, such as between 8:00 am and 5:00 pm. Something more complex like business hours that include a break for lunch and time of the session's initiation may need a schedule group because it will require multiple time ranges to make up the schedule.
Service	The services chosen represent the TCP/IP suite port numbers that will most commonly be used to transport the named protocols or groups of protocols. This is different than *Application Control* which looks more closely at the packets to determine the actual protocol used to create them. A case where either side can initiate the communication, like between two internal interfaces on the FortiGate unit, would be a more likely situation to require a policy for each direction.

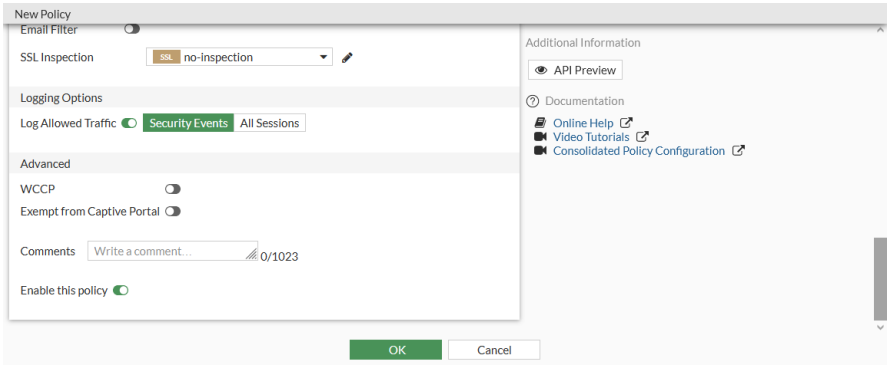
Enabling advanced policy options in the GUI

Advanced policy options can be enabled so that you can configure the options in the GUI.

To enable advanced policy options:

```
config system settings
    set gui-advanced-policy enable
end
```

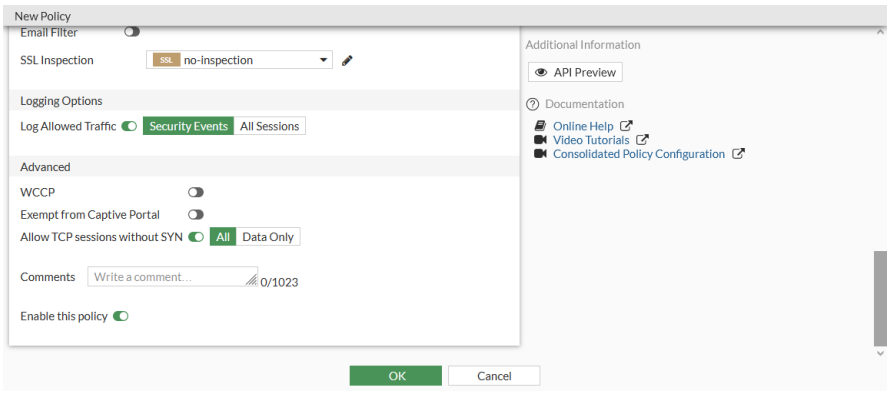
Advanced policy options are now available when creating or editing a policy in the GUI:



To enable configuring TCP sessions without SYN:

```
config system settings
    set tcp-session-without-syn enable
end
```

TCP sessions without SYN can now be configured when creating or editing a policy in the GUI:



Add Policy change summary and Policy expiration to Workflow Management

Two options, *Policy change summary* and *Policy expiration*, are included in *Workflow Management*. *Policy change summary* enforces an audit trail for changes to firewall policies. *Policy expiration* allows administrators to set a date for the firewall policy to be disabled.

There are three states for the *Policy change summary*:

- *Disable*: users will not be prompted to add a summary when editing a policy.
- *Required*: the *Policy change summary* will be enabled and will require users to add a summary when editing or creating a firewall policy.
- *Optional*: the *Policy change summary* will be enabled but users can leave the summary empty, if preferred, when editing or creating a firewall policy.

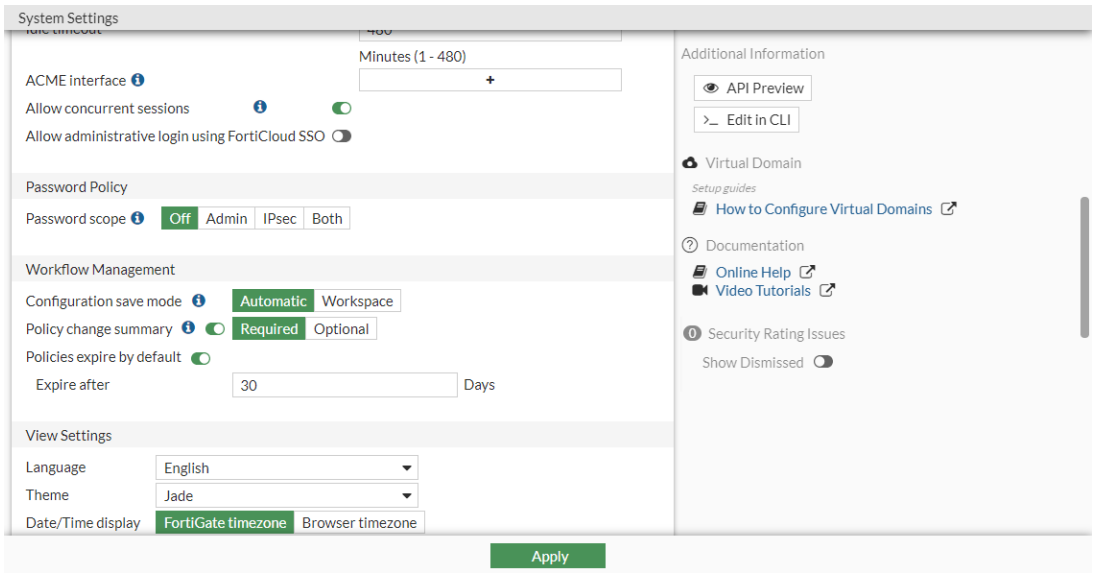
There are three states for *Policy expiration*:

- *Disable*: the firewall policy will not expire. This is the default setting for *Policy expiration*.
- *Default*: the firewall policy will expire after the default number of days.
- *Specify*: the firewall policy will expire at a set date and time.

The default value for *Policy expiration* is 30 days. This number can be changed in the CLI or in *System* > *Settings* in the GUI to any value between zero and 365 days. If the default value is set to zero, the *Default* state will disable the *Policy expiration*.

To configure the firewall policy change summary and default expiration in the GUI:

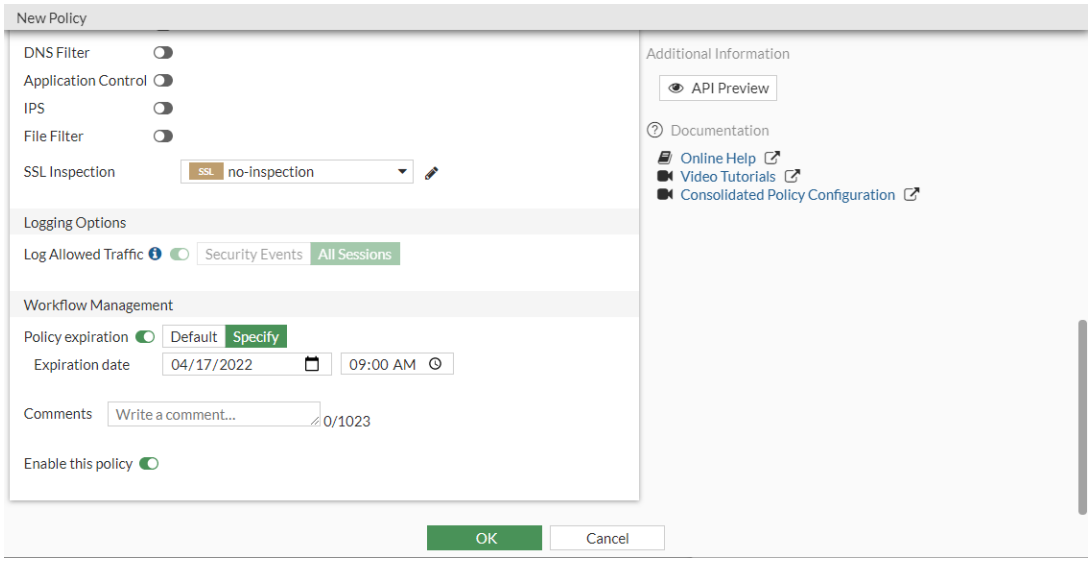
1. Go to *System* > *Feature Visibility*.
2. Enable *Workflow Management*.
3. Click *Apply*.
4. Go to *System* > *Settings*.
5. In the *Workflow Management* section, set *Policy change summary* to *Required*. *Policies expire by default* is enabled by default with an *Expire after* value of *30*.



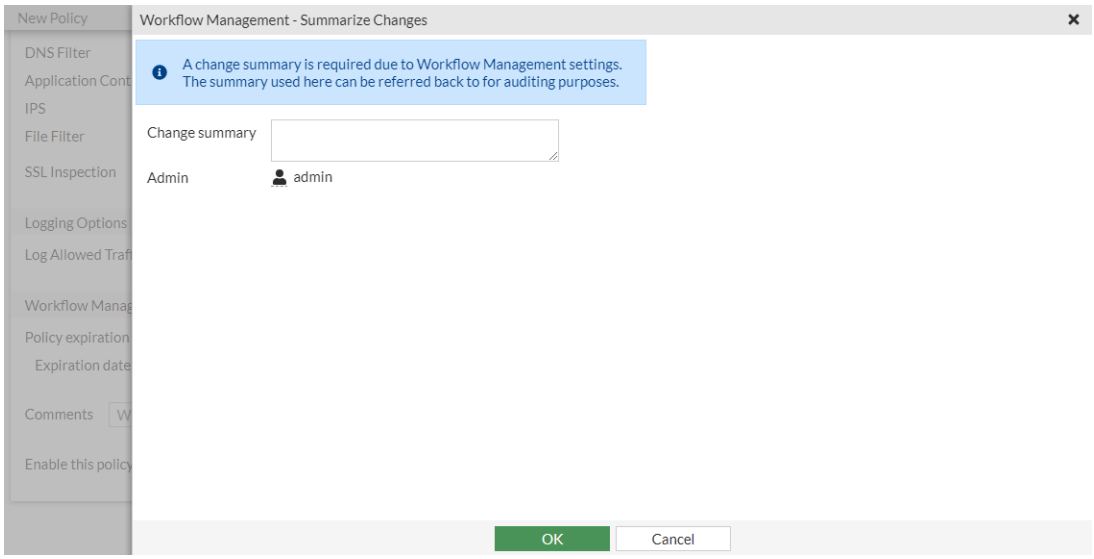
6. Click *Apply*.

To configure firewall policy expiration in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Name the policy and configure the necessary parameters.
3. Set *Policy expiration* to *Specify*. The *Expiration date* fields appears with the current date and time.



4. Select the date and time for the policy to expire from the *Expiration date* fields.
5. Click *OK*. The *Workflow Management - Summarize Changes* pane opens.



6. In the *Change summary* field, enter details about the changes made to the policy. These details can be referred to later for auditing purposes.
7. Click *OK*.

To configure the firewall policy change summary in the CLI:

```
config system settings
    set gui-enforce-change-summary {disable | require | optional}
end
```

To configure the policy expiration default value in the CLI:

```
config system settings
    set default-policy-expiry-days <integer>
end
```

To configure firewall policy expiration in the CLI:

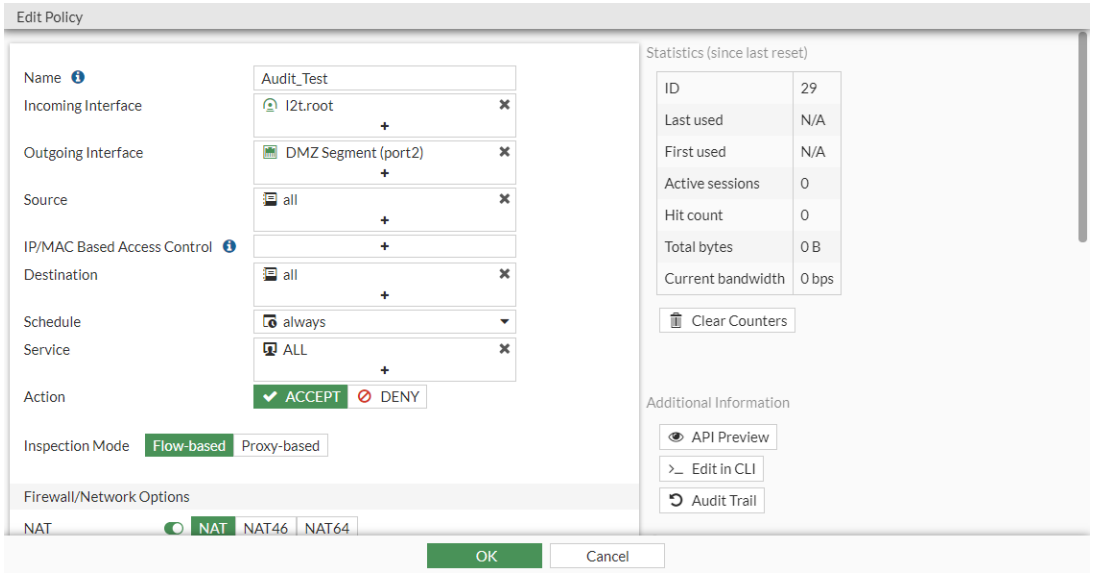
```
config firewall policy
    edit <id>
        set policy-expiry {enable | disable}
        set policy-expiry-date <YYYY-MM-DD HH:MM:SS>
    next
end
```

Policy change summaries are used to track changes made to a firewall policy. The *Audit Trail* allow users to review the policy change summaries, including the date and time of the change and which user made the change.

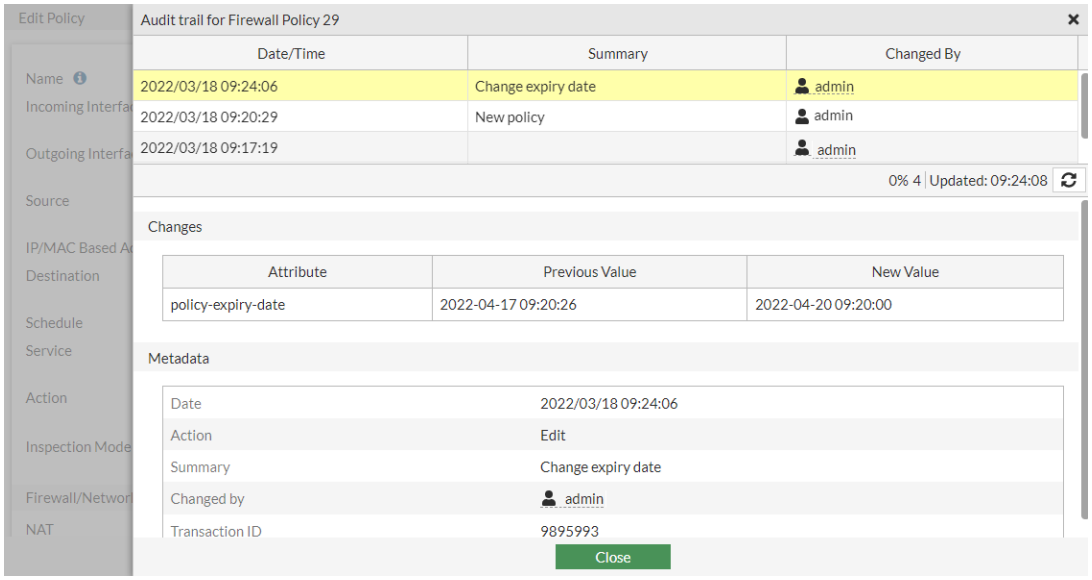
The *Audit Trail* is only supported by FortiGate models with disk logging.

To review the audit trail in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Select the policy you want to review and click *Edit*.



3. In the right-side banner, click *Audit Trail*. The *Audit trail for Firewall Policy* pane opens and displays the policy change summaries for the selected policy.



4. Select an entry to review the details of the change made.
5. When you are done reviewing the *Audit Trail*, click *Close*.
6. Click *Cancel* to exit the *Edit Policy* page.

Configurations in the CLI

Firewall policies can be created in the CLI by configuring the necessary parameters. See Configurations in the GUI on page 1320 for more information on the various parameters.

Parameter	Definition
srcintf	Incoming (ingress) interface.
dstintf	Outgoing (egress) interface.
srcaddr	Source IPv4 address and address group names.
dstaddr	Destination IPv4 address and address group names.
internet-service	Enable/disable use of Internet Services for this policy. If enabled, destination address and service are not used.
schedule	Schedule name.
service	Service and service group names.
anti-replay	Enable/disable checking of TCP flags per policy.
match-vip	Enable/disable matching of VIPs when used in a policy with a deny action.
auto-asic-offload	Enable/disable hardware acceleration. Available on select FortiGate models with Secure Processing Unit (SPU) hardware only.
tcp-mss-sender	Sender TCP maximum segment size (MSS).
tcp-mss-receiver	Receiver TCP maximum segment size (MSS).
session-ttl	Time-to-live (TTL) in seconds for session accepted by this policy.

Firewall anti-replay option per policy

When the global anti-replay option is disabled, the FortiGate does not check TCP flags in packets. The per policy anti-replay option overrides the global setting. This allows you to control whether or not TCP flags are checked per policy.

To enable the anti-replay option so TCP flags are checked using the CLI:

```
config firewall policy
    edit 1
        set name "policyid-1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set anti-replay enable
        set logtraffic all
        set nat enable
    next
end
```

Deny matching with a policy with a virtual IP applied

Preventing hosts with specific source addresses from accessing a server behind the FortiGate may be required in some cases. For this scenario, you should have previously configured a firewall policy with a virtual IP (VIP) object applied to it

to allow such access. See Destination NAT on page 1394 for details.

When denying traffic destined for a typical firewall policy without a VIP applied, you would simply configure a new firewall policy with an action of `deny` and with specific source addresses above the firewall policy that you want to prevent these hosts from accessing. However, the FortiGate matches firewall policies with VIPs applied differently than typical firewall policies. Policies with VIPs applied have priority over typical firewall policies.

Therefore, to block specific source traffic destined for a firewall policy specified with an action of `accept` and with a VIP applied, you should configure `set match-vip enable` on the firewall policy with a `deny` action that has been configured to match traffic before the firewall policy with the VIP applied. By default, new `deny` action firewall policies have `match-vip` enabled.

If the policy action is set to `accept`, `match-vip` cannot be enabled.

To block VIP traffic in a deny policy:

```
config firewall policy
    edit 1
        set name "deny-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "src-hosts-to-deny-access"
        set dstaddr "all"
        set action "deny"
        set schedule "always"
        set service "all"
        set match-vip enable
    next
    edit 2
        set name "vip-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-object-1"
        set action "accept"
        set schedule "always"
        set service "ALL"
    next
end
```

Alternatively, to block access to a firewall policy with a VIP applied, you can configure a new VIP object configured with `set src-filter <range>`. Configure a new firewall policy with a `deny` action and with this new VIP applied, and then configure this policy to match traffic before the firewall policy with the same VIP applied with an action of `accept`. In this case, the firewall policy can simply have `set match-vip disable`.

To specify a VIP with source addresses specified with a deny policy:

```
config firewall vip
    edit "vip-with-srcaddr-to-deny"
        set extip "10.1.100.199"
        set extintf "wan1"
        set mappedip "172.16.200.55"
```

```
        set src-filter "1.1.1.1/24"
    next
end
config firewall policy
    edit 3
        set name "deny-policy-3"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-with-srcaddr-to-deny"
        set action "deny"
        set match-vip disable
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "vip-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-object-1"
        set action "accept"
        set match-vip disable
        set schedule "always"
        set service "ALL"
    next
end
```

Hardware acceleration

Hardware acceleration is supported on select FortiGate devices and is enabled by default on all firewall policies to ensure optimal performance when processing network traffic traversing the FortiGate. See the Hardware Acceleration Reference Manual for details.

Typically, hardware acceleration on a specific firewall policy is disabled for one of two purposes:

- To allow CLI commands such as the packet sniffer and debug flow to display all traffic matching the policy since traffic offloaded by SPU hardware on a FortiGate device is not visible by those CLI tools.
- To troubleshoot any possible issues arising by using hardware acceleration.

To disable hardware acceleration in an IPv4 firewall policy:

```
config firewall policy
    edit 1
        set auto-asic-offload disable
    next
end
```

To disable hardware acceleration in an IPv6 firewall policy:

```
config firewall policy6
    edit 1
        set auto-asic-offload disable
    next
end
```

To disable hardware acceleration in a multicast firewall policy:

```
config firewall multicast-policy
    edit 1
        set auto-asic-offload disable
    next
end
```

TCP Maximum Segment Size (MSS)

The TCP maximum segment size (MSS) is the maximum amount of data that can be sent in a TCP segment. The MSS is the MTU size of the interface minus the 20 byte IP header and 20 byte TCP header. By reducing the TCP MSS, you can effectively reduce the MTU size of the packet.

The TCP MSS can be configured in a firewall policy, or directly on an interface. See Interface MTU packet size on page 180 for details on configuring TCP MSS directly on an interface.

To configure TCP MSS in a firewall policy:

```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set tcp-mss-sender 1448
        set tcp-mss-receiver 1448
    next
end
```

Adjusting session time-to-live (TTL)

A session is a communication channel between two devices or applications across the network. Sessions allow FortiOS to inspect and act on a sequential group of packets in a session all at once instead of inspecting each packet individually. Each session has an entry in the session table that includes important information about the session.

The session time-to-live (TTL) parameter determines how long a session of a particular protocol such as TCP, UDP, or ICMP remains in the session table. To ensure proper operation of some devices or applications, the session TTL parameter may need to be increased or decreased to allow sessions to remain active in the session table for a longer or shorter duration, respectively.

To configure a modified session TTL in a firewall policy:

```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set session-ttl 1800
```

```
    next
end
```

The session TTL can be set to zero or `never` to ensure a session never times out. See No session timeout on page 1439 for details.

Session TTL should only be set to zero or `never` after careful consideration of:

- The connected device's or application's requirements for sessions to always stay alive
- The expectation that a connected device or application will use the same session determined by traffic using a fixed source port, fixed destination port, fixed source IP address, and fixed destination IP address.

When session TTL is set to zero or `never`, then sessions will not be cleared from the session table or expire after a specified time unless the CLI commands `diagnose system session filter <filter>` and `diagnose system session clear` are used.

If this setting is used in the case when traffic through a firewall policy can generate numerous unique sessions, then this may have unintended consequences to the FortiGate's memory usage and performance due to the session table constantly growing and not clearing out idle sessions.

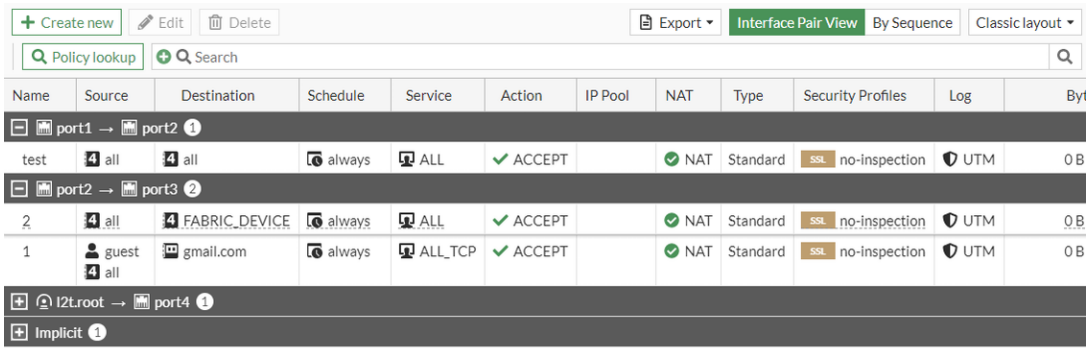
To disable session TTL in a firewall policy:

```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set session-ttl never
    next
end
```

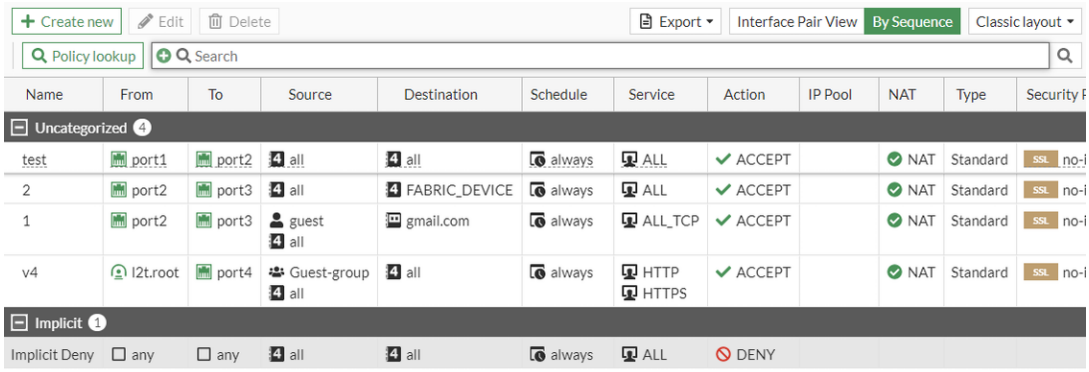
Policy views

In *Policy & Objects* policy list pages, there are two policy views: *Interface Pair View* and *By Sequence* view.

Interface Pair View displays the policies in the order that they are checked for matching traffic, grouped by the pairs of incoming and outgoing interfaces in collapsible sections. The *Interface Pair View* can be used when a policy is configured with multiple interfaces.



By Sequence displays policies in the order that they are checked for matching traffic without any grouping.

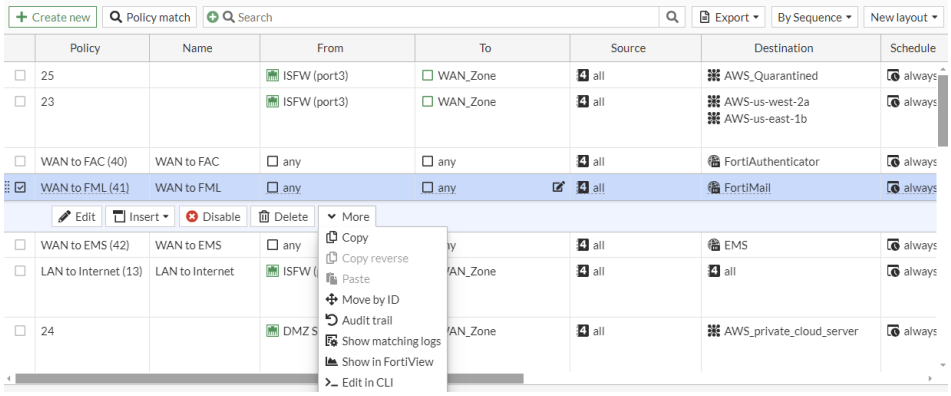


Policies can then be moved by their policy ID before or after another specified policy ID.

Moving policies by ID is only available when viewing the *Firewall Policy* page in *By Sequence* or *Sequence Grouping View*.

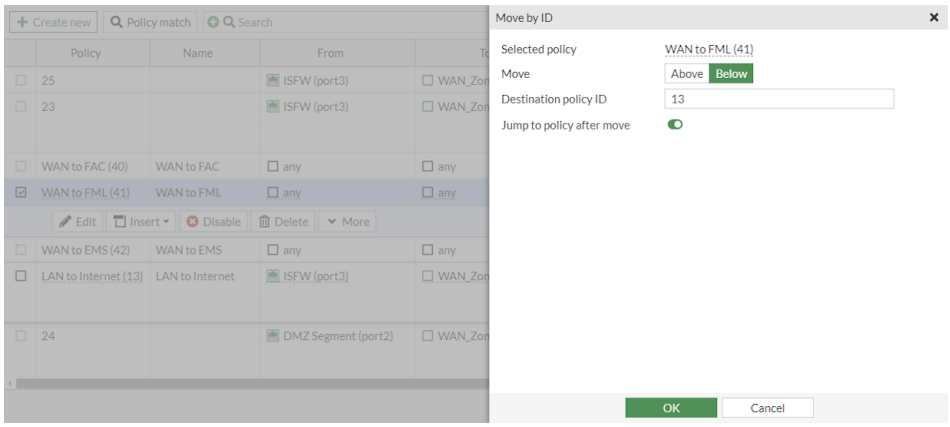
To move a policy by policy ID:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Select the policy you want to move.
3. Select *More* > *Move by ID*.



The *Move by ID* pane is displayed.

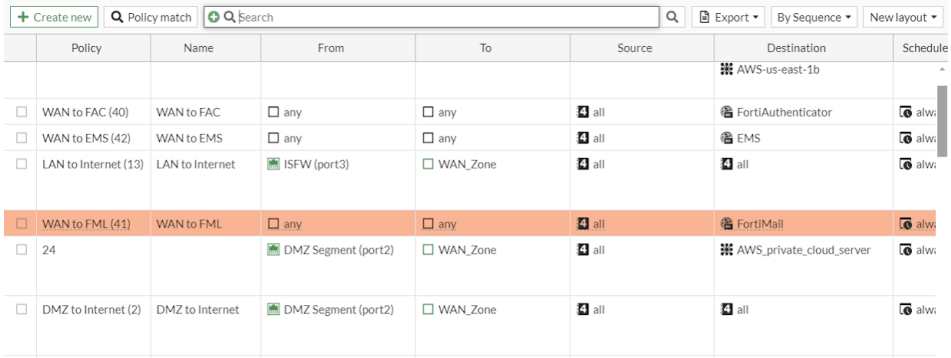
4. Define the new location of the policy:
   a. Select whether the policy should be moved *Above* or *Below* the policy ID you will define in the next step.
   b. In the *Destination policy ID* field, enter the ID of the destination policy or select it from the dropdown menu.



5. If you do not want to automatically view the new location of the policy, disable *Jump to policy after move*. This feature is enabled by default.
6. Click *OK*.

If *Workflow Management* is enabled in *System* > *Feature Visibility*, the *Workflow Management - Summarize Changes* pane is displayed. Enter a *Change summary* and click *OK* to continue.

The policy will be moved to the new location.

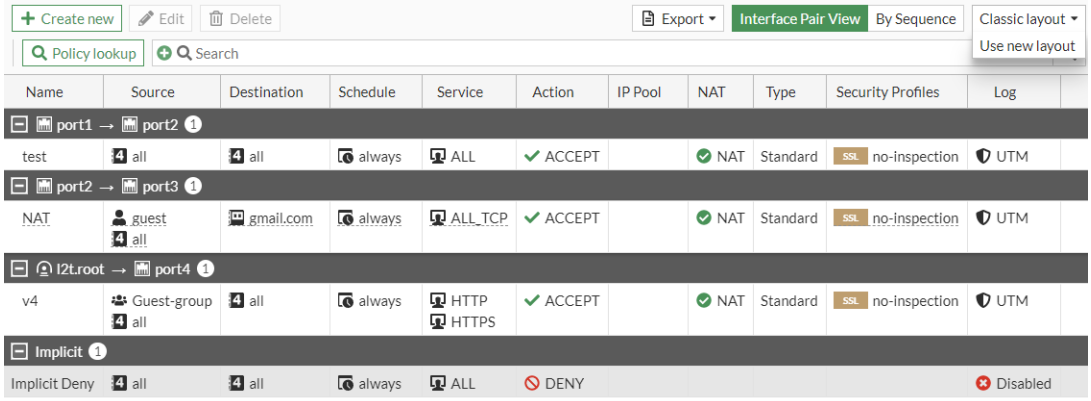


New layout for firewall policies

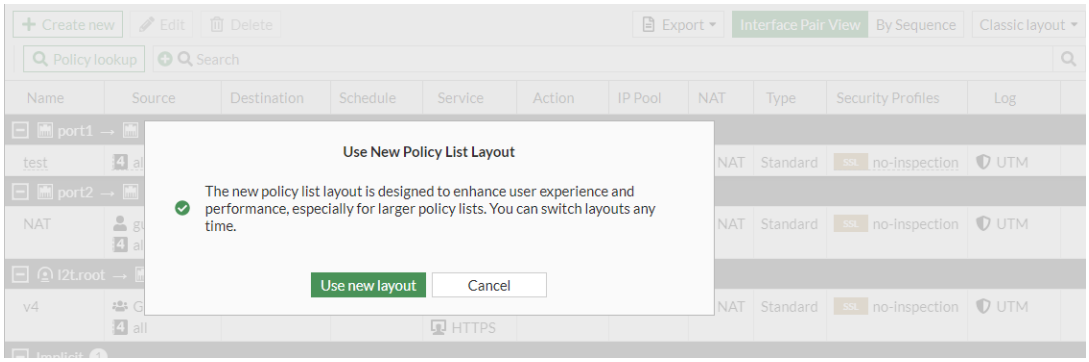
A new layout is available for the policy list with the option to alternate between the new layout and the old layout. To switch between the *Classic layout* and *New layout*, select the style from the dropdown menu.

To change from the classic layout to the new layout:

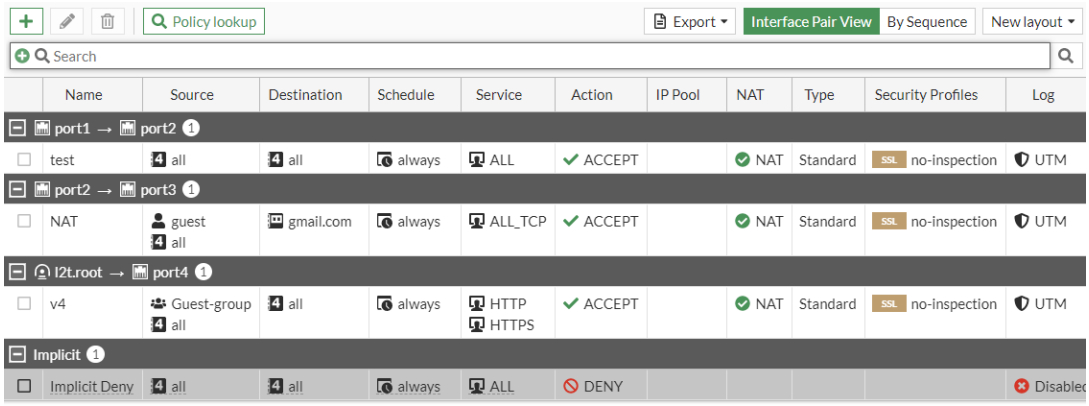
1. Go to *Policy & Objects* > *Firewall Policy*.
2. Select the *Classic layout* dropdown menu.



3. Select *Use new layout*. A confirmation message is displayed.

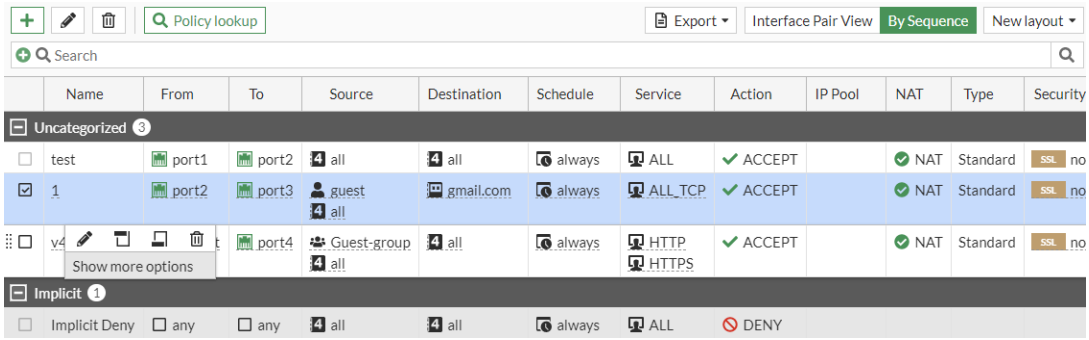


4. Click *Use new layout*. The new layout is displayed.

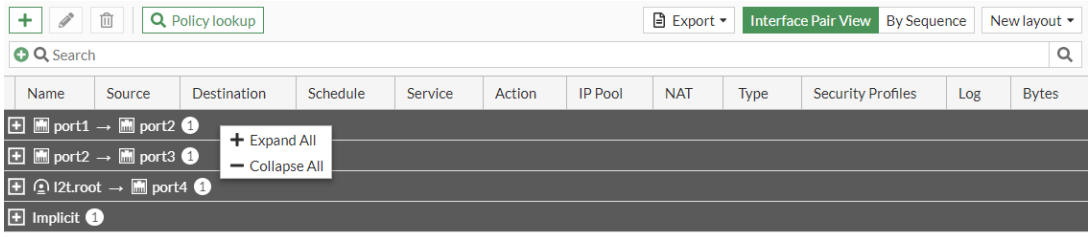


The *New layout* includes several features to enhance user experience when using the *Policy & Objects* > *Firewall Policy* page:

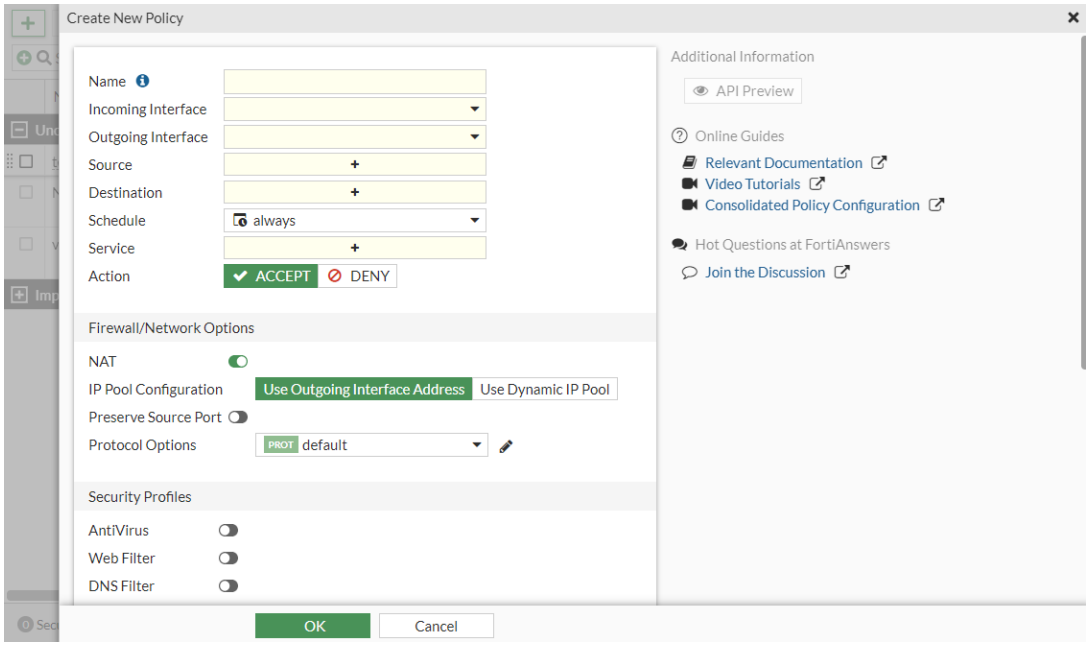
- The create, edit, and delete buttons are identified through icons instead of words. Selecting a policy also displays an inline menu with options to edit, delete, and insert policies, with the option to *Show more options* when hovered over.



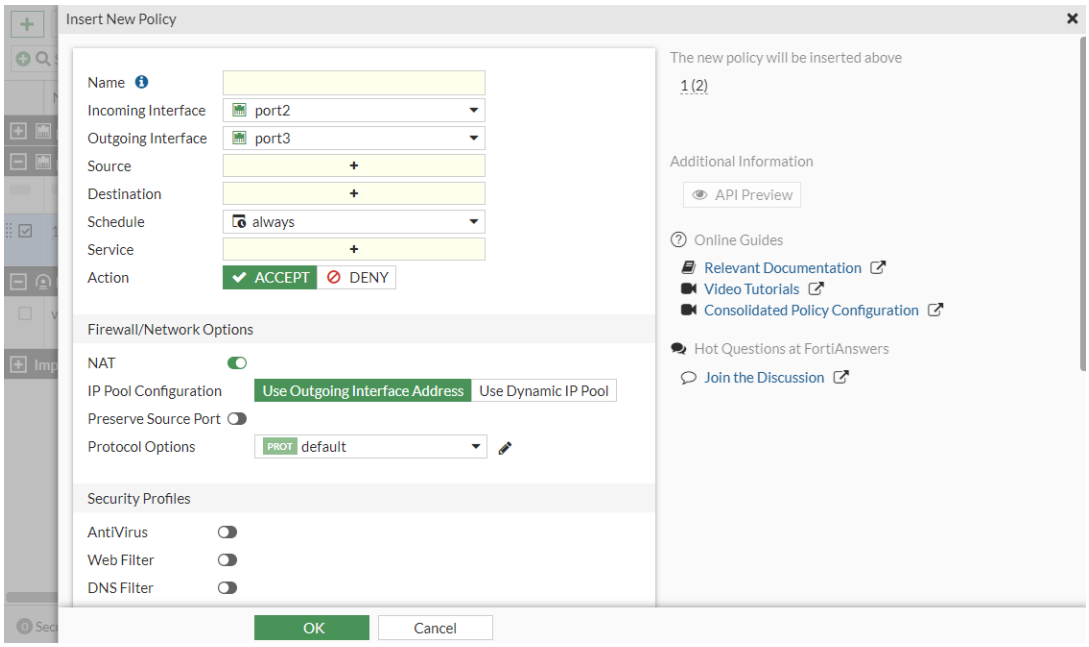
- Right-click in *Interface Pair View* to *Expand All* or *Collapse All* sections.



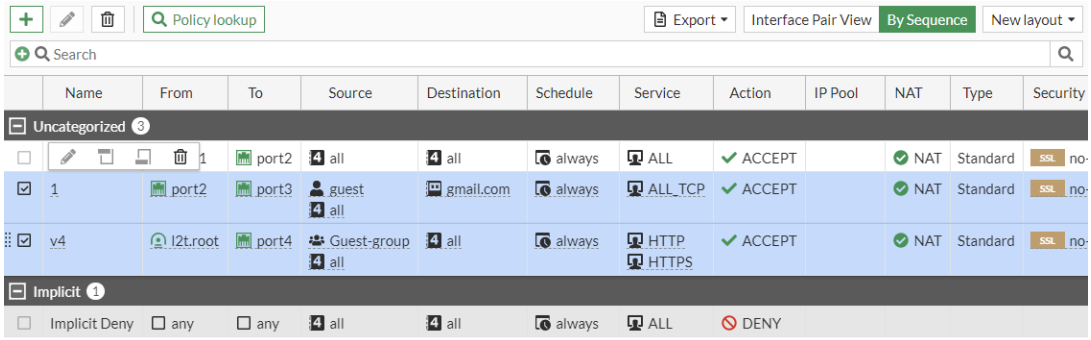
- A pane is used to create, edit, and insert policies instead of a separate page.



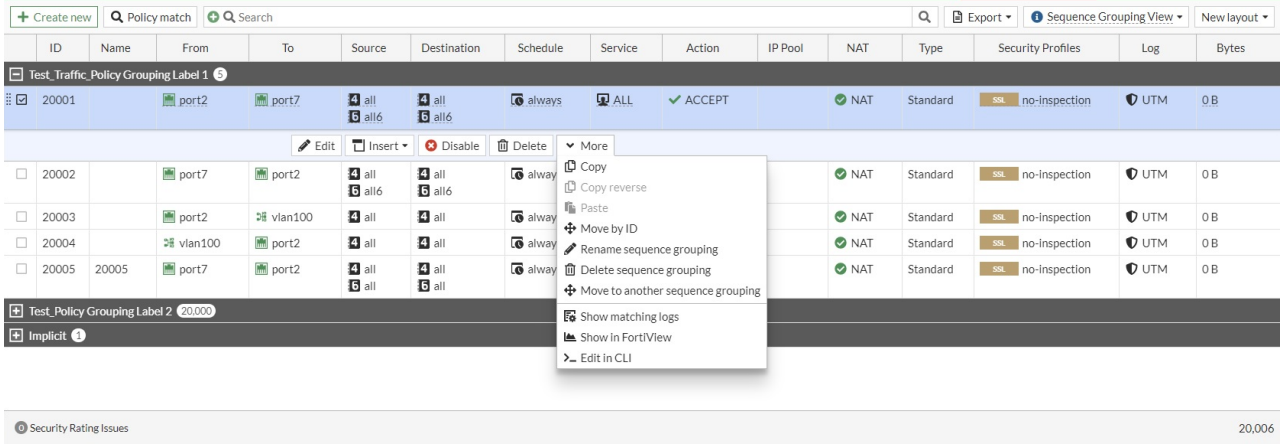
- When a policy is inserted in *Interface Pair View*, the *Incoming Interface* and *Destination Interface* fields will be automatically filled. You can confirm the location of the new policy in the right-side gutter before clicking *OK* to insert the policy.

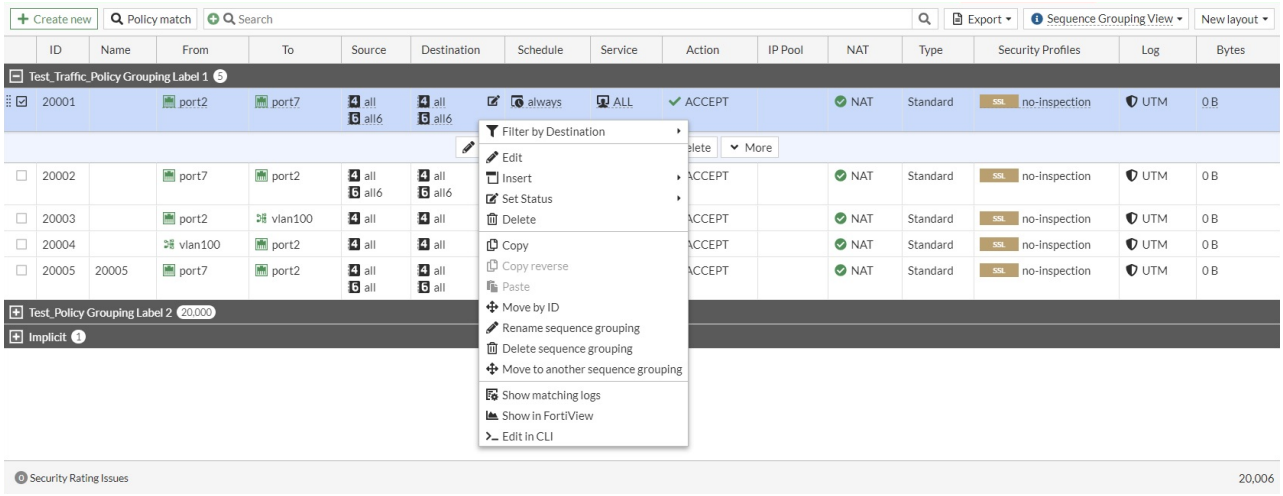


- Multiple policies can be selected at once to efficiently work with a large number of policies.

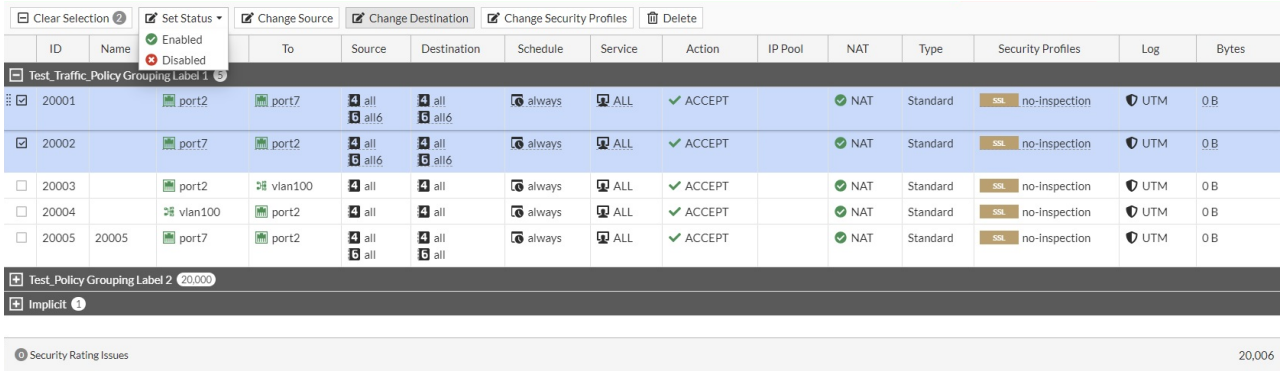


- When a single policy is selected, an inline menu opens below the row. The *More* dropdown menu includes the same expanded list of options that are available in the right-click menu.

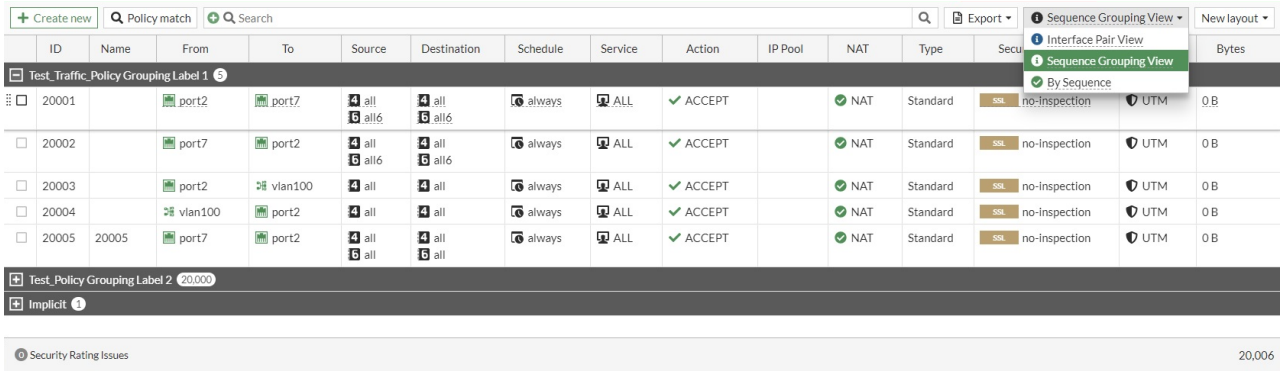




- When multiple policies are selected, the top menu bar changes to show buttons that are applicable to the multiple selections.



- The view selector drop-down includes three options: *Interface Pair View*, *Sequence Grouping View*, and *By Sequence*. For large policy tables (thousands of policies), a tooltip will specify that the *By Sequence* view will load the fastest.



Policy lookup

Firewall policy lookup is based on the `Source_interfaces/Protocol/Source_Address/Destination_Address` that matches the `source-port` and `dst-port` of the protocol. Use this tool to find out which policy matches

specific traffic from a number of policies. After completing the lookup, the matching firewall policy is highlighted on the policy list page.

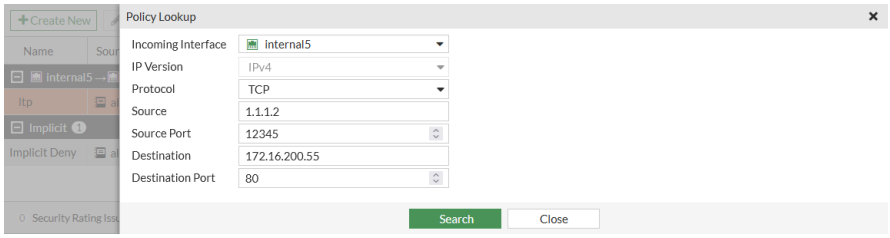
The *Policy Lookup* tool has the following requirements:

- Transparent mode does not support policy lookup function.
- When executing the policy lookup, you need to confirm whether the relevant route required for the policy work already exists.

Sample configuration

This example uses the TCP protocol to show how policy lookup works:

1. On a *Policy & Objects* policy list page, click *Policy Lookup* and enter the traffic parameters.



2. Click *Search* to display the policy lookup results.

Services

Services represent typical traffic types and application packets that pass through the FortiGate. Services include the service protocol type (TCP, UDP, ICMP, and so on), address, category, and logical destination port. Services can then be applied in a firewall policy to represent the TCP/IP suite port numbers that will most commonly be used to transport the named protocols or groups of protocols. Likewise, security profiles use service definitions to match session types.

The following services are available:

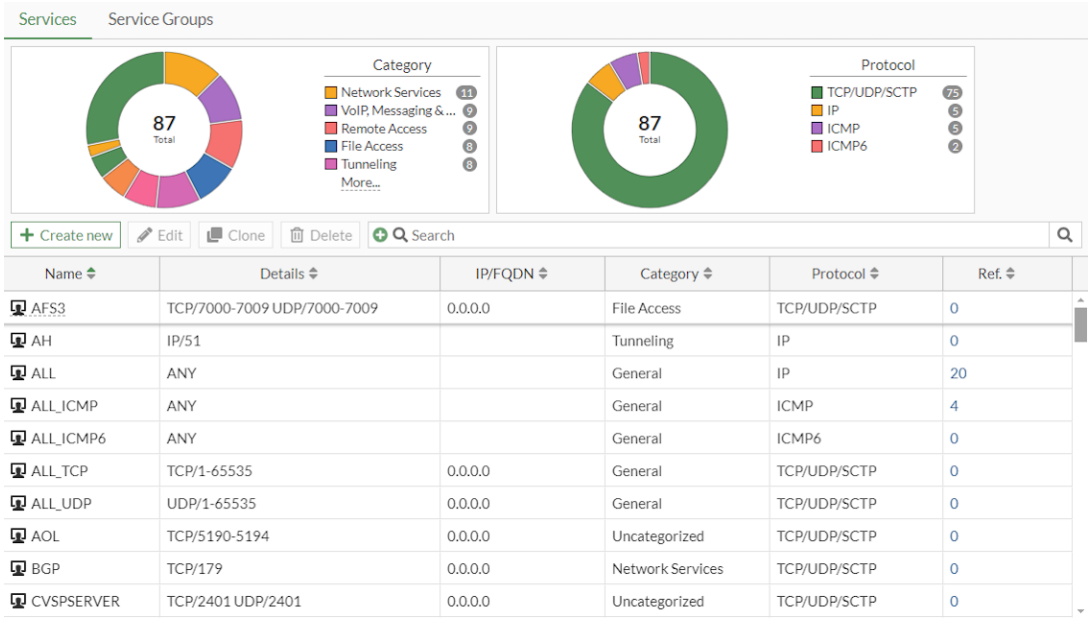
- Predefined services on page 1337
- Custom services on page 1339
- Service groups on page 1340

Predefined services

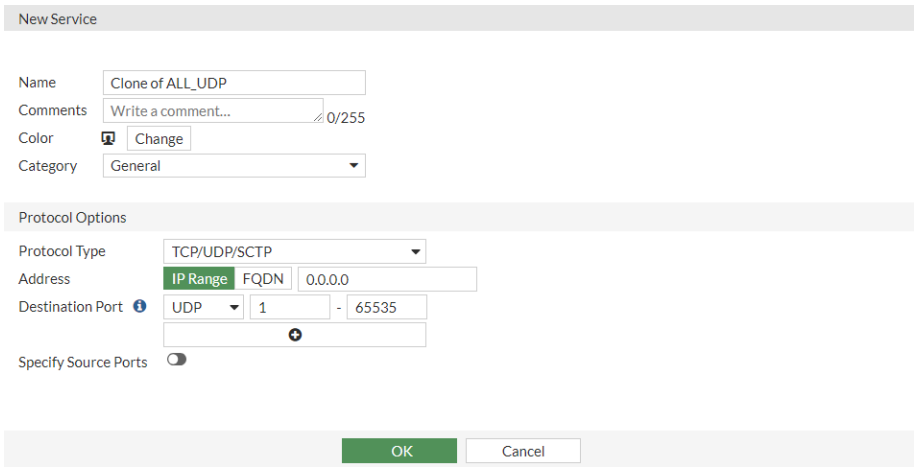
Firewall policies can be configured with default, predefined services that have been created for common traffic types. Predefined services can be edited, cloned, and deleted from the *Policy & Objects* > *Services* list. Cloning a services allows you to create a copy of the service parameters and edit it to create a similar service while still maintaining the existing service.

To clone a service:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Services* tab.



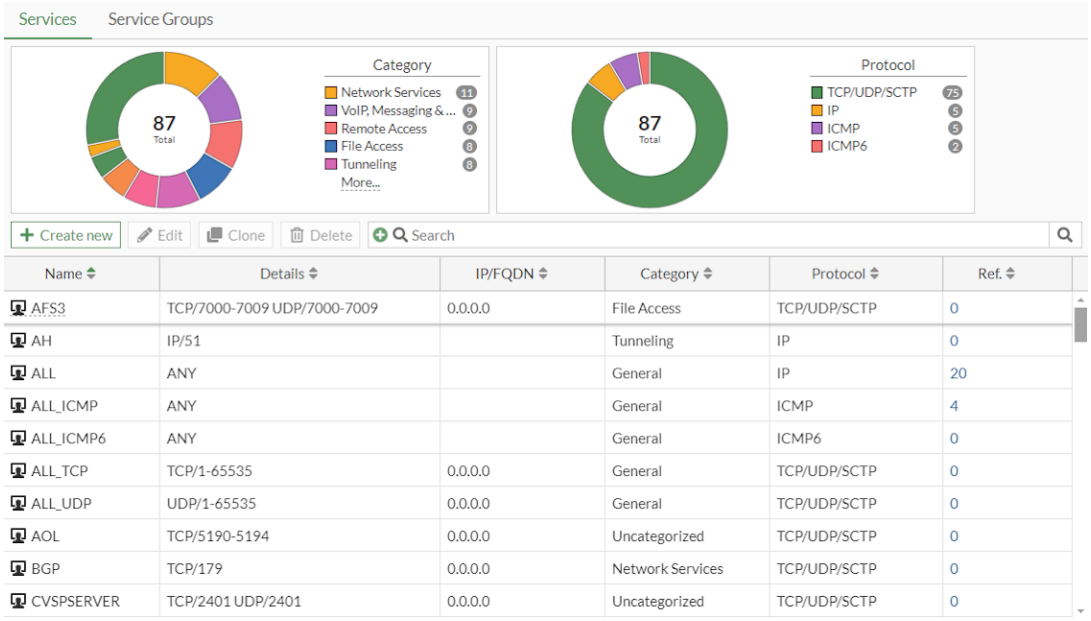
3. Select the service you want to clone.
4. Click *Clone*. The *New Service* page is displayed.



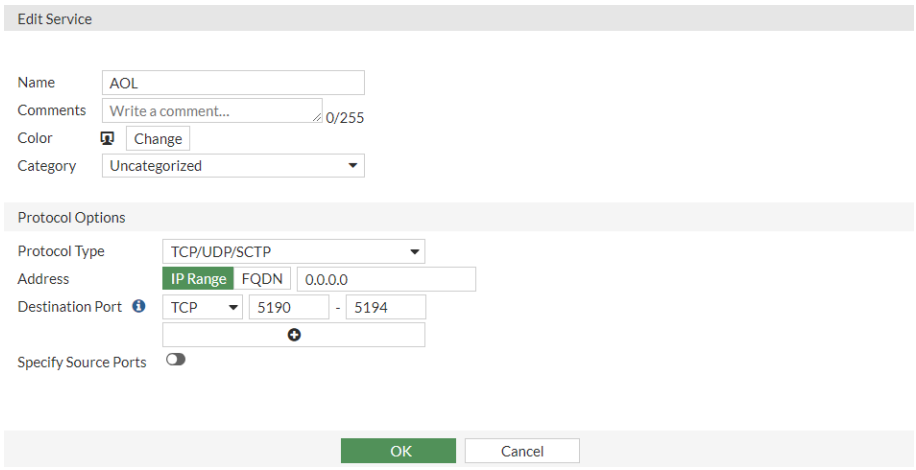
5. Edit the service details as needed.
6. Click *OK*.

To edit a service:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Services* tab.



3. Select the service you want to edit.
4. Click *Edit*. The *Edit Service* page is displayed.



5. Edit the service details as needed.
6. Click *OK*.

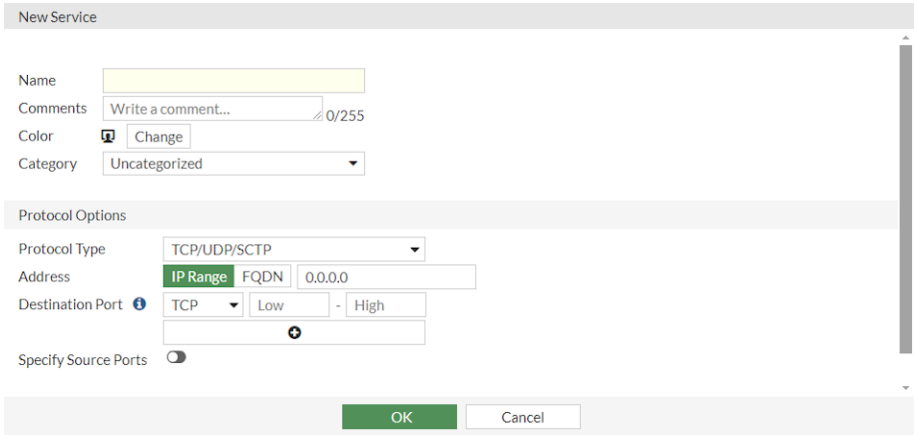
Custom services

You can create new, customized services in the *Policy & Objects* > *Services* page and the CLI. When creating a custom service, the ports, IP addresses, and protocols must be known for proper configuration. Once a service has been created, it must be applied to a firewall policy to take effect.

Custom services can also be created while configuring a new firewall policy.

To configure a custom service in the GUI:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Services* tab.
3. Click *Create new*.
4. Configure the service parameters as needed.



5. Click *OK*.

Custom services can be configured in the CLI for `TCP/UDP/SCTP`, `ICMP`, `ICMP6`, and `IP` protocols. Service parameters are dependent on the protocol type. See config firewall service custom in the CLI Reference guide for more information.

The following example demonstrates configuring a custom service with the `TCP/UDP/SCTP` protocol.

To configure a custom service in the CLI:

```
config firewall service custom
    edit <name>
        set protocol TCP/UDP/SCTP
        set tcp-portrange <destination port range>
        set udp-portrange <destination port range>
        set sctp-portrange <destination port range>
    next
end
```

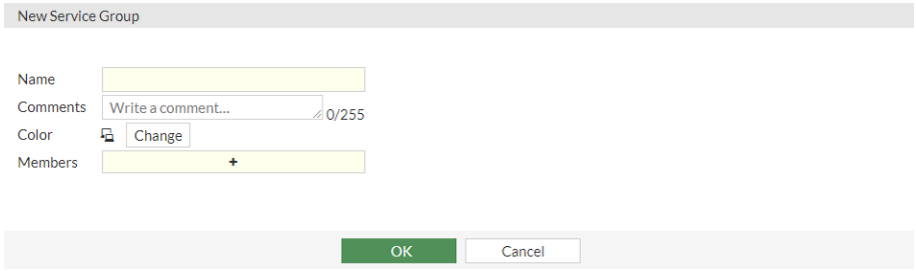
Service groups

Service groups are a collection of services and other service groups, allowing multiple services to be applied in a firewall policy at once.

Service groups can be cloned and edited in the *Service Groups* tab using the same process as services. See Predefined services on page 1337.

To configure a service group in the GUI:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Service Groups* tab.
3. Click *Create new*. The *New Service Group* page is displayed.



4. Enter the *Name*.
5. (Optional) Enter a comment and select a color for the service group.
6. Click the *Members* field and select the services and service groups to include in the group.
7. Click *OK*.

To configure a service group in the CLI:

```
config firewall service group
    edit <name>
        set fabric-object {enable | disable}
        set member <service name1>, <service name2>
        set proxy {enable | disable}
    next
end
```

NGFW policy

Profile-based next-generation firewall (NGFW) mode is the traditional mode where you create a profile (antivirus, web filter, and so on) and then apply the profile to a policy.

In policy-based NGFW mode, you allow applications and URL categories to be used directly in security policies, without requiring web filter or application control profiles. However, it is possible to select and apply web filter URL categories and groups.

In policy-based mode:

- Central NAT is always enabled. If no Central SNAT policy exists, you must create one. See Central SNAT on page 1379 for more information.
- Pre-match rules are defined separately from security policies, and define broader rules, such as SSL inspection and user authentication.
- The IPsec wizard is not supported.

If your FortiGate operates in NAT mode, rather than enabling source NAT in individual NGFW policies, go to *Policy & Objects* > *Central SNAT* and add source NAT policies that apply to all matching traffic. In many cases, you may only need one SNAT policy for each interface pair.

The NGFW mode is set per VDOM, and it is only available when the VDOM inspection mode is flow-based. You can operate your entire FortiGate or individual VDOMs in NGFW policy mode. The application default port can be set as a service port in the NGFW mode using the `default-app-port-as-service` option.

In NGFW mode, administrators can configure a security policy in learn mode to monitor traffic. See Learn mode in security policies in NGFW mode on page 1352 for more information.

Enabling policy-based NGFW mode

To enable policy-based NGFW mode without VDOMs in the GUI:

1. Go to *System* > *Settings*.
2. In *NGFW Mode*, select *Policy-based*.
3. Click *Apply*.

To enable policy-based NGFW mode with VDOMs in the GUI:

1. Go to *System* > *VDOM* .
2. Double-click a VDOM to edit the settings.
3. In *NGFW Mode*, select *Policy-based*.
4. Click *OK*.

To enable policy-based NGFW mode without VDOMs in the CLI:

```
config system settings
    set ngfw-mode policy-based
end
```

To enable policy-based NGFW mode with VDOMs in the CLI:

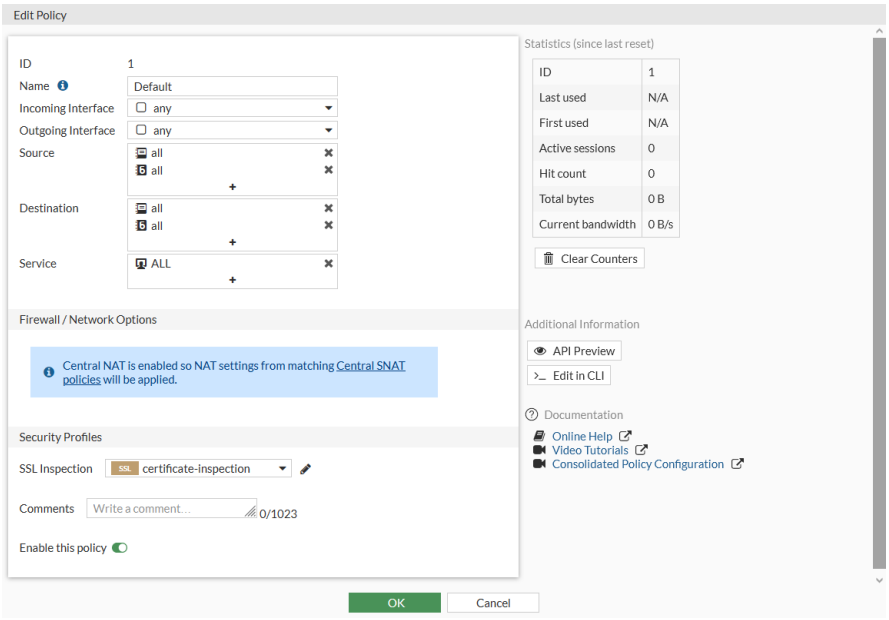
```
config vdom
    edit <vdom>
        config system settings
            set ngfw-mode policy-based
        end
    next
end
```

Security and SSL Inspection & Authentication policies

Security policies work with SSL Inspection & Authentication policies to inspect traffic. To allow traffic from a specific user or user group, both Security and SSL Inspection & Authentication policies must be configured. A default SSL Inspection & Authentication policy with the certificate-inspection SSL Inspection profile is preconfigured. Traffic will match the SSL Inspection & Authentication policy first. If the traffic is allowed, packets are sent to the IPS engine for application, URL category, user, and user group match, and then, if enabled, UTM inspection (antivirus, IPS, DLP, and email filter) is performed.

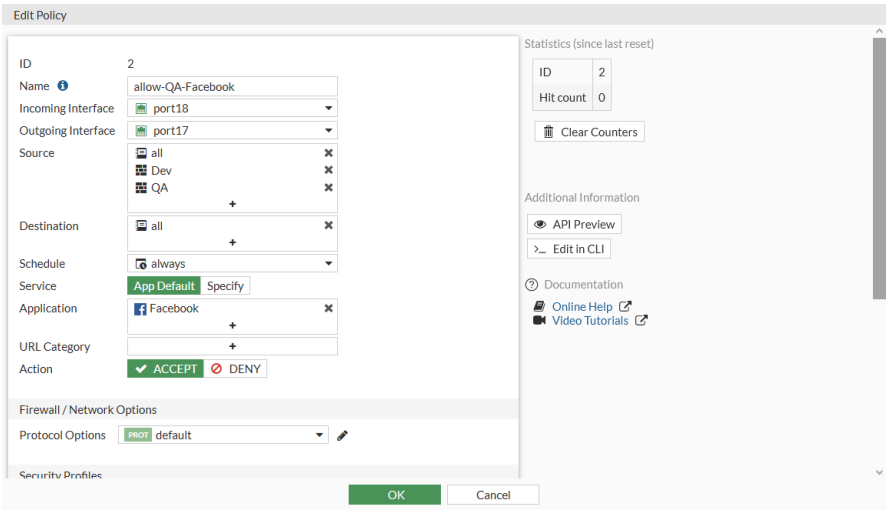
SSL Inspection & Authentication policies are used to pre-match traffic before sending the packets to the IPS engine:

- There are no schedule or action options; traffic matching the policy is always redirected to the IPS engine.
- SSL inspection, formerly configured in the VDOM settings, is configured in an SSL Inspection & Authentication policy.
- Users and user groups that require authentication must be configured in an SSL Inspection & Authentication policy.



Security policies work with SSL Inspection & Authentication policies to inspect traffic:

- Applications and URL categories can be configured directly in the policy.
- Users and user groups that require authentication must also be configured in a security policy.
- The available actions are *Accept* or *Deny*.
- The *Service* option can be used to enforce the standard port for the selected applications.
- UTM inspection is configured in a security policy.



To configure policies for Facebook and Gmail access in the CLI:

1. Configure an SSL Inspection & Authentication policy:

```
config firewall policy
    edit 1
        set name "Policy-1"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "new-deep-inspection"
        set groups "Dev" "HR" "QA" "SYS"
    next
end
```

2. Configure security policies:

```
config firewall security-policy
    edit 2
        set name "allow-QA-Facebook"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set application 15832
        set groups "Dev" "QA"
    next
    edit 4
        set name "allow-QA-Email"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set url-category 23
        set groups "QA"
    next
end
```

Logs

In the application control and web filter logs, `securityid` maps to the security policy ID.

Application control log:

```
date=2019-06-17 time=16:35:47 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1560814547702405829 tz="-0700"
appid=15832 user="Jack" group="QA" srcip=10.1.100.102 dstip=157.240.3.29 srcport=56572
dstport=443 srcintf="port18" srcintfrole="undefined" dstintf="port17"
dstintfrole="undefined" proto=6 service="P2P" direction="incoming" policyid=1
sessionid=42445 appcat="Social.Media" app="Facebook" action="pass" hostname="external-sea1-
1.xx.fbcdn.net" incidentserialno=1419629662 url="/" securityid=2 msg="Social.Media:
```

```
Facebook," apprisk="medium" scertcname="*.facebook.com" scertissuer="DigiCert SHA2 High
Assurance Server CA"
```

Web filter log:

```
date=2019-06-17 time=16:42:41 logid="0317013312" type="utm" subtype="webfilter"
eventtype="ftgd_allow" level="notice" vd="vd1" eventtime=1560814961418114836 tz="-0700"
policyid=4 sessionid=43201 user="Jack" group="QA" srcip=10.1.100.102 srcport=56668
srcintf="port18" srcintfrole="undefined" dstip=172.217.3.165 dstport=443 dstintf="port17"
dstintfrole="undefined" proto=6 service="HTTPS" hostname="mail.google.com"
action="passthrough" reqtype="direct" url="/" sentbyte=709 rcvdbyte=0 direction="outgoing"
msg="URL belongs to an allowed category in policy" method="domain" cat=23 catdesc="Web-based
Email" securityid=4
```

Traffic logs:

```
date=2019-06-17 time=16:35:53 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vd1" eventtime=1560814553778525154 tz="-0700" srcip=10.1.100.102
srcport=56572 srcintf="port18" srcintfrole="undefined" dstip=157.240.3.29 dstport=443
dstintf="port17" dstintfrole="undefined" poluuid="b740d418-8ed3-51e9-5a7b-114e99ab6370"
sessionid=42445 proto=6 action="server-rst" user="Jack" group="QA" policyid=1
policytype="consolidated" centralnatid=1 service="HTTPS" dstcountry="United States"
srccountry="Reserved" trandisp="snat" transip=172.16.200.2 transport=56572 duration=6
sentbyte=276 rcvdbyte=745 sentpkt=5 rcvdpkt=11 appid=15832 app="Facebook"
appcat="Social.Media" apprisk="medium" utmaction="allow" countapp=1 utmref=65531-294

2: date=2019-06-17 time=16:47:45 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vd1" eventtime=1560815265058557636 tz="-0700" srcip=10.1.100.102
srcport=56668 srcintf="port18" srcintfrole="undefined" dstip=172.217.3.165 dstport=443
dstintf="port17" dstintfrole="undefined" poluuid="b740d418-8ed3-51e9-5a7b-114e99ab6370"
sessionid=43201 proto=6 action="timeout" user="Jack" group="QA" policyid=1
policytype="consolidated" centralnatid=1 service="HTTPS" dstcountry="United States"
srccountry="Reserved" trandisp="snat" transip=172.16.200.2 transport=56668 duration=303
sentbyte=406 rcvdbyte=384 sentpkt=4 rcvdpkt=4 appcat="unscanned" utmaction="allow"
countweb=1 utmref=65531-3486
```

Other NGFW policy-based mode options

You can combine *Application Control* and *Web Filter* in the same NGFW mode policy.

The following security profiles can be used in NGFW policy-based mode:

- AntiVirus
- Web Filter
- Intrusion Prevention
- File Filter
- Email Filter

Logging can also be enabled in security policies.

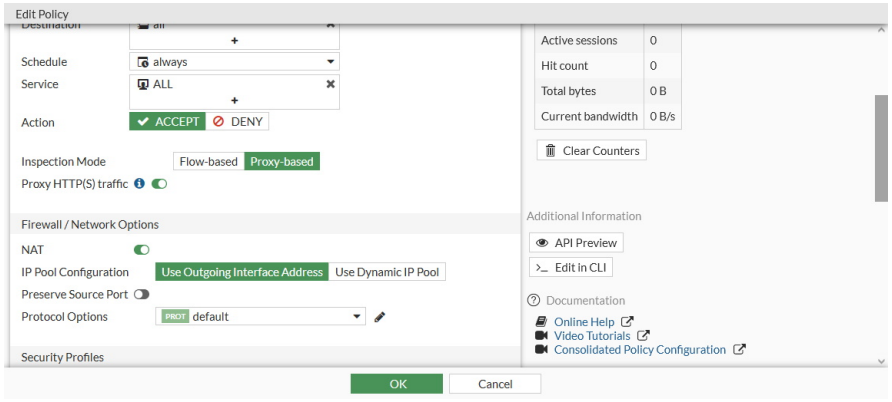
Inspection mode per policy

Inspection mode is configured on a per-policy basis in NGFW mode. This gives you more flexibility when setting up different policies.

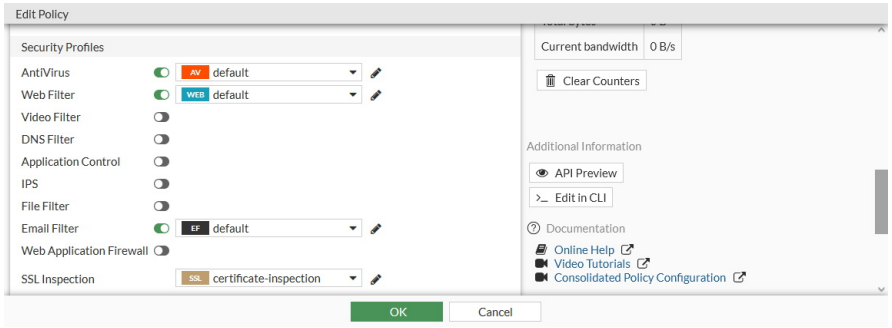
When configuring a firewall policy, you can select a *Flow-based* or *Proxy-basedInspection Mode*. The default setting is *Flow-based*.

To configure inspection mode in a policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy, or edit an existing policy.
3. Configure the policy as needed.
   a. If you change the *Inspection Mode* to *Proxy-based*, the *Proxy HTTP(S) traffic* option displays.



   b. In the *Security Profiles* section, if no security profiles are enabled, the default *SSL Inspection* is *no-inspection*.
   c. In the *Security Profiles* section, if you enable any security profile, the *SSL Inspection* changes to *certificate-inspection*.



To see the inspection mode changes using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set nat enable
    next
end
```

To see the HTTP and SSH policy redirect settings when inspection mode is set to proxy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set http-policy-redirect enable
        set ssh-policy-redirect enable
        set nat enable
    next
end
```

To see the default SSL-SSH policy set to no inspection using the CLI:

```
config firewall policy
    edit 1
       show fu | grep ssl-ssh-profile
        set ssl-ssh-profile "no-inspection"
    next
end
```

NGFW policy mode application default service

In NGFW policy-based mode, the application default service enforces applications running only on their default service port. The applications specified in the policy are monitored, and if traffic is detected from a nonstandard port, it is blocked, and a log entry is recorded with a *port-violation* event type.

If you are not using the default ports, and need to pick specific services, select *Specify* to select the required services.

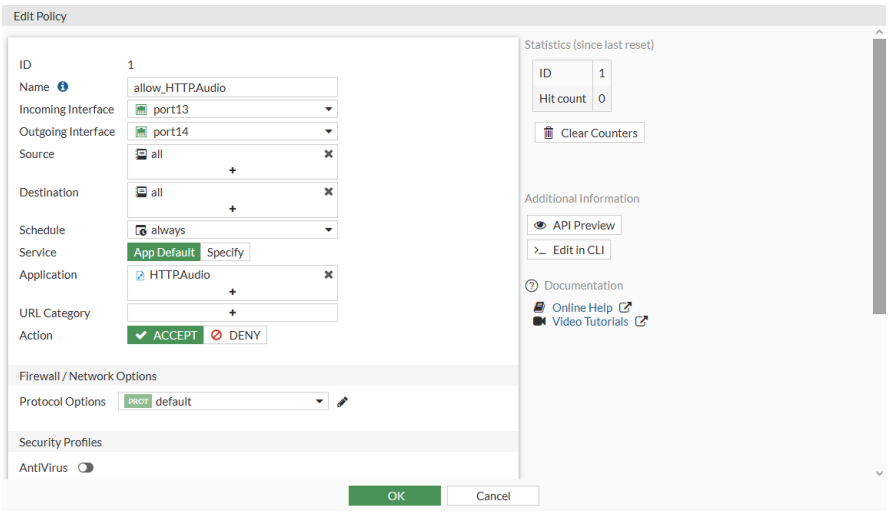
Example

In this example, the standard port is enforced for HTTPS traffic using the HTTP.Audio application.

First, an SSL Inspection & Authentication policy is created do to traffic pre-match, and then a security policy is created to allow the HTTP.Audio application when using the default port. Fetching an MP3 file from an HTTP server using port 443 is allowed, but is blocked when using a nonstandard port, such as 8443.

To enforce the HTTP.Audio application using the default port in the GUI:

1. Create a new SSL Inspection & Authentication policy, or use the default policy.
2. Go to *Policy & Objects* > *Security Policy*, and click *Create New*.
3. Enter a name for the policy, such as *allow_HTTP.Audio*.
4. Configure the ports as needed.
5. Set *Service* to *App Default*.
6. In the *Application* field, select *HTTP.Audio*.
7. Set the *Action* to *Accept*.



8. Click *OK*.

To enforce the HTTP.Audio application using the default port in the CLI:

1. Create a firewall policy:

```
config firewall policy
    edit 1
        set name "consolidated_all"
        set srcintf "port13"
        set dstintf "port14"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "new-deep-inspection"
    next
end
```

2. Create a security policy:

```
config firewall security-policy
    edit 1

        set name "allow_HTTP.Audio"
        set srcintf "port13"
        set dstintf "port14"
        set srcaddr "all"
        set enforce-default-app-port enable
        set action accept
        set schedule "always"
        set logtraffic all
        set application 15879
    next
end
```

Logs

The application logs show logs with an event type of `port-violation` for traffic on port 8443 that is blocked, and an event type of `signature` for traffic on port 443 that is allowed.

Blocked:

```
2: date=2019-06-18 time=16:15:40 logid="1060028736" type="utm" subtype="app-ctrl"
eventtype="port-violation" level="warning" vd="vd1" eventtime=1560899740218875746 tz="-0700"
appid=15879 srcip=10.1.100.22 dstip=172.16.200.216 srcport=52680 dstport=8443
srcintf="port13" srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6
service="HTTPS" direction="incoming" policyid=1 sessionid=5041 appcat="Video/Audio"
app="HTTP.Audio" action="block" hostname="172.16.200.216" incidentserialno=1906780850
url="/app_data/story.mp3" securityid=2 msg="Video/Audio: HTTP.Audio," apprisk="elevated"
```

Allowed:

```
1: date=2019-06-18 time=16:15:49 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1560899749258579372 tz="-0700"
appid=15879 srcip=10.1.100.22 dstip=172.16.200.216 srcport=54527 dstport=443
srcintf="port13" srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6
service="HTTPS" direction="incoming" policyid=1 sessionid=5064 appcat="Video/Audio"
app="HTTP.Audio" action="pass" hostname="172.16.200.216" incidentserialno=1139663486
url="/app_data/story.mp3" securityid=2 msg="Video/Audio: HTTP.Audio," apprisk="elevated"
```

Add option to set application default port as a service port

The `default-app-port-as-service` option can be used in NGFW mode to set the application default port as a service port. This allows applications to match the policy and be blocked immediately the first time that traffic hits the firewall. When this option is enabled, the NGFW policy aggregates the ports used by the applications in the policy and performs a pre-match on the traffic.

```
config system settings
    set default-app-port-as-service {enable | disable}
end
```

This option can be configured on a per-VDOM level.

This setting is enabled by default on new installations. When upgrading, the setting is disabled to retain the previous behavior.

To configure the application default port as service port:

1. Configure the VDOM settings:

```
config system settings
    set vdom-type traffic
    set opmode nat
    set ngfw-mode policy-based
    set block-land-attack disable
    set default-app-port-as-service enable
    set application-bandwidth-tracking disable
end
```

2. Configure the NGFW policy:

```
config firewall security-policy
    edit 1
```

```
        set name "test"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set internet-service-src disable
        set enforce-default-app-port enable
        set action accept
    next
end
```

Sample logs

The following logging behavior occurs in NGFW mode with `default-app-port-as-service`:

- When `default-app-port-as-service` and `enforce-default-app-port` are enabled, traffic that does not match the default port is blocked immediately. Only a traffic log is generated.

  Log with SSH and FTP traffic:

  ```
  1: date=2022-02-24 time=11:16:36 eventtime=1645730197145603994 tz="-0800"
  logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vd1"
  srcip=10.1.100.12 srcport=40402 srcintf="port2" srcintfrole="undefined"
  dstip=172.16.200.55 dstport=21 dstintf="port1" dstintfrole="undefined"
  srccountry="Reserved" dstcountry="Reserved" sessionid=6811 proto=6 action="deny"
  policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-4093cb91e67b"
  policyname="Default" centralnatid=1 service="FTP" trandisp="snat" transip=172.16.200.4
  transport=40402 duration=10 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned"
  ```

  Log with SSH and FTP traffic with port 2121:

  ```
  1: date=2022-02-24 time=11:19:20 eventtime=1645730360685614031 tz="-0800"
  logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vd1"
  srcip=10.1.100.12 srcport=41362 srcintf="port2" srcintfrole="undefined"
  dstip=172.16.200.55 dstport=2121 dstintf="port1" dstintfrole="undefined"
  srccountry="Reserved" dstcountry="Reserved" sessionid=7213 proto=6 action="deny"
  policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-4093cb91e67b"
  policyname="Default" centralnatid=1 service="tcp/2121" trandisp="snat"
  transip=172.16.200.4 transport=41362 duration=9 sentbyte=60 rcvdbyte=0 sentpkt=1
  rcvdpkt=0 appcat="unscanned"
  ```

- When `default-app-port-as-service` is disabled and `enforce-default-app-port` is enabled, traffic that does not match the default port is not blocked immediately. Application and traffic logs are generated.

  Traffic log with SSH and FTP traffic:

  ```
  1: date=2022-02-24 time=11:21:51 eventtime=1645730511325606916 tz="-0800"
  logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vd1"
  srcip=10.1.100.12 srcport=40408 srcintf="port2" srcintfrole="undefined"
  dstip=172.16.200.55 dstport=21 dstintf="port1" dstintfrole="undefined"
  srccountry="Reserved" dstcountry="Reserved" sessionid=7522 proto=6 action="deny"
  policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-4093cb91e67b"
  policyname="Default" centralnatid=1 service="FTP" trandisp="snat" transip=172.16.200.4
  transport=40408 duration=14 sentbyte=164 rcvdbyte=171 sentpkt=3 rcvdpkt=2 appid=15896
  ```

```
app="FTP" appcat="Network.Service" apprisk="elevated" utmaction="block" countapp=1
utmref=65501-0
```

Application log with SSH and FTP traffic:

```
2: date=2022-02-24 time=11:21:39 eventtime=1645730499338228209 tz="-0800"
logid="1059028705" type="utm" subtype="app-ctrl" eventtype="signature" level="warning"
vd="vd1" appid=15896 srcip=10.1.100.12 srccountry="Reserved" dstip=172.16.200.55
dstcountry="Reserved" srcport=40408 dstport=21 srcintf="port2" srcintfrole="undefined"
dstintf="port1" dstintfrole="undefined" proto=6 service="FTP" direction="outgoing"
policyid=0 sessionid=7522 action="block" appcat="Network.Service" app="FTP"
incidentserialno=188744239 msg="Network.Service: FTP" apprisk="elevated"
```

Traffic log with SSH and FTP traffic with port 2121:

```
1: date=2022-02-24 time=11:24:25 eventtime=1645730665235613912 tz="-0800"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vd1"
srcip=10.1.100.12 srcport=41366 srcintf="port2" srcintfrole="undefined"
dstip=172.16.200.55 dstport=2121 dstintf="port1" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=7876 proto=6 action="deny"
policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-4093cb91e67b"
policyname="Default" centralnatid=1 service="tcp/2121" trandisp="snat"
transip=172.16.200.4 transport=41366 duration=11 sentbyte=112 rcvdbyte=171 sentpkt=2
rcvdpkt=2 appid=15896 app="FTP" appcat="Network.Service" apprisk="elevated"
utmaction="block" countapp=1 utmref=65500-0
```

Application log with SSH and FTP traffic with port 2121:

```
2: date=2022-02-24 time=11:24:16 eventtime=1645730656426052412 tz="-0800"
logid="1060028736" type="utm" subtype="app-ctrl" eventtype="port-violation"
level="warning" vd="vd1" appid=15896 srcip=10.1.100.12 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcport=41366 dstport=2121 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 service="FTP"
direction="outgoing" policyid=0 sessionid=7876 action="block" appcat="Network.Service"
app="FTP" incidentserialno=188744241 msg="Network.Service: FTP, non-default port used:
2121" apprisk="elevated"
```

Application logging in NGFW policy mode

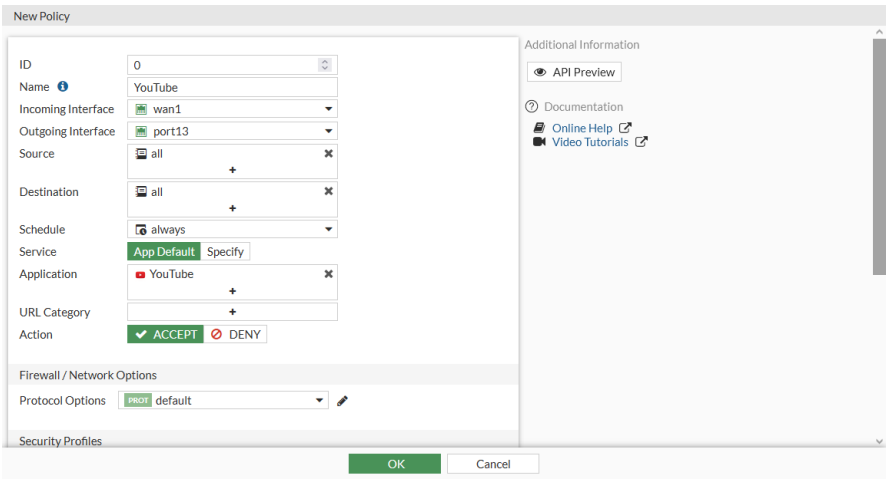
In NGFW policy mode, if an application, application category, or application group is selected on a security policy, and traffic logging is set to *UTM* or *All*, then application control logs will be generated. In addition, when a signature is set to the *ACCEPT* action under a security policy, all corresponding child signatures will be assessed and logged as well.

Under NGFW, with `default-app-port-as-service` enabled, enable APP Default. The traffic which doesn't match the default port will be blocked immediately, and there is only traffic log generated.

Under NGFW, with `default-app-port-as-service` disabled, enable APP Default. The traffic which doesn't match the default port will not be blocked immediately, and there is app and traffic logs generated.

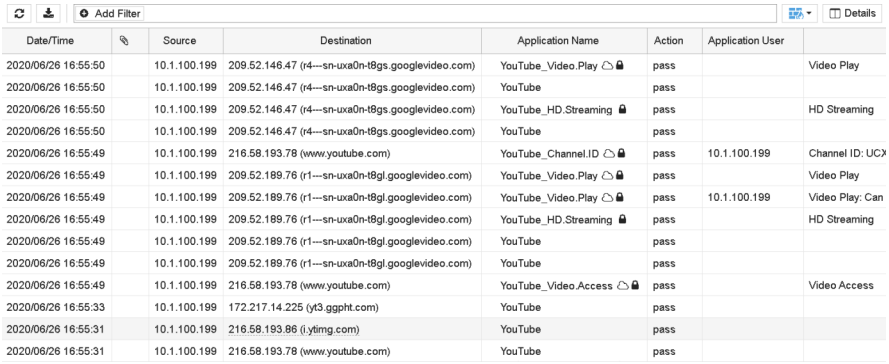
To verify application logging:

1. Go to *Policy & Objects* > *Security Policy* and configure a new policy for YouTube.
2. Set *Action* to *ACCEPT* and *Log Allowed Traffic* to *Security Events*.



3. Configure the remaining settings as required, then click *OK*.
4. On a client system, play some YouTube videos.
5. On FortiOS, go to *Log & Report* > *Security Events* and view the *Application Control* logs.

   There are logs not only for *YouTube*, but also for *YouTube_Video.Play*, *YouTube_Video.Access*, and so on, as verified from the *Application Name* column.



Date/Time		Source	Destination	Application Name	Action	Application User	
2020/06/26 16:55:50		10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube_Video.Play	pass		Video Play
2020/06/26 16:55:50		10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:50		10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube_HD.Streaming	pass		HD Streaming
2020/06/26 16:55:50		10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49		10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube_Channel.ID	pass	10.1.100.199	Channel ID: UCX
2020/06/26 16:55:49		10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_Video.Play	pass		Video Play
2020/06/26 16:55:49		10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_Video.Play	pass	10.1.100.199	Video Play: Can
2020/06/26 16:55:49		10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_HD.Streaming	pass		HD Streaming
2020/06/26 16:55:49		10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49		10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49		10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube_Video.Access	pass		Video Access
2020/06/26 16:55:33		10.1.100.199	172.217.14.225 (yt3.ggpht.com)	YouTube	pass		
2020/06/26 16:55:31		10.1.100.199	216.58.193.86 (i.ytimg.com)	YouTube	pass		
2020/06/26 16:55:31		10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube	pass		

Learn mode in security policies in NGFW mode

In NGFW mode, administrators can configure a security policy in learn mode to monitor traffic that passes through the source and destination interfaces. The learn mode uses a special prefix in the `policymode` and `profile` fields in traffic and UTM logs for use by FortiAnalyzer and the Policy Analyzer Management Extension Application (MEA) that is available with FortiManager.

When enabled on FortiManager, Policy Analyzer MEA works with security policies in learning mode to analyze logs sent from a managed FortiGate to FortiAnalyzer. Based on the analyzed traffic, FortiManager administrators can choose to automatically create a policy in FortiManager for the managed FortiGate. For more information about Policy Analyzer MEA, see the Policy Analyzer Administration Guide.

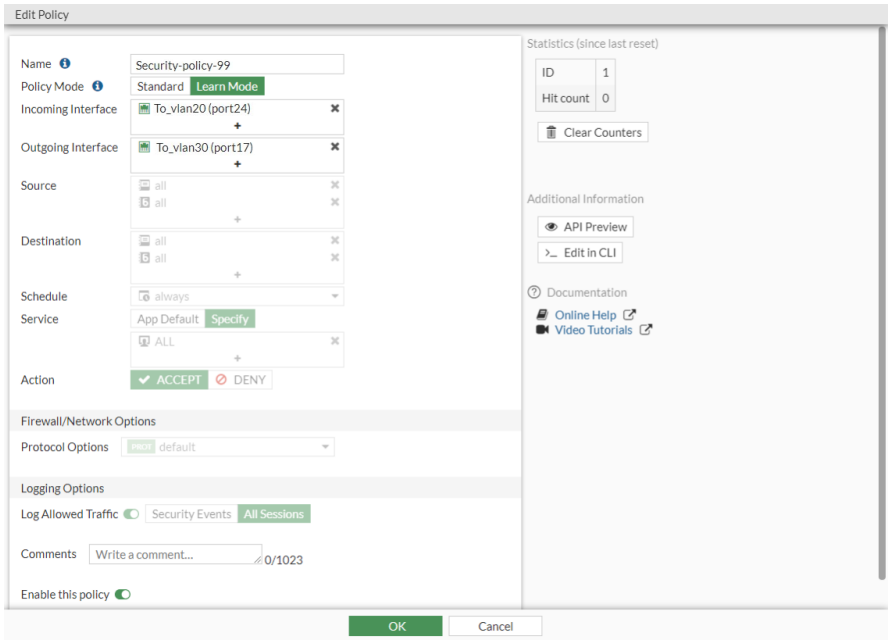
The following limitations apply when learn mode is enabled in a security policy:

- Only interfaces with `device-identification enable` can be used as source interfaces in a security policy with learning mode enabled.

- Incoming and outgoing interfaces do not support `any`.
- Internet service is not supported.
- NAT46 and NAT64 are not supported.
- Users and groups are not supported.
- Some negate options are not supported.

To enable learn mode in the GUI:

1. Enable policy-based NGFW mode:
   a. Go to *System* > *Settings*.
   b. Set the *NGFW Mode* to *Policy-based* and click *Apply*.
2. Go to *Policy & Objects* > *Security Policy*, and open a security policy for editing.
3. Set the *Policy Mode* to *Learn Mode*.



4. Select an *Incoming Interface*.
5. Select an *Outgoing Interface*.
6. (Optional) Type a comment in the *Comments* box.
7. Toggle on *Enable this policy*.
8. Click *OK* to save the security policy.

To enable learn mode in the CLI:

1. Enable policy-based NGFW mode:

```
config system settings
    set ngfw-mode policy-based
end
```

2. Enable learn mode in a security policy:

```
config firewall security-policy
    edit <id>
        set learning-mode enable
    next
end
```

To view learn mode fields in logs in the CLI:

1. Filter and view fields in traffic logs:

```
# execute log filter category 0

# execute log display

1 logs found.

1 logs returned.

1: date=2022-03-21 time=10:21:11 eventtime=1647883271150012188 tz="-0700"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="root"
srcip=10.1.100.41 srcport=43296 srcintf="port24" srcintfrole="undefined"
dstip=172.16.200.55 dstport=80 dstintf="port17" dstintfrole="wan"
srccountry="Reserved" dstcountry="Reserved" sessionid=33934 proto=6
policymode="learn" action="accept" policyid=99 policytype="security-policy"
poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c" policyname="Security-policy-99"
centralnatid=3 service="HTTP" trandisp="snat" transip=172.16.200.9 transport=43296
duration=1 sentbyte=412 rcvdbyte=529 sentpkt=6 rcvdpkt=4 appid=15893
app="HTTP.BROWSER" appcat="Web.Client" apprisk="medium" utmaction="allow"
countweb=1 countav=1 countips=3 countapp=1 crscore=50 craction=2
srchwvendor="VMware" devtype="Computer" osname="Debian"
mastersrcmac="00:0c:29:b5:92:8d" srcmac="00:0c:29:b5:92:8d" srcserver=0
utmref=65534-0
```

2. Filter and view fields in UTM logs:

```
# execute log filter category 2

# execute log display

1 logs found.

1 logs returned.

1: date=2022-03-21 time=10:21:09 eventtime=1647883270101403283 tz="-0700"
logid="0211008193" type="utm" subtype="virus" eventtype="infected" level="notice"
vd="root" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c"
policytype="security-policy" policymode="learn" msg="File is infected."
action="monitored" service="HTTP" sessionid=33934 srcip=10.1.100.41
dstip=172.16.200.55 srcport=43296 dstport=80 srccountry="Reserved"
dstcountry="Reserved" srcintf="port24" srcintfrole="undefined" dstintf="port17"
dstintfrole="wan" proto=6 direction="incoming" filename="eicar.com"
```

```
quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-
engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE" virusid=2172
url="http://172.16.200.55/virus/eicar.com" profile="learn-av" agent="curl/7.35.0"
httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical" rawdata="Response-
Content-Type=application/x-msdos-program"
```

3. Filter and view fields in UTM-IPS logs:

```
# execute log filter category 4

# execute log display

3 logs found.

3 logs returned.

1: date=2022-03-21 time=10:21:09 eventtime=1647883270101485354 tz="-0700"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="info" srcip=10.1.100.41 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcintf="port24" srcintfrole="undefined"
dstintf="port17" dstintfrole="wan" sessionid=33934 action="detected" proto=6
service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c"
policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 agent="curl/7.35.0" httpmethod="GET" direction="incoming"
attackid=29844 profile="learn-ips" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=158335134 attackcontextid="2/2"
attackcontext="YW0NCg0KWDVPIVAlQEFQWzRcUFpYNTQoUF4pN0NDKTd9JEVJQ0FSLVNUQU5EQVJELUFO
VElWSVJVUy1URVNULUZJTEUhJEgrSCo8L1BBQ0tFVD4="

2: date=2022-03-21 time=10:21:09 eventtime=1647883270101484791 tz="-0700"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="info" srcip=10.1.100.41 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcintf="port24" srcintfrole="undefined"
dstintf="port17" dstintfrole="wan" sessionid=33934 action="detected" proto=6
service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c"
policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 agent="curl/7.35.0" httpmethod="GET" direction="incoming"
attackid=29844 profile="learn-ips" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=158335134 attackcontextid="1/2"
attackcontext="PFBBVFRFUk5TPiBYNU8hUCVAQVBbNFxQWlg1NChQXik3Q0MpN30kRUlDQVItU1RBTkRB
UkQtQU5USVZJUlVTLVRFU1QtRklMRSEkSCtIKjtYNU8hUCVAQVBbNFxQWlg1NChQXik3Q0MpN30kRUlDQVI
tU1RBTkRBUkQtQU5USVZJUlVTLVRFU1QtRklMRSEkSCtIKjwvUEFUVEVSTlM+CjxVUkk+IDwvVVJJPgo8SE
VBREVSPiBIVFRQLzEuMSAyMDAgT0sNCkRhdGU6IE1vbiwgMjEgTWFyIDIwMjIgMTc6MjE6MTAgR01UDQpTZ
XJ2ZXI6IEFwYWNoZS8yLjQuMTggKFVidW50dSkNCkxhc3QtTW9kaWZpZWQ6IFRodSwgMDEgRGVjIDIwMTYg
MDE6MjY6MzUgR01UDQpFVGFnOiAiNDQtNTQyOGViNjU4MDk3YSINCkFjY2VwdC1SYW5nZXM6IGJ5dGVzDQp
Db250ZW50LUxlbmd0aDogNjgNCkNvbnRlbnQtVHlwZTogYXBwbGljYXRpb24veC1tc2Rvcy1wcm9ncmFtDQ
oNCjwvSEVBREVSPgo8Qk9EWT4gWDVPIVAlQEFQWzRcUFpYNTQoUF4pN0NDKTd9JEVJQ0FSLVNUQU5EQVJEL
```

```
UFOVElWSVJVUy1URVNULUZJTEUhJEgrSCo8L0JPRFk+CjxQQUNLRVQ+IEhUVFAvMS4xIDIwMCBPSw0KRGF0
ZTogTW9uLCAyMSBNYXIgMjAyMiAxNzoyMToxMCBHTVQNClNlcnZlcjogQXBhY2hlLzIuNC4xOCAoVWJ1bnR
1KQ0KTGFzdC1Nb2RpZmllZDogVGh1LCAwMSBEZWMgMjAxNiAwMToyNjozNSBHTVQNCkVUYWc6ICI0NC01ND
I4ZWI2NTgwOTdhIg0KQWNjZXB0LVJhbmdlczogYnl0ZXMNCkNvbnRlbnQtTGVuZ3RoOiA2OA0KQ29udGVud
C1UeXBlOiBhcHBsaWNhdGlvbi94LW1zZG9zLXByb2dy"

3: date=2022-03-21 time=10:21:09 eventtime=1647883270101483279 tz="-0700"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="info" srcip=10.1.100.41 srccountry="Reserved"
dstip=172.16.200.55 dstcountry="Reserved" srcintf="port24" srcintfrole="undefined"
dstintf="port17" dstintfrole="wan" sessionid=33934 action="detected" proto=6
service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c"
policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 hostname="172.16.200.55" url="/virus/eicar.com"
agent="curl/7.35.0" httpmethod="GET" direction="incoming" attackid=29844
profile="learn-ips" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=158335134 msg="file_transfer: Eicar.Virus.Test.File"
attackcontextid="0/2" rawdataid="1/1" rawdata="Response-Content-Type=application/x-
msdos-program"
```

Filter and view fields in UTM-webfilter logs:

```
# execute log filter category 3

# execute log display

2 logs found.

2 logs returned.

2: date=2022-03-21 time=10:21:09 eventtime=1647883270100329681 tz="-0700"
logid="0319013317" type="utm" subtype="webfilter" eventtype="urlmonitor" level="notice"
vd="root" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c"
policytype="security-policy" policymode="learn" sessionid=33934 srcip=10.1.100.41
srcport=43296 srccountry="Reserved" srcintf="port24" srcintfrole="undefined"
dstip=172.16.200.55 dstport=80 dstcountry="Reserved" dstintf="port17" dstintfrole="wan"
proto=6 httpmethod="GET" service="HTTP" hostname="172.16.200.55" agent="curl/7.35.0"
profile="learn-webf" action="passthrough" reqtype="direct"
url="http://172.16.200.55/virus/eicar.com" sentbyte=92 rcvdbyte=0 direction="outgoing"
msg="URL has been visited" ratemethod="domain" cat=255 catdesc="Unknown"
```

Dynamic address tags

Tags for dynamic addresses, including EMS (normal and local EMS tags), FortiPolicy, FortiVoice, and FortiNAC can be used as the source or destination address in security policies. Once these tags are used in security policies, run `diagnose ips pme dynamic-address list` to show the addresses that are used in the policy. The following example uses an EMS tag.

To apply an EMS tag object to a security policy in the GUI:

1. Go to *Policy & Objects* > *Security Policy*.
2. Click *Create new* or edit an existing policy.
3. In the *Source* field, click the + and select *EMS1_ZTNA_ZT_OS_WIN*.
4. Configure the other settings as needed.
5. Click *OK*.

To apply an EMS tag object to a security policy in the CLI:

1. Configure the policy:

```
config firewall security-policy
    edit 1
        set name "ddd"
        set srcintf "port8"
        set dstintf "port7"
        set srcaddr "EMS1_ZTNA_ZT_OS_WIN"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

2. Verify which IP addresses are used in the policy:

```
# diagnose ips pme dynamic-address list
EMS1_ZTNA_ZT_OS_WIN [vdom=0 type=IP]:
```

Local-in policy

While security profiles control traffic flowing through the FortiGate, local-in policies control inbound traffic that is going to a FortiGate interface.

Administrative access traffic (HTTPS, PING, SSH, and others) can be controlled by allowing or denying the service in the interface settings. Trusted hosts can be configured under an administrator to restrict the hosts that can access the administrative service.

Local-in policies allow administrators to granularly define the source and destination addresses, interface, and services. Traffic destined for the FortiGate interface specified in the policy that meets the other criteria is subject to the policies action.

Local-in policies can be used to restrict administrative access or other services, such as VPN, that can be specified as services. You can define source addresses or address groups to restrict access from. For example, by using a geographic type address you can restrict a certain geographic set of IP addresses from accessing the FortiGate. An IP Address threat feed can also be used as either a source or destination address; see Applying an IP address threat feed in a local-in policy on page 3509 for more information.

Local-in policies can also use virtual patching to mitigate known vulnerabilities targeted at the FortiGate. Vulnerability rules are scanned on local-in traffic on the specified interface, and all matched local-in traffic is dropped accordingly. See Virtual patching on the local-in management interface on page 1448 for more information.

Local-in policies can only be created or edited in the CLI. You can view the existing local-in policies in the GUI by enabling it in *System* > *Feature Visibility* under the *Additional Features* section. This page does not list the custom local-in policies.

To configure a local-in policy using the CLI:

```
config firewall {local-in-policy | local-in-policy6}
    edit <policy_number>
        set intf <interface>
        set srcaddr <source_address> [source_address] ...
        set dstaddr <destination_address> [destination_address] ...
        set action {accept | deny}
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set virtual-patch {enable | disable}
        set comments <string>
    next
end
```

For example, to prevent the source subnet 10.10.10.0/24 from pinging port1, but allow administrative access for PING on port1:

```
config firewall address
    edit "10.10.10.0"
        set subnet 10.10.10.0 255.255.255.0
    next
end
config firewall local-in-policy
    edit 1
        set intf "port1"
        set srcaddr "10.10.10.0"
        set dstaddr "all"
        set service "PING"
        set schedule "always"
    next
end
```

To test the configuration:

1. From the PC at 10.10.10.12, start a continuous ping to port1:

   ```
   ping 192.168.2.5 –t
   ```

2. On the FortiGate, enable debug flow:

   ```
   # diagnose debug flow filter addr 10.10.10.12
   # diagnose debug flow filter proto 1
   # diagnose debug enable
   # diagnose debug flow trace start 10
   ```

3. The output of the debug flow shows that traffic is dropped by local-in policy 1:

   ```
   # id=20085 trace_id=1 func=print_pkt_detail line=5746 msg="vd-root:0 received a packet
   (proto=1, 10.10.10.12:1->192.168.2.5:2048) from port1. type=8, code=0, id=1, seq=128."
   id=20085 trace_id=1 func=init_ip_session_common line=5918 msg="allocate a new session-
   0017c5ad"
   ```

```
id=20085 trace_id=1 func=vf_ip_route_input_common line=2615 msg="find a route:
flag=80000000 gw-192.168.2.5 via root"
id=20085 trace_id=1 func=fw_local_in_handler line=474 msg="iprope_in_check() check
failed on policy 1, drop"
```

Implicit deny rule

If a local-in-policy is not functioning correctly and traffic that should be blocked is being allowed through, the issue may be that the implicit deny local-in-policy has not been created. Unlike IPv4 policies, there is no default implicit deny policy. The implicit deny policy should be placed at the bottom of the list of local-in-policies. Local-in-policies are created for each interface, but if you want to create a general implicit deny rule for all interfaces for a specific service, source, address, or destination address, use the `any` interface.

By default, no local-in policies are defined, so there are no restrictions on local-in traffic. When you define a local-in policy, if no action is set manually, then the action will default to `deny`.

For example, to allow only the source subnet 172.16.200.0/24 to ping port1:

```
config firewall address
    edit "172.16.200.0"
        set subnet 172.16.200.0 255.255.255.0
    next
end
config firewall local-in-policy
    edit 2
        set intf "port1"
        set srcaddr "172.16.200.0"
        set dstaddr "all"
        set action accept
        set service "PING"
        set schedule "always"
    next
    edit 3
        set intf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "PING"
        set schedule "always"
    next
end
```

To test the configuration:

1. From the PC at 172.16.200.2, start a continuous ping to port1:

   ```
   ping 172.16.200.1 -t
   ```

2. On the FortiGate, enable debug flow:

   ```
   # diagnose debug flow filter proto 1
   # diagnose debug enable
   # diagnose debug flow trace start 10
   ```

3. The output of the debug flow shows that ping traffic coming from the 172.16.200.0 subnet is allowed:

```
# id=65308 trace_id=25 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.16.200.2:5->172.16.200.1:2048) tun_id=0.0.0.0 from port1. type=8, code=0,
id=5, seq=0."
id=65308 trace_id=25 func=init_ip_session_common line=6121 msg="allocate a new session-
00029409, tun_id=0.0.0.0"
id=65308 trace_id=25 func=__vf_ip_route_input_rcu line=2012 msg="find a route:
flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=25 func=ip_session_confirm_final line=3189 msg="npu_state=0x0, hook=1"
id=65308 trace_id=26 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.16.200.1:5->172.16.200.2:0) tun_id=0.0.0.0 from local. type=0, code=0,
id=5, seq=0."
id=65308 trace_id=26 func=resolve_ip_tuple_fast line=6027 msg="Find an existing session,
id-00029409, reply direction"
id=65308 trace_id=27 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.16.200.2:5->172.16.200.1:2048) tun_id=0.0.0.0 from port1. type=8, code=0,
id=5, seq=1."
id=65308 trace_id=27 func=resolve_ip_tuple_fast line=6027 msg="Find an existing session,
id-00029409, original direction"
id=65308 trace_id=28 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.16.200.1:5->172.16.200.2:0) tun_id=0.0.0.0 from local. type=0, code=0,
id=5, seq=1."
```

4. From the PC at 172.20.120.13, start a continuous ping to port1:

```
ping 172.16.200.1 -t
```

5. The output of the debug flow shows that ping traffic coming from subnets other than 172.16.200.0 is dropped by local-in policy 3:

```
# id=65308 trace_id=21 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.20.120.13:1->172.16.200.1:2048) tun_id=0.0.0.0 from port2. type=8, code=0,
id=1, seq=8."
id=65308 trace_id=21 func=init_ip_session_common line=6121 msg="allocate a new session-
0002929d, tun_id=0.0.0.0"
id=65308 trace_id=21 func=__vf_ip_route_input_rcu line=2012 msg="find a route:
flag=80000000 gw-0.0.0.0 via root"
id=65308 trace_id=21 func=__iprope_tree_check line=520 msg="gnum-100004, use int hash,
slot=51, len=2"
id=65308 trace_id=21 func=fw_local_in_handler line=545 msg="iprope_in_check() check
failed on policy 3, drop"
```

Additional options

To disable or re-enable the local-in policy, use the `set status {enable | disable}` command.

To dedicate the interface as an HA management interface, use the `set ha-mgmt-intf-only enable` command.

Example:

```
config firewall local-in-policy
    edit 1
        set ha-mgmt-intf-only enable
        set intf port4
        set srcaddr all
        set dstaddr all
```

```
        set service ALL
        set schedule always
        set action accept
        set status enable
    next
end
```

If a user tries to set the HA reserved management interface during the local-in policy an error is generated. Use the `set ha-mgmt-intf-only enable` command to avoid the error.

TTL policies

You can configure a time-to-live (TTL) policy to block attack traffic with high TTLs. This feature only applies to local-in traffic and does not apply to traffic passing through the FortiGate. You can use `srcintf` to set the interface that the local-in traffic hits. See config firewall ttl-policy.

To configure a TTL policy using the CLI:

```
config firewall ttl-policy
    edit <id>
        set status {enable | disable}
        set action {accept | deny}
        set srcintf <interface>
        set srcaddr <source_address> [source_address] ...
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set ttl <value/range>
    next
end
```

Internet service as source addresses - NEW

An internet service can be used as the source address in a local-in policy. This allows for more flexibility and control when managing local traffic, enhancing network security and efficiency.

```
config firewall local-in-policy
    edit <id>
        set internet-service-src {enable | disable}
        set internet-service-src-name <string>
        set internet-service-src-group <string>
        set internet-service-src-custom <string>
        set internet-service-src-custom-group <string>
        set internet-service-src-negate {enable | disable}
    next
end
```

internet-service-src {enable | disable}	Enable/disable use of Internet Services in source for this local-in policy. If enabled, the source address is not used.
internet-service-src-name <string>	Internet Service source name.

internet-service-src-group <string>	Internet Service source group name.
internet-service-src-custom <string>	Custom Internet Service source name.
internet-service-src-custom-group <string>	Custom Internet Service source group name.
internet-service-src-negate {enable | disable}	When enabled, `internet-service-src` specifies what the service must NOT be.

DoS policy

A Denial of Service (DoS) policy examines network traffic arriving at a FortiGate interface for anomalous patterns, which usually indicates an attack.

A denial of service occurs when an attacking system starts an abnormally large number of sessions with a target system. The large number of sessions slows down or disables the target system, preventing legitimate users from using it.

DoS policies are checked before security policies, preventing attacks from triggering more resource intensive security protection and slowing down the FortiGate.

DoS anomalies

Predefined sensors are setup for specific anomalous traffic patterns. New DoS anomalies cannot be added by the user.

The predefined anomalies that can be used in DoS policies are:

Anomaly	Description	Recommended Threshold
tcp_syn_flood	If the SYN packet rate of new TCP connections, including retransmission, to one destination IP address exceeds the configured threshold value, the action is executed.	2000 packets per second.
tcp_port_scan	If the SYN packet rate of new TCP connections, including retransmission, from one source IP address exceeds the configured threshold value, the action is executed.	1000 packets per second.
tcp_src_session	If the number of concurrent TCP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
tcp_dst_session	If the number of concurrent TCP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
udp_flood	If the UDP traffic to one destination IP address exceeds the configured threshold value, the action is executed.	2000 packets per second.

Anomaly	Description	Recommended Threshold
udp_scan	If the UDP sessions setup rate originating from one source IP address exceeds the configured threshold value, the action is executed.	2000 sessions per second.
udp_src_session	If the number of concurrent UDP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
udp_dst_session	If the number of concurrent UDP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
icmp_flood	If the number of ICMP packets sent to one destination IP address exceeds the configured threshold value, the action is executed.	250 packets per second.
icmp_sweep	If the ICMP sessions setup rate originating from one source IP address exceeds the configured threshold value, the action is executed.	100 sessions per second.
icmp_src_session	If the number of concurrent ICMP connections from one source IP address exceeds the configured threshold value, the action is executed.	300 concurrent sessions
icmp_dst_session	If the number of concurrent ICMP connections to one destination IP address exceeds the configured threshold value, the action is executed.	1000 concurrent sessions
ip_src_session	If the number of concurrent IP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
ip_dst_session	If the number of concurrent IP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
sctp_flood	If the number of SCTP packets sent to one destination IP address exceeds the configured threshold value, the action is executed.	2000 packets per second
sctp_scan	If the number of SCTP sessions originating from one source IP address exceeds the configured threshold value, the action is executed.	1000 packets per second
sctp_src_session	If the number of concurrent SCTP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions
sctp_dst_session	If the number of concurrent SCTP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions

For thresholds based on the number of concurrent sessions, blocking the anomaly will not allow more than the number of concurrent sessions to be set as the threshold.

For example, if the period for a particular anomaly is 60 seconds, such as those where the threshold is measured in concurrent sessions, after the 60 second timer has expired the number of allowed sessions that match the anomaly criteria is reset to zero. This means that, if you allow 10 sessions through before blocking, after the 60 seconds has elapsed, another 10 sessions will be allowed. The attrition of sessions from expiration should keep the allowed sessions from reaching the maximum.

For rate based thresholds, where the threshold is measured in packets per second, the *Block* action prevents anomalous traffic from overwhelming the firewall in two ways:

- continuous: Block packets once an anomaly is detected, and continue to block packets while the rate is above the threshold. This is the default setting.
- periodical: After an anomaly is detected, allow the configured number of packets per second.

For example, if a DoS policy is configured to block icmp_flood with a threshold of 10pps, and a continuous ping is started at a rate of 20pps for 1000 packets:

- In continuous mode, the first 10 packets are passed before the DoS sensor if triggered, and then the remaining 990 packets are blocked.
- In periodical mode, 10 packets are allowed to pass per second, so 500 packets are blocked in the 50 seconds during which the ping is occuring.

The actual numbers of passed and blocked packets may not be exact, as fluctuations in the rates can occur, but the numbers should be close to the defined threshold.

To configure the block action for rate based anomaly sensors:

```
config ips global
    set anomaly-mode {continuous | periodical}
end
```

DoS policies

A DoS policy can be configured to use one or more anomalies.

To configure a DoS policy in the GUI:

1. Go to *Policy & Objects* > *IPv4 DoS Policy* or *Policy & Objects* > *IPv6 DoS Policy* and click *Create New*.
   If the option is not visible, enable *DoS Policy* in *Feature Visibility*. See Feature visibility on page 3062 for details.

2. Configure the following:

Name	Enter a name for the policy.
Incoming Interface	Enter the interface that the policy applies to.
Source Address	Enter the source address.
Destination Address	Enter the destination address. This is the address that the traffic is addressed to. In this case, it must be an address that is associated with the firewall interface. For example, it could be an interface address, a secondary IP address, or the address assigned to a VIP address.
Service	Select the services or service groups. The ALL service can used or, to optimize the firewall resources, only the services that will be answered on an interface can be used.

L3 Anomalies
L4 Anomalies
	Configure the anomalies:
	• *Logging*: Enable/disable logging for specific anomalies or all of them. Anomalous traffic will be logged when the action is *Block* or *Monitor*.
	• *Action*: Select the action to take when the threshold is reached:
	  • *Disable*: Do not scan for the anomaly.
	  • *Block*: Block the anomalous traffic.
	  • *Monitor*: Allow the anomalous traffic, but record a log message if logging is enabled.
	• *Threshold*: The number of detected instances that triggers the anomaly action.

Comments	Optionally, enter a comment.

3. Enable the policy, then click *OK*.

The quarantine option is only available in the CLI. See Quarantine on page 1366 for information.

To configure a DoS policy in the GUI:

```
config firewall DoS-policy
    edit 1
        set name "Flood"
        set interface "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set quarantine attacker
                set quarantine-expiry 1d1h1m
```

```
                set quarantine-log enable
                set threshold 100
            next
        end
    next
end
```

`name <string>`	Enter a name for the policy.
`interface <string>`	Enter the interface that the policy applies to.
`srcaddr <string>`	Enter the source address.
`dstaddr <string>`	Enter the destination address. This is the address that the traffic is addressed to. In this case, it must be an address that is associated with the firewall interface. For example, it could be an interface address, a secondary IP address, or the address assigned to a VIP address.
`service <string>`	Enter the services or service groups. The `ALL` service can used or, to optimize the firewall resources, only the services that will be answered on an interface can be used.
`status {enable | disable}`	Enable/disable this anomaly.
`log {enable | disable}`	Enable/disable anomaly logging. When enabled, a log is generated whenever the anomaly action is triggered, regardless of which action is configured.
`action {pass | block}`	Set the action to take when the threshold is reached: • `pass`: Allow traffic, but record a log message if logging is enabled. • `block`: Block traffic if this anomaly is found.
`quarantine {none | attacker}`	Set the quarantine method (see Quarantine on page 1366): • `none`: Disable quarantine. • `attacker`: Block all traffic from the attacker's IP address, and add the attacker's IP address to the banned user list.
`quarantine-expiry <###d##h##m>`	Set the duration of the quarantine, in days, hours, and minutes (###d##h##m) (1m - 364d23h59m, default = 5m). This option is available if `quarantine` is set `attacker`.
`quarantine-log {enable | disable}`	Enable/disable quarantine logging (default = disable). This option is available if `quarantine` is set `attacker`.
`threshold <integer>`	The number of detected instances - packets per second or concurrent session number - that triggers the anomaly action.

Quarantine

Quarantine is used to block any further traffic from a source IP address that is considered a malicious actor or a source of traffic that is dangerous to the network. Traffic from the source IP address is blocked for the duration of the quarantine, and the source IP address is added to the banned user list.

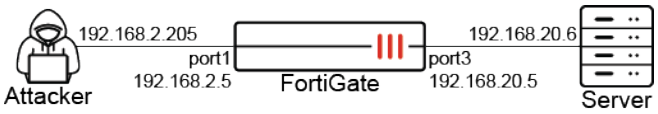
The banned user list is kept in the kernel, and used by Antivirus, Data Loss Prevention (DLP), DoS, and Intrusion Prevention System (IPS). Any policies that use any of these features will block traffic from the attacker's IP address.

To view the quarantined user list:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
192.168.2.205     Wed Nov 25 12:47:54 2020 Wed Nov 25 12:57:54 2020 DOS
```

Troubleshooting DoS attacks

The best way to troubleshoot DoS attacks is with Anomaly logs and IPS anomaly debug messages.



To test an icmp_flood attack:

1. From the Attacker, launch an icmp_flood with 50pps lasting for 3000 packets.
2. On the FortiGate, configure continuous mode and create a DoS policy with an icmp_flood threshold of 30pps:

```
config firewall DoS-policy
    edit 1
        set name icmpFlood
        set interface "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set threshold 30
            next
        end
    next
end
```

3. Configure the debugging filter:

```
# diagnose ips anomaly config
DoS sensors in kernel vd 0:
DoS id 1 proxy 0
  0 tcp_syn_flood status 0 log 0 nac 0 action 0 threshold 2000
  ...
  7 udp_dst_session status 0 log 0 nac 0 action 0 threshold 5000
  8 icmp_flood status 1 log 1 nac 0 action 7 threshold 30
  9 icmp_sweep status 0 log 0 nac 0 action 0 threshold 100
  ...
total # DoS sensors: 1.

# diagnose ips anomaly filter id 8
```

4. Launch the icmp_flood from a Linux machine. This example uses Nmap:

```
$ sudo nping --icmp --rate 50 -c 3000 192.168.2.50
SENT (0.0522s) ICMP [192.168.2.205 > 192.168.2.50 Echo request (type=8/code=0) id=8597
seq=1] IP [ttl=64 id=47459 iplen=28 ]
...
Max rtt: 11.096ms | Min rtt: 0.028ms | Avg rtt: 1.665ms
Raw packets sent: 3000 (84.000KB) | Rcvd: 30 (840B) | Lost: 2970 (99.00%)
Nping done: 1 IP address pinged in 60.35 seconds
```

5. During the attack, check the anomaly list on the FortiGate:

```
# diagnose ips anomaly list
list nids meter:
id=icmp_flood         ip=192.168.2.50 dos_id=1 exp=998 pps=46 freq=50

total # of nids meters: 1.
```

id=icmp_flood	The anomaly name.
ip=192.168.2.50	The IP address of the host that triggered the anomaly. It can be either the client or the server. For icmp_flood, the IP address is the destination IP address. For icmp_sweep, it would be the source IP address.
dos_id=1	The DoS policy ID.
exp=998	The time to be expired, in jiffies (one jiffy = 0.01 seconds).
pps=46	The number of packets that had been received when the diagnose command was executed.
freq=50	For session based anomalies, freq is the number of sessions. For packet rate based anomalies (flood, scan): • In continuous mode: freq is the greater of pps, or the number of packets received in the last second. • In periodic mode: freq is the pps.

6. Go to *Log & Report* > *Security Events* and download the *Anomaly* logs:

```
date=2020-11-20 time=14:38:39 eventtime=1605911919824184594 tz="-0800"
logid="0720018433" type="utm" subtype="anomaly" eventtype="anomaly" level="alert"
vd="root" severity="critical" srcip=192.168.2.205 srccountry="Reserved"
dstip=192.168.2.50 srcintf="port1" srcintfrole="undefined" sessionid=0 action="clear_
session" proto=1 service="PING" count=1307 attack="icmp_flood" icmpid="0x2195"
icmptype="0x08" icmpcode="0x00" attackid=16777316 policyid=1 policytype="DoS-policy"
ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly: icmp_flood, 31 > threshold
30, repeats 28 times" crscore=50 craction=4096 crlevel="critical"

date=2020-11-20 time=14:39:09 eventtime=1605911949826224056 tz="-0800"
logid="0720018433" type="utm" subtype="anomaly" eventtype="anomaly" level="alert"
vd="root" severity="critical" srcip=192.168.2.205 srccountry="Reserved"
dstip=192.168.2.50 srcintf="port1" srcintfrole="undefined" sessionid=0 action="clear_
session" proto=1 service="PING" count=1497 attack="icmp_flood" icmpid="0x2195"
icmptype="0x08" icmpcode="0x00" attackid=16777316 policyid=1 policytype="DoS-policy"
ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly: icmp_flood, 50 > threshold
30, repeats 1497 times" crscore=50 craction=4096 crlevel="critical"
```

Analysis

In the first log message:

msg="anomaly: icmp_flood, 31 > threshold 30	At the beginning of the attack, a log is recorded when the threshold of 30pps is broken.
repeats 28 times	The number of packets that has exceeded the threshold since the last time a log was recorded.
srcip=192.168.2.205 dstip=192.168.2.50	The source and destination IP addresses of the attack.
action="clear_session"	Equivalent to block. If `action` was set to `monitor` and logging was enabled, this would be `action="detected"`.

In the second log message:

- Because it is an ongoing attack, the FortiGate generates one log message for multiple packets every 30 seconds..
- It will not generate a log message if:
  - The same attack ID happened more than once in a five second period, or
  - The same attack ID happened more than once in a 30 second period and the actions are the same and have the same source and destination IP addresses.

msg="anomaly: icmp_flood, 50 > threshold 30	In the second before the log was recorded, 50 packets were detected, exceeding the configured threshold.
repeats 1497 times	The number of packets that has exceeded the threshold since the last time a log was recorded

Access control lists

An access control list (ACL) is a granular, targeted blocklist that is used to block IPv4 and IPv6 packets on a specified interface based on the criteria configured in the ACL policy.

On FortiGate models with ports that are connected through an internal switch fabric with TCAM capabilities, ACL processing is offloaded to the switch fabric and does not use CPU resources. VLAN interfaces that are based on physical switch fabric interfaces are also supported. Interfaces that are connected through an internal switch fabric usually have names prefixed with *port* or *lan*, such as *port1* or *lan2*; other interfaces are not supported.

The packets will be processed by the CPU when offloading is disabled or not possible, such as when a port on a supported model does not connect to the internal fabric switch.

ACL is supported on the following FortiGate models:

- 100D, 100E, 100EF, 101E
- 140D, 140D-POE, 140E, 140E-POE
- 1500D, 1500DT
- 3000D, 3100D, 3200D, 3700D, 3800D
- All 300E and larger E-series models
- All 100F and larger F-series models

Example

To block all IPv4 and IPv6 telnet traffic from port2 to Company_Servers:

```
config firewall acl
    edit 1
       set interface "port2"
       set srcaddr "all"
       set dstaddr "Company_Servers"
       set service "TELNET"
    next
end
config firewall acl6
    edit 1
        set interface "port2"
        set srcaddr "all"
        set dstaddr "Company_Servers_v6"
        set service "TELNET"
    next
end
```

Diagnose commands

To check the number of packets dropped by an ACL:

```
# diagnose firewall acl counter
ACL id 1 dropped 0 packets
```

```
# diagnose firewall acl counter6
ACL id 2 dropped 0 packets
```

To clear the packet drop counters:

```
# diagnose firewall acl clearcounter
```

```
# diagnose firewall acl clearcounter6
```

Interface policies

Interface policies are implemented before the security policies and are only flow-based. They are configured in the CLI.

This feature allows you to attach a set of IPS policies with the interface instead of the forwarding path, so packets can be delivered to IPS before entering the firewall. This feature is used for following IPS deployments:

- One-Arm: By defining interface policies with IPS and DoS anomaly checks and enabling sniff-mode on the interface, the interface can be used for one-arm IDS.
- IPv6 IPS: IPS inspection can be enabled through interface IPv6 policy.
- Scan traffic that is destined to the FortiGate.
- Scan and log traffic that are silently dropped or flooded by Firewall or Multicast traffic.

IPS sensors can be assigned to an interface policy. Both incoming and outgoing packets are inspected by IPS sensor (signature).

To configure an interface policy:

```
config firewall interface-policy
    edit 1
        set status enable
        set comments 'test interface policy #1'
        set logtraffic utm
        set interface "port2"
        set srcaddr all
        set dstaddr all
        set service "ALL"
        set application-list-status disable
        set ips-sensor-status disable
        set dsri disable
        set av-profile-status enable
        set av-profile default
        set webfilter-profile-status disable
    next
end
```

Source NAT

Network Address Translation (NAT) is the process that enables a single device, such as a router or firewall, to act as an agent between the internet or public network and a local or private network. This agent acts in real-time to translate the source or destination IP address of a client or server on the network interface. Source IP translation enables a single, public address to represent a significantly larger number of private addresses. Destination IP translation enables the firewall to translate a public, destination address to a private address. So we don't have to configure a real public IP address for the server deployed in a private network.

NAT can be subdivided into two types:

- Source NAT (SNAT)
- Destination NAT (DNAT)

This section is about SNAT. Three NAT working modes are supported: static SNAT, dynamic SNAT, and central SNAT. For information about DNAT, see Destination NAT on page 1394.

The following topics provide instructions on configuring policies with source NAT:

- Static SNAT on page 1372
- Dynamic SNAT on page 1372
- Central SNAT on page 1379
- Configuring an IPv6 SNAT policy on page 1390
- SNAT policies with virtual wire pairs on page 1392
- Configuring PCP port mapping with SNAT and DNAT on page 1453

Static SNAT

In static SNAT all internal IP addresses are always mapped to the same public IP address. This is a port address translation, Since we have 60416 available port numbers, this one public IP address can handle the conversion of 60,416 internal IP addresses to the same service, where a service is defined by a specified protocol, destination IP address, and destination port.

Internal Source IP	Source Port	Translated Source IP	Translated Source Port
10.1.100.1	11110	172.16.200.1	5117
10.1.100.1	11111	172.16.200.1	5118
10.1.100.2	11112	172.16.200.1	5119
.....	.....	172.16.200.1	.....
.....	.....	172.16.200.1	65533

FortiGate firewall configurations commonly use the Outgoing Interface address.

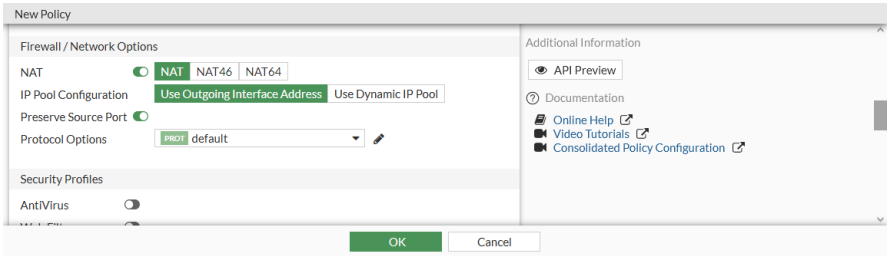
Sample configuration

The following example of static SNAT uses an internal network with subnet 10.1.100.0/24 (vlan20) and an external/ISP network with subnet 172.16.200.0/24 (vlan30).

When the clients in internal network need to access the servers in external network, We need to translate IP addresses from 10.1.100.0/24 to an IP address 172.16.200.0/24, In this example, we implement static SNAT by creating a firewall policy.

To configure static NAT:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the required policy parameters.
3. Enable *NAT* and select *Use Outgoing Interface Address*. For packets that match this policy, its source IP address is translated to the IP address of the outgoing interface.
4. If needed, enable *Preserve Source Port* to keep the same source port for services that expect traffic to come from a specific source port. Disable *Preserve Source Port* to allow more than one connection through the firewall for that service.



5. Click *OK*.

Dynamic SNAT

Dynamic SNAT maps the private IP addresses to the first available public address from a pool of addresses. In the FortiGate firewall, this can be done by using IP pools. IP pools is a mechanism that allows sessions leaving the FortiGate firewall to use NAT. An IP pool defines a single IP address or a range of IP addresses to be used as the source address

for the duration of the session. These assigned addresses are used instead of the IP address assigned to that FortiGate interface.

IP pool types

FortiGate uses four types of IPv4 IP pools. This topic focuses on some of the differences between them.

Overload

This type of IP pool is similar to static SNAT mode. We need to define an external IP range that contains one or more IP addresses. When there is only one IP address it is almost the same as static SNAT, the outgoing interface address is used. When it contains multiple IP addresses, it is equivalent to an extended mode of static SNAT.

For instance, if we define an overload type IP pool with two external IP addresses (172.16.200.1—172.16.200.2), since there are 60,416 available port numbers per IP, this IP pool can handle 60,416*2 internal IP addresses to the same service, where a service is defined by a specific protocol, destination IP address, and destination port.

Original Source IP	Original Source Port	Translated Source IP	Translated Source Port
10.1.100.1	11110	172.16.200.1	5117
10.1.100.2	11111	172.16.200.1	5118
.....	.....	172.16.200.1	.....
.....	.....	172.16.200.1	65533
.....	.....	172.16.200.2	5117
.....	.....	.....	.....
.....	.....	172.16.200.2	65533

The mapped IP address can be calculated from the source IP address. The index number of the address in the pool is the remainder of the source IP address, in decimal, divided by the number addresses in the pool.

To calculate the decimal value of the source IP address, either use an online calculator, or use the following equation:

$$a.b.c.d = a * (256)^3 + b * (256)^2 + c * (256) + d$$

For example:

$$192.168.0.1 = 192 * (256)^3 + 168 * (256)^2 + 0 * (256) + 1 = 3232235521$$

If there is one IP pool, where:

- P_1 = the first address in the IP pool
- R_1 = the number of IP addresses in the IP pool
- X = the source IP address as a decimal number
- Y = the mapped IP address

Then the equation to determine the mapped address is:

$$Y = P_1 + X \bmod R_1$$

For example:

IP pool	Source IP address
172.26.73.20 to 172.26.73.90	192.168.1.200

1. Convert the source IP address to a decimal number:
$$192 * (256)^3 + 168 * (256)^2 + 1 * (256) + 200 = 3232235976$$
2. Determine the number of IP addresses in the pool:
$$172.26.73.90 - 172.26.73.20 = 71$$
3. Find the remainder of the source IP address divided by the number of addresses in the pool:
$$3232235976 \bmod 71 = 26$$
4. Add the remainder to the first IP address in the pool:
$$172.26.73.20 + 26 = 172.26.73.46$$
So, the mapped IP address is **172.26.73.46**.

If there are multiple IP pools, the calculation is similar to when there is only one pool.

If there are two IP pools, where:

- P_1 = the first address in the first IP pool
- P_2 = the first address in the second IP pool
- R_1 = the number of IP addresses in the first IP pool
- R_2 = the number of IP addresses in the second IP pool
- X = the source IP address as a decimal number
- Y = the mapped IP address

Then the equations to determine the mapped address are:

If $X \bmod (R_1 + R_2) >= R_1$, then $Y = P_2 + X \bmod R_2$

If $X \bmod (R_1 + R_2) < R_1$, then $Y = P_1 + X \bmod R_1$

For example:

IP pools	Source IP address
pool01: 172.26.73.20 to 172.26.73.90	192.168.1.200
pool02: 172.26.75.50 to 172.26.75.150	

1. Convert the source IP address to a decimal number:
$$192 * (256)^3 + 168 * (256)^2 + 1 * (256) + 200 = 3232235976$$
2. Determine the total number of IP addresses in the pools:
$$(172.26.73.90 - 172.26.73.20) + (172.26.75.50 - 172.26.75.150) = 71 + 101 = 172$$
3. Find the remainder of the source IP address divided by the number of addresses in the pools:
$$3232235976 \bmod 172 = 108$$
4. The remainder is greater than the number of addresses in pool01, so the address is selected from pool02 and the remainder is recalculated based only on pool02:
$$3232235976 \bmod 101 = 40$$
5. Add the new remainder to the first IP address in pool02:
$$172.26.75.50 + 40 = 172.26.75.90$$
So, the mapped IP address is **172.26.75.90**.

One-to-one

This type of IP pool means that the internal IP address and the external (translated) IP address match one-to-one. The port address translation (PAT) is disabled when using this type of IP pool. For example, if we define a one-to-one type IP pool with two external IP addresses (172.16.200.1 - 172.16.200.2), this IP pool only can handle two internal IP addresses.

Fixed port range

For the overload and one-to-one IP pool types, we do not need to define the internal IP range. For the fixed port range type of IP pool, we can define both internal IP range and external IP range. Since each external IP address and the number of available port numbers is a specific number, if the number of internal IP addresses is also determined, we can calculate the port range for each address translation combination. So we call this type fixed port range. This type of IP pool is a type of port address translation (PAT).

For instance, if we define one external IP address (172.16.200.1) and ten internal IP addresses (10.1.100.1-10.1.100.10), we have translation IP+Port combination like following table:

Original Source IP	Original Source Port	Translated Source IP	Translated Source Port Range
10.1.100.1	.....	172.16.200.1	5117~11157
10.1.100.2	.....	172.16.200.1	11158~17198
10.1.100.3	.....	172.16.200.1	.....
10.1.100.4	.....	172.16.200.1	.....
10.1.100.5	.....	172.16.200.1	.....
10.1.100.6	.....	172.16.200.1	.....
10.1.100.7	.....	172.16.200.1	.....
10.1.100.8	.....	172.16.200.1	.....
10.1.100.9	.....	172.16.200.1	53445~59485
10.1.100.10	.....	172.16.200.1	59486~65526

Port block allocation

This type of IP pool is also a type of port address translation (PAT). It gives users a more flexible way to control the way external IPs and ports are allocated. Users need to define *Block Size/Block Per User* and external IP range. *Block Size* means how many ports each Block contains. *Block per User* means how many blocks each user (internal IP) can use.

The following is a simple example:

- **External IP Range**: 172.16.200.1—172.16.200.1
- **Block Size**: 128
- **Block Per User**: 8

Result:

- **Total-PBAs**: 472 (60416/128)
- **Maximum ports can be used per User (Internal IP Address)**: 1024 (128*8)
- **How many Internal IP can be handled**: 59 (60416/1024 or 472/8)

Interim logs can be configured for port block allocation (PBA) NAT logging. This enables continuous access to PBA event logs during an ongoing session, and provides comprehensive logging throughout a session's lifespan.

PBA event logs are generated periodically based on the configured time interval:

```
config firewall ippool
    edit pba-ippool
        set type port-block-allocation
        set pba-interim-log <integer>
    next
end
```

For example, when the PBA interim log interval is set to 600 seconds, event logs are obtained every ten minutes:

- Configure the PBA IP pool with a time interval:

```
config firewall ippool
    edit "pba-ippool"
        set type port-block-allocation
        set startip 172.16.200.151
        set endip 172.16.200.151
        set block-size 64
        set num-blocks-per-user 1
        set pba-interim-log 600
    next
end
```

- Check the event logs:

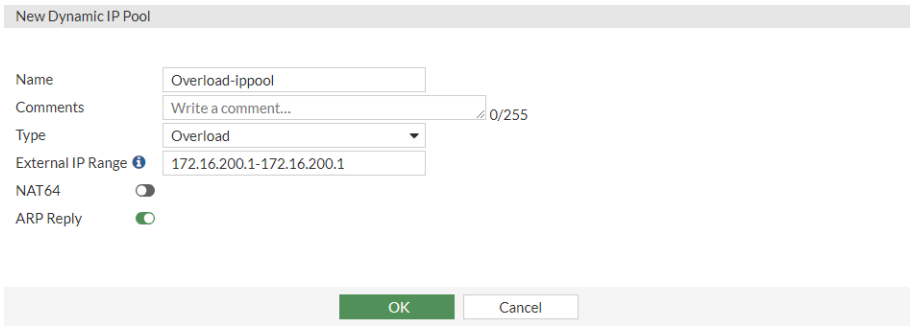
```
# execute log display

2 logs found.

2 logs returned.

1: date=2024-02-04 time=13:34:04 eventtime=1707082444264865326 tz="-0800"
logid="0100022024" type="event" subtype="system" level="notice" vd="vdom1" logdesc="IP
pool PBA interim log" action="ippool-interim" saddr="10.1.100.42" nat=172.16.200.151
portbegin=5117 portend=5180 poolname="pba-ippool" duration=1200 msg="IPpool interim"

2: date=2024-02-04 time=13:24:03 eventtime=1707081844204865060 tz="-0800"
logid="0100022024" type="event" subtype="system" level="notice" vd="vdom1" logdesc="IP
pool PBA interim log" action="ippool-interim" saddr="10.1.100.42" nat=172.16.200.151
portbegin=5117 portend=5180 poolname="pba-ippool" duration=600 msg="IPpool interim"
```

Sample configuration

When an IP pool object is created with *ARP Reply* enabled, the object does not need to be referenced in any policies before a FortiGate interface starts responding to ARP requests for the addresses in the IP pool.

To configure overload IP pool in the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Overload*.
4. Enter the external IP range separated by a hyphen (*172.16.200.1-172.16.200.1*).



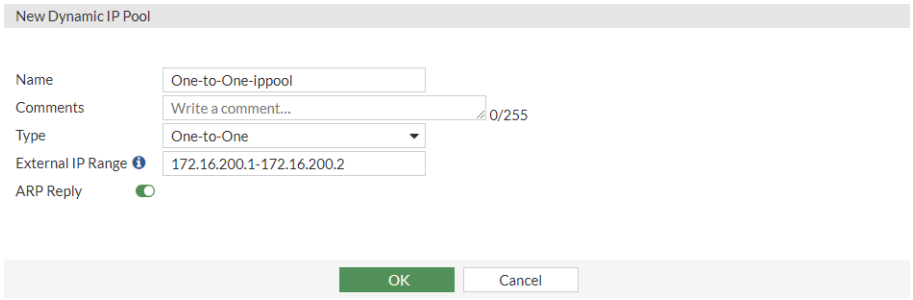
5. Click *OK*.

To configure overload IP pool in the CLI:

```
config firewall ippool
    edit "Overload-ippool"
        set startip 172.16.200.1
        set endip 172.16.200.1
    next
end
```

To configure one-to-one IP pool using the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *One-to-One*.
4. Enter the external IP range separated by a hyphen (*172.16.200.1-172.16.200.2*).



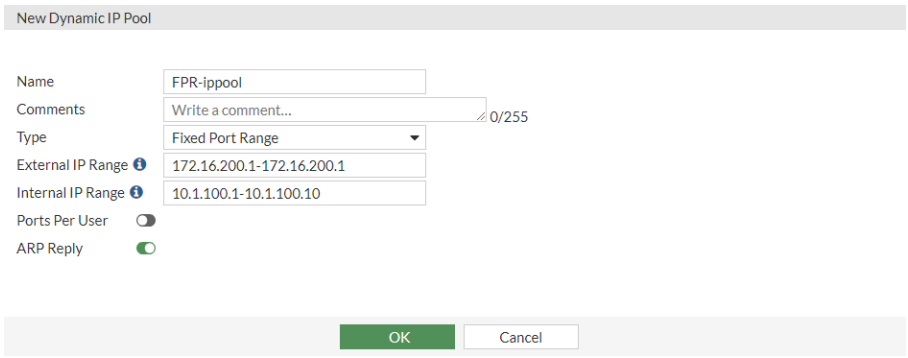
5. Click *OK*.

To configure one-to-one IP pool in the CLI:

```
config firewall ippool
    edit "One-to-One-ippool"
        set type one-to-one
        set startip 172.16.200.1
        set endip 172.16.200.2
    next
end
```

To configure fixed port range IP pool in the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Fixed Port Range*.
4. Enter the external IP range separated by a hyphen *172.16.200.1-172.16.200.1*).
5. Enter the internal IP range separated by a hyphen *10.1.100.1-10.1.100.10*).



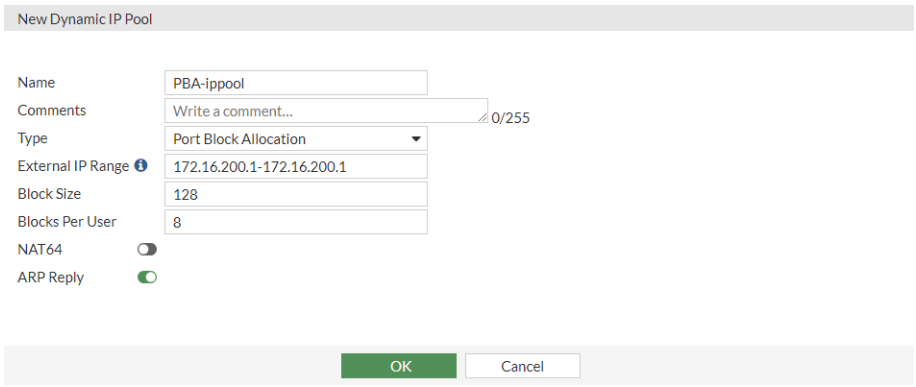
6. Click *OK*.

To configure fixed port range IP pool in the CLI:

```
config firewall ippool
    edit "FPR-ippool"
        set type fixed-port-range
        set startip 172.16.200.1
        set endip 172.16.200.1
        set source-startip 10.1.100.1
        set source-endip 10.1.100.10
   next
end
```

To configure port block allocation IP pool in the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Port Block Allocation*.
4. Enter the external IP range separated by a hyphen *172.16.200.1-172.16.200.1*).



5. Click *OK*.

To configure port block allocation IP pool in the CLI:

```
config firewall ippool
    edit PBA-ippool
        set type port-block-allocation
        set startip 172.16.200.1
        set endip 172.16.200.1
        set block-size 128
        set num-blocks-per-user 8
    next
end
```

IP pools and VIPs as local IP addresses

IP pools and VIPs are considered local IP addresses if responding to ARP requests on these external IP addresses is enabled (`set arp-reply enable`, by default). In this case, the FortiGate is considered a destination for those IP addresses and can receive reply traffic at the application layer successfully.

However, as a side-effect, once an IP pool or VIP has been configured, even if it is never used in a firewall policy, the FortiGate considers it as a local address and will not forward traffic based on the routing table. Therefore, any unused IP pools or VIPs should be deleted to prevent any unexpected behavior.

For a history of behavior changes related to IP pools and VIPs, see Technical Tip: IP pool and virtual IP behavior changes in FortiOS 6.4, 7.0, 7.2, and 7.4.

Central SNAT

The central SNAT table enables you to define and control (with more granularity) the address translation performed by FortiGate. With the NAT table, you can define the rules for the source address or address group, and which IP pool the destination address uses.

FortiGate reads the NAT rules from the top down until it hits a matching rule for the incoming address. This enables you to create multiple NAT policies that dictate which IP pool is used based on source address, destination address, and source port. NAT policies can be rearranged within the policy list. NAT policies are applied to network traffic after a security policy.

The central SNAT table allows you to create, edit, delete, and clone central SNAT entries.

Central SNAT notes

- The central NAT feature is not enabled by default.
- If central NAT is enabled, the NAT option under IPv4 policies is skipped and SNAT must be done via `central-snat-map`. The firewall policy list and dialog boxes have messages and redirection links to show this information.
- If NGFW mode is policy-based, then it is assumed that central NAT (specifically SNAT) is enabled implicitly.

Sample configuration

To enable central SNAT from the GUI:

1. In *System* > *Settings*, under *System Operations Settings*, enable *Central SNAT*.
2. Click *Apply*.

To enable or disable central SNAT using the CLI:

```
config system settings
   set central-nat {enable | disable}
end
```

When central NAT is enabled, *Policy & Objects* displays the Central SNAT section.

The Central SNAT policy has many options:

Field	Description
Type	Specify whether you are performing SNAT on IPv4 or IPv6. This option only appears when IPv6 is enabled under *Feature Visibility*.
Incoming Interface	Specify one or more interfaces for the ingress traffic.
Outgoing Interface	Specify one or more interfaces for the egress traffic.
Source Address	Specify the address or address group of the source.
Destination Address	Specify the address or address group of the destination.
NAT	Enable or disable to perform NAT. When disabled, no source address translation will occur.
IP Pool Configuration	Use outgoing interface address: • Use the address of the outgoing interfaces as source address. Use Dynamic IP Pool: • Choose an IP Pool to perform source NAT.
Protocol	Choose from any, TCP, UDP, SCTP, or specify the protocol number to match. For example, for ICMP, click *specify* with the protocol number 1.
Explicit port mapping	Enable in order to match this NAT policy only when the following ports are a match: • Choose an original source port from one to 65535. NAT'd port will be chosen by the FortiGate based on the IP Pool configuration.

Field	Description
	Explicit port mapping cannot apply to some protocols which do not use ports, such as ICMP. When enabling a NAT policy which uses Explicit port mapping, always consider that ICMP traffic will not match this policy. When using IP Pools, only the Overload type IP Pool allows Explicit port mapping. When Explicit port mapping is applied, you must define an original source port range and a translated sort port range. The source port will map one to one with the translated port. Refer to Dynamic SNAT to understand how each IP Pool type works.
Comments	Enter comments for this NAT policy.
Enable this policy	Enable or disable this policy.

To configure central SNAT using the CLI:

```
config firewall central-snat-map
    edit <policyID number>
        set status {enable | disable}
        set orig-addr <valid address object preconfigured on the FortiGate>
        set srcintf <name of interface on the FortiGate>
        set dst-addr <valid address object preconfigured on the FortiGate>
        set dstintf <name of interface on the FortiGate>
        set protocol <integer for protocol number>
        set dst-port <integer for destination port or port range>
        set orig-port <integer for original port number>
        set nat-port <integer for translated port number>
        set comments <string>
    next
end
```

Setting the destination port for traffic matching is available when the protocols are TCP, UDP, or SCTP.

The following examples demonstrate configuring central SNAT:

- Example one: Apply SNAT to all traffic on page 1381
- Example two: Apply an IP pool to all TCP traffic on page 1382
- Example three: Apply an IP pool to all traffic with a specific original port range on page 1383
- Example four: Create two central SNAT rules on page 1385
- Example five: Fine-tuning source port behavior NEW on page 1386

Example one: Apply SNAT to all traffic

Apply SNAT to all traffic from port2 to port3.

To configure from the CLI:

```
config firewall central-snat-map
    edit 1
```

```
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

Example two: Apply an IP pool to all TCP traffic

Apply an IP pool to all traffic from port3 to port2 that are TCP. NAT all other traffic using the outgoing interface IP.

To configure from the CLI:

```
config firewall ippool
    edit "Overload-IPPOOL"
        set startip 192.168.2.201
        set endip 192.168.2.202
    next
end
config firewall central-snat-map
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
        set protocol 6
        set nat-ippool "Overload-IPPOOL"
    next
    edit 2
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

To collect session table output from the CLI:

```
diagnose sys session list
```

The TCP session (protocol 6) is NAT'd with Overload-IPPOOL to 192.168.2.201:

```
session info: proto=6 proto_state=05 duration=14 expire=0 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=860/7/1 reply=555/8/1 tuples=2
tx speed(Bps/kbps): 60/0 rx speed(Bps/kbps): 38/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:49531->23.57.57.114:443(192.168.2.201:61776)
hook=pre dir=reply act=dnat 23.57.57.114:443->192.168.2.201:61776(192.168.0.10:49531)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5f:a2:2a
```

```
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011065 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

A UDP session (protocol 17) is NAT'd to the outgoing interface IP address 192.168.2.86:

```
session info: proto=17 proto_state=01 duration=16 expire=163 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=dns-udp vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=59/1/1 reply=187/1/1 tuples=2
tx speed(Bps/kbps): 3/0 rx speed(Bps/kbps): 11/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:52177->4.2.2.1:53(192.168.2.86:61770)
hook=pre dir=reply act=dnat 4.2.2.1:53->192.168.2.86:61770(192.168.0.10:52177)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011061 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Example three: Apply an IP pool to all traffic with a specific original port range

Apply an IP Pool to all traffic from port3 to port2 that have a specific original port range, mapping the ports to the same NAT'd port range. Nat all other traffic using the outgoing interface IP.

To configure from the CLI:

```
config firewall central-snat-map
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
        set orig-port 50000-65535
        set nat-ippool "Overload-IPPOOL"
        set nat-port 50000-65535
    next
    edit 2
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

To collect session table output from the CLI:

```
diagnose sys session list
```

Traffic with original port in the range between 50000-65535 will be NAT'd with the Overload type IP Pool. The mapped port is in the same port range:

```
session info: proto=17 proto_state=01 duration=3 expire=176 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=dns-udp vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=71/1/1 reply=123/1/1 tuples=2
tx speed(Bps/kbps): 23/0 rx speed(Bps/kbps): 40/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:52540->4.2.2.1:53(192.168.2.201:52540)
hook=pre dir=reply act=dnat 4.2.2.1:53->192.168.2.201:52540(192.168.0.10:52540)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011399 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Traffic with original port outside the range of 50000-65535 will be NAT'd to the outgoing interface IP:

```
session info: proto=6 proto_state=01 duration=3 expire=3597 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=2262/10/1 reply=2526/11/1 tuples=2
tx speed(Bps/kbps): 741/5 rx speed(Bps/kbps): 828/6
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:49805->142.250.68.66:443(192.168.2.86:62214)
hook=pre dir=reply act=dnat 142.250.68.66:443->192.168.2.86:62214(192.168.0.10:49805)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=0001139a tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Protocols which do not use ports, such as ICMP, will be NAT'd to the outgoing interface IP:

```
session info: proto=1 proto_state=00 duration=7 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=480/8/1 reply=480/8/1 tuples=2
tx speed(Bps/kbps): 66/0 rx speed(Bps/kbps): 66/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:1->4.2.2.1:8(192.168.2.86:62209)
hook=pre dir=reply act=dnat 4.2.2.1:62209->192.168.2.86:0(192.168.0.10:1)
```

```
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=0001138b tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Example four: Create two central SNAT rules

In the following example, two central SNAT rules will be created:

- Rule 3 will have a destination port set and IP pool `test-ippool4-3` applied.
- Rule 5 will have IP pool `test-ippool4-1` applied but will not set the destination port.

Example traffic will then be passed to see how the rule is matched.

To test central SNAT rule destination port support:

1. Configure central SNAT rule 3 with the destination port range specified:

```
config firewall ippool
    edit "test-ippool4-3"
        set startip 172.16.200.150
        set endip 172.16.200.150
    next
end
config firewall central-snat-map
    edit 3
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set protocol 6
        set nat-ippool "test-ippool4-3"
        set dst-port 80-443
    next
end
```

2. Configure central SNAT rule 5:

```
config firewall ippool
    edit "test-ippool4-1"
        set startip 172.16.200.151
        set endip 172.16.200.151
    next
end
config firewall central-snat-map
    edit 5
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set nat-ippool "test-ippool4-1"
    next
end
```

3. Send HTTP traffic to pass through the FortiGate that is expected to match central SNAT rule 3. IP pool `test-ippool4-3` will perform source NAT.
4. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=2 expire=3599 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=1800/31/1 reply=77304/60/1 tuples=2
tx speed(Bps/kbps): 602/4 rx speed(Bps/kbps): 25854/206
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24
gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:46731->172.16.200.55:80(172.16.200.150:46731)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.150:46731(10.1.100.42:46731)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003c37 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

5. Send PING traffic to pass through the FortiGate that is expected to match central SNAT rule 5. IP pool `test-ippool4-1` will perform source NAT.
6. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=2 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 99/0 rx speed(Bps/kbps): 99/0
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24
gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:36732->172.16.200.55:8(172.16.200.151:36732)
hook=pre dir=reply act=dnat 172.16.200.55:36732->172.16.200.151:0(10.1.100.42:36732)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003f62 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Example five: Fine-tuning source port behavior - NEW

FortiOS supports maintaining or altering the original source port in SNAT using the `port-preserve` command:

- When `port-preserve` is enabled, SNAT will use the original source port if it is not already in use. This is the default.
- When `port-preserve` is disabled, SNAT will always change the source port to use the next higher, available port in the range. When the highest available port is reached, the counter will roll back to the first available port in the range. This allows ports to remain free until the counter rolls back to them.

The `port-preserve` command is available for the central SNAT or for firewall policies when NAT is enabled.

To configure source port behavior for central SNAT:

```
config firewall central-snat-map
    edit 1
        set port-preserve {enable | disable}
    next
end
```

To preserve the original source port in a firewall policy:

1. Enable original source port preservation in the policy:

```
config firewall policy
    edit 2
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set port-preserve enable
    next
end
```

2. Check the session after the first traffic passes through the FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=7 expire=3594 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty src-vis
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 21/0 rx speed(Bps/kbps): 14/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5162)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5162(10.1.100.42:20042)
po/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0001cf04 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
```

```
no_ofld_reason:  mac-host-check disabled-by-policy
total session: 1
```

SNAT uses source port 5162.

3. Clear the old session.
4. Send traffic again with the same source port from the client.
5. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=4 expire=3598 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty src-vis
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 41/0 rx speed(Bps/kbps): 28/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5162)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5162(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0001d0bf tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  mac-host-check disabled-by-policy
total session: 1
```

The same source port has been used.

To alter the original source port in a firewall policy:

1. Disable original source port preservation in the policy:

```
config firewall policy
    edit 2
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set port-preserve disable
    next
end
```

2. Check the session after the first traffic passes through the FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=34 expire=113 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
```

```
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=269/5/1 reply=164/3/1 tuples=2
tx speed(Bps/kbps): 4/0 rx speed(Bps/kbps): 2/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5149)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5149(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a004 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

SNAT uses source port 5149.

3. Clear the old session.
4. Send traffic again with the same source port from the client.
5. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=3 expire=3597 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 49/0 rx speed(Bps/kbps): 33/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5151)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5151(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a1a5 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

A new source port has been used.

6. Clear the old session again.
7. Send traffic again with the same source port from the client.
8. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=20 expire=3581 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
```

```
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 8/0 rx speed(Bps/kbps): 5/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5153)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5153(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a519 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

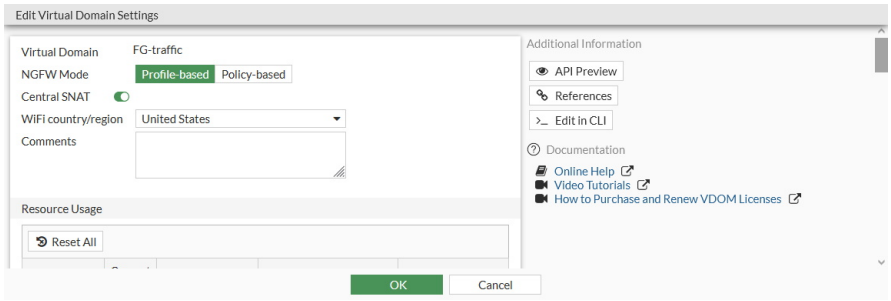
Another new source port has been used.

Configuring an IPv6 SNAT policy

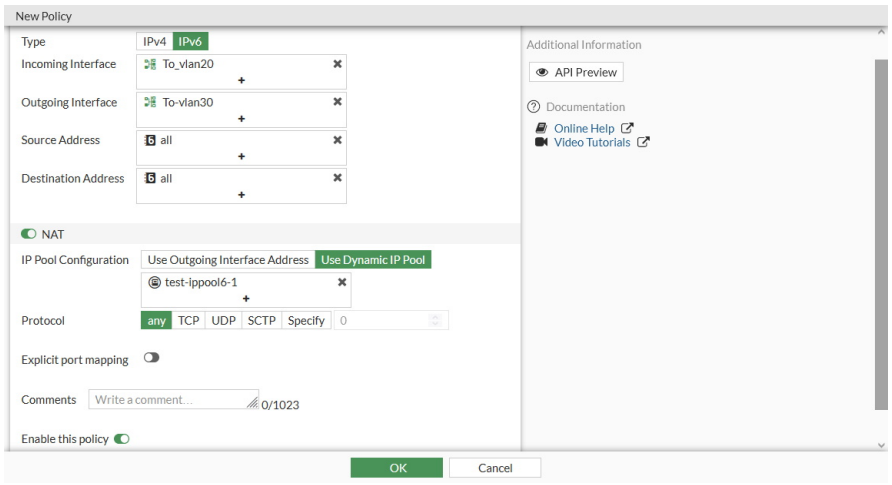
IPv4 and IPv6 central SNAT maps are displayed in the same table.

To configure an IPv6 policy with central SNAT in the GUI:

1. Enable central SNAT:
   a. In the Global VDOM, go to *System* > *VDOM*.
   b. Select a VDOM and click *Edit*. The *Edit Virtual Domain Settings* pane opens.
   c. Enable *Central SNAT*.



   d. Click *OK*.
2. In the VDOM with central SNAT enabled (FG-traffic in this example), go to *Policy & Objects* > *Central SNAT* and click *Create New*.
3. Configure the policy settings:
   a. For *Type*, select *IPv6*.
   b. Enter the interface, address, and IP pool information.
   c. Configure the remaining settings as needed.



d. Click *OK*.

The matching SNAT traffic will be handled by the IPv6 central SNAT map.

To configure an IPv6 policy with central SNAT in the CLI:

1. Enable central SNAT:

```
config vdom
    edit FG-traffic
        config system settings
            set central-nat enable
        end
    next
end
```

2. Create an IPv6 central SNAT policy:

```
config vdom
    edit FG-traffic
        config firewall central-snat-map
            edit 2
                set type ipv6
                set srcintf "wan2"
                set dstintf "wan1"
                set orig-addr6 "all"
                set dst-addr6 "all"
                set nat-ippool6 "test-ippool6-1"
            next
        end
    next
end
```

3. Verify the SNAT traffic:

```
(FG-traffic) # diagnose sniffer packet any icmp6 4
interfaces=[any]
filters=[icmp6]
3.602891 wan2 in 2000:10:1:100::41 -> 2000:172:16:200::55: icmp6: echo request seq 0
3.602942 wan1 out 2000:172:16:200::199 -> 2000:172:16:200::55: icmp6: echo request seq 0
3.603236 wan1 in 2000:172:16:200::55 -> 2000:172:16:200::199: icmp6: echo reply seq 0
3.603249 wan2 out 2000:172:16:200::55 -> 2000:10:1:100::41: icmp6: echo reply seq 0
```

```
4.602559 wan2 in 2000:10:1:100::41 -> 2000:172:16:200::55: icmp6: echo request seq 1
4.602575 wan1 out 2000:172:16:200::199 -> 2000:172:16:200::55: icmp6: echo request seq 1
4.602956 wan1 in 2000:172:16:200::55 -> 2000:172:16:200::199: icmp6: echo reply seq 1
4.602964 wan2 out 2000:172:16:200::55 -> 2000:10:1:100::41: icmp6: echo reply seq 1
^C
8 packets received by filter
0 packets dropped by kernel
```

SNAT policies with virtual wire pairs

Source NAT (SNAT) can be configured in IPv4 and IPv6 policies with virtual wire pair (VWP) interfaces, and between VWP interfaces when central NAT is enabled.

To configure a policy using SNAT and a VWP interface when central NAT is disabled:

1. Create the VWP interface:

```
config system virtual-wire-pair
    edit "test-vw-1"
        set member "port1" "port4"
    next
end
```

2. Create the IP pool. The IP pool must have a different subnet than the VWP peers.

```
config firewall ippool
    edit "vwp-pool-1"
        set startip 172.16.222.99
        set endip 172.16.222.100
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 88
        set srcintf "port4"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set ippool enable
        set poolname "vwp-pool-1"
    next
end
```

4. Verify the IP pool functions as expected and traffic passes through:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
23.438095 port4 in 172.16.200.11 -> 172.16.200.156: icmp: echo request
23.438126 port1 out 172.16.222.100 -> 172.16.200.156: icmp: echo request
```

```
23.438492 port1 in 172.16.200.156 -> 172.16.222.100: icmp: echo reply
23.438501 port4 out 172.16.200.156 -> 172.16.200.11: icmp: echo reply
24.439305 port4 in 172.16.200.11 -> 172.16.200.156: icmp: echo request
24.439319 port1 out 172.16.222.100 -> 172.16.200.156: icmp: echo request
24.439684 port1 in 172.16.200.156 -> 172.16.222.100: icmp: echo reply
24.439692 port4 out 172.16.200.156 -> 172.16.200.11: icmp: echo reply

8 packets received by filter
0 packets dropped by kernel
```

To configure a SNAT between VWP interfaces when central NAT is enabled:

1. Enable central NAT:

```
config system settings
    set central-nat enable
end
```

2. Create the VWP interface:

```
config system virtual-wire-pair
    edit "test-vw-1"
        set member "port1" "port4"
    next
end
```

3. Create the IP pool. The IP pool must have a different subnet than the VWP peers.

```
config firewall ippool
    edit "vwp-pool-1"
        set startip 172.16.222.99
        set endip 172.16.222.100
    next
end
```

4. Configure the SNAT policy:

```
config firewall central-snat-map
    edit 2
        set srcintf "port4"
        set dstintf "port1"
        set orig-addr "all"
        set dst-addr "all"
        set nat-ippool "vwp-pool-1"
    next
end
```

5. Configure the firewall policy:

```
config firewall policy
    edit 90
        set srcintf "port4"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
```

```
        next
    end
```

Destination NAT

Network Address Translation (NAT) is the process that enables a single device, such as a router or firewall, to act as an agent between the internet or public network and a local or private network. This agent acts in real-time to translate the source or destination IP address of a client or server on the network interface. NAT can be subdivided into two types:

- Source NAT (SNAT)
- Destination NAT (DNAT)

This section is about DNAT. For information about SNAT, see Source NAT on page 1371.

A virtual IP (VIP) maps external IP addresses to internal IP addresses for DNAT. See Configuring VIPs on page 1396 and Configuring VIP groups on page 1399.

The following types of VIPs can be created:

Static VIP	A virtual IP that maps an IP address or range to another IP address or range. Custom settings can allow the VIP to be filtered by Source Address and/or services, so that the VIP only applies to the filtered traffic. See Static virtual IPs on page 1395.
Static VIP with services	A virtual IP that defines services for a single port number mapping. See Virtual IP with services on page 1400.
Static VIP with port forwarding	A virtual IP that hides the port number for an internal server or maps several internal servers to the same public IP address. See Virtual IPs with port forwarding on page 1401.
FQDN-based VIP	A virtual IP mapped to an FQDN. See Configure FQDN-based VIPs on page 1414.
Virtual server load balancing	A special type of virtual IP used to implement server load balancing. See Virtual server load balance on page 1403. Virtual IPs can also be used for server load balance multiplexing. See Virtual server load balance multiplexing on page 1412.
Central DNAT	Where DNAT is configured by creating virtual IPs and selecting the VIPs in firewall policies, central NAT is not configured in the firewall policy. Central NAT is enabled in *System Settings*. When enabled, the *Policy & Objects* tree displays the *Central SNAT* policy option. Use the *Central SNAT* policy to configure VIPs as separate objects. During use, FortiGate reads the enabled NAT rules from the top down, until it locates a matching rule. See Central DNAT on page 1415.

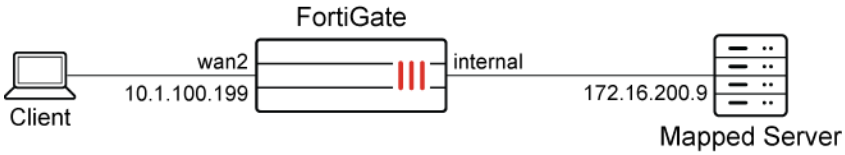
See also Configuring PCP port mapping with SNAT and DNAT on page 1453.

Static virtual IPs

Static Virtual IPs (VIP) are used to map external IP addresses to internal IP addresses. This is also called destination NAT, where a packet's destination is being NAT'd, or mapped, to a different address.

Static VIPs are commonly used to map public IP addresses to resources behind the FortiGate that use private IP addresses. A static one-to-one VIP is when the entire port range is mapped. A port forwarding VIP is when the mapping is configured on a specific port or port range.

Sample configuration



To create a virtual IP in the GUI:

1. In *Policy & Objects* > *Virtual IPs*.
2. Select the *Virtual IP* or *IPv6 Virtual IP* tab based on the IP versions used.
3. Click *Create new*.
4. Enter a unique name for the virtual IP.
5. Enter values for the external IP address/range and map to IPv4/IPv6 address/range fields.
6. Click *OK*.

To create a virtual IP in the CLI:

```
config firewall vip
    edit "Internal_WebServer"
        set extip 10.1.100.199
        set extintf "any"
        set mappedip "172.16.200.55"
    next
end
```

To apply a virtual IP to policy in the CLI:

```
config firewall policy
    edit 8
        set name "Example_Virtual_IP_in_Policy"
        set srcintf "wan2"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "Internal_WebServer"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

IP pools and VIPs as local IP addresses

IP pools and VIPs are considered local IP addresses if responding to ARP requests on these external IP addresses is enabled (`set arp-reply enable`, by default). In this case, the FortiGate is considered a destination for those IP addresses and can receive reply traffic at the application layer successfully.

However, as a side-effect, once an IP pool or VIP has been configured, even if it is never used in a firewall policy, the FortiGate considers it as a local address and will not forward traffic based on the routing table. Therefore, any unused IP pools or VIPs should be deleted to prevent any unexpected behavior.

For a history of behaviour changes related to IP pools and VIPs, see Technical Tip: IP pool and virtual IP behaviour changes in FortiOS 6.4, 7.0, 7.2, and 7.4.

Configuring VIPs

Virtual IPs can be configured for IPv4 and IPv6. After creating the VIP, add it to a firewall policy.

FortiOS does not check whether VIPs overlap. As a result, you can configure multiple VIPs with the same external interface and IP. However, you can view overlapping VIPs in the security rating report. See Viewing VIP overlap in security rating reports on page 1398.

To configure a VIP in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs*.
2. Select the *Virtual IP* tab, and click *Create New*.

3. Configure the following settings:

Name		Enter a name for the VIP.
Comments		Enter a description of the VIP.
Color		Click *Change* to select a color for the VIP.
Network		
	Interface (`extintf`)	The external interface that the firewall policy source interface must match. For example, if the external interface is port1, then the VIP can be used in a policy from port1 to port3, but not in a policy from port2 to port3. If the external interface is *any*, then the VIP can be used in any firewall policy.
	Type (`type`)	• Static NAT - Use an external IP address or address range. • FQDN - Use an external IP or FQDN address. • load-balance (CLI only) - Load balance traffic. • server-load-balance - Load balance traffic across multiple servers. SSL processing can be offloaded to the FortiGate. This type of VIP is configure from *Policy & Objects* > *Virtual Servers*. • dns-translation (CLI only) - DNS translation. • access-proxy - Used for ZTNA. See ZTNA HTTPS access proxy example on page 1229 for details.
	External IP address/range (`extip`)	In a static NAT VIP, the external IP address is the IP address that the FortiGate listens for traffic on. When the external interface is not *any*, 0.0.0.0 can be used to make the external IP address equivalent to the external interface's IP address. The external IP address is also used to perform SNAT for the mapped server when the server outbound traffic with a destination interface that matches the external interface. The firewall policy must also have NAT enabled.
Map to		
	IPv4 address/range (`mappedip`)	The IPv4 address or range that the internal resource is being mapped to.
	IPv6 address/range (`ipv6-mappedip`)	The IPv6 address or range that the internal resource is being mapped to.
Optional Filters		Enable to access additional options.
	Source address (`src-filter`)	Restrict the source IP address, address range, or subnet that is allowed to access the VIP.
	Services (`service`)	Set the services that are allowed to be mapped.

Port Forwarding (`portforward`)		Enable port forwarding and display additional options. Enable port forwarding to specify the port (`mappedport`) to map to.
	Protocol (`protocol`)	Select the protocol to use when forwarding packets to the port.
	Port Mapping Type	• One to one - Each external service port is mapped to one port. A range is allowed, but the number of ports should be the same. • Many to Many - The port mapping can be one to one, one to many, or many to one. There are no restrictions on how many external ports must map to internal ports.
	External service port (`extport`)	Enter the external service port range to be mapped to a port range on the destination network.
	Map to IPv4 port (`mappedport`)	Enter the mapped IPv4 port range on the destination network.
	Map to IPv6 port (`ipv6-mappedport`)	Enter the mapped IPv6 port range on the destination network.

4. Click *OK* to save the VIP.

Viewing VIP overlap in security rating reports

There is no overlap check for VIPs, so there are no constraints when configuring multiple VIPs with the same external interface and IP. A new security rating report alerts users of any VIP overlaps.

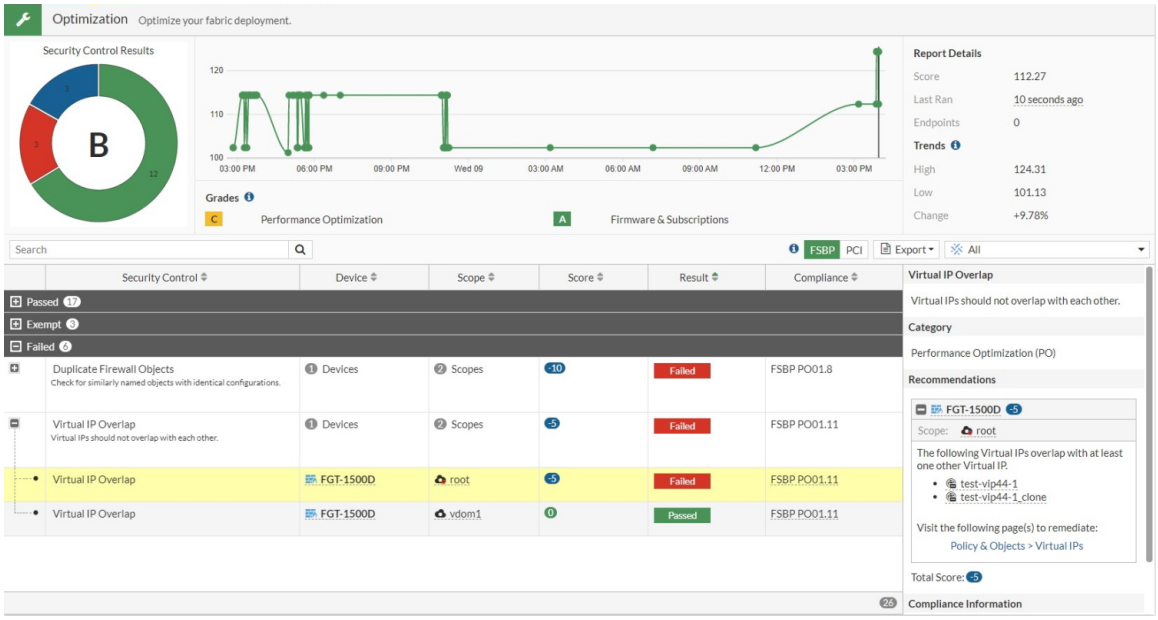
To configure two VIPs with the same external interface and IP:

```
config firewall vip
    edit "test-vip44-1"
        set extip 10.1.100.154
        set mappedip "172.16.200.156"
        set extintf "port24"
    next
    edit "test-vip44-1_clone"
        set extip 10.1.100.154
        set mappedip "172.16.200.156"
        set extintf "port24"
        set src-filter 10.1.100.11
    next
end
```

No error message appears regarding the overlapping VIPs.

To view the security rating report:

1. Go to *Security Fabric* > *Security Rating* and click the *Optimization* scorecard.
2. Expand the *Failed* section. The *Virtual IP Overlap* results show an overlap (*test-vip44-1* and *test-vip44-1_clone*) on the root FortiGate.



Configuring VIP groups

Virtual IP addresses (VIPs) can be organized into groups. After creating the VIP group, add it to a firewall policy.

VIP groups are useful when multiple VIPs are used together in firewall policies. If the VIP group members change, or a group member's settings change (such as the IP address, port, or port mapping type), then those changes are automatically updated in the corresponding firewall policies.

The following table summarizes which VIP types are allowed and not allowed to be members of a VIP group:

Group type	VIP types allowed as members	VIP types not allowed as members
IPv4	• Static NAT • Load balance • DNS translation • FQDN	• Access proxy • Server load balance
IPv6	• Static NAT	• Access proxy • Server load balance

Different VIP types can be added to the same group.

To configure a VIP group in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs*.
2. Navigate to the *Virtual IP Group* or *IPv6 Virtual IP Group* tab.

3. Click *Create new*.
4. Enter a name.
5. Optionally, enter additional information in the *Comments* field.
6. For IPv4 groups, select the *Interface*. Select a specific interface if all of the VIPs are on the same interface; otherwise, select *any*.
7. Click the + in the *Members* field and select the members to add to the group.
8. Click *OK*.

To configure an IPv4 VIP group in the CLI:

```
config firewall vipgrp
    edit <name>
        set interface <name>
        set member <vip1> <vip2> ...
    next
end
```

To configure an IPv6 VIP group in the CLI:

```
config firewall vipgrp6
    edit <name>
        set member <vip1> <vip2> ...
    next
end
```

Virtual IP with services

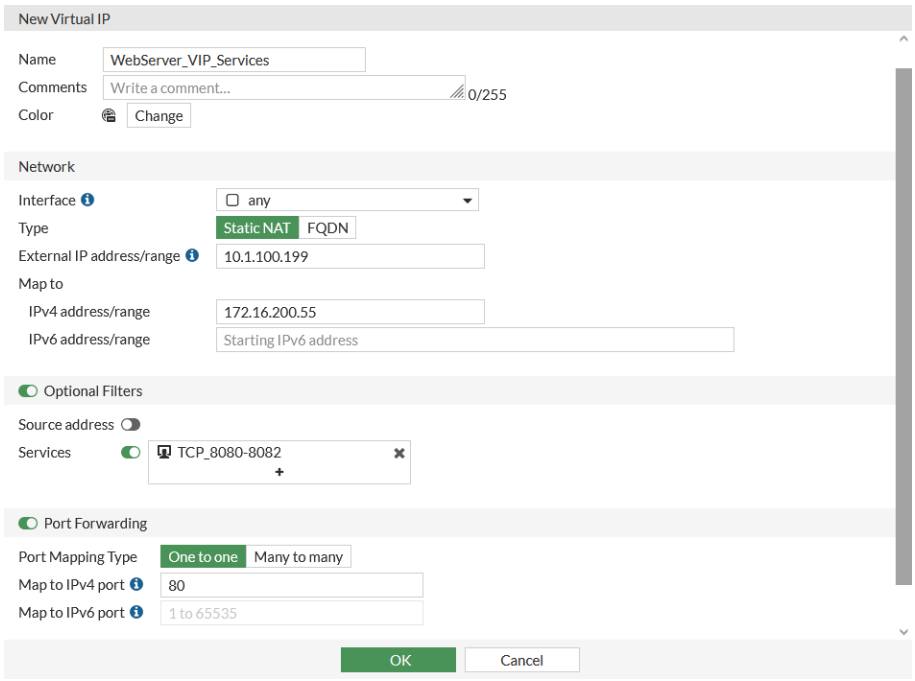
Virtual IP with services is a more flexible virtual IP mode. This mode allows users to define services to a single port number mapping.

This topic shows how to use virtual IP with services enabled. This example has one public external IP address. We map TCP ports 8080, 8081, and 8082 to an internal WebServer TCP port 80. This allows remote connections to communicate with a server behind the firewall.

Sample configuration

To create a virtual IP with services in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a unique name for the virtual IP.
4. Configure the fields in the *Network* section. For example:
   - Set *Interface* to *any*.
   - Set *External IP Address/Range* to *10.1.100.199*.
   - Set *Mapped IP Address/Range* to *172.16.200.55*.
5. Enable *Optional Filters* and then enable *Services*.
6. In the *Services* field, add TCP ports 8080, 8081, 8082. See Internet service customization on page 1591 for information about creating a custom port range service.
7. Enable *Port Forwarding* and set *Map to IPv4 port* to *80*.



8. Click *OK*.

To see the results:

1. Apply the above virtual IP to the firewall policy.
2. The results are:
   - Access 10.1.100.199:8080 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
   - Access 10.1.100.199:8081 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
   - Access 10.1.100.199:8082 from external network and FortiGate maps to 172.16.200.55:80 in internal network.

To create a virtual IP with services in the CLI:

```
config firewall vip
    edit "WebServer_VIP_Services"
        set service "TCP_8080-8082"
        set extip 10.1.100.199
        set extintf "any"
        set portforward enable
        set mappedip "172.16.200.55"
        set mappedport 80
    next
end
```

Virtual IPs with port forwarding

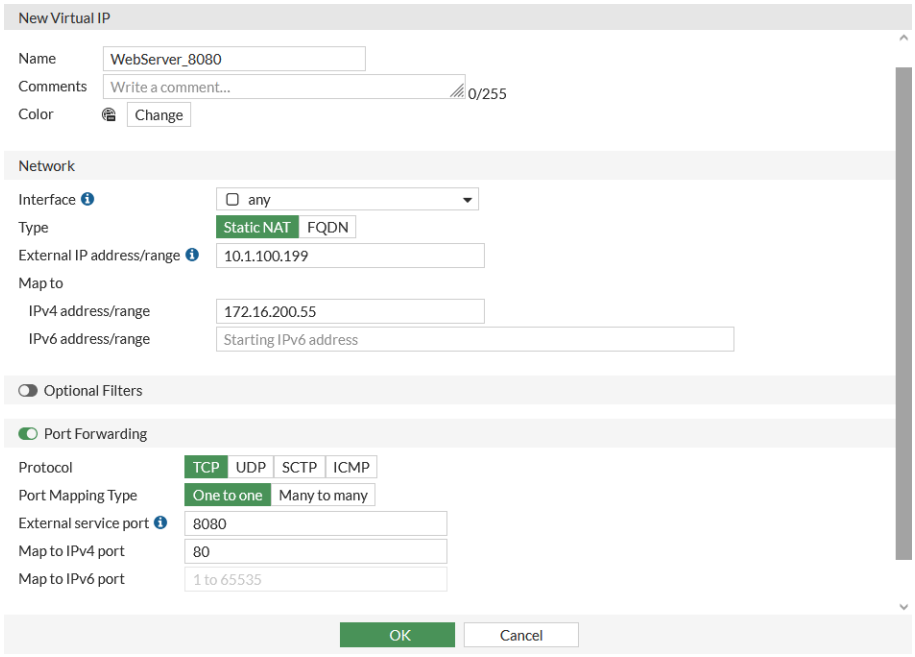
If you need to hide the internal server port number or need to map several internal servers to the same public IP address, enable port-forwarding for Virtual IP.

This topic shows how to use virtual IPs to configure port forwarding on a FortiGate unit. This example has one public external IP address. We map TCP ports 8080, 8081, and 8082 to different internal WebServers' TCP port 80. This allows remote connections to communicate with a server behind the firewall.

Sample configuration

To create a virtual IP with port forwarding in the GUI:

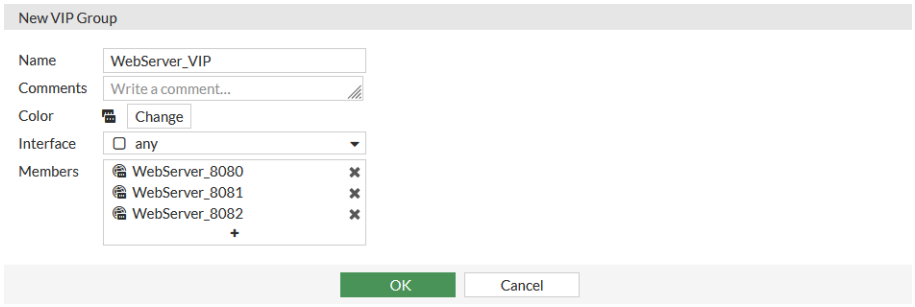
1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a unique name for the virtual IP.
4. Configure the fields in the *Network* section. For example:
   - Set *Interface* to *any*.
   - Set *External IP Address/Range* to *10.1.100.199*.
   - Set *Mapped IP Address/Range* to *172.16.200.55*.
5. Leave *Optional Filters* disabled.
6. Enable *Port Forwarding* and configure the fields. For example:
   - Set *Protocol* to *TCP*.
   - Set *External Service Port* to *8080*.
   - Set *Map to IPv4 port* to *80*.



7. Click *OK*.
8. Follow the above steps to create two additional virtual IPs.
   a. For one virtual IP:
      - Use a different *Mapped IP Address/Range*, for example *172.16.200.56*.
      - Set *External Service Port* to *8081*.
      - Use the same *Map to IPv4 port* number: *80*.
   b. For the other virtual IP:

- Use a different *Mapped IP Address/Range*, for example *172.16.200.57*.
- Set *External Service Port* to *8082*.
- Use the same *Map to IPv4 port* number: *80*.

9. Create a *Virtual IP Group* and put the above three virtual IPs into that group:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP Group* tab.
   b. Click *Create new*.
   c. Enter a name for the group.
   d. Add the three previously created virtual IPs as members.



   e. Click *OK*.

To see the results:

1. Apply the above virtual IP to the Firewall policy.
2. The results are:
   - Access 10.1.100.199:8080 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
   - Access 10.1.100.199:8081 from external network and FortiGate maps to 172.16.200.56:80 in internal network.
   - Access 10.1.100.199:8082 from external network and FortiGate maps to 172.16.200.57:80 in internal network

Virtual server load balance

This topic shows a special virtual IP type: virtual server. Use this type of VIP to implement server load balancing.

The FortiOS server load balancing contains all the features of a server load balancing solution. You can balance traffic across multiple backend servers based on multiple load balancing schedules including:

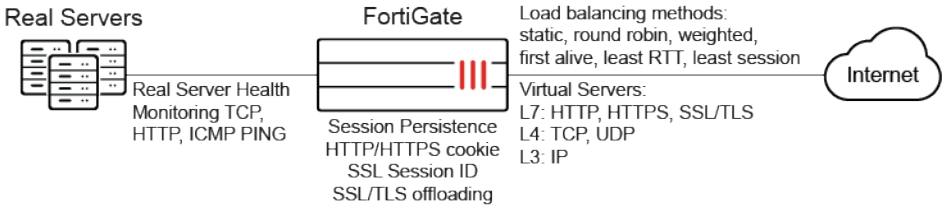
- Static (failover)
- Round robin
- Weighted (to account for different sized servers or based on the health and performance of the server including round trip time and number of connections)

The load balancer supports HTTP, HTTPS, IMAPS, POP3S, SMTPS, SSL/TLS, and generic TCP/UDP and IP protocols. Session persistence is supported based on the SSL session ID based on an injected HTTP cookie, or based on the HTTP or HTTPS host. SSL/TLS load balancing includes protection from protocol downgrade attacks. Server load balancing is supported on most FortiGate devices and includes up to 10,000 virtual servers on high-end systems.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models NEW on page 103 for more information.

FortiOS HTTP and HTTPS server load balancing does not support load balancing based on URL routing. You can use FortiWeb server pools or FortiADC server load balancing to load balance sessions to two or more URL based routes.

Sample topology



SSL/TLS offloading

FortiGate SSL/TLS offloading is designed for the proliferation of SSL/TLS applications. The key exchange and encryption/decryption tasks are offloaded to the FortiGate unit where they are accelerated using FortiASIC technology which provides significantly more performance than a standard server or load balancer. This frees up valuable resources on the server farm to give better response to business operations. Server load balancing offloads most SSL/TLS versions including SSL 3.0, TLS 1.0, and TLS 1.2, and supports full mode or half mode SSL offloading with DH key sizes up to 4096 bits.

FortiGate SSL offloading allows the application payload to be inspected before it reaches your servers. This prevents intrusion attempts, blocks viruses, stops unwanted applications, and prevents data loss. SSL/TLS content inspection supports TLS versions 1.0, 1.1, and 1.2 and SSL versions 1.0, 1.1, 1.2, and 3.0.

Virtual server requirements

When creating a new virtual server, you must configure the following options:

- Virtual Server Type.
- Load Balancing Methods.
- Health check monitoring (optional).
- Session persistence (optional).
- Virtual Server IP (External IP Address).
- Virtual Server Port (External Port).
- Real Servers (Mapped IP Address & Port).

Virtual server types

Select the protocol to be load balanced by the virtual server. If you select a general protocol such as IP, TCP, or UDP, the virtual server load balances all IP, TCP, or UDP sessions. If you select specific protocols such as HTTP, HTTPS, or SSL, you can apply additional server load balancing features such as *Persistence* and *HTTP Multiplexing*.

HTTP	Select *HTTP* to load balance only HTTP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 80 for HTTP sessions). You can enable *HTTP Multiplexing*. You can also set *Persistence* to *HTTP Cookie* to enable cookie-based persistence.
HTTPS	Select *HTTPS* to load balance only HTTPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 443 for HTTPS sessions). You can enable *HTTP Multiplexing*. You can also set *Persistence* to *HTTP Cookie* to enable cookie-based persistence, or you can set *Persistence* to *SSL Session ID*.
IMAPS	Select *IMAPS* to load balance only IMAPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 993 for IMAPS sessions). You can also set *Persistence* to *SSL Session ID*.
POP3S	Select *POP3S* to load balance only POP3S sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 995 for POP3S sessions). You can also set *Persistence* to *SSL Session ID*.
SMTPS	Select *SMTPS* to load balance only SMTPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 465 for SMTPS sessions). You can also set *Persistence* to *SSL Session ID*.
SSL	Select *SSL* to load balance only SSL sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced. You can also set *Persistence* to *SSL Session ID*.
TCP	Select *TCP* to load balance only TCP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced.
UDP	Select *UDP* to load balance only UDP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced.
IP	Select *IP* to load balance all sessions accepted by the security policy that contains this virtual server.

Load balancing methods

The load balancing method defines how sessions are load balanced to real servers.

All load balancing methods do not send traffic to real servers that are down or not responding. FortiGate can only determine if a real server is not responding by using a health check monitor. You should always add at least one health check monitor to a virtual server or to real servers; otherwise load balancing might try to distribute sessions to real servers that are not functioning.

Static	The traffic load is statically spread evenly across all real servers. Sessions are not assigned according to how busy individual real servers are. This load balancing method provides some persistence because all sessions from the same source address always go to the same real server. Because the distribution is stateless, so if a real server is added, removed, or goes up or down, the distribution is changed and persistence might be lost.
Round Robin	Directs new requests to the next real server. This method treats all real servers as equals regardless of response time or the number of connections. This method does not direct requests to real servers that down or non responsive.
Weighted	Real servers with a higher weight value receive a larger percentage of connections. Set the real server weight when adding a real server.
Least Session	Directs requests to the real server that has the least number of current connections. This method works best in environments where the real servers or other equipment you are load balancing all have similar capabilities. This load balancing method uses the FortiGate session table to track the number of sessions being processed by each real server. The FortiGate unit cannot detect the number of sessions actually being processed by a real server.
Least RTT	Directs sessions to the real server with the lowest round trip time. The round trip time is determined by a ping health check monitor. The default is 0 if no ping health check monitors are added to the virtual server.
First Alive	Directs sessions to the first live real server. This load balancing schedule provides real server failover protection by sending all sessions to the first live real server. If a real server fails, all sessions are sent to the next live real server. Sessions are not distributed to all real servers so all sessions are processed by the first real server only.
HTTP Host	Load balances HTTP host connections across multiple real servers using the host's HTTP header to guide the connection to the correct real server.

Health check monitoring

In the FortiGate GUI, you can configure health check monitoring so that the FortiGate unit can verify that real servers are able respond to network connection attempts. If a real server responds to connection attempts, the load balancer continues to send sessions to it. If a real server stops responding to connection attempts, the load balancer assumes that the server is down and does not send sessions to it. The health check monitor configuration determines how the load balancer tests real servers. You can use a single health check monitor for multiple load balancing configurations. You can configure TCP, HTTP, DNS, and ping health check monitors. You usually set the health check monitor to use the same protocol as the traffic being load balanced to it. For example, for an HTTP load balancing configuration, you would normally use an HTTP health check monitor.

Session persistence

Use persistence to ensure a user is connected to the same real server every time the user makes an HTTP, HTTPS, or SSL request that is part of the same user session. For example, if you are load balancing HTTP and HTTPS sessions to a collection of eCommerce web servers, when users make a purchase, they will be starting multiple sessions as they navigate the eCommerce site. In most cases, all the sessions started by this user during one eCommerce session should be processed by the same real server. Typically, the HTTP protocol keeps track of these related sessions using cookies. HTTP cookie persistence ensure all sessions that are part of the same user session are processed by the same real server.

When you configure persistence, the FortiGate unit load balances a new session to a real server according to the load balance method. If the session has an HTTP cookie or an SSL session ID, the FortiGate unit sends all subsequent sessions with the same HTTP cookie or SSL session ID to the same real server.

Real servers

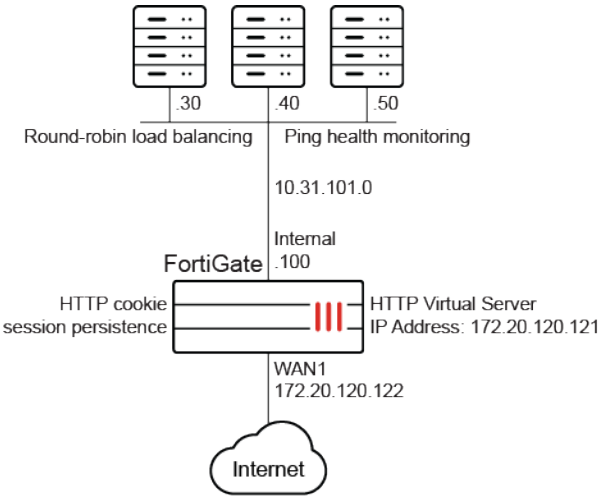
Add real servers to a load balancing virtual server to provide information the virtual server requires to send sessions to the server. A real server configuration includes the IP address of the real server and port number the real server receives sessions on. The FortiGate unit sends sessions to the real server's IP address using the destination port number in the real server configuration.

When configuring a real server, you can also specify the weight (if the load balance method is set to *Weighted*) and you can limit the maximum number of open connections between the FortiGate unit and the real server. If the maximum number of connections is reached for the real server, the FortiGate unit automatically switches all further connection requests to other real servers until the connection number drops below the limit. Setting *Maximum Connections* to 0 means that the FortiGate unit does not limit the number of connections to the real server.

Sample of HTTP load balancing to three real web servers

This example describes the steps to configure the load balancing configuration below. In this configuration, a FortiGate unit is load balancing HTTP traffic from the Internet to three HTTP servers on the internal network. HTTP sessions are accepted at the wan1 interface with destination IP address 172.20.120.121 on TCP port 8080, and forwarded from the internal interface to the web servers. When forwarded, the destination address of the session is translated to the IP address of one of the web servers.

This load balancing configuration also includes session persistence using HTTP cookies, round-robin load balancing, and TCP health monitoring for the real servers. Ping health monitoring consists of the FortiGate unit using ICMP ping to ensure the web servers can respond to network traffic.



General steps:

1. Create a health check monitor.

   A ping health check monitor causes the FortiGate to ping the real servers every 10 seconds. If one of the servers does not respond within 2 seconds, the FortiGate unit will retry the ping 3 times before assuming that the HTTP server is not responding.

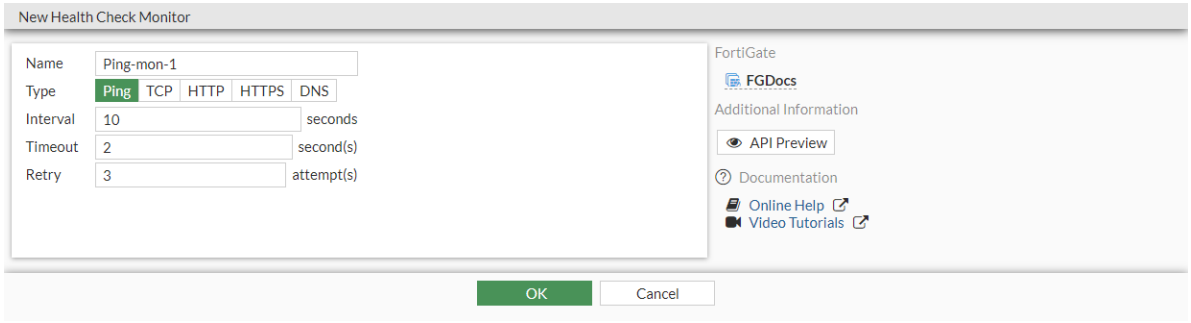
2. Create a load balance virtual server with three real servers.
3. Add the load balancing virtual server to a policy as the destination address.

To see the virtual servers and health check monitors options in the GUI, *Load Balance* must be selected in *Feature Visibility* > *Additional Features*. See Feature visibility on page 3062 on page 1 for details.

Configure a load balancing virtual server in the GUI

To create a health check monitor:

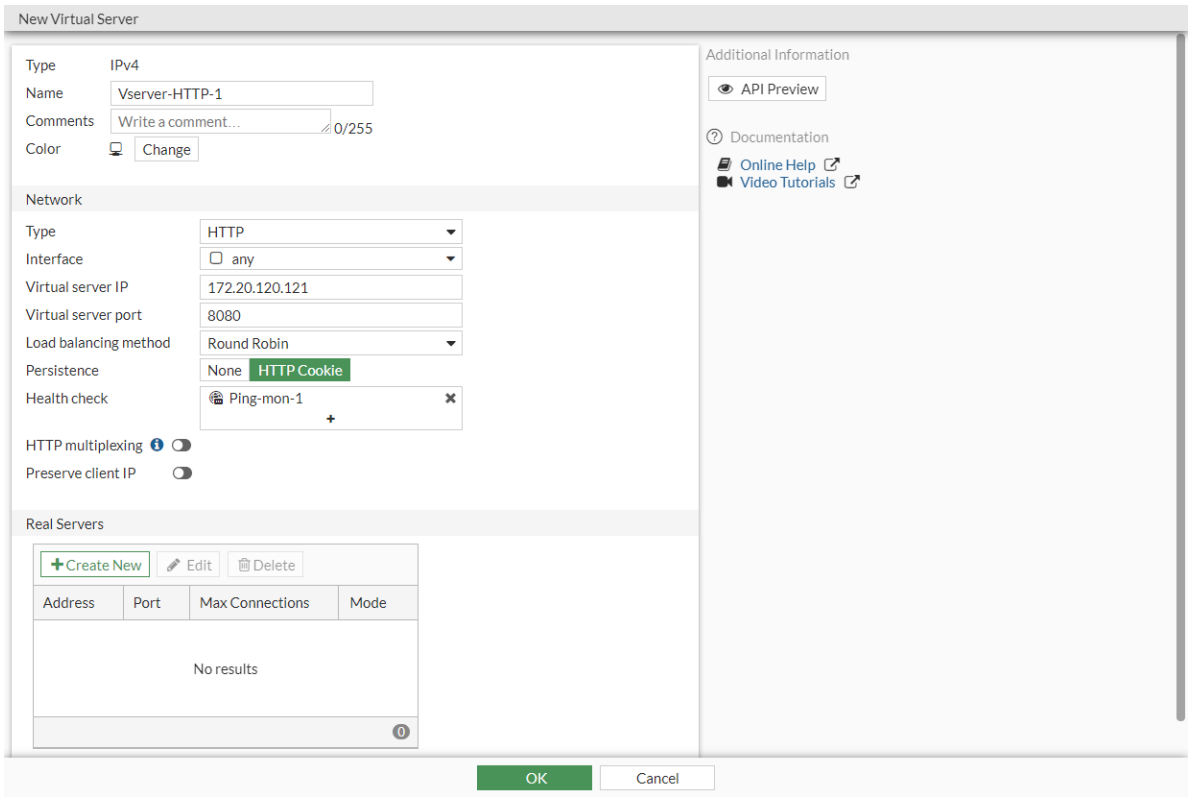
1. Go to *Policy & Objects* > *Health Check*.
2. Click *Create New*.
3. Set the following:
   - *Name* to *Ping-mon-1*
   - *Type* to *Ping*
   - *Interval* to *10* seconds
   - *Timeout* to *2* seconds
   - *Retry* to *3* attempt(s)



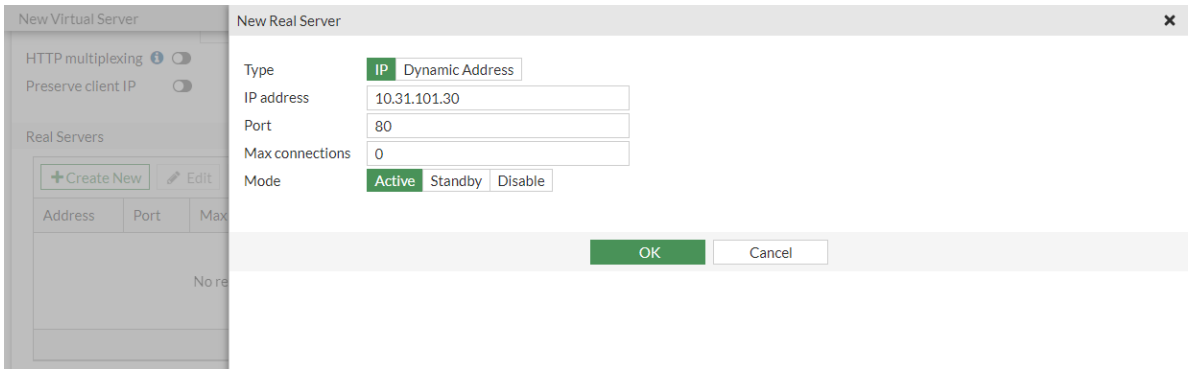
4. Click *OK*.

To create a virtual server:

1. Go to *Policy & Objects* > *Virtual Servers*.
2. Click *Create New*.
3. Set the following:
   - *Name* to *Vserver-HTTP-1*
   - *Type* to *HTTP*
   - *Interface* to *wan1*
   - *Virtual Server IP* to *172.20.120.121*
   - *Virtual Server Port* to *8080*
   - *Load Balance Method* to *Round Robin*
   - *Persistence* to *HTTP Cookie*
   - *Health Check* to *Ping-mon-1*



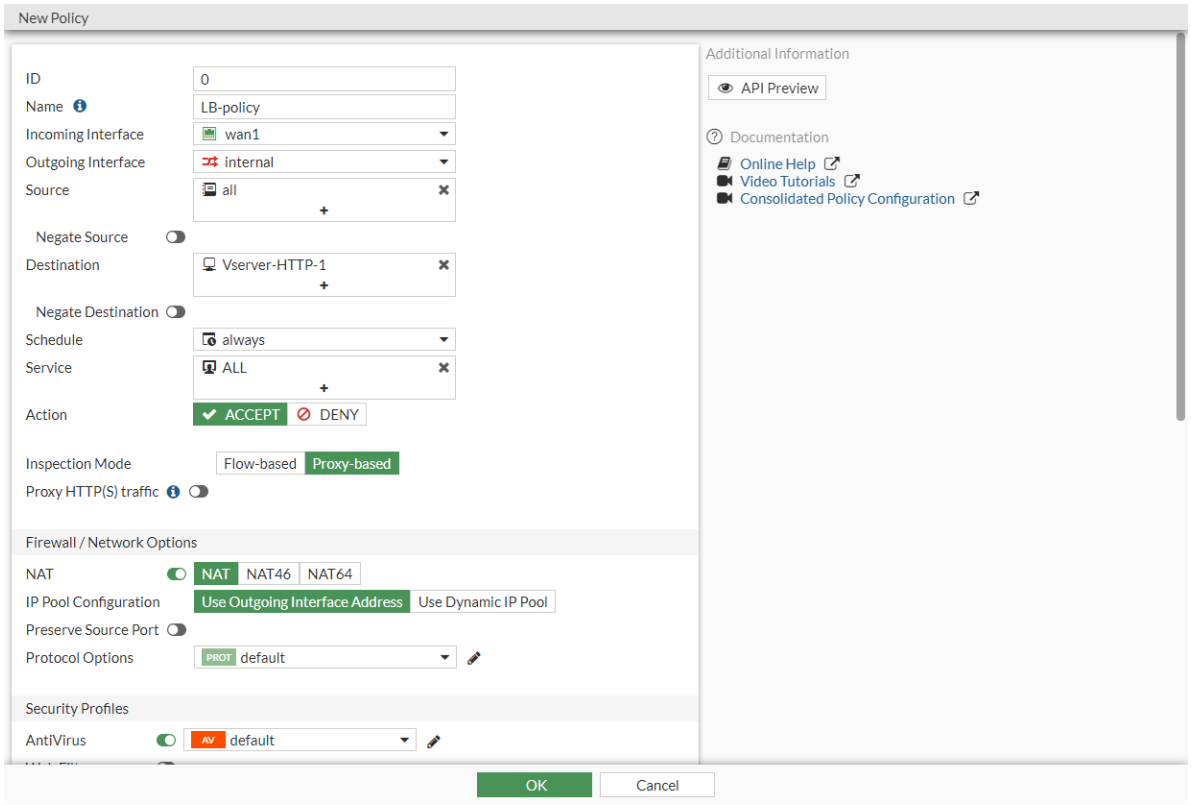
4. In the *Real Servers* table, click *Create New*.
5. Set the following for the first real server:
   - *Type* to *IP*
   - *IP Address* to *10.31.101.30*
   - *Port* to *80*
   - *Max Connections* to *0*
   - *Mode* to *Active*



6. Click *OK*. Configure two more real servers with IP addresses 10.31.101.40 and 10.31.101.50, and the same settings as the first real server.
7. Click *OK*.

To create a security policy that includes the load balance virtual server as the destination address:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Set the *Inspection Mode* to *Proxy-based*. The new virtual server will not be available if the inspection mode is *Flow-based*.
4. Set the following:
   - *Name* to *LB-policy*
   - *Incoming Interface* to *wan1*
   - *Outgoing Interface* to *internal*
   - *Source* to *all*
   - *Destination* to *Vserver-HTTP-1*
   - *Schedule* to *always*
   - *Service* to *ALL*
   - *Action* to *ACCEPT*
5. Enable *NAT* and set *IP Pool Configuration* to *Use Outgoing Interface Address*.
6. Enable *AntiVirus* and select an antivirus profile.



7. Click *OK*.

Configure a load balancing virtual server in the CLI

To configure HTTP load balancing to three real web servers in the CLI:

1. Create a health check monitor:

```
config firewall ldb-monitor
    edit "Ping-mon-1"
        set type ping
        set interval 10
        set timeout 2
        set retry 3
    next
end
```

2. Create a virtual server:

```
config firewall vip
    edit "Vserver-HTTP-1"
        set type server-load-balance
        set extip 172.20.120.121
        set extintf "any"
        set server-type http
        set monitor "Ping-mon-1"
        set ldb-method round-robin
        set persistence http-cookie
        set extport 8080
        config realservers
            edit 1
                set type ip
                set ip 10.31.101.30
                set port 80
            next
            edit 2
                set type ip
                set ip 10.31.101.40
                set port 80
            next
            edit 3
                set type ip
                set ip 10.31.101.50
                set port 80
            next
        end
    next
end
```

3. Add the load balancing virtual server to a policy as the destination address:

```
config firewall policy
    edit 2
        set name "LB-policy"
        set inspection-mode proxy
        set srcintf "wan1"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "Vserver-HTTP-1"
```

```
            set action accept
            set schedule "always"
            set service "ALL"
            set utm-status enable
            set ssl-ssh-profile "certificate-inspection"
            set av-profile "default"
            set fsso disable
            set nat enable
        next
    end
```

Results

Traffic accessing 172.20.120.121:8080 is forwarded in turn to the three real servers.

If the access request has an http-cookie, FortiGate forwards the access to the corresponding real server according to the cookie.

Virtual server load balance multiplexing

HTTP2 connection coalescing and concurrent multiplexing allows multiple HTTP2 requests to share the same TLS connection when the destination IP is the same.

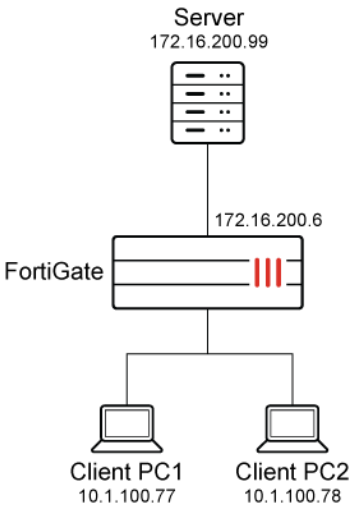
To configure the load balanced virtual server:

```
config firewall vip
    edit <name>
        set type server-load-balance
        set server-type {http | https}
        set http-multiplex {enable | disable}
        set http-multiplex-ttl <integer>
        set http-multiplex-max-request <integer>
        set http-supported-max-version {http1 | http2}
    next
end
```

`http-multiplex {enable | disable}`	Enable/disable HTTP multiplexing.
`http-multiplex-ttl <integer>`	Set the time-to-live for idle connections to servers (in seconds, 0 - 2147483647, default = 15).
`http-multiplex-max-request <integer>`	Set the maximum number of requests that the multiplex server can handle before disconnecting (0 - 2147483647, default = 0).
`http-supported-max-version {http1 | http2}`	Set the maximum supported HTTP version: • http1: support HTTP 1.1 and HTTP1. • http2: support HTTP2, HTTP 1.1, and HTTP1 (default).

Example

In this example, multiple clients submit requests in HTTP2. The requests hit the VIP address, and then FortiGate opens a session between itself (172.16.200.6) and the server (172.16.200.99). The coalescing occurs in this session as the multiple streams share the same TLS session to connect to the same destination server.



To configure connection coalescing and concurrent multiplexing with virtual server load balancing:

1. Configure the virtual server:

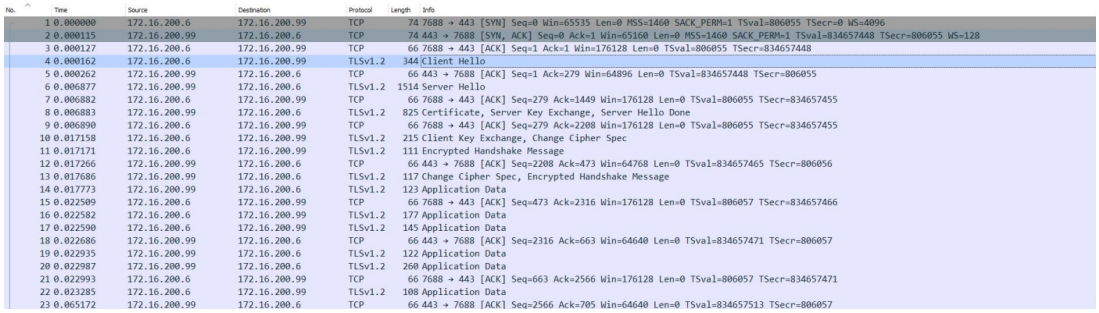
```
config firewall vip
    edit "vip-test"
        set type server-load-balance
        set extip 10.1.100.222
        set extintf "port2"
        set server-type https
        set extport 443
        config realservers
            edit 1
                set ip 172.16.200.99
                set port 443
            next
        end
        set http-multiplex enable
        set ssl-mode full
        set ssl-certificate "Fortinet_SSL"
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "vip-test"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
        set logtraffic all
```

```
        set nat enable
    next
end
```

3. Get the clients to access the VIP address (10.1.100.222). The FortiGate shares the first TLS connection with second TLS connection.
4. Verify the sniffer packet capture on the FortiGate server side. There is one client hello.

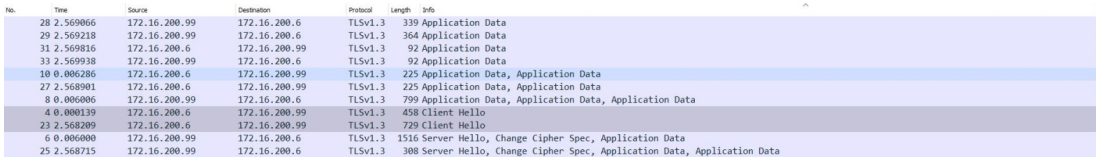


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.6	172.16.200.99	TCP	74	7688 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=806055 TSecr=0 WS=4096
2	0.000115	172.16.200.99	172.16.200.6	TCP	74	443 → 7688 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=834657448 TSecr=806055 WS=128
3	0.000127	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=806055 TSecr=834657448
4	0.000162	172.16.200.6	172.16.200.99	TLSv1.2	344	Client Hello
5	0.000262	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=834657448 TSecr=806055
6	0.006877	172.16.200.99	172.16.200.6	TLSv1.2	1514	Server Hello
7	0.006882	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=1449 Win=176128 Len=0 TSval=806055 TSecr=834657455
8	0.006883	172.16.200.99	172.16.200.6	TLSv1.2	825	Certificate, Server Key Exchange, Server Hello Done
9	0.006890	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=2208 Win=176128 Len=0 TSval=806055 TSecr=834657455
10	0.017158	172.16.200.6	172.16.200.99	TLSv1.2	215	Client Key Exchange, Change Cipher Spec
11	0.017171	172.16.200.6	172.16.200.99	TLSv1.2	111	Encrypted Handshake Message
12	0.017266	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2208 Ack=473 Win=64768 Len=0 TSval=834657465 TSecr=806056
13	0.017686	172.16.200.99	172.16.200.6	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
14	0.017773	172.16.200.99	172.16.200.6	TLSv1.2	123	Application Data
15	0.022509	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=473 Ack=2316 Win=176128 Len=0 TSval=806057 TSecr=834657466
16	0.022582	172.16.200.6	172.16.200.99	TLSv1.2	177	Application Data
17	0.022590	172.16.200.6	172.16.200.99	TLSv1.2	145	Application Data
18	0.022686	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2316 Ack=663 Win=64640 Len=0 TSval=834657471 TSecr=806057
19	0.022935	172.16.200.99	172.16.200.6	TLSv1.2	122	Application Data
20	0.022987	172.16.200.99	172.16.200.6	TLSv1.2	260	Application Data
21	0.022993	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=663 Ack=2566 Win=176128 Len=0 TSval=806057 TSecr=834657471
22	0.023285	172.16.200.6	172.16.200.99	TLSv1.2	108	Application Data
23	0.065172	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2566 Ack=705 Win=64640 Len=0 TSval=834657513 TSecr=806057

5. Disable HTTP multiplexing:

```
config firewall vip
    edit "vip-test"
        config realservers
            edit 1
                set type ip
                set ip 172.16.200.99
                set port 443
            next
        end
        set http-multiplex disable
    next
end
```

6. Verify the sniffer packet capture. This time, the FortiGate does reuse the TLS connection, so there are two client hellos sent to the real server.



No.	Time	Source	Destination	Protocol	Length	Info
28	2.569066	172.16.200.99	172.16.200.6	TLSv1.3	339	Application Data
29	2.569218	172.16.200.99	172.16.200.6	TLSv1.3	364	Application Data
31	2.569816	172.16.200.6	172.16.200.99	TLSv1.3	92	Application Data
33	2.569938	172.16.200.99	172.16.200.6	TLSv1.3	92	Application Data
10	0.006286	172.16.200.6	172.16.200.99	TLSv1.3	225	Application Data, Application Data
27	2.568901	172.16.200.6	172.16.200.99	TLSv1.3	225	Application Data, Application Data
8	0.006006	172.16.200.99	172.16.200.6	TLSv1.3	799	Application Data, Application Data, Application Data
4	0.000139	172.16.200.6	172.16.200.99	TLSv1.3	458	Client Hello
23	2.568209	172.16.200.6	172.16.200.99	TLSv1.3	729	Client Hello
6	0.006000	172.16.200.99	172.16.200.6	TLSv1.3	1516	Server Hello, Change Cipher Spec, Application Data
25	2.568715	172.16.200.99	172.16.200.6	TLSv1.3	308	Server Hello, Change Cipher Spec, Application Data, Application Data

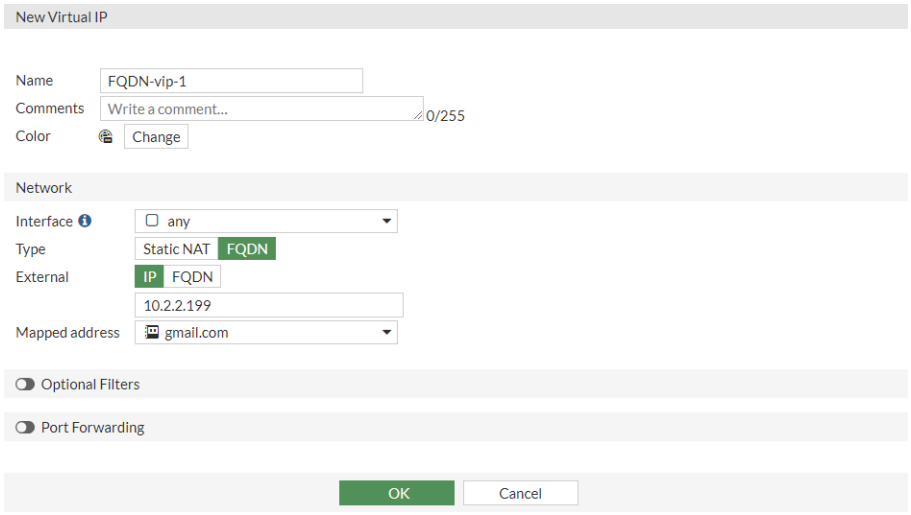
Configure FQDN-based VIPs

In public cloud environments, sometimes it is necessary to map a VIP to an FQDN address.

To configure an FQDN-based VIP in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a name for the VIP.
4. Select an interface.
5. For *Type*, select *FQDN*.
6. For *External*, select *IP* and enter the external IP address.

7. For *Mapped address*, select an FQDN address.



8. Click *OK*.

To configure an FQDN-based VIP in the CLI:

```
config firewall vip
    edit "FQDN-vip-1"
        set type fqdn
        set extip 10.2.2.199
        set extintf "any"
        set mapped-addr "destination"
    next
end
```

Central DNAT

Central NAT allows for the central configuration of SNAT (source NAT) and DNAT (destination NAT).

To enable central NAT in the GUI:

1. Go to *System* > *Settings*.
2. In the *System Operation Settings*, enable *Central SNAT*.
3. Click *Apply*.

To enable central NAT in the CLI:

```
config system settings
    set central-nat {enable | disable}
end
```

When central NAT is enabled, virtual IPs (VIPs) are not configured in the firewall policy. The VIPs are configured as separate objects where their status must be enabled.

This option is only available for IPv4 VIP and VIP46 objects.

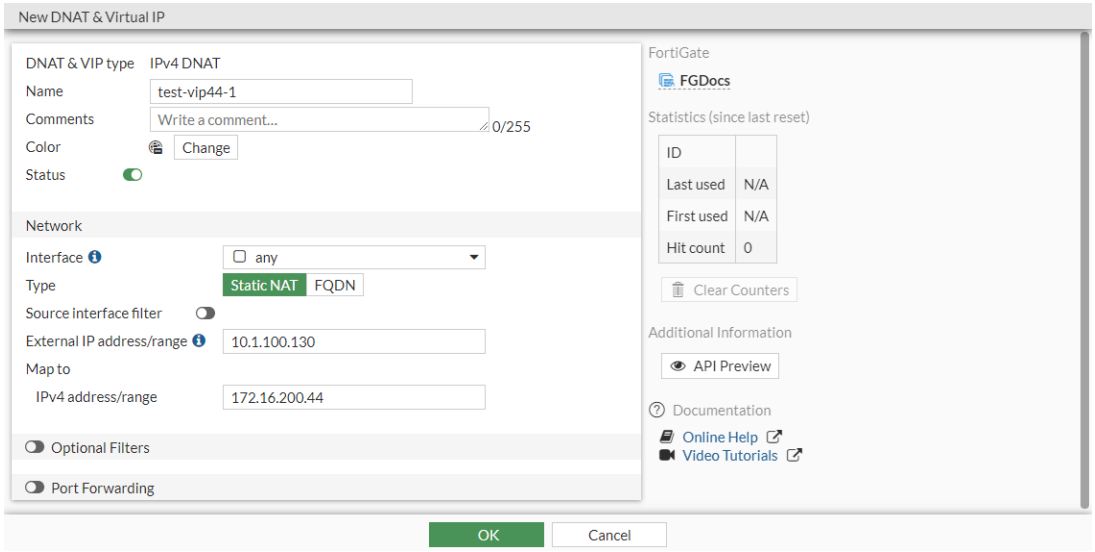
Configuring a DNAT and VIP object in central NAT mode is similar to configuring a VIP when central NAT is disabled. See Static virtual IPs on page 1395 for more information on each setting.

VIP objects can carry over when switching from non-central NAT mode to central NAT mode or vice-versa. However, if a VIP is assigned to a firewall policy in non-central NAT mode, it must be unassigned before switching to central NAT mode.

In this example, a DNAT and VIP are configured to forward traffic from 10.1.100.130 to 172.16.200.44. This example assumes that the firewall address, Addr_172.16.200.44/32, has already been configured.

To configure DNAT and a VIP in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *DNAT & Virtual IPs* and click *Create New* > *DNAT & Virtual IP*.
   b. Enter a name (*test-vip44-1*).
   c. Set the *External IP address/range* to *10.1.100.130*.
   d. Set the *Map to IPv4 address/range* to *172.16.200.44*.

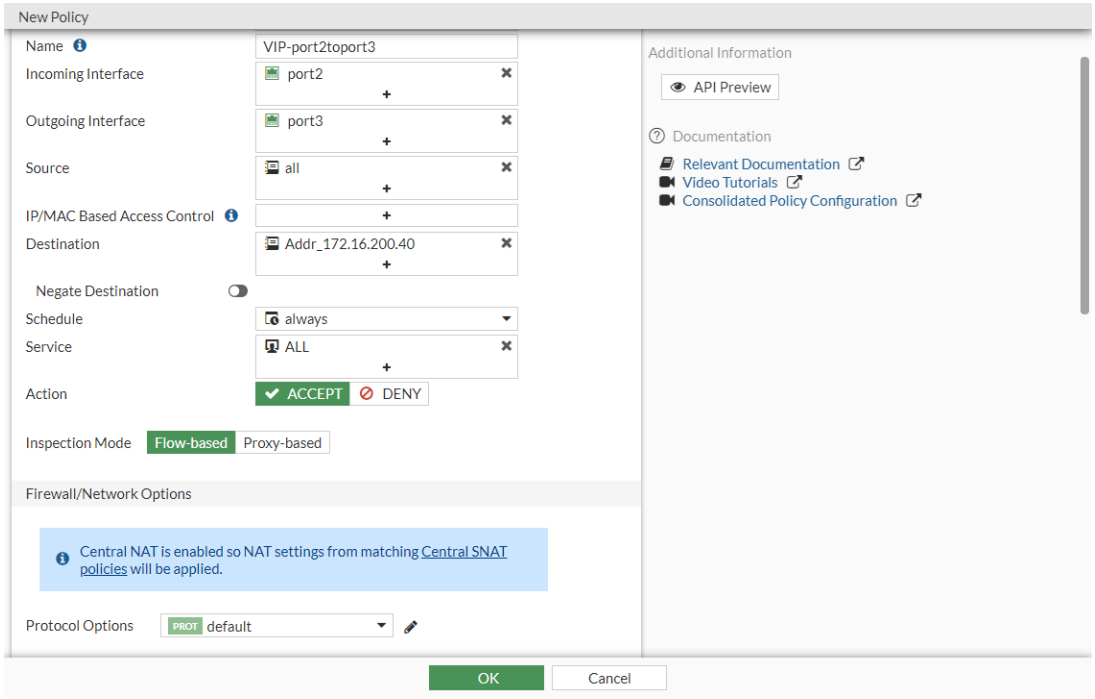


   e. Click *OK*.
2. Configure a firewall policy that allows traffic in the direction of the VIP:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Configure the following settings:

Name	*VIP-port2toport3*
Source	*all*
Destination	*Addr_172.16.200.40*

Schedule	*always*
Service	*ALL*
Action	*ACCEPT*

c. Configure the other settings as needed. There is no SNAT configuration section, so central SNAT policies will be applied.



d. Click *OK*.

To configure DNAT and a VIP in the CLI:

1. Configure the VIP:

```
config firewall vip
    edit "test-vip44-1"
        set extip 10.1.100.130
        set mappedip "172.16.200.44"
        set extintf "any"
        set status enable
    next
end
```

2. Configure a firewall policy that allows traffic in the direction of the VIP:

```
config firewall policy
    edit 3
        set name "VIP-port2toport3"
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "Addr_172.16.200.40"
```

```
            set schedule "always"
            set service "ALL"
        next
    end
```

To verify the DNAT and VIP:

If the VIP status is enabled, it will appear in the VIP table:

```
# diagnose firewall iprope list 100000
policy index=7 uuid_idx=625 action=accept
flag (8000104): f_p nat pol_stats
cos_fwd=0  cos_rev=0
group=00100000 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 10.1.100.130-10.1.100.130, uuid_idx=625,
service(1):
        [0:0x0:0/(0,0)->(0,0)] helper:auto
nat(1): flag=0 base=10.1.100.130:0 172.16.200.44-172.16.200.44(0:0)
```

If the VIP status is disabled, it will not appear in the VIP table.

In this example, a one-to-one static NAT is enabled. Send a ping to 10.1.100.130, and the traffic will be forwarded to the destination 172.16.200.44.

Examples and policy actions

The following topics provide examples and instructions on policy actions:

- NAT46 and NAT64 policy and routing configurations on page 1419
- Mirroring SSL traffic in policies on page 1429
- Recognize anycast addresses in geo-IP blocking on page 1431
- Matching GeoIP by registered and physical location on page 1432
- HTTP to HTTPS redirect for load balancing on page 1433
- Use Active Directory objects directly in policies on page 1435
- No session timeout on page 1439
- MAP-E support on page 1440
- Seven-day rolling counter for policy hit counters on page 1444
- Cisco Security Group Tag as policy matching criteria on page 1445
- Virtual patching on the local-in management interface on page 1448
- Configuring PCP port mapping with SNAT and DNAT on page 1453
- Refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction on page 1458
- Per-policy disclaimer messages on page 1460

NAT46 and NAT64 policy and routing configurations

Multiple NAT46 and NAT64 related objects are consolidated into regular objects. A per-VDOM virtual interface, naf.<vdom>, is automatically added to process NAT46/NAT64 traffic. The features include:

- `vip46` and `vip64` settings are consolidated in `vip` and `vip6` configurations.
- `policy46` and `policy64` settings are consolidated in `firewall policy` settings.
- `nat46`/`nat64` are included in `firewall policy` settings.
- `ippool` and `ippool6` support NAT46 and NAT64 (when enabled, the IP pool should match a subnet).
- Central SNAT supports NAT46 and NAT64.
- `add-nat46-route` in `ippool6` and `add-nat64-route` in `ippool` are enabled by default. The FortiGate generates a static route that matches the IP range in `ippool6` or `ippool` for the naf tunnel interface.

Automatic processing of the naf tunnel interface is not supported in security policies.

To configure NAT46/NAT64 translation, use the standard `vip`/`vip6` setting, apply it in a firewall policy, enable NAT46/NAT64, and enter the IP pool to complete the configuration.

The external IP address cannot be the same as the external interface IP address.

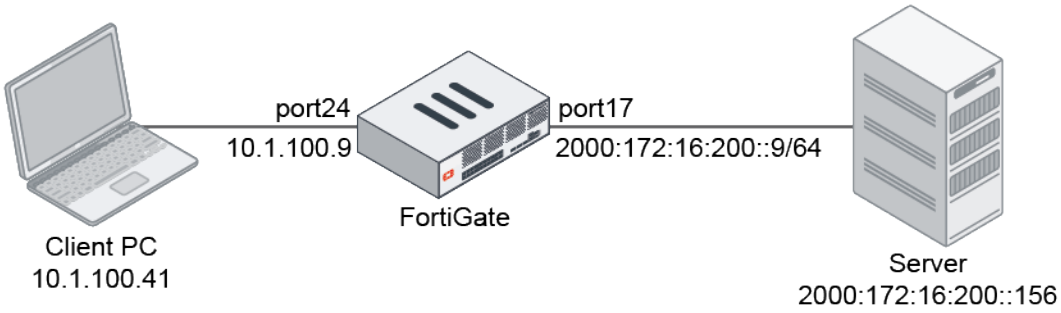
Examples

IPv6 must be enabled to configure these examples. In the GUI, so go to *System* > *Feature Visibility* and enable *IPv6*. In the CLI, enter the following:

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

NAT46 policy

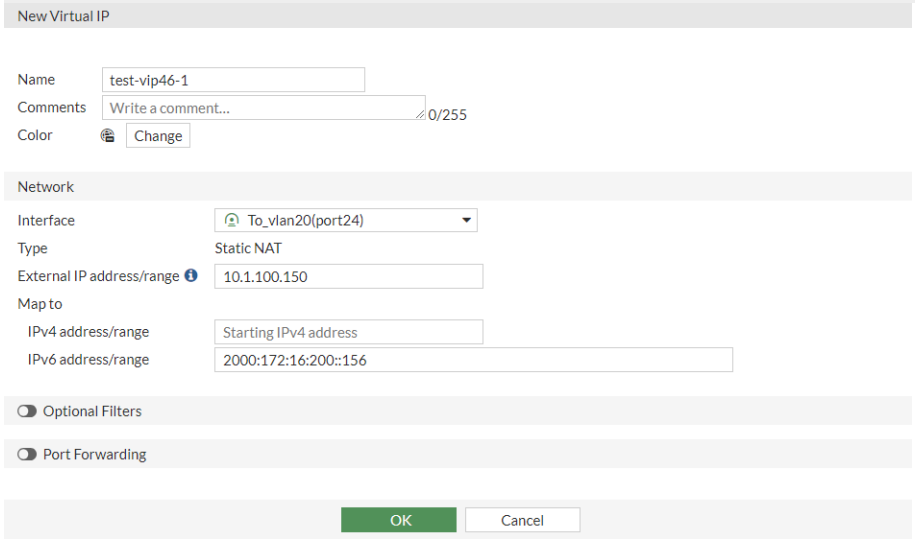
In this example, a client PC is using IPv4 and an IPv4 VIP to access a server that is using IPv6. The FortiGate uses NAT46 to translate the request from IPv4 to IPv6 using the virtual interface naf.root. An `ippool6` is applied so that the request is SNATed to the `ippool6` address (2000:172:16:101::1 - 2000:172:16:101::1).



To create a NAT46 policy in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-vip46-1
Interface	To_vlan20
Type	Static NAT
External IP address/range	10.1.100.150
Map to IPv6 address/range	2000:172:16:200::156



   d. Click *OK*.
2. Configure the IPv6 pool:
   a. Go to *Policy & Objects* > *IP Pools* and select the *IPv6 IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

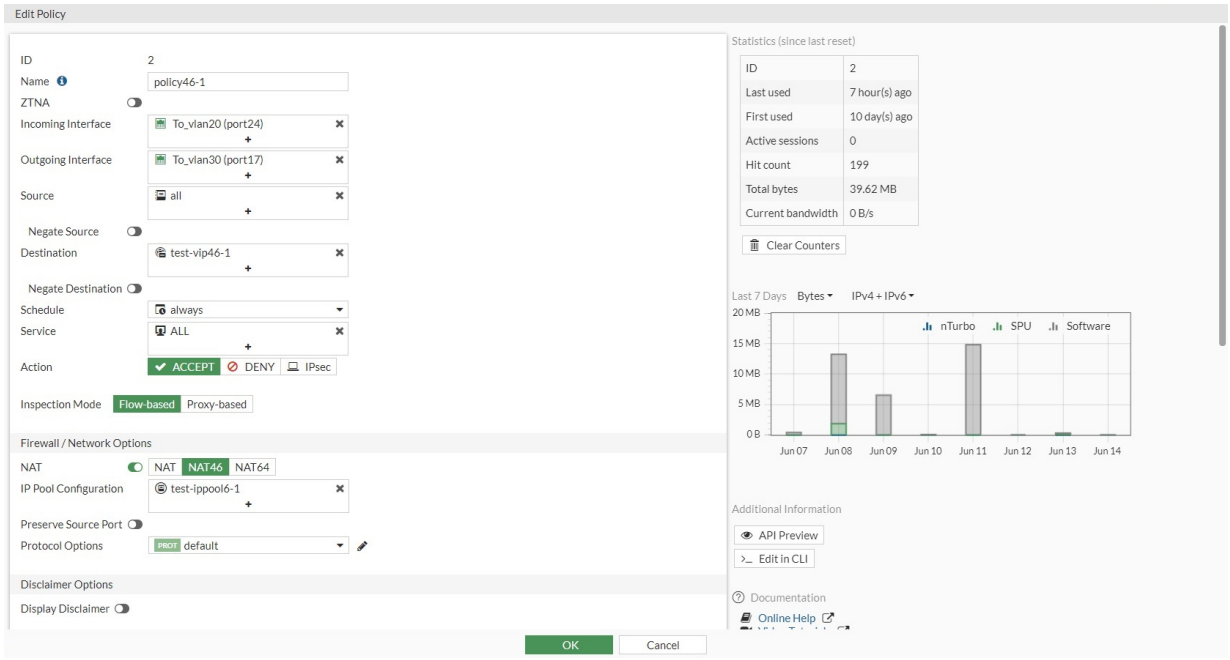
Name	test-ippool6-1
External IP address/range	2000:172:16:101::1-2000:172:16:101::1
NAT46	Enable

   d. Click *OK*.
3. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New* or edit an existing policy.
   b. Enter the following:

Name	policy46-1
Incoming Interface	To_vlan20

Outgoing Interface	To_vlan30
Source	all
Destination	test-vip46-1
Schedule	always
Service	ALL
Action	ACCEPT
NAT	NAT46
IP Pool Configuration	test-ippool6-1

c. Configure the other settings as needed.



d. Click *OK*.

To create a NAT46 policy in the CLI:

1. Configure the VIP:

```
config firewall vip
    edit "test-vip46-1"
        set extip 10.1.100.150
        set nat44 disable
        set nat46 enable
        set extintf "port24"
        set arp-reply enable
        set ipv6-mappedip 2000:172:16:200::156
    next
end
```

2. Configure the IPv6 pool:

```
config firewall ippool6
    edit "test-ippool6-1"
        set startip 2000:172:16:101::1
        set endip 2000:172:16:101::1
        set nat46 enable
        set add-nat46-route enable
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 2
        set name "policy46-1"
        set srcintf "port24"
        set dstintf "port17"
        set action accept
        set nat46 enable
        set srcaddr "all"
        set dstaddr "test-vip46-1"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname6 "test-ippool6-1"
    next
end
```

To verify the traffic and session tables:

1. Verify the traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
2.593302 port24 in 10.1.100.41 -> 10.1.100.150: icmp: echo request
2.593344 naf.root out 10.1.100.41 -> 10.1.100.150: icmp: echo request
2.593347 naf.root in 2000:172:16:101::1 -> 2000:172:16:200::156: icmp6: echo request seq
1
2.593383 port17 out 2000:172:16:101::1 -> 2000:172:16:200::156: icmp6: echo request seq
1
2.593772 port17 in 2000:172:16:200::156 -> 2000:172:16:101::1: icmp6: echo reply seq 1
2.593788 naf.root out 2000:172:16:200::156 -> 2000:172:16:101::1: icmp6: echo reply seq
1
2.593790 naf.root in 10.1.100.150 -> 10.1.100.41: icmp: echo reply
2.593804 port24 out 10.1.100.150 -> 10.1.100.41: icmp: echo reply
11 packets received by filter
0 packets dropped by kernel
```

2. Verify the session tables for IPv4 and IPv6:

```
(root) # diagnose sys session list
session info: proto=1 proto_state=00 duration=2 expire=59 timeout=0 flags=00000000
```

```
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 106/0 rx speed(Bps/kbps): 106/0
orgin->sink: org pre->post, reply pre->post dev=24->53/53->24
gwy=10.1.100.150/10.1.100.41
hook=pre dir=org act=noop 10.1.100.41:29388->10.1.100.150:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.150:29388->10.1.100.41:0(0.0.0.0:0)
peer=2000:172:16:101::1:29388->2000:172:16:200::156:128 naf=1
hook=pre dir=org act=noop 2000:172:16:101::1:29388->2000:172:16:200::156:128(:::0)
hook=post dir=reply act=noop 2000:172:16:200::156:29388->2000:172:16:101::1:129(:::0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00012b77 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session 1
```

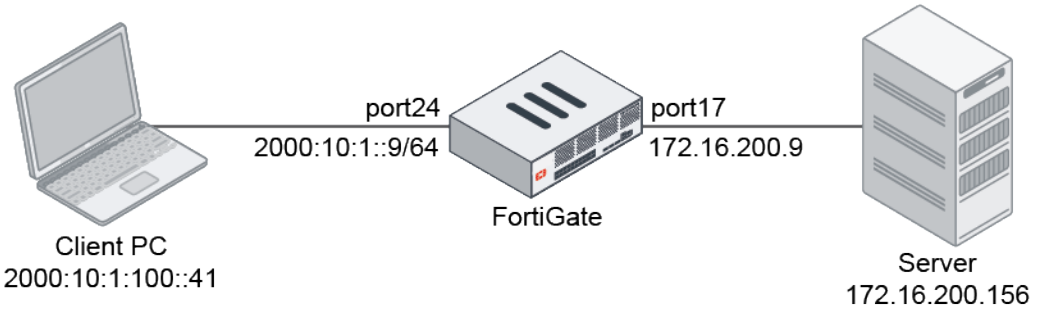
```
(root) # diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=5 expire=56 timeout=0 flags=00000000
sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty
statistic(bytes/packets/allow_err): org=312/3/0 reply=312/3/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=53->17/17->53
hook=pre dir=org act=noop 2000:172:16:101::1:29388->2000:172:16:200::156:128(:::0)
hook=post dir=reply act=noop 2000:172:16:200::156:29388->2000:172:16:101::1:129(:::0)
peer=10.1.100.150:29388->10.1.100.41:0 naf=2
hook=pre dir=org act=noop 10.1.100.41:29388->10.1.100.150:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.150:29388->10.1.100.41:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00001bbc tos=ff/ff ips_view=1024 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

The IPv4 session is between the incoming physical interface port24 and naf.root. The IPv6 session is between the naf.root and the outgoing physical interface port17.

NAT64 policy

In this example, a client PC is using IPv6 and an IPv6 VIP to access a server that is using IPv4. The FortiGate uses NAT64 to translate the request from IPv6 to IPv4 using the virtual interface naf.root. An `ippool` is applied so that the request is SNATed to the `ippool` address (172.16.101.2 - 172.16.101.3).

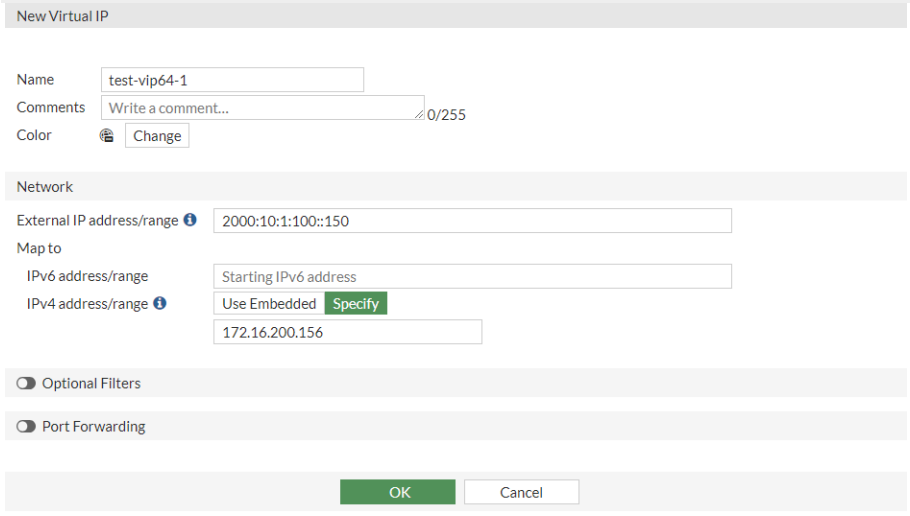
An embedded VIP64 object is used in this configuration so a specific IPv4 mapped IP does not need to be set. The lower 32 bits of the external IPv6 address are used to map to the IPv4 address. Only an IPv6 prefix is defined. In this example, the IPv6 prefix is 2001:10:1:100::, so the IPv6 address 2001:10:1:100::ac10:c89c will be translated to 172.16.200.156.



To create a NAT64 policy in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *IPv6 Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-vip64-1
External IP address/range	2000:10:1:100::150
Map to IPv4 address/range	**Specify**: 172.16.200.156



   d. Click *OK*.
2. Configure the VIP with the embedded IPv4 address enabled:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *IPv6 Virtual IP* tab.
   b. Click *Create new*.

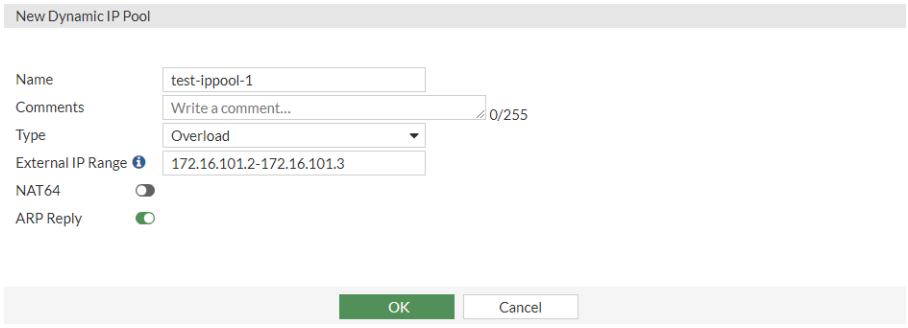
c. Enter the following:

Name	test-vip64-2
External IP address/range	2001:10:1:100::-2001:10:1:100::ffff:ffff
Map to IPv4 address/range	Use Embedded

d. Click *OK*.

3. Configure the IP pool:
   a. Go to *Policy & Objects* > *IP Pools* and select the *IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-ippool4-1
Type	Overload
External IP address/range	172.16.101.2-172.16.101.3
NAT64	Enable



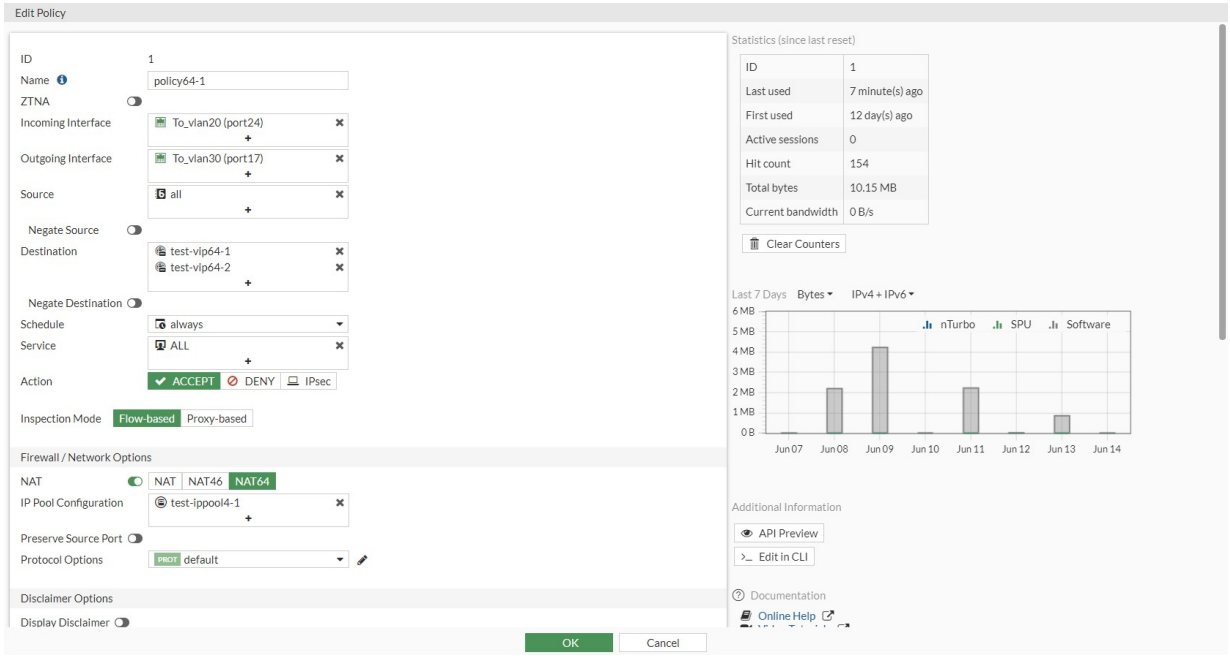
d. Click *OK*.

4. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New* or edit an existing policy.
   b. Enter the following:

c.

Name	policy64-1
Incoming Interface	To_vlan20
Outgoing Interface	To_vlan30
Source	all
Destination	test-vip64-1 test-vip64-2
Schedule	always
Service	ALL
Action	ACCEPT
NAT	NAT64
IP Pool Configuration	test-ippool4-1

d. Configure the other settings as needed.



e. Click *OK*.

To create a NAT64 policy in the CLI:

1. Configure the VIP:

```
config firewall vip6
    edit "test-vip64-1"
        set extip 2000:10:1:100::150
        set nat66 disable
        set nat64 enable
        set ipv4-mappedip 172.16.200.156
    next
end
```

2. Configure the VIP with the embedded IPv4 address enabled:

```
config firewall vip6
    edit "test-vip64-2"
        set extip 2001:10:1:100::-2001:10:1:100::ffff:ffff
        set nat66 disable
        set nat64 enable
        set embedded-ipv4-address enable
    next
end
```

3. Configure the IP pool:

```
config firewall ippool
    edit "test-ippool4-1"
        set startip 172.16.101.2
        set endip 172.16.101.3
        set nat64 enable
        set add-nat64-route enable
```

```
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "policy64-1"
        set srcintf "port24"
        set dstintf "port17"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "test-vip64-1" "test-vip64-2"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname "test-ippool4-1"
    next
end
```

To verify the traffic and session tables:

1. Verify the VIP64 traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
20.578417 port24 in 2000:10:1:100::41 -> 2000:10:1:100::150: icmp6: echo request seq 1
20.578495 naf.root out 2000:10:1:100::41 -> 2000:10:1:100::150: icmp6: echo request seq
1
20.578497 naf.root in 172.16.101.2 -> 172.16.200.156: icmp: echo request
20.578854 port17 out 172.16.101.2 -> 172.16.200.156: icmp: echo request
20.579083 port17 in 172.16.200.156 -> 172.16.101.2: icmp: echo reply
20.579093 naf.root out 172.16.200.156 -> 172.16.101.2: icmp: echo reply
20.579095 naf.root in 2000:10:1:100::150 -> 2000:10:1:100::41: icmp6: echo reply seq 1
20.579377 port24 out 2000:10:1:100::150 -> 2000:10:1:100::41: icmp6: echo reply seq 1
11 packets received by filter
0 packets dropped by kernel
```

2. Verify the session tables for IPv6 and IPv4:

```
(root) # diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=5 expire=56 timeout=0 flags=00000000
sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty
statistic(bytes/packets/allow_err): org=312/3/0 reply=312/3/0 tuples=2
tx speed(Bps/kbps): 55/0 rx speed(Bps/kbps): 55/0
orgin->sink: org pre->post, reply pre->post dev=24->53/53->24
hook=pre dir=org act=noop 2000:10:1:100::41:29949->2000:10:1:100::150:128(:::0)
```

```
hook=post dir=reply act=noop 2000:10:1:100::150:29949->2000:10:1:100::41:129(:::0)
peer=172.16.101.2:45392->172.16.200.156:8 naf=1
hook=pre dir=org act=noop 172.16.101.2:45392->172.16.200.156:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.156:45392->172.16.101.2:0(0.0.0.0:0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=000021ec tos=ff/ff ips_view=1024 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session 1
```

```
(root) # diagnose sys session list
session info: proto=1 proto_state=00 duration=7 expire=54 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=53->17/17->53
gwy=172.16.200.156/172.16.101.2
hook=pre dir=org act=noop 172.16.101.2:45392->172.16.200.156:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.156:45392->172.16.101.2:0(0.0.0.0:0)
peer=2000:10:1:100::150:29949->2000:10:1:100::41:129 naf=2
hook=pre dir=org act=noop 2000:10:1:100::41:29949->2000:10:1:100::150:128(:::0)
hook=post dir=reply act=noop 2000:10:1:100::150:29949->2000:10:1:100::41:129(:::0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=0001347f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

The IPv6 session is between the incoming physical interface port24 and naf.root. The IPv4 session is between the naf.root and the outgoing physical interface port17.

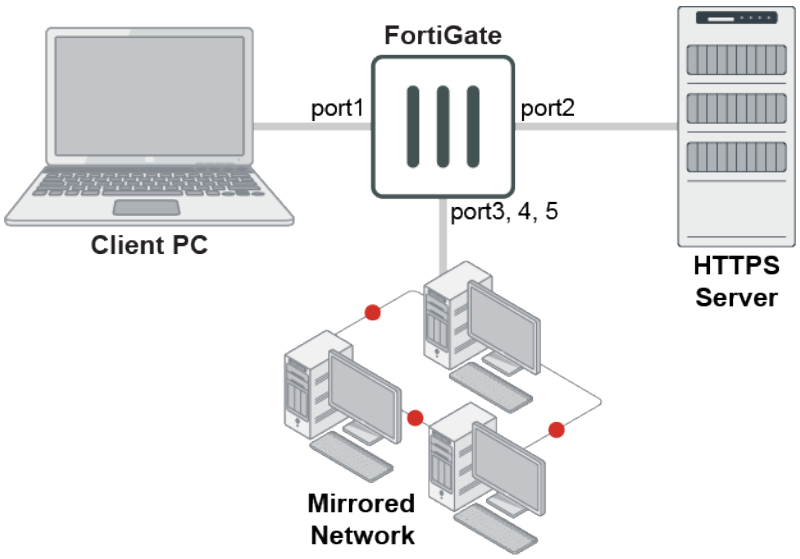
3. Verify the embedded VIP64 traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
7.696010 port24 in 2000:10:1:100::41 -> 2001:10:1:100::ac10:c89c: icmp6: echo request
seq 1
7.696057 naf.root out 2000:10:1:100::41 -> 2001:10:1:100::ac10:c89c: icmp6: echo request
seq 1
7.696060 naf.root in 172.16.101.2 -> 172.16.200.156: icmp: echo request
7.696544 port17 out 172.16.101.2 -> 172.16.200.156: icmp: echo request
7.696821 port17 in 172.16.200.156 -> 172.16.101.2: icmp: echo reply
7.696839 naf.root out 172.16.200.156 -> 172.16.101.2: icmp: echo reply
7.696841 naf.root in 2001:10:1:100::ac10:c89c -> 2000:10:1:100::41: icmp6: echo reply
seq 1
7.697167 port24 out 2001:10:1:100::ac10:c89c -> 2000:10:1:100::41: icmp6: echo reply seq
1
11 packets received by filter
0 packets dropped by kernel
```

Mirroring SSL traffic in policies

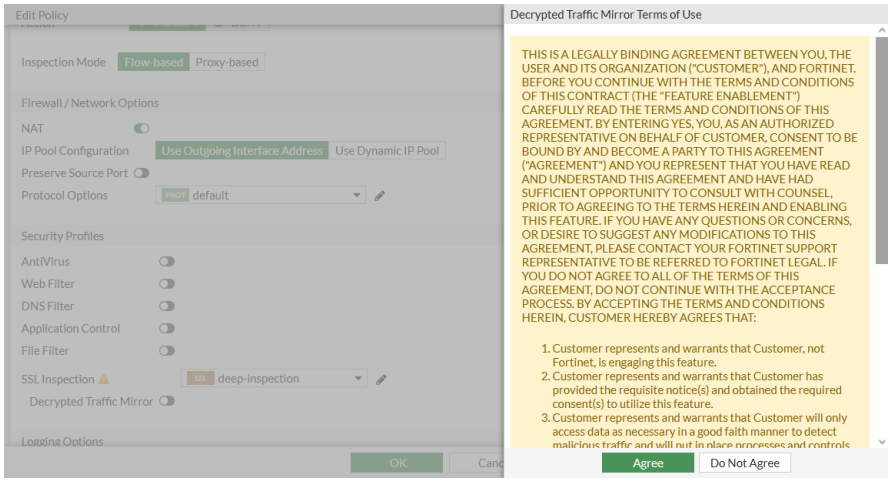
SSL mirroring allows the FortiGate to decrypt and mirror traffic to a designated port. A new decrypted traffic mirror profile can be applied to IPv4, IPv6, and explicit proxy firewall policies in both flow and proxy mode. Full SSL inspection must be used in the policy for the traffic mirroring to occur.

SSL inspection is automatically enabled when you enable a security profile on the policy configuration page.



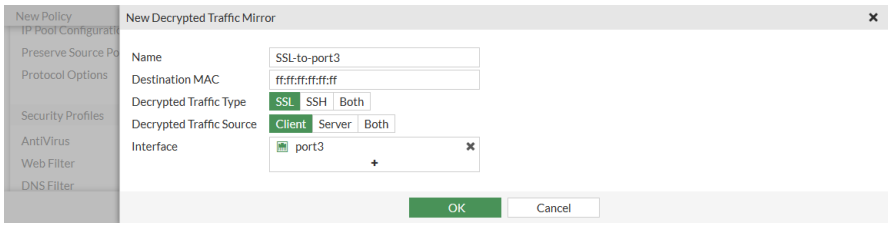
To configure SSL mirroring in a policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy, or edit an existing one.
3. Configure the interfaces, sources, and other required information.
4. In the *Security Profiles* section, for *SSL Inspection*, select *deep-inspection*, or another profile that uses *Full SSL Inspection*.
5. Enable *Decrypted Traffic Mirror* . The terms of use agreement opens.



6. Click *Agree* to accept the terms.

7. In the drop-down list, select a decrypted traffic mirror, or click *Create* to create a new one.
   In this example, a new decrypted traffic mirror is created using the port3 interface.



8. Click *OK* to save the policy.

To configure SSL mirroring in proxy mode in the CLI:

1. Create the decrypted traffic mirror profile:

```
config firewall decrypted-traffic-mirror
    edit SSL-to-port3
        set dstmac ff:ff:ff:ff:ff:ff
        set traffic-type ssl
        set traffic-source client
        set interface port3
    next
end
```

2. Configure the policy to enable SSL traffic mirroring:

```
config firewall policy
    edit 1
        set name "mirror-policy"
        set srcintf "port1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set ssl-ssh-profile "deep-inspection"
        set decrypted-traffic-mirror "SSL-to-port3"

THIS IS A LEGALLY BINDING AGREEMENT BETWEEN YOU, THE USER AND ITS ORGANIZATION
("CUSTOMER"), AND FORTINET. BEFORE YOU CONTINUE WITH THE TERMS AND CONDITIONS OF THIS
CONTRACT (THE "FEATURE ENABLEMENT") CAREFULLY READ THE TERMS AND CONDITIONS OF THIS
AGREEMENT. BY ENTERING YES, YOU, AS AN AUTHORIZED REPRESENTATIVE ON BEHALF OF CUSTOMER,
CONSENT TO BE BOUND BY AND BECOME A PARTY TO THIS AGREEMENT ("AGREEMENT") AND YOU
REPRESENT THAT YOU HAVE READ AND UNDERSTAND THIS AGREEMENT AND HAVE HAD SUFFICIENT
OPPORTUNITY TO CONSULT WITH COUNSEL, PRIOR TO AGREEING TO THE TERMS HEREIN AND ENABLING
THIS FEATURE. IF YOU HAVE ANY QUESTIONS OR CONCERNS, OR DESIRE TO SUGGEST ANY
MODIFICATIONS TO THIS AGREEMENT, PLEASE CONTACT YOUR FORTINET SUPPORT REPRESENTATIVE TO
BE REFERRED TO FORTINET LEGAL. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS
AGREEMENT, DO NOT CONTINUE WITH THE ACCEPTANCE PROCESS. BY ACCEPTING THE TERMS AND
CONDITIONS HEREIN, CUSTOMER HEREBY AGREES THAT:

1. Customer represents and warrants that Customer, not Fortinet, is engaging this
feature.
```

```
2. Customer represents and warrants that Customer has provided the requisite notice(s)
and obtained the required consent(s) to utilize this feature.

3. Customer represents and warrants that Customer will only access data as necessary in
a good faith manner to detect malicious traffic and will put in place processes and
controls to ensure this occurs.

4. Customer represents and warrants that Customer has the right to enable and utilize
this feature, and Customer is fully in compliance with all applicable laws in so doing.

5. Customer shall indemnify Fortinet in full for any of the above certifications being
untrue.

6. Customer shall promptly notify Fortinet Legal in writing of any breach of these Terms
and Conditions and shall indemnify Fortinet in full for any failure by Customer or any
of its employees or representatives to abide in full by the Terms and Conditions above.

7. Customer agrees that these Terms and Conditions shall be governed by the laws of the
State of California, without regards to the choice of laws provisions thereof and
Customer hereby agrees that any dispute related to these Terms and Conditions shall be
resolved in Santa Clara County, California, USA, and Customer hereby consents to
personal jurisdiction in Santa Clara County, California, USA.

Do you want to continue? (y/n) y
    next
end
```

Recognize anycast addresses in geo-IP blocking

An anycast IP can be advertised from multiple locations and the router selects a path based on latency, distance, cost, number of hops, and so on. This technique is widely used by providers to route users to the closest server. Since the IP is hosted in multiple geographic locations, there is no way to specify one single location to that IP.

Anycast IP address ranges can be bypassed in geo-IP blocking. The ISDB contains a list of confirmed anycast IP ranges that can be used for this purpose.

When the source or destination is set to `geoip`, you can enable the `geoip-anycast` option. Once enabled, IPs where the anycast option is set to `1` in `geoip_db` are bypassed in country matching and blocking.

You can only use the CLI to configure this feature.

To enable the geoip-anycast option using the CLI:

```
config firewall policy
   edit 1
      set name "policyid-1"
      set srcintf "wan2"
      set dstintf "wan1"
      set srcaddr "all"
      set dstaddr "test-geoip-CA_1"
      set action accept
      set schedule "always"
```

```
      set service "ALL"
      set geoip-anycast enable
      set logtraffic all
      set nat enable
   next
end
```

To check the geoip-anycast option for an IP address using the CLI:

```
diagnose geoip ip2country 1.0.0.1
   1.0.0.1 - Australia, is anycast ip
```

The anycast IP is `1.0.0.1`.

Matching GeoIP by registered and physical location

IP addresses have both a physical and registered location in the geography IP database. Sometimes these two locations are different. The `geoip-match` command allows users to match an IPv4 address in an firewall policy to its physical or registered location when a GeoIP is used as a source or destination address. IPv6 policies currently support geography address objects but do not support `geoip-match`.

In the following example, the physical location of 220.243.219.10 is CA (Canada), the registered location is CN (China), and it is not an anycast IP.

To configure GeoIP matching based on registered location:

1. Create a firewall policy to match the IP:

```
config firewall policy
    edit 1
        set name "policy_id_1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "test-geoip-CA"
        set action accept
        set schedule "always"
        set service "ALL"
        set geoip-match registered-location
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

   Since CA is applied as a destination address and registered location IP matching is enabled, if the destination IP of the traffic is 220.243.219.10, then the traffic will be blocked because the registered location is CN.

2. Verify that the policy is blocking traffic from the IP address:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
5.383798 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
6.381982 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
7.382608 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
```

```
^C
3 packets received by filter
0 packets dropped by kernel
```

To configure GeoIP matching based on physical location:

1. Create a firewall policy to match the IP:

```
config firewall policy
    edit 1
        set name "policy_id_1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "test-geoip-CA"
        set action accept
        set schedule "always"
        set service "ALL"
        set geoip-match physical-location
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

   Since CA is applied as a destination address and physical location IP matching is enabled, if the destination IP of the traffic is 220.243.219.10, then the traffic will pass through.

2. Verify that the policy is allowing traffic from the IP address:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
5.273985 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
5.274176 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
6.274426 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
6.274438 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
7.273978 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
7.273987 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
^C
6 packets received by filter
0 packets dropped by kernel
```

HTTP to HTTPS redirect for load balancing

You can configure a virtual server with HTTP to HTTPS redirect enabled. When enabled, a virtual server can convert a client's HTTP requests to HTTPS requests. Through this mandatory conversion, HTTP traffic is converted to HTTPS traffic. This conversion improves the security of the user network.

You can only enable this feature by using the CLI. After you enable this feature, traffic flows as follows:

- When FortiGate receives an HTTP request for an external IP, such as 10.1.100.201 in the following example, FortiGate sends an HTTP 303 response back to the original client and redirects HTTP to HTTPS, instead of forwarding the HTTP request to the real backend servers.
- The client browser restarts the TCP session to HTTPS.

- The HTTPS session comes to the FortiGate where a matching firewall policy allows the HTTPS traffic and establishes a secure SSL connection, and then forwards the request to the real backend servers.

To configure virtual server with HTTPS redirect enabled:

1. Create a virtual server with `server-type` set to `http`:
```
config firewall vip
   edit "virtual-server-http"
      set type server-load-balance
      set extip 10.1.100.201
      set extintf "wan2"
      set server-type http
      set ldb-method round-robin
      set extport 80
      config realservers
         edit 1
            set ip 172.16.200.44
            set port 80
         next
         edit 2
            set ip 172.16.200.55
            set port 80
         next
      end
   next
end
```
2. Create a virtual server with `server-type` set to `https` and with the same external IP address:
```
config firewall vip
   edit "virtual-server-https"
      set type server-load-balance
      set extip 10.1.100.201
      set extintf "wan2"
      set server-type https
      set ldb-method round-robin
      set extport 443
         config realservers
            edit 1 set ip 172.16.200.44
            set port 443
         next
         edit 2
            set ip 172.16.200.55
            set port 443
         next
      end
      set ssl-certificate "Fortinet_CA_SSL"
   next
end
```
3. Enable the `http-redirect` option for the virtual server with `server-type` set to `http`:
```
config firewall vip
   edit "virtual-server-http"
      set http-redirect enable
   next
end
```
4. Add the two virtual servers to a policy:
```
config firewall policy
```

```
    edit 9
       set srcintf "wan2"
       set dstintf "wan1"
       set srcaddr "all"
       set dstaddr "virtual-server-http" "virtual-server-https"
       set action accept
       set schedule "always"
       set service "ALL"
       set inspection-mode proxy set logtraffic all
       set auto-asic-offload disable
       set nat enable
    next
end
```

Use Active Directory objects directly in policies

Active Directory (AD) groups can be used directly in identity-based firewall policies. You do not need to add remote AD groups to local FSSO groups before using them in policies.

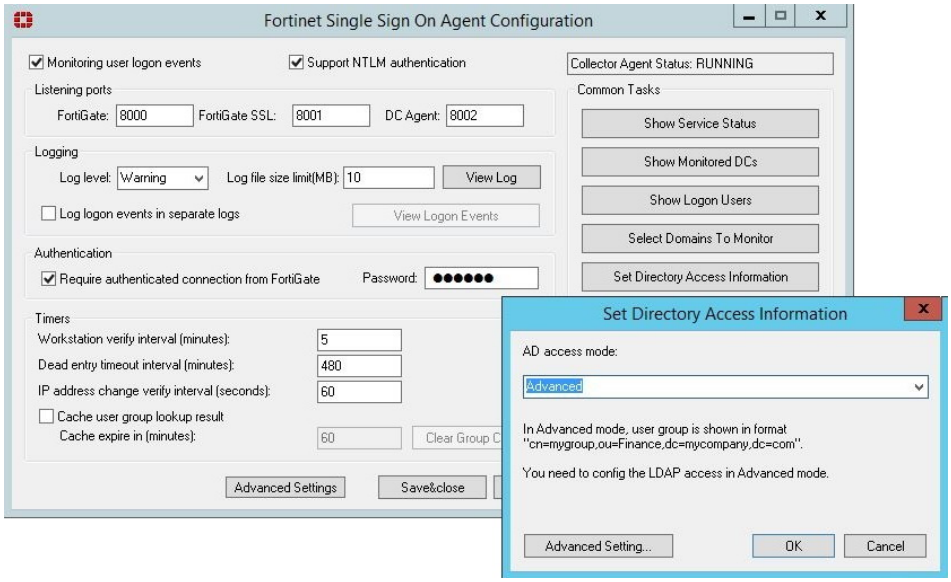
FortiGate administrators can define how often group information is updated from AD LDAP servers.

To retrieve and use AD user groups in policies:

1. Set the FSSO Collector Agent AD access mode on page 1435
2. Add an LDAP server on page 1436
3. Create the FSSO collector that updates the AD user groups list on page 1437
4. Use the AD user groups in a policy on page 1438

Set the FSSO Collector Agent AD access mode

To use this feature, you must set FSSO Collector Agent to *Advanced* AD access mode. If the FSSO Collector Agent is running in the default mode, FortiGate cannot correctly match user group memberships.



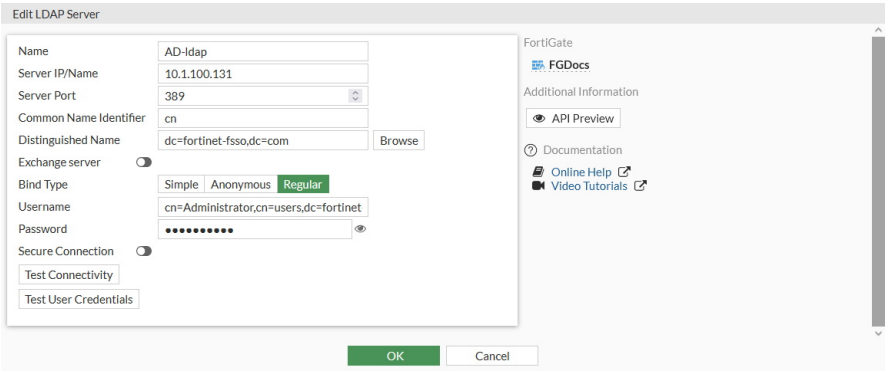
Add an LDAP server

When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 2568 and Configuring client certificate authentication on the LDAP server on page 2582.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 2575.

To add an LDAP server in the GUI:

1. Go to *User & Authentication* > *LDAP Servers*.
2. Click *Create New*.
3. Configure the settings as needed.



4. If secure communication over TLS is supported by the remote AD LDAP server:
   a. Enable *Secure Connection* .
   b. Select the protocol.
   c. Select the certificate from the CA that issued the AD LDAP server certificate.
   If the protocol is LDAPS, the port will automatically change to 636.
5. Click *OK*.

To add an LDAP server in the CLI:

```
config user ldap
    edit "AD-ldap"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password XXXXXXXXXXXXXXXXXXXXXXXX
    next
end
```

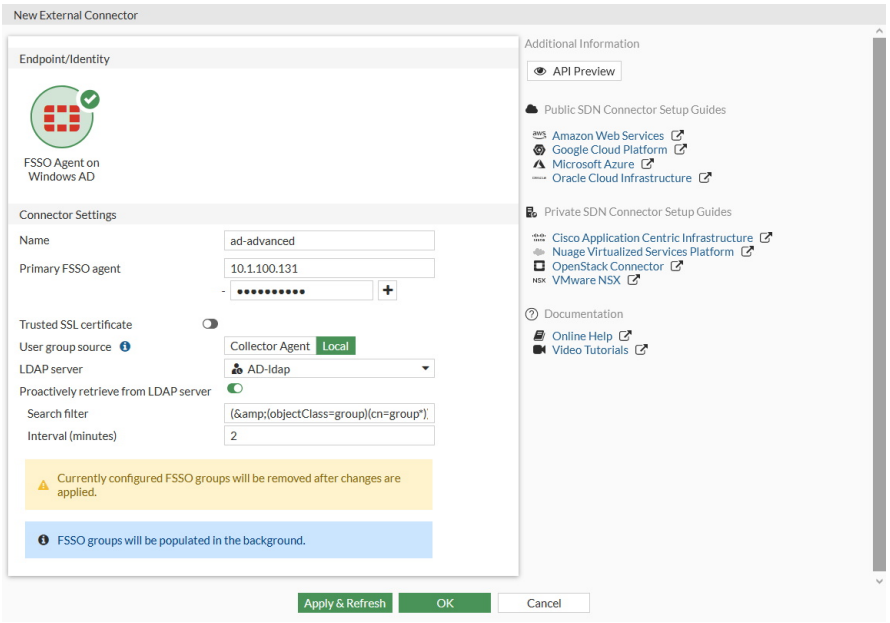
Create the FSSO collector that updates the AD user groups list

To create an FSSO agent connector in the GUI:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. In the *Endpoint/Identity* section, click *FSSO Agent on Windows AD*.
4. Fill in the *Name*
5. Set the *Primary FSSO Agent* to the IP address of the FSSO Collector Agent, and enter its password.
6. Set the *User Group Source* to *Local*.
7. Set the *LDAP Server* to the just created *AD-ldap* server.
8. Enable *Proactively Retrieve from LDAP Server*.
9. Set the *Search Filter* to *(&(objectClass=group)(cn=group*))*.

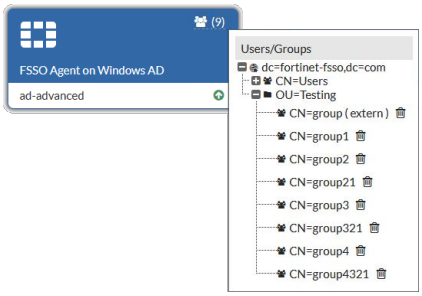
   The default search filter retrieves all groups, including Microsoft system groups. In this example, the filter is configured to retrieve *group1*, *group2*, etc, and not groups like *grp199*.

   The filter syntax is not automatically checked; if it is incorrect, the FortiGate might not retrieve any groups.
10. Set the *Interval (minutes)* to configure how often the FortiGate contacts the remote AD LDAP server to update the group information.



11. Click *OK*.
12. To view the AD user groups that are retrieved by the FSSO agent, hover the cursor over the group icon on the fabric

connector listing.



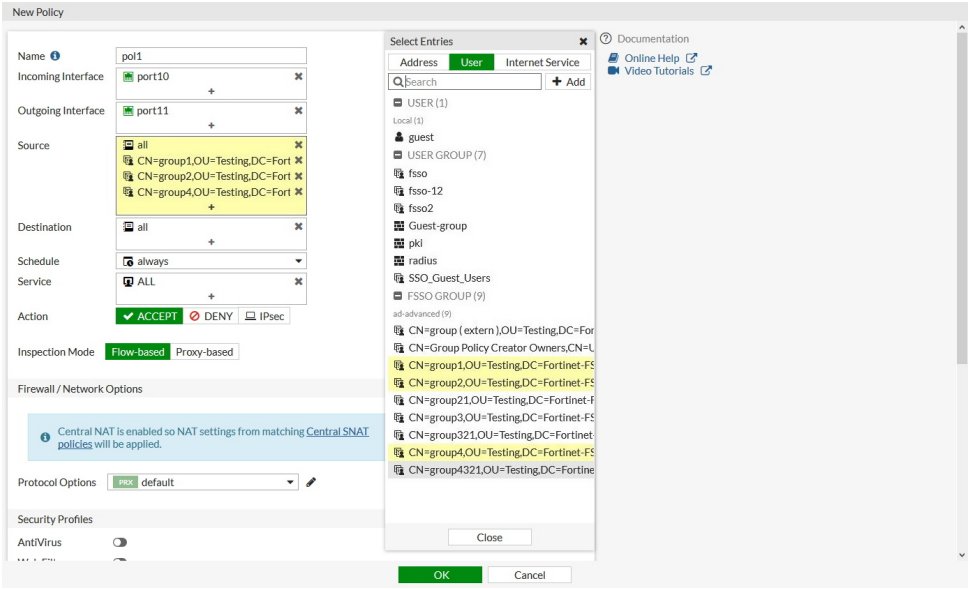
To create an FSSO agent connector in the CLI:

```
config user fsso
    edit "ad-advanced"
        set server "10.1.100.131"
        set password XXXXXXXXXXXXXX
        set ldap-server "AD-ldap"
        set ldap-poll enable
        set ldap-poll-interval 2
        set ldap-poll-filter "(&amp;(objectClass=group)(cn=group*))"
    next
end
```

You can view the retrieved AD user groups with the `show user adgrp` command.

Use the AD user groups in a policy

The AD user groups retrieved by the FortiGate can be used directly in firewall policies.



No session timeout

To allow clients to permanently connect with legacy medical applications and systems that do not have keepalive or auto-reconnect features, the session timeout can be set to never for firewall services, policies, and VDOMs.

The options to disable session timeout are hidden in the CLI.

To set the session TTL value of a custom service to never:

```
config firewall service custom
    edit "tcp_23"
        set tcp-portrange 23
        set session-ttl never
    next
end
```

To set the session TTL value of a policy to never:

```
config firewall policy
    edit 201
        set srcintf "wan1"
        set dstintf "wan2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "TCP_8080"
        set logtraffic disable
        set session-ttl never
        set nat enable
    next
end
```

To set the session TTL value of a VDOM to never:

```
config system session-ttl
    set default never
    config port
        edit 1
            set protocol 6
            set timeout never
            set start-port 8080
            set end-port 8080
        next
    end
end
```

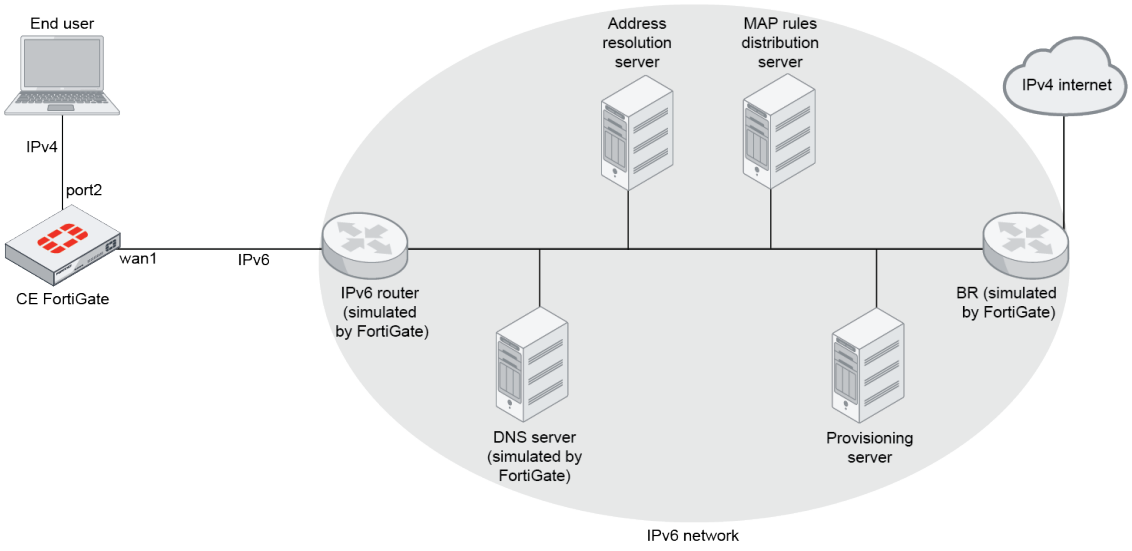
To view a session list with the timeout set to never:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=9 expire=never timeout=never flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
```

```
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=2290/42/1 reply=2895/34/1 tuples=2
tx speed(Bps/kbps): 238/1 rx speed(Bps/kbps): 301/2
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:34256->172.16.200.55:23(172.16.200.10:34256)
hook=pre dir=reply act=dnat 172.16.200.55:23->172.16.200.10:34256(10.1.100.41:34256)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=9 auth_info=0 chk_client_info=0 vd=1
serial=00000b27 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

MAP-E support

On a customer edge (CE) FortiGate, an IPv4-over-IPv6 (MAP-E) tunnel can be created between the FortiGate and the border relay (BR) operating in an IPv6 network. A tunnel interface is created between the FortiGate and BR, which can be applied to firewall policies and IPsec VPN.



To configure a MAP-E tunnel between the FortiGate and the BR:

1. Configure fixed IP mode.
   a. Configure IPv6 on the interface:

```
config system interface
    edit "wan1"
        config ipv6
            set autoconf enable
            set unique-autoconf-addr enable
            set interface-identifier ::6f:6c1f:3400:0
```

```
        end
    next
end
```

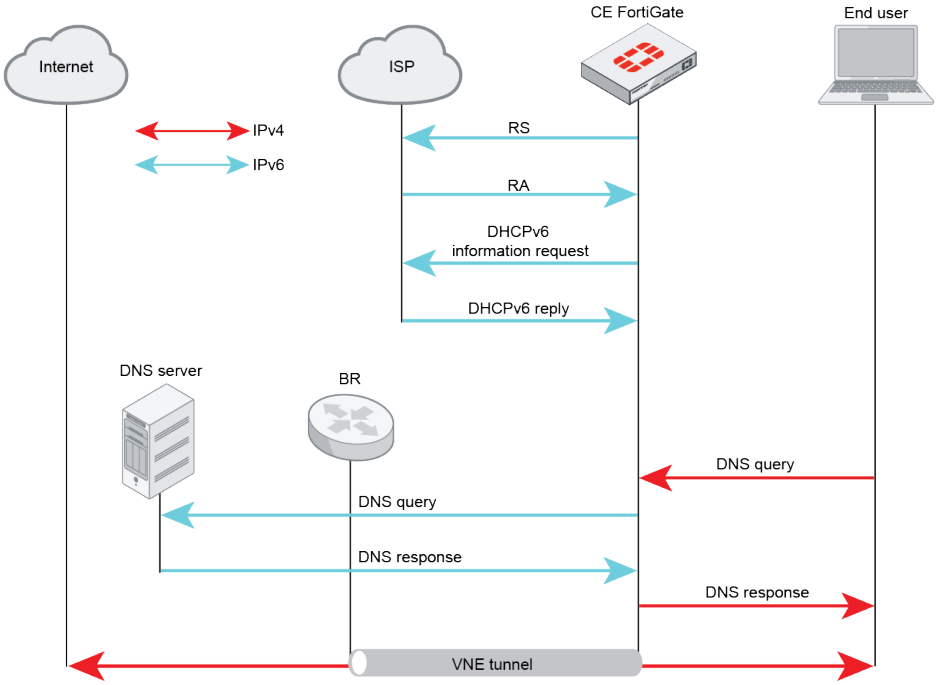
The `interface-identifier` is an IPv6 address. Its last 64-bit will be kept and the rest will be cleared automatically. It will combine with the IPv6 prefix it gets from the IPv6 router to generate the IPv6 address of the interface.

By default, `unique-autoconf-addr` is disabled. It must be enabled so it can handle IPv6 prefix changing.

b. Configure the VNE tunnel:

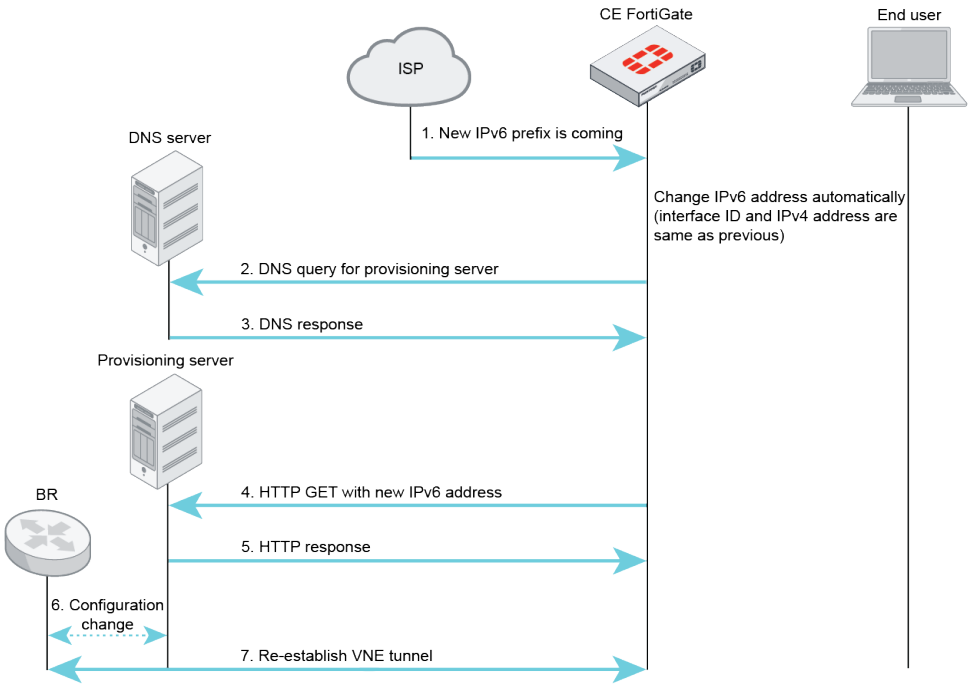
```
config system vne-tunnel
    set status enable
    set interface "wan1"
    set mode fixed-ip
    set ipv4-address 10.10.81.81 255.255.255.0
    set br 2001:160::82
    set update-url "http://qa.forosqa.com/update?user=xxxx&pass=yyyy"
end
```

Initial sequence overview of VNE tunnel under fixed IP mode:



Once the IPv6 address of the FortiGate changes, the tunnel will be down because the BR does not know the FortiGate's new IPv6 address. The FortiGate uses `update-url` to update the new IPv6 address to the provisioning server. The provisioning server updates the FortiGate's IPv6 address to the BR so the VNE tunnel can be re-established.

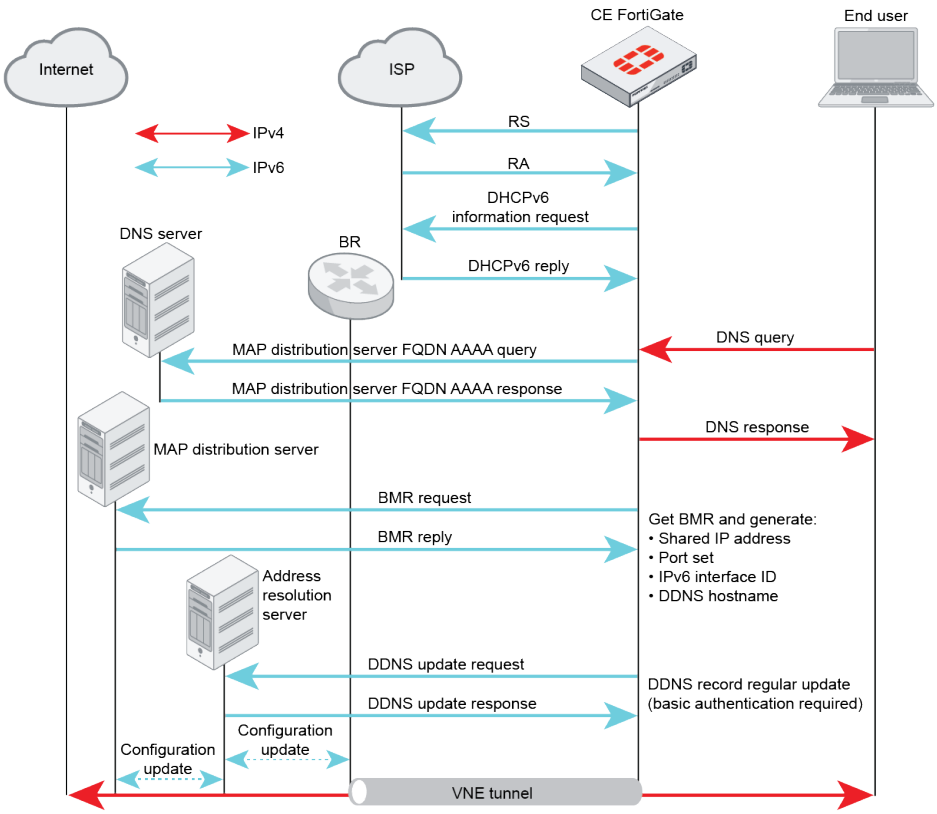
Communication sequence overview of re-establishing VNE tunnel:



2. Configure the VNE tunnel to use MAP-E mode:

```
config system vne-tunnel
    set status enable
    set interface 'wan1'
    set ssl-certificate "Fortinet_Factory"
    set bmr-hostname ********
    set auto-asic-offload enable
    set mode map-e
end
```

Initial sequence overview of VNE tunnel under MAP-E mode:



The FortiGate sends a MAP rule request to the MAP distribution server once the IPv6 address is configured on the FortiGate by RS/RA. Next, the FortiGate will send an AAAA query to get the IPv6 address of the MAP distribution server. After sending the BMR request to the MAP distribution server, the FortiGate will get the IPv4 address, port set, BR IPv6 address, and hostname of the address resolution server from the BMR reply. The VNE tunnel between the FortiGate and BR is now established.

The address resolution server is actually a dynamic DNS. The hostname is used for the FortiGate to maintain an IPv6 address when it changes.

The FortiGate updates the DDNS server with its IPv6 address whenever it updates, which in turn provides the update to the MAP distribution server and BR so they know how to resolve the FortiGate by hostname.

Once the VNE tunnel is established, a tunnel interface is created (`vne.root`), and an IPv4-over-IPv6 tunnel is set up between the FortiGate and BR. The route, firewall policy, and DNS server can now be configured to let the traffic go through the VNE tunnel and the and protect the end-user. The VNE tunnel can also be used in IPsec phase 1.

3. Configure the route:

```
config router static
    edit 1
        set device "vne.root"
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 111
        set name "ff"
        set srcintf "port2"
        set dstintf "vne.root"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set nat enable
    next
end
```

5. Configure the DNS server:

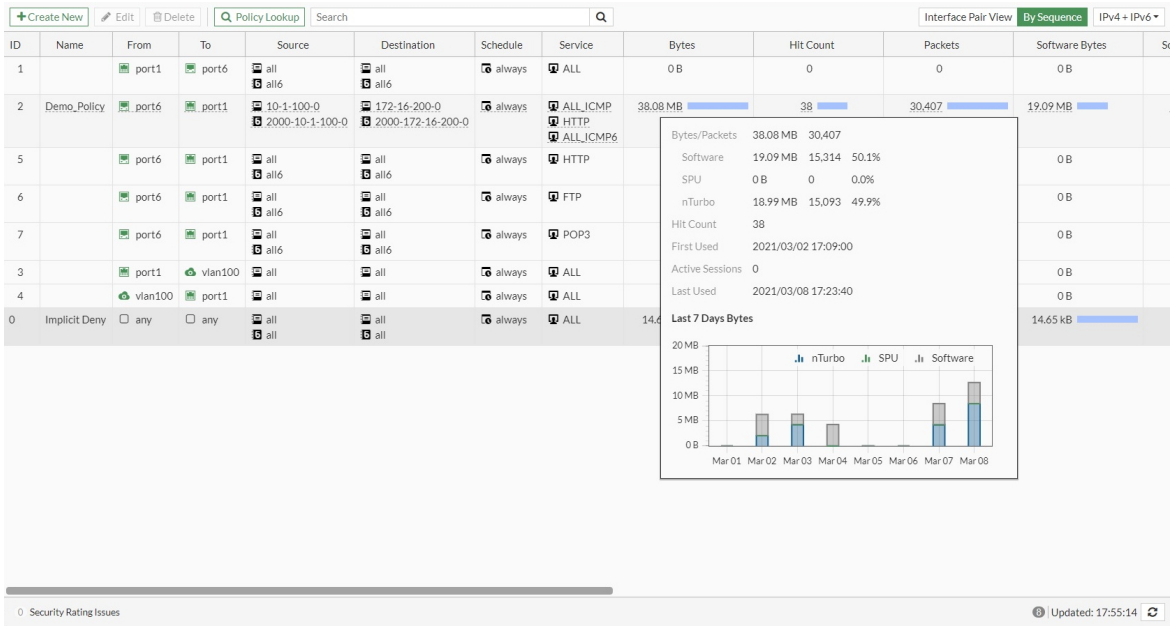
```
config system dns-server
    edit "port2"
    next
end
```

Seven-day rolling counter for policy hit counters

Instead of storing a single number for the hit count and byte count collected since the inception of each policy, seven numbers for the last seven days and an active counter for the current day are stored. The past seven-day hit count is displayed in the policy list and policy pages. A seven-day bar chart shows statistics on each policy page. This feature is currently supported in firewall and multicast policies, but not security policies.

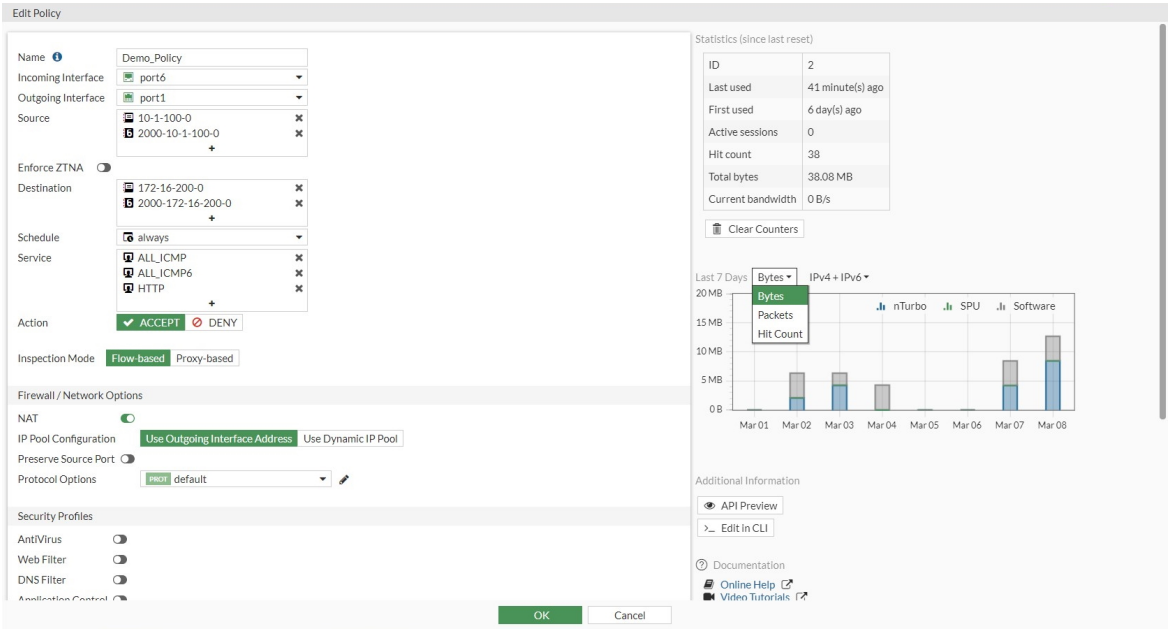
To view the rolling counter information in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* or *Policy & Objects* > *Multicast Policy*.
2. Select a policy and hover over the *Bytes*, *Packets*, or *Hit Count* values to view the tooltip with the corresponding traffic statistics and bar graph (this example uses firewall policies).



3. Click *Edit*. The policy traffic statistics appear in the right-hand side of the page.

4. Use the dropdowns to filter the bar graph data by counter (*Bytes*, *Packets*, or *Hit Count*) and policy type (*IPv4*, *IPv6*, or *IPv4 + IPv6*).



5. Optionally, click *Clear Counters* to delete the traffic statistics for the policy.
6. Click *OK*.

To view the rolling counter information in the CLI:

```
# diagnose firewall iprope show 100004 2
idx=2 pkts/bytes=14709/18777329 asic_pkts/asic_bytes=8087/10413737 nturbo_pkts/nturbo_
bytes=8087/10413737 flag=0x0 hit count:19 (4 7 0 1 1 3 3 0)
    first:2021-03-02 17:09:00 last:2021-03-08 17:23:40
 established session count:0
    first est:2021-03-02 17:11:20 last est:2021-03-08 17:23:40

# diagnose firewall iprope6 show 100004 2
idx=2 pkts/bytes=15698/19307164 asic_pkts/asic_bytes=7006/8578911 nturbo_pkts/nturbo_
bytes=7006/8578911 flag=0x0 hit count:19 (4 7 0 1 3 2 2 0)
    first:2021-03-02 17:10:32 last:2021-03-08 17:23:33
 established session count:0
    first est:2021-03-02 17:11:43 last est:2021-03-08 17:23:33
```

Cisco Security Group Tag as policy matching criteria

The FortiGate can read the Cisco Security Group Tag (SGT) in Ethernet frames, and use them as matching criteria in firewall policies. A policy can match based on the presence of an SGT, or the detection of a specific ID or IDs.

When a packet with a SGT passes through and a session is established, the `ext_header_type=0xc5:0xc5` flag is included in the session table.

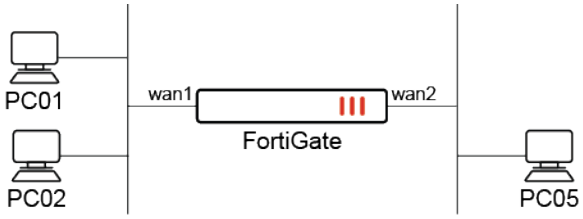
This feature is available in flow mode policies for virtual wire pair policies or policies in transparent mode VDOMs.

To configure a firewall policy to detect SGTs in Ethernet frames:

```
config firewall policy
    edit 1
        set sgt-check {enable | disable}
        set sgt <ID numbers>
    next
end
```

Examples

In these examples, wan1 and wan2 are in a virtual wire pair. Firewall policies are created that pass traffic with SGTs with a specific ID number, any ID number, or either of two specific ID numbers.



To configure the virtual wire pair:

```
config system virtual-wire-pair
    edit "test-vwp-1"
        set member "wan1" "wan2"
        set wildcard-vlan enable
    next
end
```

To configure a firewall policy to match frames that have an SGT with ID 20 and allow them through:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
        set sgt 20
    next
end
```

To configure a firewall policy to match frames that have an SGT with any ID:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
```

```
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
    next
end
```

To configure a firewall policy to match frames that have the SGT with IDs 20 or 21:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
        set sgt 20 21
    next
end
```

Processing only Ethernet frames with a Cisco Security Group Tag

In this example, an Ethernet frame is sent from PC01 with an SGT tag (ID 20), which can pass through to PC05 based on any of the firewall policies in the previous examples.

To verify the configuration:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=10 expire=3593 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty br dst-vis f00
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 10/0 rx speed(Bps/kbps): 5/0
orgin->sink: org pre->post, reply pre->post dev=13->10/10->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.1.11:36970->10.1.2.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.2.11:80->10.1.1.11:36970(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=00:b0:e1:22:cf:e4
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=1
serial=0000183c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
ext_header_type=0xc5:0xc5
total session 1
```

Processing Ethernet frames with a Cisco Security Group Tag and VLAN tag

The FortiGate has the ability to process Ethernet frames with both the Cisco Security Group Tag and VLAN tag.

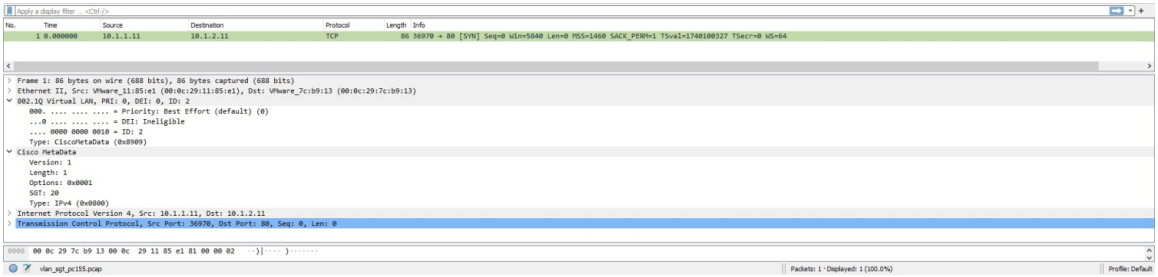
In this example, PC02 is connected to a switch port configured for VLAN 2. An Ethernet frame is sent from PC02 with an SGT tag (ID 20) and VLAN ID (2), which can pass through to PC05 based on any of the firewall policies in the previous examples.

To verify the configuration:

1. Check the session list:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=2007 expire=3482 timeout=3600
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br
statistic(bytes/packets/allow_err): org=164/3/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.1.11:36970->10.1.2.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.2.11:80->10.1.1.11:36970(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=572 auth_info=0 chk_client_info=0 vd=0
serial=0432fb8f tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
vlanid=2
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
            ext_header_type=0xc5:0xc5
```

2. Perform a packet capture on PC05 (Wireshark is used in this example) and check that the packet includes the VLAN ID and Cisco SGT fields.



Virtual patching on the local-in management interface

Virtual patching is a method of mitigating vulnerability exploits by using the FortiGate's IPS engine to block known vulnerabilities. Virtual patching can be applied to traffic destined to the FortiGate by applying the FMWP (Firmware Virtual Patch) database to the local-in interface using local-in policies. Attacks geared towards GUI and SSH management access, for example, can be mitigated using the FMWP database pushed from FortiGuard, thereby virtually patching these vulnerabilities.

When the `virtual-patch` option is enabled in a local-in policy, the IPS engine queries the FortiGuard API server to:

- Obtain a list of vulnerabilities targeting the FortiGate on a particular version
- Determine whether the session destined to the local-in interface on the FortiGate requires a scan by identifying and tagging services in the session. The session's port number and protocol are used to identify the services. Currently only SSL VPN and web GUI services are tagged in a session.

If a tagged session lacks vulnerability signatures for the FortiOS version, then the IPS engine bypasses the session. This optimizes performance by only scanning and dropping sessions that are exploiting a vulnerability.

To configure virtual patching:

```
config firewall local-in-policy
    edit <id>
        set action accept
        set virtual-patch {enable | disable}
    next
end
```

The FortiGate must have a valid FMWR (Firmware) license to install the FMWP database. The FMWP database can be viewed by running the `diagnose autoupdate versions` command.

```
# diagnose autoupdate versions
FMWP Definitions
---------
Version: 23.00084 signed
Contract Expiry Date: Wed Jan  1 2031
Last Updated using manual update on Wed Sep  6 15:19:11 2023
Last Update Attempt: Wed Sep  6 15:40:08 2023
Result: No Updates
```

Once `virtual-patch` is enabled, the WAD process will periodically query vulnerability items from the FortiGuard API server at "productapi.corp.fortinet.com" and forward it to IPS.

Sample vulnerability item found on the FortiGuard API server


```
{"ID":918630,"product":"fortios","vendor":"fortinet","max_version":"7.2.5","min_
version":"7.2.0","severity":"high","vuln_type":"Format String","refs":
["https://www.fortiguard.com/psirt/FG-IR-23-137"],"description":"This indicates detection of
a Zero-Day vulnerability protected by a signature from Fortinet's FortiGuard Labs. This
signature should help mitigate the threat proactively both prior to, and after an official
statement is available from the vendor. Once an official advisory or statement is available
from the vendor, the signature name and its description will be updated to provide more
details regarding this vulnerability. Further details may also be made available in an
advisory on FortiGuard Center (http://www.fortiguard.com).","patch_sig_id":10004065,"patch_
sig_ids":[],"detection_sig_ids":null,"date_added":"2023-08-22T13:09:11","date_
updated":"2023-08-22T13:09:11"}
```


FortiGuard can be queried from the FortiOS CLI for a list of vulnerability rules while specifying parameters for the vendor, version, product, and model by running the `diagnose wad dev-vuln query` command. For example, to query Fortinet Inc.'s FortiOS 7.2.5:

```
# diagnose wad dev-vuln query vendor=fortinet&version=7.2.5&product=fortios
Dev-Vuln Lookup result: success, cache: found, fgd: unknown, item: 0x7fb474e0b4a0
Vulnerability details:
info entry (1):
        'vendor' = fortinet
```

```
       'product' = fortios
         'model' = N/A
   'version.min' = 7.2.0
   'version.max' = 7.2.5
      'firmware' = N/A
         'build' = N/A
    'date_added' = 2023-08-22T13:09:11
  'date_updated' = 2023-08-22T13:09:11
       'sig_id' = 10004065
      'vuln_id' = 918630
     'severity' = 3
...
```

After receiving the vulnerability rules from the WAD process, the IPS engine marks them as virtual patch rules mapped to each CVE vulnerability signature. For example:

FortiOS.NodeJS.Proxy.Authentication.Bypass(CVE-2022-40684)

FortiOS.SSL.VPN.Web.Portal.Password.Improper.Authentication(CVE-2018-13382)

FortiOS.SSL.VPN.Web.Protoal.Pathname.Information.Disclosure(CVE-2018-13379)

To show the list of available FMWP signatures from the FMWP database:

```
# get rule fmwp status
rule-name: "FortiOS.Fclicense.Daemon.Format.String."
rule-id: 10004067
rev: 23.082
date: 1697644800
action: block
status: enable
log: disable
log-packet: disable
severity: 3.high
service: TCP, HTTP
location: server
os: Linux
application: Other
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
vuln_type: Format String
cve: 202329181
fos_comp: Web-GUI
....
```

The following are the diagnose commands:

```
# diagnose ips vpatch {fmwp-status | fmwp-enable-all  | fmwp-reset}
```

`fmwp-status`	Shows the current status of enabled FMWP signatures.
`fmwp-enable-all`	Enable all FMWP signatures in FMWP database.
`fmwp-reset`	Revert the results of `fmwp-enable-all`.

Example

In this example, virtual patching is enabled for the local-in policy and the following scenarios are described:

- FortiGate with an SSL VPN vulnerability
- FortiGate with a web GUI vulnerability
- FortiGate with both an SSL VPN and web GUI vulnerability

To enable virtual patching:

1. Enable virtual patching in the local-in policy:

```
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "ALL"
        set schedule "always"
        set virtual-patch enable
    next
end
```

Because the IPS engine can currently only tag services related to SSL VPN and web GUI signatures, all other protocols are scanned when `service` is set to `ALL`. However, you can bypass scanning of other protocols, such as SSH and FTP, by setting `service` to only `HTTPS`.

2. Observe the outcome of the following scenarios:
   - In this example, FortiOS has an SSL VPN vulnerability. The IPS engine drops SSL VPN traffic to the local-in interface on the FortiGate and bypasses web GUI traffic. Traffic for other services is scanned and passed to the interface.

     Following is a log of the SSL VPN traffic that was dropped because of the vulnerability. Bypassed web GUI traffic did not generate any logs.

     ```
     # diagnose ips vpatch fmwp-status
     Enabled FMWP signatures: 3

       10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.

     1: date=2023-11-07 time=14:53:44 eventtime=1699325624346021995 tz="+1200"
     logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
     vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
     dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
     dstintf="root" dstintfrole="undefined" sessionid=284 action="dropped" proto=6
     service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
     srcport=53250 dstport=11443 hostname="myfortigate.example" url="/error"
     httpmethod="POST" direction="outgoing" attackid=10002887
     ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=99614721 msg="vPatch:
     FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"
     ```

   - In this example, FortiOS has a web GUI vulnerability. The IPS engine drops web GUI traffic to the local-in interface on the FortiGate and bypasses SSL VPN traffic. Traffic for other services is scanned and passed to

the interface.

Following is a log of the web GUI traffic that was dropped because of the vulnerability. Bypassed SSL VPN traffic did not generate any logs.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 2

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=14:55:15 eventtime=1699325715311370215 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=304 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=53622 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin"
agent="Node.js" httpmethod="GET" direction="outgoing" attackid=10002156
ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=99614722 msg="vPatch:
FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096
crlevel="critical"
```

- In this example, FortiOS has an SSL VPN and a web GUI vulnerability. The IPS engine drops both SSL VPN and web GUI traffic to the local-in interface on the FortiGate. Traffic for other services is scanned and passed to the interface.

  Following is a log of the SSL VPN and web GUI traffic that was dropped because of the vulnerability.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=06:42:44 eventtime=1699296164649894963 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=1094 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow."
srcport=44164 dstport=10443 hostname="myfortigate.example" url="/error"
httpmethod="POST" direction="outgoing" attackid=10002887
ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=116392250 msg="vPatch:
FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096 crlevel="critical"

2: date=2023-11-07 time=06:42:09 eventtime=1699296129458704870 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert"
vd="root" severity="critical" srcip=10.1.100.22 srccountry="Reserved"
dstip=10.1.100.1 dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="root" dstintfrole="undefined" sessionid=1066 action="dropped" proto=6
service="HTTPS" policyid=1 attack="FortiOS.NodeJS.Proxy.Authentication.Bypass."
srcport=42352 dstport=443 hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin"
agent="Node.js" httpmethod="GET" direction="outgoing" attackid=10002156
ref="http://www.fortinet.com/ids/VID10002156" incidentserialno=116392236 msg="vPatch:
FortiOS.NodeJS.Proxy.Authentication.Bypass." crscore=50 craction=4096
crlevel="critical"
```

Configuring PCP port mapping with SNAT and DNAT

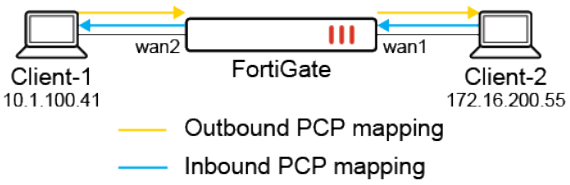
FortiOS supports the Port Control Protocol (PCP) by allowing the FortiGate to act as a PCP server, and dynamically manage network addresses and port translations for PCP clients. The PCP server must be enabled with a pool (`config system pcp-server`). In the firewall policy, enable either `pcp-outbound` or `pcp-inbound` mode and assign the pool.

```
config system pcp-server
    set status {enable | disable}
    config pools
        edit <name>
            set client-subnet <ip_address/subnet>
            set ext-intf <string>
            set extip ip>[-<ip>]
            set extport <port>[-<port>]
            set minimal-lifetime <integer>
            set maximal-lifetime <integer>
            set client-mapping-limit <integer>
            set mapping-filter-limit <integer>
            set allow-opcode {map peer announce}
            set third-party {allow | disallow}
            set multicast-announcement {enable | disable}
            set announcement-count <integer>
            set intl-intf <string>
            set recycle-delay <integer>
        next
    end
end
```

`client-subnet <ip_address/subnet>`	Enter the IP address with subnet from which PCP requests are accepted.
`ext-intf <string>`	Enter the external interface name.
`extip <ip>[-<ip>]`	Enter the IP address or address range on the external interface to map to an address on the internal network.
`extport <port>[-<port>]`	Enter the incoming port number or port range to map to a port number on the internal network.
`minimal-lifetime <integer>`	Set the minimal lifetime of a PCP mapping, in seconds (60 - 300, default = 120).
`maximal-lifetime <integer>`	Set the maximal lifetime of a PCP mapping, in seconds (3600 - 604800, default = 86400).
`client-mapping-limit <integer>`	Mapping limit per client (0 - 65535, default = 0, 0 = unlimited).
`mapping-filter-limit <integer>`	Filter limit per mapping (0 - 5, default = 1).
`allow-opcode {map peer announce}`	Set the allowed PCP OpCode: • `map`: allow MAP OpCode • `peer`: allow PEER OpCode • `announce`: allow ANNOUNCE OpCode

`third-party {allow | disallow}`	Allow/disallow the third-party option.
`multicast-announcement {enable | disable}`	Enable/disable multicast announcements.
`announcement-count <integer>`	Set the number of multicast announcements (3 - 10, default = 3).
`intl-intf <string>`	Enter the internal interface name.
`recycle-delay <integer>`	Set the minimum delay the PCP server will wait before recycling mappings that have expired, in seconds (0 - 3600, default = 0).

The following topology is used to demonstrate two use cases of PCP mapping: with SNAT and DNAT.



Example 1: PCP mapping with SNAT

This example demonstrates how PCP mapping works with SNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-outbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP outbound mapping. When traffic is sent from Client-1 to Client-2, SNAT is performed by the PCP outbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 999
        set name "Outbound-pcp-policy999"
        set srcintf "wan2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
```

```
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set pcp-outbound enable
        set pcp-poolname "pcp-pool1"
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:

   ```
   root@pc41:~# pcp -i 10.1.100.41:41111 -p 172.16.200.55:80 -s 10.1.100.8
   ```

3. On the FortiGate, verify the PCP outbound mappings list:

   ```
   # diagnose firewall pcp-mapping list outbound
   PCP outbound mappings (vdom=root):
   pool:1 nonce:04307eb4037e0448317dc8b7 protocol:6 duration:8 lifetime:900 expiry:893
   intl:10.1.100.41:41111 ext:172.16.200.231:50000 remote:172.16.200.55:80
   ```

4. Send HTTP traffic that passes through the FortiGate and access Client-2 (172.16.200.55:80) from Client-1.
5. On the FortiGate, verify the session list. The source IP address of Client-1 is translated to 172.16.200.231:50000, which follows the PCP outbound mapping:

   ```
   # diagnose sys session list
   session info: proto=6 proto_state=01 duration=8 expire=3599 timeout=3600 flags=00000000
   socktype=0 sockport=0 av_idx=0 use=3
   origin-shaper=
   reply-shaper=
   per_ip_shaper=
   class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
   state=log may_dirty f00 pcp_outbound
   statistic(bytes/packets/allow_err): org=1812/33/1 reply=124168/92/1 tuples=2
   tx speed(Bps/kbps): 204/1 rx speed(Bps/kbps): 13998/111
   orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.55/10.1.100.41
   hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.55:80(172.16.200.231:50000)
   hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.231:50000(10.1.100.41:41111)
   pos/(before,after) 0/(0,0), 0/(0,0)
   misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
   serial=0000b4f8 tos=ff/ff app_list=0 app=0 url_cat=0
   rpdb_link_id=00000000 ngfwid=n/a
   npu_state=0x4000001 no_offload
   no_ofld_reason:  disabled-by-policy
   total session 1
   ```

6. Send HTTP traffic that passes through the FortiGate and access another server from Client-1.
7. On the FortiGate, verify the session list. This time, the source IP address of Client-1 is not translated to 172.16.200.231:50000, since the traffic does not match the existing PCP outbound mapping:

   ```
   # diagnose sys session list
   session info: proto=6 proto_state=01 duration=6 expire=3596 timeout=3600 flags=00000000
   ```

```
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=1449/26/1 reply=98808/72/1 tuples=2
tx speed(Bps/kbps): 215/1 rx speed(Bps/kbps): 14703/117
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.155/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.155:80(172.16.200.8:41111)
hook=pre dir=reply act=dnat 172.16.200.155:80->172.16.200.8:41111(10.1.100.41:41111)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
serial=0000b596 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Example 2: PCP mapping with DNAT

This example demonstrates how PCP mapping works with DNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-inbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP inbound mapping. When traffic is sent from Client-2 to access the external IP of Client-1 (172.16.200.231:50000), traffic passes by due to the PCP inbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 998
        set name "Inbound-pcp-policy998"
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
```

```
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set pcp-inbound enable
        set pcp-poolname "pcp-pool1"
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:

   ```
   root@pc41:~# pcp -i 10.1.100.41:80 -s 10.1.100.8
   ```

3. On the FortiGate, verify the PCP inbound mappings list:

   ```
   # diagnose firewall pcp-mapping list inbound
   PCP inbound mappings (vdom=root):
   pool:1 nonce:35e2ff035b959f7a4e669791 protocol:6 duration:3 lifetime:900 expiry:900
   intl:10.1.100.41:80 ext:172.16.200.231:50000
   ```

4. From Client-2 (172.16.200.55:80), send traffic that passes through the FortiGate and access the external IP of Client-1 (172.16.200.231:50000).
5. On the FortiGate, run a sniffer trace. The traffic is allowed through policy 998, and the destination IP:port is translated from 172.16.200.231:50000 to 10.1.100.41:80, which follows the PCP inbound mapping:

   ```
   # diagnose sniffer packet any 'tcp and port 50000 or port 80' 4
   interfaces=[any]
   filters=[tcp and port 50000 or port 80]
   2.959915 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: syn 3480016601
   2.960051 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: syn 3480016601
   2.960390 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: syn 2813145613 ack 3480016602
   2.960447 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: syn 2813145613 ack
   3480016602
   2.960644 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145614
   2.960664 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145614
   2.961194 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: psh 3480016602 ack
   2813145614
   2.961209 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: psh 3480016602 ack 2813145614
   2.961516 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: ack 3480016686
   2.961533 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: ack 3480016686
   2.993623 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: psh 2813145614 ack 3480016686
   2.993637 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: psh 2813145614 ack
   3480016686
   2.993947 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145875
   2.993962 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145875
   2.995677 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: fin 3480016686 ack
   2813145875
   2.995691 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: fin 3480016686 ack 2813145875
   2.996059 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: fin 2813145875 ack 3480016687
   2.996075 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: fin 2813145875 ack
   3480016687
   2.996230 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145876
   ```

```
2.996245 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145876
```

Only traffic matching the PCP inbound mapping will be forwarded by policy 998. Any other traffic is dropped.

Refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction

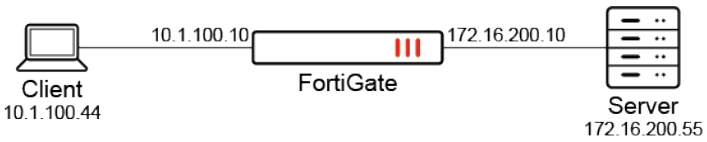
Active sessions can be refreshed for specific protocols and port ranges per VDOM in a specified direction. This option can help prevent potential denial of service (DoS) attacks by controlling the direction of traffic that refreshes existing sessions.

```
config system session-ttl
    config port
        edit <id>
            set protocol <integer>
            set timeout <timeout_value>
            set refresh-direction {both | outgoing | incoming}
        next
    end
end
```

Setting the `refresh-direction` to `outgoing` will use the original direction, while `incoming` will use the reply direction. To refresh in both directions, select `both`.

Example

In this example, active sessions for UDP port 5001 will be refreshed in the incoming direction.



To refresh active sessions for UDP port 5001 in the incoming direction:

1. Configure the global session TTL timer:

```
config system session-ttl
    set default 3600
    config port
        edit 5001
            set protocol 17
            set timeout 5001
            set refresh-direction incoming
            set start-port 5001
            set end-port 5001
        next
    end
end
```

2. Send UDP 5001 traffic from the client to the server.
3. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=77 expire=4923 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=58/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

The timeout and refresh for the reply direction are attached to the session.

4. Send UDP 5001 traffic again from the client to the server.
5. Verify the diagnostics.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 5001' 4
interfaces=[any]
filters=[udp and port 5001]
3.387747 wan2 in 10.1.100.41.2041 -> 172.16.200.55.5001: udp 1
3.387757 wan1 out 172.16.200.10.62458 -> 172.16.200.55.5001: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

   b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=119 expire=4881 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 1/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18
gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
```

```
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the client to the server (outgoing), the expiration timer continues to count down and is not refreshed.

6. Send reverse UDP 5001 traffic from the server to the client.
7. Verify the diagnostics again.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 62458 or port 2041' 4
interfaces=[any]
filters=[udp and port 62458 or port 2041]
3.237328 wan1 in 172.16.200.55.5001 -> 172.16.200.10.62458: udp 1
3.237339 wan2 out 172.16.200.55.5001 -> 10.1.100.41.2041: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

   b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=1710 expire=4995 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=116/4/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18
gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the server to the client (incoming), the expiration timer is refreshed.

Per-policy disclaimer messages

FortiOS supports a customizable captive portal to direct users to install or enable required software.

Per-policy custom disclaimers in each VDOM are supported. For example, you may want to configure three firewall policies, each of which matches traffic from endpoints with different FortiClient statuses:

Endpoint status	FortiOS behavior
Endpoint does not have FortiClient installed.	Traffic matches a firewall policy that displays an in-browser warning to install FortiClient from the provided link.

Endpoint status	FortiOS behavior
Endpoint has FortiClient installed, registered to EMS, and connected to the FortiGate.	Traffic matches a dynamic firewall policy which allows the endpoint to reach its destination via this policy.
Endpoint is deregistered from EMS and disconnected from the FortiGate.	Traffic matches another dynamic firewall policy that displays warning to register FortiClient to EMS.

The replacement message groups and policy disclaimer settings must be enabled.

To enable per-policy disclaimer messages in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Enable *Replacement Message Groups* and *Policy Disclaimer*.
3. Click *Apply*.

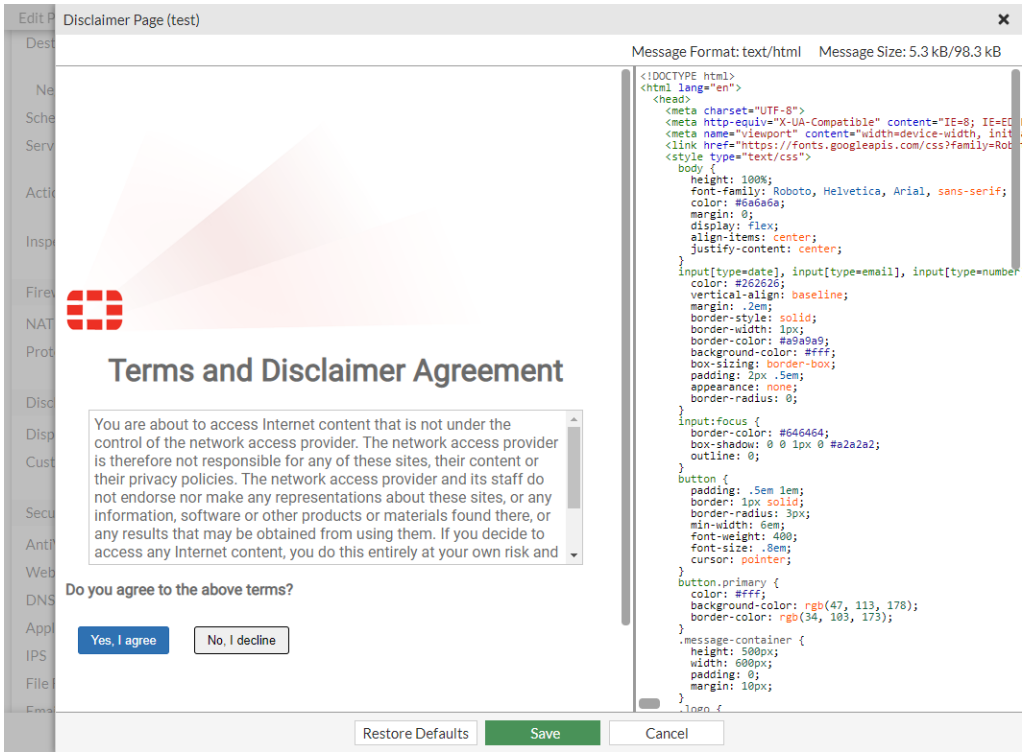
To enable per-policy disclaimer messages in the CLI:

```
config system global
    set gui-replacement-message-groups enable
end
config system settings
    set gui-policy-disclaimer enable
end
```

To configure per-policy disclaimers in the GUI:

1. Ensure the per-policy disclaimer messages option is enabled.
2. Go to *Policy & Objects* > *Firewall Policy*.
3. Edit the policy that applies when an endpoint does not have FortiClient installed.
4. Under *Disclaimer Options*, enable *Display Disclaimer* and *Customize Messages*.
5. Add a replacement message group:
   a. Select an existing replacement message group from the dropdown and click *Edit Disclaimer Message*.
   b. Click *Create*, enter a name, and click *OK*. Select the replacement message group and click *Edit*

Disclaimer Message.



6. Edit the message to warn users to install FortiClient, and provide the FortiClient download link.
7. Click *Save*.
8. Repeat the above steps for each policy that requires a custom disclaimer message.

To configure per-policy disclaimers in the CLI:

```
config firewall policy
    edit 1
        set name "111"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "pc155_address"
        set action accept
        set schedule "always"
        set service "ALL"
        set wsso disable
        set groups "ems_03_group"
        set disclaimer enable
        set replacemsg-override-group "test"
        set nat enable
    next
    edit 4
        set name "44"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "pc5-address"
        set action accept
```

```
        set schedule "always"
        set service "ALL"
        set wsso disable
        set groups "ems_03_group"
        set disclaimer enable
        set replacemsg-override-group "test2"
        set nat enable
    next
    edit 6
        set name "66"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set status disable
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set fsso disable
        set block-notification enable
        set replacemsg-override-group "endpoint-override"
    next
end
```

Address objects

Addresses define sources and destinations of network traffic and can be used in many functions such as firewall policies, ZTNA, etc.

To view the possible uses list of address object usage:

1. Go to *Policy & Objects* > *Addresses*.
2. Click the number under *Ref*. The *Usage of Address:<Predefined address>* pane opens, where *<Predefined address>* is one of the predefined addresses, such as SSLVPN_TUNNEL_ADDR1.
3. In the *Usage of Address:<Predefined address>* pane, click *Possible Uses* to view the list.

When properly set up, these address objects can be used with great flexibility to make the configuration of different functions simpler and more intuitive. When used in a firewall policy, the FortiGate compares the IP addresses contained in packet headers with a policy's source and destination addresses to determine if the policy matches the traffic. The matching of IP addresses in packet headers is also performed for other FortiGate functions configured with address objects.

Address Types

When creating an IPv4 address, there are several different types of addresses that can be specified. Which one is chosen will depend on which method most easily yet accurately describes the addresses that you are trying to include with as few entries as possible based on the information that you have. For instance, if you are trying to describe the addresses of a specific company's web server but do not know how extensive their web server farm is, you would be

more likely to use a Fully Qualified Domain Name (FQDN) rather than a specific IP address. On the other hand, some computers do not have FQDNs and a specific IP address must be used.

The following table provides a short description of the different types of addresses:

Address type	Description
Subnet	The subnet type of address is expressed using a host address and a subnet mask. This is the most flexible of the address types because the address can refer to as little as one individual address (x.x.x.x/32) or as many as all of the available addresses (0.0.0.0/0). See Subnet on page 1466 and Dynamic policy — Fabric devices on page 1467 for more information.
IP range	The IP range type can be used to define a continuous set of IP addresses between one specific IP address and another (inclusive). It is a flexible way to describe a continuous set of addresses while being specific and granular, without needing to fall within the boundaries of standard subnets. See IP range on page 1469 for more information.
FQDN	The Fully Qualified Domain Name (FQDN) address type accepts an address string and resolves it to one or more IP addresses. It relies on DNS to keep up with address changes without having to manually change the IP addresses on the FortiGate. See FQDN addresses on page 1469 for more information. FQDN can also be specified as wildcard addresses such as *.example.com. See Using wildcard FQDN addresses in firewall policies on page 1470 for more information.
Geography	Geography addresses are those determined by the country/region of origin. The IPs for the country/region is automatically determined from the Geography IP database. See Geography based addresses on page 1473 and IPv6 geography-based addresses on page 1475 for more information.
Dynamic	Dynamic address objects are collections of addresses that are integrated from different external sources or other modules within the FortiGate. They can be used in policies that support the dynamic address type and come in different subtypes. • ClearPass: IP addresses gathered from the ClearPass Policy Manager. See ClearPass integration for dynamic address objects on page 1487 for more information. • Device & OS Identification: MAC addresses gathered from device detection that can be filtered by hardware vendor, model, OS, and OS version. See MAC addressed-based policies on page 1497 for more information. • Fabric Connector Address: IP addresses retrieved from SDN connectors, such as public and private cloud connectors. See Getting started with public and private SDN connectors on page 3410 for more information. • FortiNAC Tag: IP addresses collected from FortiNAC. See FortiNAC tag dynamic address on page 1490 for more information.

Address type	Description
	• FortiVoice Tag: IP and MAC addresses collected from FortiVoice. See FortiVoice tag dynamic address on page 1494 for more information. • FortiPolicy Tag: IP addresses pushed from FortiPolicy. See Configuring FortiPolicy on page 3221 for more information. • FortiVoice Tag: IP addresses collected from FortiVoice. • Fortinet Single Sign-On (FSSO): IP addresses of authenticated users from a FSSO collector agent, CPPM by FortiManager, or FortiNAC. See FSSO dynamic address subtype on page 1484 for more information. • Switch Controller NAC Policy Tag: MAC addresses collected from NAC policies.
Device (Mac address)	A MAC address is a link layer-based address type and it cannot be forwarded across different IP segments. In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range. See MAC addressed-based policies on page 1497, Adding MAC-based addresses to devices on page 129, ISDB well-known MAC address list on page 1499, and IPv6 MAC addresses and usage in firewall policies on page 1501 for more information.
Wildcard (CLI only)	Wildcard addresses are addresses that identify ranges of IP addresses, reducing the amount of firewall addresses and security policies required to match some of the traffic on your network. See Wildcard addressing on page 1477 for more information.
Interface subnet (CLI only)	For all interfaces set to a LAN or DMZ role, an option is available and enabled by default to automatically create an address object for the connected network. If the interface's subnet changes, the address object subnet changes too. See Interface subnet on page 1478 for more information.

Address Group

Address groups are designed for ease of use in the administration of the device. If you have several addresses or address ranges that will commonly be treated the same or require the same security policies, you can put them into address groups, rather than entering multiple individual addresses in each policy that refers to them.

There are two different types of address groups and the following table provides a short description of each type:

Address group type	Description
Group	Members of an address group type group can belong to multiple address groups. See Address group on page 1480, Allow empty address groups on page 1482, and Address group exclusions on page 1483 for more information.
Folder	Members or an address group type folder can only belong to a single address folder. See Address folders on page 1481 for more information.

When an address group with no members is configured in a firewall policy, the policy will not match any traffic and will just match the implicit deny policy. See Allow empty address groups on page 1482 for more information.

Subnet

A subnet address object is usually used to refer internal networks or addresses which are defined by the network administrator.

A subnet address usually consists of a network address and a netmask, for example, 192.168.1.0 255.255.255.0. In this example, the network address is 192.168.1.0 and the netmask is 255.255.255.0. The network address defines the network to match and the netmask specify the IP address to match on the network.

In the above example, the subnet address 192.168.1.0 255.255.255.0 would match the following IP addresses:

```
192.168.1.1
192.168.1.2
192.168.1.3
...
192.168.1.255
```

For defining a subnet address object the valid format of IP address and netmask could be either:

`x.x.x.x/x.x.x.x`, such as 192.168.1.0/255.255.255.0

or

`x.x.x.x/x`, such as 192.168.1.0/24

To define a single address using subnet, use the netmask 255.255.255.255 or /32. A warning message will be shown if any other netmask is used and will not let the user save the address object.

To create a subnet address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Subnet* from the dropdown menu.
5. In the *IP/Netmask* field, enter the address and subnet mask according to the format x.x.x.x/x.x.x.x or the short hand format of x.x.x.x/x
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

Dynamic policy — Fabric devices

The dynamic address group represents the configured IP addresses of all Fortinet devices connected to the Security Fabric. It currently includes FortiManager, FortiAnalyzer, FortiClient EMS, FortiMail, FortiAP(s), and FortiSwitch(es). Like other dynamic address groups for fabric connectors, it can be used as an IPv4 address in firewall policies and objects.

The list of firewall addresses includes a default address object called `FABRIC_DEVICE`. You can apply the `FABRIC_DEVICE` object to the following types of policies:

- Firewall policy, including virtual wire pairs, NAT 46, and NAT 64 (IPv4 only)
- IPv4 shaping policy
- IPv4 ACL policy
- Security policy (NGFW mode)

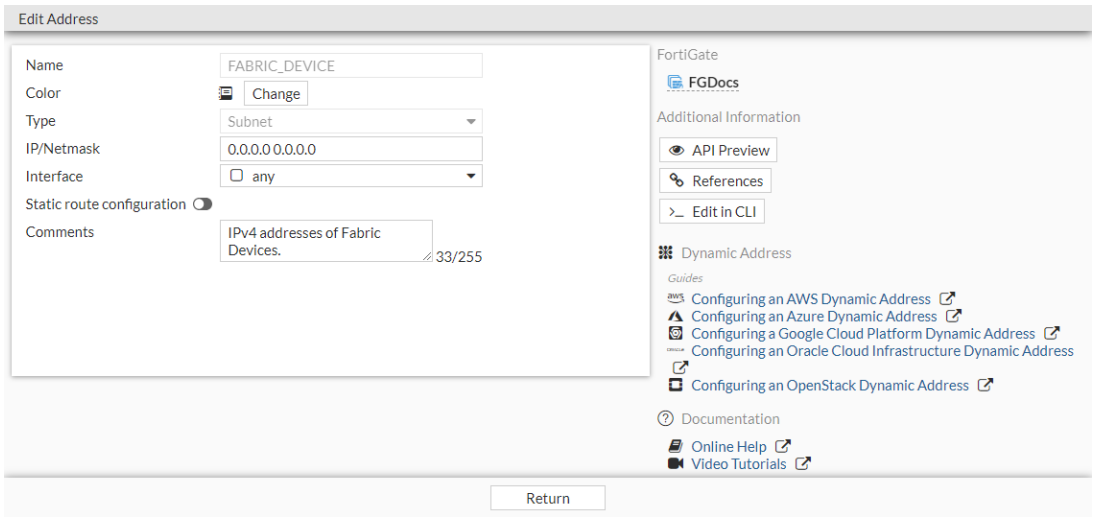
You cannot apply the `FABRIC_DEVICE` object to the following types of policies:

- IPv4 explicit proxy policy

You also cannot use the `FABRIC_DEVICE` object with the following settings:

- Custom extension on `internet-service`
- Exclusion of `addrgrp`

Initially the `FABRIC_DEVICE` object does not have an address value. The address value is populated dynamically as things change. As a result, you cannot edit the `FABRIC_DEVICE` object, add any addresses to the object, or remove any addresses from the object. The *Edit Address* pane in the GUI only has a *Return* button because the object is read-only:



The `FABRIC_DEVICE` object address values are populated based on:

- FortiAnalyzer IP (from the *Fabric Settings* pane)
- FortiManager IP (from the *Fabric Settings* pane)
- FortiMail IP (from the *Fabric Settings* pane)
- FortiClient EMS IP (from the *Fabric Settings* pane)
- FortiAP IPs (from the *FortiAP Setup* pane or DHCP)
- FortiSwitch IPs (from the *FortiSwitch Setup* page or DHCP)

To apply the FABRIC_DEVICE object to a firewall policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy or edit an existing policy.
3. For the *Destination* field, select *FABRIC_DEVICE* from the list of address entries.
4. Configure the rest of the policy as needed.
5. Click *OK*.

To apply the FABRIC_DEVICE object to a firewall policy using the CLI:

```
config firewall address
    edit "FABRIC_DEVICE"
        set type ipmask
        set comment "IPv4 addresses of Fabric Devices."
        set visibility enable
        set associated-interface ''
        set color 0
        set allow-routing disable
        set subnet 0.0.0.0 0.0.0.0
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "FABRIC_DEVICE"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set fsso disable
        set nat enable
    next
end
```

Diagnose commands

You can run diagnose commands to list IP addresses of Fortinet devices that are configured in the Security Fabric or used in a security policy.

To view the IP addresses of Fabric devices:

```
(root) # diagnose firewall sf-addresses list

FabricDevices: 172.18.64.48
FortiAnalyzer: 172.18.60.25
FortiSandbox: 172.18.52.154
FortiManager: 172.18.28.31
FortiClientEMS: 172.18.62.6
FortiAP:
FortiSwitch:
FortiAP/SW-DHCP:
```

To view which IP addresses are used in a security policy:

```
(root) # diagnose ips pme fabric-address list
VDOM 0:
- builtin [mask=0x1e]:
  - type=4: 172.18.62.213
  - type=4: 172.18.62.219
  - type=2: 172.18.70.82
- query:
  - 168.254.1.2
  - 0.0.0.0
  - 168.254.1.2
```

IP range

The IP range type of address can describe a group of addresses while being specific and granular. It does this by specifying a continuous set of IP addresses between one specific IP address and another.

The format would be:

`x.x.x.x-x.x.x.x`, such as 192.168.110.100-192.168.110.120

To create an IP range address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *IP Range* from the dropdown menu.
5. In the *IP Range* field, enter the range of addresses in the following format: x.x.x.x-x.x.x.x (no spaces)
6. In the *Interface* field, leave as the default any or select a specific interface from the drop down menu.
7. Enter any additional information in the *Comments* field.
8. Click *OK*.

FQDN addresses

By using Fully Qualified Domain Name (FQDN) addressing you can take advantage of the dynamic ability of DNS to keep up with address changes without having to manually change the addresses on the FortiGate. FQDN addresses are most often used with external web sites but they can be used for internal web sites as well if there is a trusted DNS server that can be accessed. FQDN addressing also comes in handy for large web sites that may use multiple addresses and load balancers for their web sites. The FortiGate firewall automatically maintains a cached record of all the addresses resolved by the DNS for the FQDN addresses used.

For example, if you were doing this manually and you wanted to have a security policy that involved Google, you could track down all of the IP addresses that they use across multiple countries. Using the FQDN address is simpler and more convenient.

When representing hosts by an FQDN, the domain name can also be a subdomain, such as mail.example.com.

Valid FQDN formats include:

- <host_name>.<top_level_domain_name>, such as example.com
- <host_name>.<second_level_domain_name>.<top_level_domain_name>, such as mail.example.com.

The FortiGate firewall keeps track of the DNS TTLs so as the entries change on the DNS servers the IP address will effectively be updated for the FortiGate. As long as the FQDN address is used in a security policy, it stores the address in the DNS cache.

There is a possible security downside to using FQDN addresses. Using a fully qualified domain name in a security policy means that your policies are relying on the DNS server to be accurate and correct. Should the DNS server be compromised, security policies requiring domain name resolution may no longer function properly.

To create a Fully Qualified Domain Name address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *FQDN* from the dropdown menu.
5. Enter the domain name in the *FQDN* field.
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

Using wildcard FQDN addresses in firewall policies

You can use wildcard FQDN addresses in firewall policies. IPv4, IPv6, ACL, local, shaping, NAT64, NAT46, and NGFW policy types support wildcard FQDN addresses.

For wildcard FQDN addresses to work, the FortiGate should allow DNS traffic to pass through.

Initially, the wildcard FQDN object is empty and contains no addresses. When the client tries to resolve a FQDN address, the FortiGate will analyze the DNS response. The IP address(es) contained in the answer section of the DNS response will be added to the corresponding wildcard FQDN object. It is therefore necessary to have the DNS session-helpers defined in the `config system session-helper` setting.

Since FortiGate must analyze the DNS response, it does not work with DNS over HTTPS.

In FortiOS 7.0 and later, FortiGate supports DNS over TLS. It is possible to analyze DNS responses sent over DoT, as long as there is a firewall policy that allows the DNS traffic from the client and is configured with a DNS filter that supports DoT. For information on configuring this, see DNS inspection with DoT and DoH on page 1742.

When the wildcard FQDN gets the resolved IP addresses, FortiOS loads the addresses into the firewall policy for traffic matching.

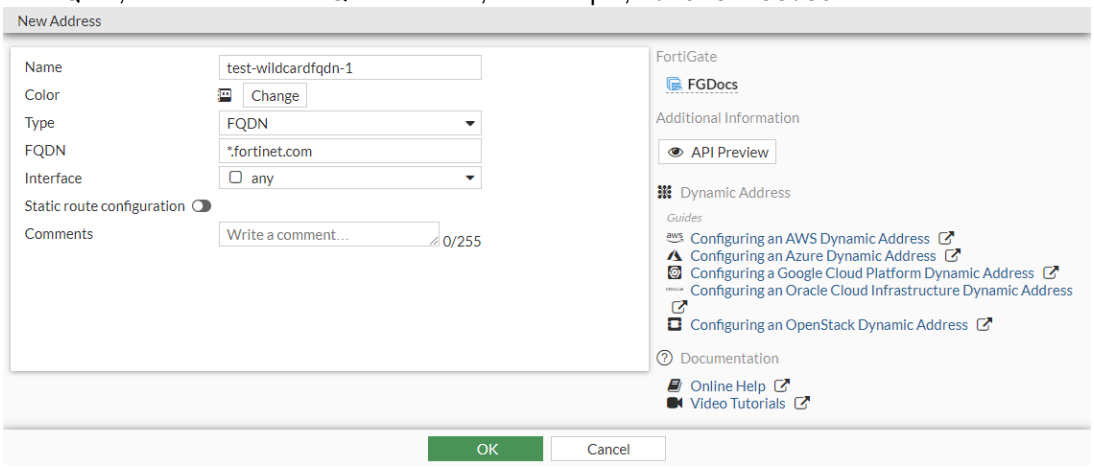
The FortiGate will keep the IP addresses in the FQDN object table as long as the DNS entry itself has not expired. Once it expires, the IP address is removed from the wildcard FQDN object until another query is made. At any given time, a single wildcard FQDN object may have up to 1000 IP addresses.

The DNS expiry TTL value is set by the authoritative name server for that DNS record. If the TTL for a specific DNS record is very short and you would like to cache the IP address longer, then you can extend it with the CLI. See To extend the TTL for a DNS record in the CLI: on page 1472

Wildcard FQDN IPs are synchronized to other autoscale members whenever a peer learns of a wildcard FQDN address.

To create a wildcard FQDN using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Specify a *Name*.
4. For *Type*, select *FQDN*.
5. For *FQDN*, enter a wildcard FQDN address, for example, `*.fortinet.com`.



6. Click *OK*.

To use a wildcard FQDN in a firewall policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. For *Destination*, select the wildcard FQDN.
3. Configure the rest of the policy as needed.
4. Click *OK*.

To create a wildcard FQDN using the CLI:

```
config firewall address
    edit "test-wildcardfqdn-1"
        set type fqdn
```

```
        set fqdn "*.fortinet.com"
    next
end
```

To use wildcard FQDN in a firewall policy using the CLI:

```
config firewall policy
    edit 2
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "test-wildcardfqdn-1"
        set action accept
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
        set nat enable
    next
end
```

To use the diagnose command to list resolved IP addresses of wildcard FQDN objects:

```
# diagnose firewall fqdn list
List all FQDN:
*.fortinet.com: ID(48) ADDR(96.45.36.159) ADDR(192.168.100.161) ADDR(65.39.139.161)
```

Alternatively:

```
# diagnose test application dnsproxy 6
worker idx: 0
vfid=0 name=*.fortinet.com ver=IPv4  min_ttl=3266:0, cache_ttl=0 , slot=-1, num=3,
wildcard=1
       96.45.36.159 (ttl=68862:68311:68311) 192.168.100.161 (ttl=3600:3146:3146)
65.39.139.161
(ttl=3600:3481:3481)
```

To use the diagnose command for firewall policies which use wildcard FQDN:

```
# diagnose firewall iprope list 100004
...
destination fqdn or dynamic address (1):*.fortinet.com ID(48) uuid_idx=57 ADDR
(208.91.114.104) ADDR(208.91.114.142) ADDR(173.243.137.143) ADDR(65.104.9.196) ADDR
(96.45.36.210)
...
```

To extend the TTL for a DNS record in the CLI:

The TTL for DNS records can be configured globally, or for a specific FQDN address. If it is configured for an FQDN address, that setting will supersede the global setting for that address. See Important DNS CLI commands on page 274 for information about configuring a global TTL.

In this the example the `set cache-ttl` value has been extended to 3600 seconds.

```
config firewall address
    edit "fortinet.com"
        set type fqdn
        set fqdn "www.fortinet.com"
        set cache-ttl 3600
    next
end
```

Geography based addresses

Geography addresses are those determined by country of origin. The IP for the country or region is automatically determined from the Geography IP database.

To view IP Geography database:

```
#diagnose autoupdate versions | grep -A 6 "IP Geography DB"
IP Geography DB
---------
Version: 3.00152
Contract Expiry Date: n/a
Last Updated using manual update on Thu Nov 17 17:52:00 2022
Last Update Attempt: Wed Nov 23 10:56:46 2022
Result: No Updates
```

Without a valid license, local IP geography database will continue to work. However the FortiGate will stop receiving geography IP updates from the FortiGuard servers and the geography IP database will no longer be updated. IP geolocation service is part of base services included with all FortiCare support contracts. See FortiGuard Security Services for more information.

To create a geography address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Geography* from the dropdown menu.
5. In the *Country/Region* field, select a single country from the dropdown menu.
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enter any additional information in the *Comments* field.
8. Click *OK*.

Overrides

It is possible to assign a specific IP address range to a customized country ID. Generally, geographic addressing is done at the VDOM level; it could be considered global if you are using the root VDOM, but the `geoip-override setting` is a global setting.

To configure a geography IP override:

1. Assign a specific IP address range to a customized country ID:

```
config system geoip-override
    edit "MyCustomCountry"
        config ip-range
            edit 1
                set start-ip 1.1.1.1
                set end-ip 1.1.1.2
            next
        end
    next
end
```

2. Use get sys geoip-country XX to determine the name corresponding to the custom 2-digit country code A0:

```
# get sys geoip-country A0
id                   : A0
name                 : MyCustomCountry
```

3. Show the full configuration of the geography IP override just created to show that it corresponds to country code A0:

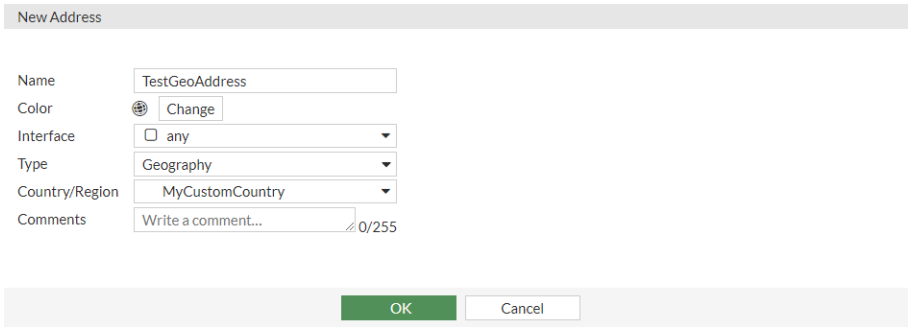
```
# show full sys geoip-override
config system geoip-override
    edit "MyCustomCountry"
        set description ''
        set country-id "A0"
        config ip-range
            edit 1
                set start-ip 1.1.1.1
                set end-ip 1.1.1.2
            next
        end
    next
end
```

To configure a geography address:

1. Enable debug to display the CLI commands running on the backend in response to certain GUI configuration:

```
# diagnose debug enable
# diagnose debug cli 7
Debug messages will be on for 30 minutes.
```

2. Go to *Policy & Objects* > *Addresses* and create a geography address using the previously created custom country code:



3. Observe the corresponding CLI commands run on the backend:

```
FGT # 0: config firewall address
0: edit "TestGeoAddress"
0: set type geography
0: set country "A0"
0: end
```

Diagnose commands

There are a few diagnose commands used with geographic addresses:

```
diagnose firewall ipgeo [country-list | ip-list | ip2country | override | copyright-notice]
```

Diagnose command	Description
country-list	List of all countries.
ip-list	List of the IP addresses associated with the country.
ip2country	Used to determine the physical and registered locations of the IP address as well and if the type is anycast.
override	List of user defined geography data; items configured with the `config system geoip-override` command.
copyright-notice	Shows the copyright notice.

```
diagnose geoip [geoip-query | ip2country | iprange]
```

Diagnose command	Description
geoip-query	Used to determine the complete geolocation of a specific IP address from the FortiGuard IP Geography DB.
ip2country	Used to determine which country a specific IP address is assigned to.
Iprange	List the IP addresses or IP ranges associated with the country.

For more details and examples using these diagnose commands, see the Fortinet Community article Technical Tip: Commands to verify GeoIP information and troubleshoot GeoIP database.

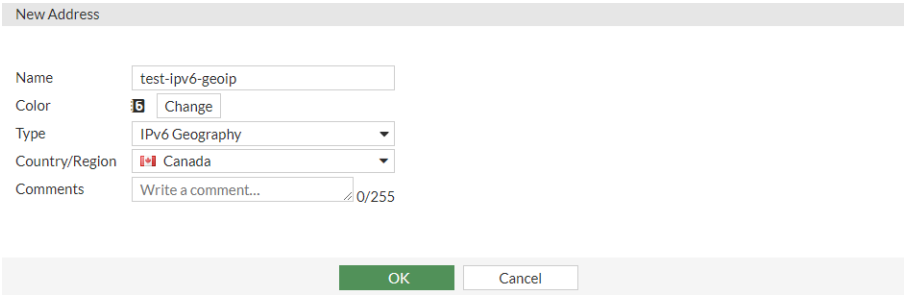
IPv6 geography-based addresses

Geography-based IPv6 addresses can be created and applied to IPv6 firewall policies.

IPv6 geography-based addresses do not support `geoip-override` or `geoip-anycast`.

To create an IPv6 geography-based address in the GUI:

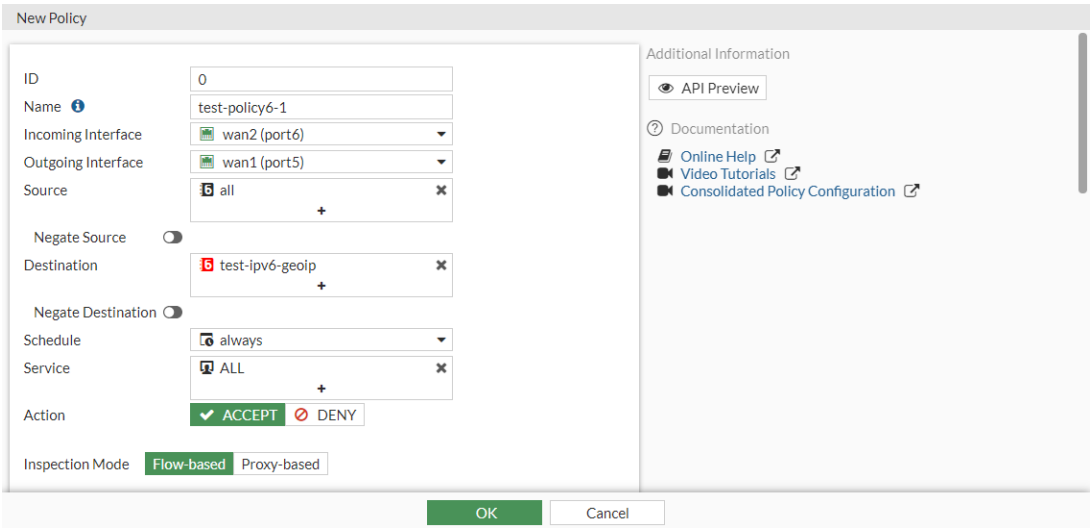
1. Go to *Policy and Objects* > *Addresses* and select *IPv6 Address*.
2. Click *Create new*.
3. Enter a name for the address.
4. Set *Type* to *IPv6 Geography*.
5. Select the *Country/Region* from the list.
6. Optionally, enter comments.



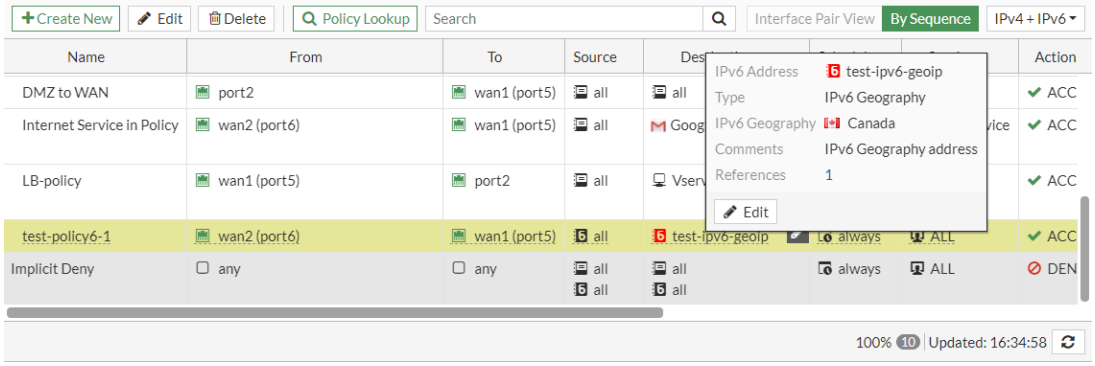
7. Click *OK*.

To use the IPv6 geography address in a policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit an existing policy, or create a new one, using the IPv6 geography address as the *Source* or *Destination Address*.



3. In the policy list, hover over the address to view details.



To configure an IPv6 geography-based address in the CLI:

1. Create an IPv6 geography-based address:

```
config firewall address6
    edit "test-ipv6-geoip"
        set type geography
        set color 6
        set comment "IPv6 Geography address"
        set country "CA"
    next
end
```

2. Use the IPv6 geography-based address in a policy:

```
config firewall policy
    edit 1
        set name "test-policy6-1"
        set srcintf "port6"
        set dstintf "port5"
        set srcaddr6 "all"
        set dstaddr6 "test-ipv6-geoip"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Wildcard addressing

Wildcard addresses are addresses that identify ranges of IP addresses, reducing the amount of firewall addresses and security policies required to match some of the traffic on your network. Wildcard addresses are an advanced feature, usually required only for complex networks with complex firewall filtering requirements. By using these wildcard addresses in the firewall configuration, administrators can eliminate creating multiple, separate IP based address objects and then grouping them to then apply to multiple security policies.

A wildcard address consists of an IP address and a wildcard netmask, for example, 192.168.0.56 255.255.0.255. In this example, the IP address is 192.168.0.56 and the wildcard netmask is 255.255.0.255. The IP address defines the networks to match and the wildcard netmask defines the specific addresses to match on these networks.

In a wildcard netmask, zero denotes ignoring the value of the octet in the IP address. This means the wildcard firewall address matches any number in this address octet. This also means that the number included in this octet of IP address is ignored and can be any number. Usually, if the octet in the wildcard netmask is zero, the corresponding octet in the IP address is also zero.

In a wildcard netmask, a number denotes matching addresses according to how the numbers translate into binary addresses. For example, the wildcard netmask is 255; the wildcard address will only match addresses with the value for this octet that is in the IP address part of the wildcard address. So, if the first octet of the IP address is 192 and the first octet of the wildcard netmask is 255, the wildcard address will only match addresses with 192 in the first octet.

In the above example, the wildcard address 192.168.0.56 255.255.0.255 would match the following IP addresses:

```
192.168.0.56
192.168.1.56
192.168.2.56
...
192.168.255.56
```

The wildcard addresses 192.168.0.56 255.255.0.255 and 192.168.1.56 255.255.0.255 define the same thing since the 0 in the wildcard mask means to match any address in the third octet.

The following is an example of how to configure a wildcard firewall address.

```
config firewall address
    edit example_wildcard_address
        set type wildcard
        set wildcard 192.168.0.56 255.255.0.255
    next
end
```

Wildcard firewall addresses are initially configured in the CLI. You cannot choose wildcard in the GUI when creating the address, but after the address is created in the CLI, it will show up in the GUI. The *Type* field shows a grayed-out value of *Wildcard* and the settings, other than the *Type*, can be edited.

Interface subnet

Interface subnet address type enables an address object to be created automatically for the interface with which it is associated. Once created, the address object is updated when the interface IP/netmask changes on the associated interface.

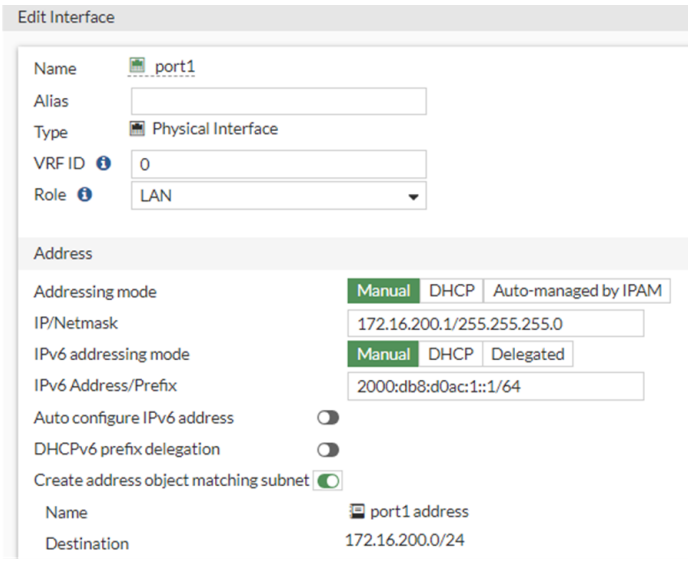
To create the interface subnet address type object, create or edit an interface under *Network* > *Interfaces*, and enable the *Create address object matching subnet* option.

The *Create address object matching subnet* option is automatically enabled and displayed in the GUI when *Role* is set to *LAN* or *DMZ*.

When you disable the *Create address object matching subnet* option, the feature is disabled, and the associated firewall address is deleted.

To create an interface subnet:

1. Go to *Network* > *Interfaces*.
2. Select *Create New* > *Interface* or select existing interface and *Edit*.
3. Set *Role* to either *LAN* or *DMZ*.
4. Verify that *Create address object matching subnet* is available and automatically enabled.



5. Click *OK*.

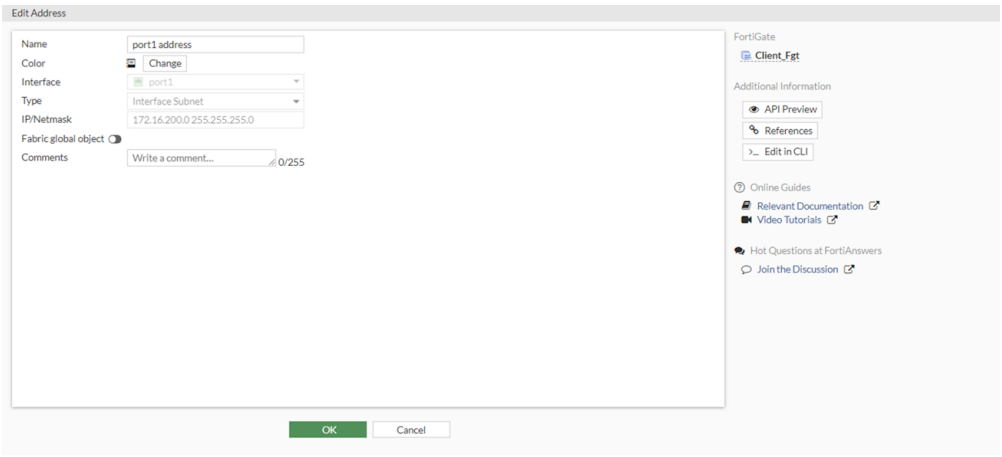
The following is an example of how to configure an interface subnet firewall address on the CLI:

```
config firewall address
    edit "port1 address"
        set type interface-subnet
        set interface "port1"
    next
end
```

Interface subnet addresses are automatically created when *Role* is set to *LAN* or *DMZ* in the *Interface* page, or you can manually configure interface subnet addresses in the CLI. You cannot choose *Interface Subnet* in the GUI when creating the address, but after the address is created, *Interface Subnet* displays in the GUI. However, all the settings are grayed out, except *Name* and *Comments*, which can be edited.

When *Role* is set to *LAN* or *DMZ* in the *Interface* page, the new address object displays on the *Policy & Objects* > *Address* > *Interface Subnet* page.

After the address is created, the subnet is dynamically assigned to the address object, which can be seen in both GUI and CLI. If the interface address changes, the subnet will update dynamically.




```
config firewall address
    edit "port1 address"
        set type interface-subnet
        set subnet 172.16.200.0 255.255.255.0
        set interface "port1"
    next
end
```

Address group

The use of groups is not mandatory. However, adding individual addresses to a policy sometimes becomes tedious. If you use several different addresses with a given policy, these address objects can be grouped into an address group as it is much easier to add or subtract addresses from the group.

Security policies require addresses with homogenous network interfaces. Therefore, address groups should contain only addresses bound to the same network interface or Any.

For example, if address 1.1.1.1 is associated with port1, and address 2.2.2.2 is associated with port2, they cannot be in the same group. However, if 1.1.1.1 and 2.2.2.2 are configured with an interface of Any, they can be grouped, even if the addresses involve different networks.

To create an address group:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Go to *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Group*.
5. Select the + in the *Members* field. The *Select Entries* pane opens.
6. Select members of the group. It is possible to select more than one entry. Select the *x* icon in the field to remove an entry.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

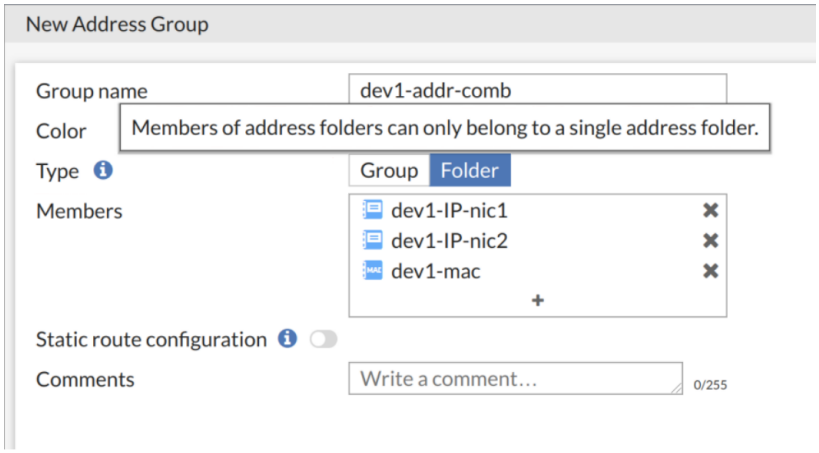
Address folders

Some address objects logically belong to the same device, such as two IPs from the same computer. These address objects can be grouped into an address folder, which is an exclusive list of address objects that do not appear in other address groups or folders.

In the CLI, the folder type can be set after the member list is already populated. If the member list contains an incompatible entry, then the setting will be discarded when the `next`/`end` command is issued. If the folder type is set before the member list is populated, then the possible member entry list will be filtered according to the selected type.

To create an address folder in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Click *Create new* and enter a name.
3. For *Type*, select *Folder*.
4. For *Members*, click the + to add the addresses. Address folders and groups are exclusive, so the *Select Entries* window filters out address objects that are a member of an existing group or folder.



5. Click *OK*.
6. In the address table, expand the *Address Group* section to view the folder (*dev1-addr-comb*). The expandable folder view shows the address folder's child objects:

	safe-network1-devices	Address Group (Folder)	2 entries		0
	dev1-addr-comb	Address Group (Folder)	3 entries		1
	dev1-IP-nic1	Subnet	192.168.1.25/32		1
	dev1-IP-nic2	Subnet	192.168.1.22/32		1
	dev1-mac	Device (MAC Address)	00:0a:95:9d:68:16		1
	dev2-addr-comb	Address Group (Folder)	4 entries		1
	dev2-IP-nic1	Subnet	192.168.1.101/32		1
	dev2-IP-nic2	Subnet	192.168.1.102/32		1
	dev2-IP-nic3	Subnet	192.168.1.103/32		1
	dev2-mac	Device (MAC Address)	11:5b:12:2c:87:02		1

To configure an address folder in the CLI:

```
config firewall addrgrp
    edit "safe-network1-devices"
        set type folder
```

```
        set member "dev1-addr-comb" "dev2-addr-comb"
        set comment ''
        set exclude disable
        set color 13
    next
end

config firewall addrgrp
    edit "dev1-addr-comb"
        set type folder
        set member "dev1-IP-nic1" "dev1-IP-nic2" "dev1-mac"
        set comment ''
        set exclude disable
        set color 18
    next
end

config firewall addrgrp
    edit "dev2-addr-comb"
        set type folder
        set member "dev2-IP-nic1" "dev2-IP-nic2" "dev2-IP-nic3" "dev2-mac"
        set comment ''
        set exclude disable
        set color 5
    next
end
```

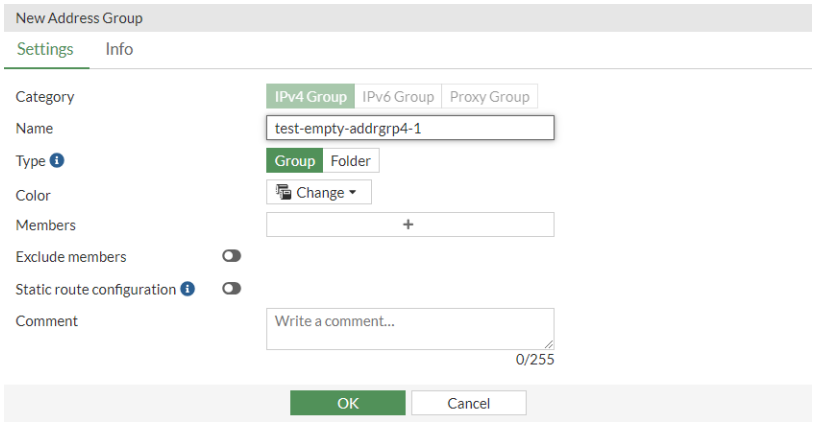
Allow empty address groups

Address groups with no members can be configured in the GUI, CLI, and through the API. In previous versions of FortiOS, error messages appear for empty address groups and they cannot be configured.

When an address group with no members is configured in a firewall policy, the policy will not match any traffic. In this case, policy matching logic will proceed down the list of firewall policies until matching the implicit deny policy.

To create an empty address group in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Click *Create new* .
3. Enter a name.



4. Click *OK*. The *This field is required.* error is not displayed under the *Members* field.

To create an empty address group in the CLI:

```
config firewall addrgrp
    edit "test-empty-addrgrp4-1"
    next
end
```

No error message is returned in the console.

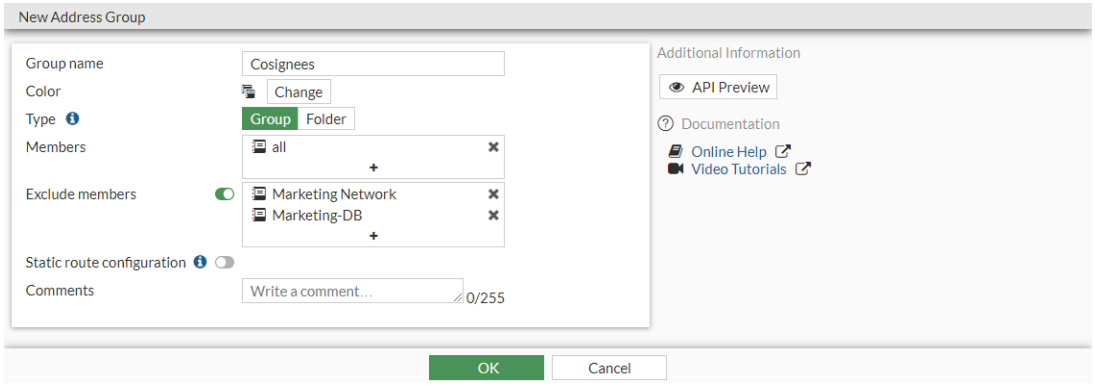
Address group exclusions

Specific IP addresses or ranges can be subtracted from the address group with the *Exclude Members* setting in IPv4 address groups.

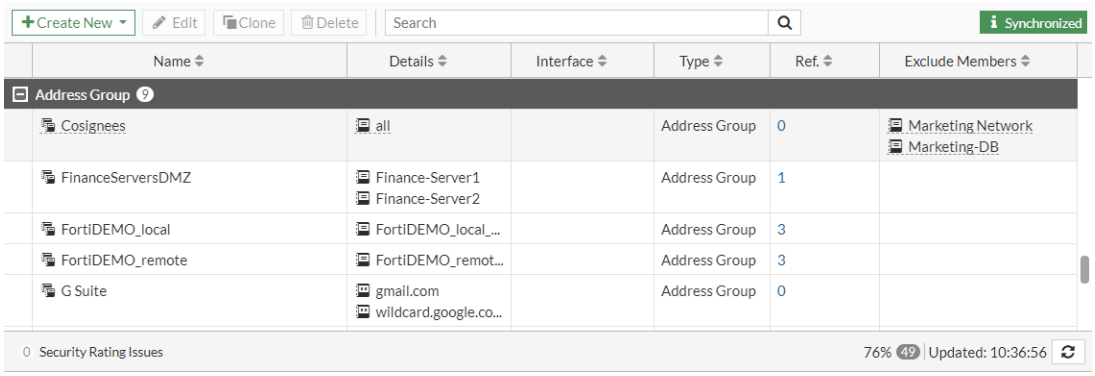
This feature is only supported for IPv4 address groups, and only for addresses with a *Type* of *IP Range* or *Subnet*.

To exclude addresses from an address group using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Create a new address group, or edit an existing address group.
3. Enable *Exclude Members* and click the + to add entries.
4. Configure the other settings as needed.
5. Click *OK*.



The excluded members are listed in the *Exclude Members* column.



To exclude addresses from an address group using the CLI:

```
config firewall addrgrp
    edit <address group>
        set exclude enable
        set exclude-member <address> <address> ... <address>
    next
end
```

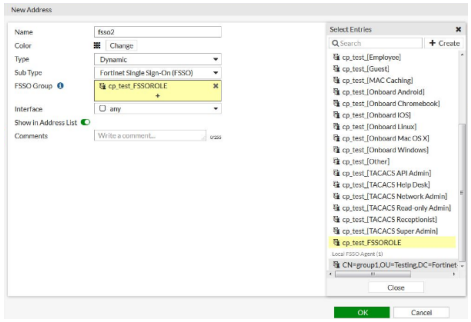
FSSO dynamic address subtype

The Fortinet Single Sign-ON (FSSO) dynamic firewall address subtype can be used in policies that support dynamic address types. The FortiGate will update the dynamic address used in firewall policies based on the source IP information for the authenticated FSSO users.

It can also be used with FSSO group information that is forwarded by ClearPass Policy Manager (CPPM) via FortiManager, and other FSSO groups provided by the FSSO collector agent or FortiNAC. Up to 3000 dynamic FSSO IP addresses are supported per dynamic FSSO group.

To configure FSSO dynamic addresses with CPPM and FortiManager in the GUI:

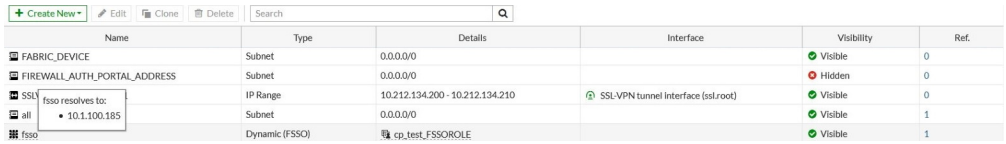
1. Create the dynamic address object:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. For *Type*, select *Dynamic*.
   d. For *Sub Type*, select *Fortinet Single Sign-On (FSSO)*.
   e. Select one or more groups.
   f. Click *OK* to save the configuration.



In the address table, there will be an error message for the address you just created (*Unresolved dynamic address: fsso*). This is expected because there are currently no authenticated FSSO users (based on source IP) in the local FSSO user list.

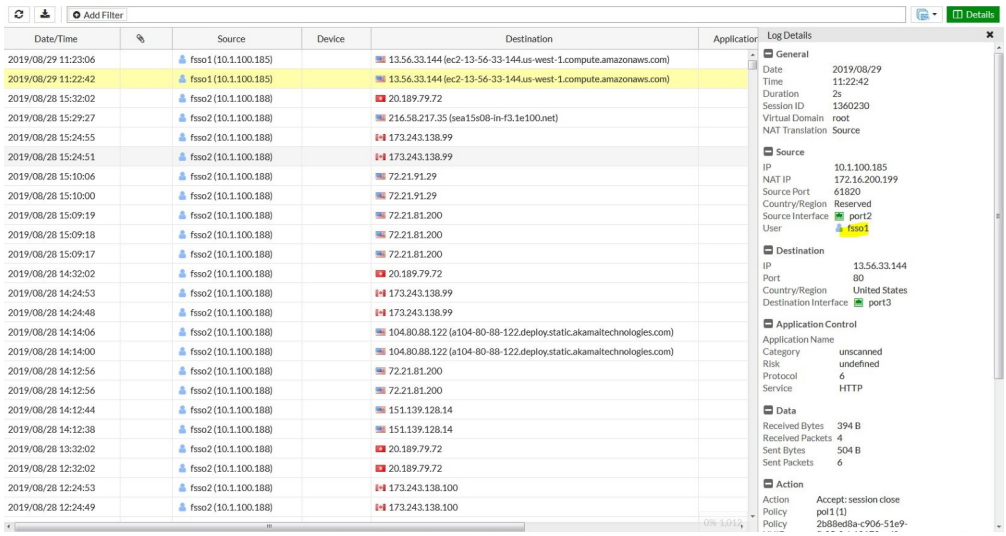
2. Add the dynamic address object to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Create a new policy or edit an existing policy.
   c. For *Source*, select Address from the dropdown list and add the dynamic FSSO address object you just created.
   d. Configure the rest of the policy as needed.
   e. Click *OK* to save your changes.
3. Test the authentication to add a source IP address to the FSSO user list:
   a. Log in as user and use CPPM for user authentication to connect to an external web server. After successful authentication, CPPM forwards the user name, source IP address, and group membership to the FortiGate via FortiManager.
   b. Go to *Monitor* > *Firewall User Monitor* to view the user name (*fsso1*) and IP address.

   c. Go to *Policy & Objects* > *Addresses* to view the updated address table. The error message no longer appears.
   d. Hover over the dynamic FSSO address to view the IP address (*fsso resolves to: 10.1.100.185*).

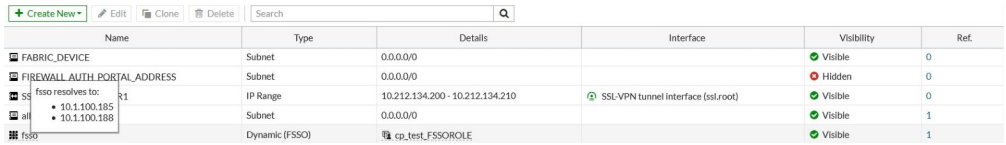


To verify user traffic in the GUI:

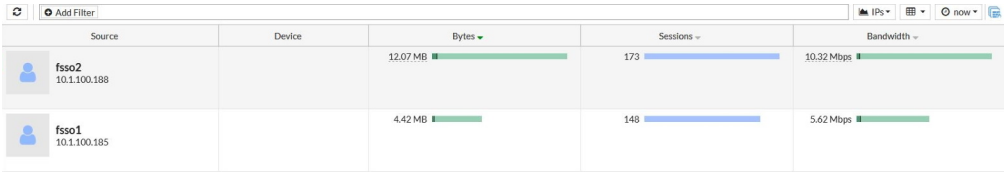
1. Go to *Log & Report* > *Forward Traffic*.
   Details for the user *fsso1* are visible in the traffic log:



   - If another user is authenticated by CPPM, then the dynamic address *fsso* entry in the address table will be updated. The IP address for user *fsso2* (*10.1.100.188*) is now visible:



2. Go to *FortiView* > *Sources* to verify that the users were able to successfully pass the firewall policy.



If a user logs off and CPPM receives log off confirmation, then CPPS updates the FortiGate FSSO user list via FortiManager. The user IP address is deleted from the dynamic FSSO address, and the user is no longer be able to pass the firewall policy.

To configure FSSO dynamic addresses with CPPM and FortiManager in the CLI:

1. Create the dynamic address object:

```
config firewall address
    edit "fsso"
        set type dynamic
        set sub-type fsso
        set fsso-group "cp_test_FSSOROLE"
    next
end
```

2. Add the dynamic address object to a policy:

```
config firewall policy
    edit 1
        set name "pol1"
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "fsso"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set fsso disable
        set nat enable
    next
end
```

To verify user traffic in the CLI:

1. Check the FSSO user list:

```
diagnose debug authd fsso list
----FSSO logons----
IP: 10.1.100.185  User: fsso1  Groups: cp_test_FSSOROLE  Workstation:  MemberOf: FSSO-
CPPM cp_test_FSSOROLE
```

```
Total number of logons listed: 1, filtered: 0
----end of FSSO logons----
```

2. Check the authenticated firewall users list:

```
diagnose firewall auth list
10.1.100.185, fsso1
type: fsso, id: 0, duration: 2928, idled: 2928
server: FortiManager
packets: in 0 out 0, bytes: in 0 out 0
group_id: 2 33554433
group_name: FSSO-CPPM cp_test_FSSOROLE
----- 1 listed, 0 filtered ------
```

After user traffic passes through the firewall, the nu

```
diagnose firewall auth list
10.1.100.185, fsso1
type: fsso, id: 0, duration: 3802, idled: 143
server: FortiManager
packets: in 1629 out 1817, bytes: in 2203319 out 133312
group_id: 2 33554433
group_name: FSSO-CPPM cp_test_FSSOROLE
----- 1 listed, 0 filtered ------
```

ClearPass integration for dynamic address objects

ClearPass Policy Manager (CPPM) can gather information about the statuses of network hosts, for example, the latest patches or virus infections. Based on this information, CPPM send the IP addresses and current states, such as Healthy or Infected, to the FortiGate.

On the FortiGate, the IP addresses received from CPPM are added to a dynamic firewall address with the *clearpass-spt* subtype. This address can be used in any policy that supports dynamic addresses, such as Firewall or SSL-VPN policies.

In this example, you create two dynamic IP addresses that are used in two firewall policies (deny and allow). One policy allows traffic (host state = Healthy), and the other denies traffic (host state = Infected). When CPPM sends the information, the IP addresses are assigned according to their host state: Healthy or Infected.

You can then verify that traffic from the Infected host is denied access by the deny policy, and traffic from the Healthy host is allowed access by the allow policy.

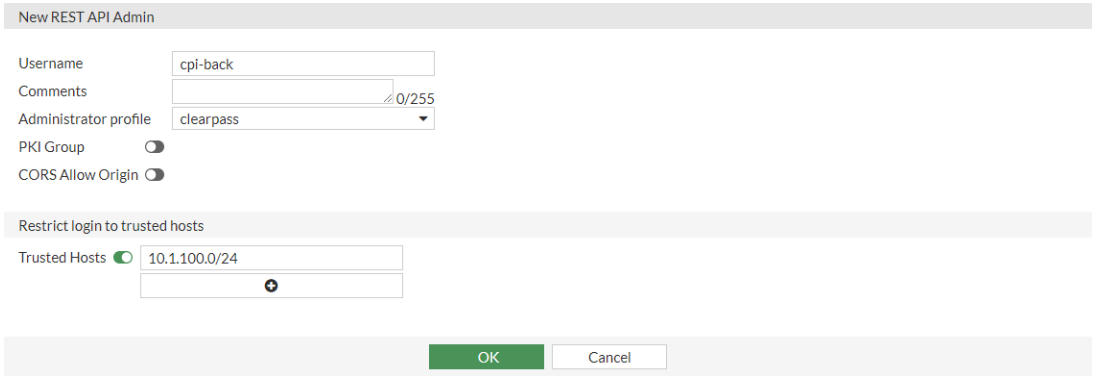
Create a REST API administrator

A REST API administrator is required to generate an authorization token for REST API messages, and to limit hosts that can send REST API messages to the FortiGate.

To create a REST API administrator in the GUI:

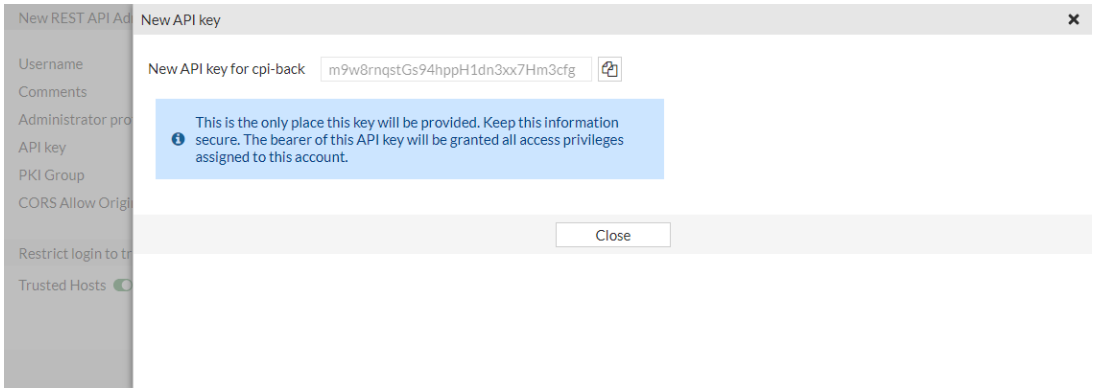
1. Go to *System* > *Administrators*.
2. Click *Create New* > *REST API Admin*.
3. Configure the *Username* and other information as needed.
4. Disable *PKI Group*.

5. In the *Trusted Hosts* field, enter *10.1.100.0/24*.



   For this example, an administrator profile called *clearpass* was created with full read/write access. See Administrator profiles on page 2745 for details.

6. Click *OK*.
   The *New API key* pane opens.



   The API key is the REST API authorization token that is used in REST API messages sent by CPPM to the FortiGate.

7. Copy the API key to a secure location. A new key can be generated if this one is lost or compromised.
8. Click *Close*.

To create a REST API administrator in the CLI:

```
config system api-user
    edit "cpi-back"
        set accprofile "clearpass"
        config trusthost
            edit 1
                set ipv4-trusthost 10.1.100.0 255.255.255.0
            next
        end
    next
end

execute api-user generate-key cp-api
    New API key: 0f1HxGHh9r9p74k7qgfHNH40p51bjs
    NOTE: The bearer of this API key will be granted all access privileges assigned to the
api-user cp-api.
```

Create dynamic IP addresses with the clearpass subtype

Two dynamic IP addresses are required, one for the allow policy, and the other for the deny policy.

To create the dynamic IP addresses:

```
config firewall address
    edit "cppm"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt healthy
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
    edit "cppm-deny"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt infected
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
end
```

Create firewall policies

Two firewall policies are required, one to accept traffic (*cppm-allow*), and the other to deny traffic (*cppm-deny*).

To create the firewall policies in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Configure the allow policy:
   a. Click *Create New*.
   b. Enter a name for the policy.
   c. Set *Source* set to *cppm*.
   d. Set *Action* to *ACCEPT*.
   e. Configure the remaining settings as needed.
   f. Click *OK*.
3. Configure the deny policy:
   a. Click *Create New*.
   b. Enter a name for the policy.
   c. Set *Source* set to *cppm-deny*.
   d. Set *Action* to *DENY*.
   e. Configure the remaining settings as needed.
   f. Click *OK*.

To create the firewall policies in the CLI:

```
config firewall address
    edit "cppm"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt healthy
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
    edit "cppm-deny"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt infected
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
end
```

Verification

Go to *Log & Report* > *Forward Traffic* to review traffic logs and ensure that traffic is allowed or denied as expected.

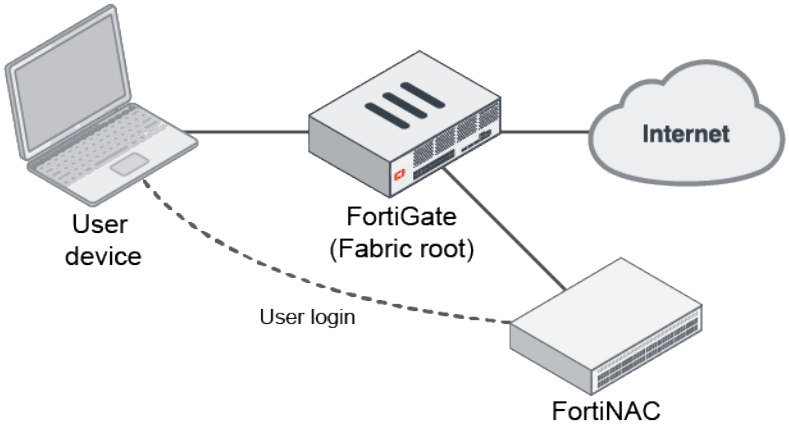
To verify that FortiGate addresses are assigned correctly, enter the following:

```
# diagnose firewall dynamic list
List all dynamic addresses:
cppm-deny: ID(141)
        ADDR(10.1.100.188)

cppm: ID(176)
        ADDR(10.1.100.185)
        ADDR(10.1.100.186)
```

FortiNAC tag dynamic address

The FortiNAC tag dynamic firewall address type is used to store the device IP, FortiNAC firewall tags, and FortiNAC group information sent from FortiNAC by the REST API when user logon and logoff events are registered.

In the following example, the user connecting to the network will be required to first log on to the FortiNAC. When the login succeeds, the logon information is synchronized to the FortiGate using the REST API. The FortiGate updates the dynamic firewall address object with the user and IP information of the user device. This firewall address is used in firewall policies to dynamically allow network access for authenticated users, thereby allowing SSO for the end user.

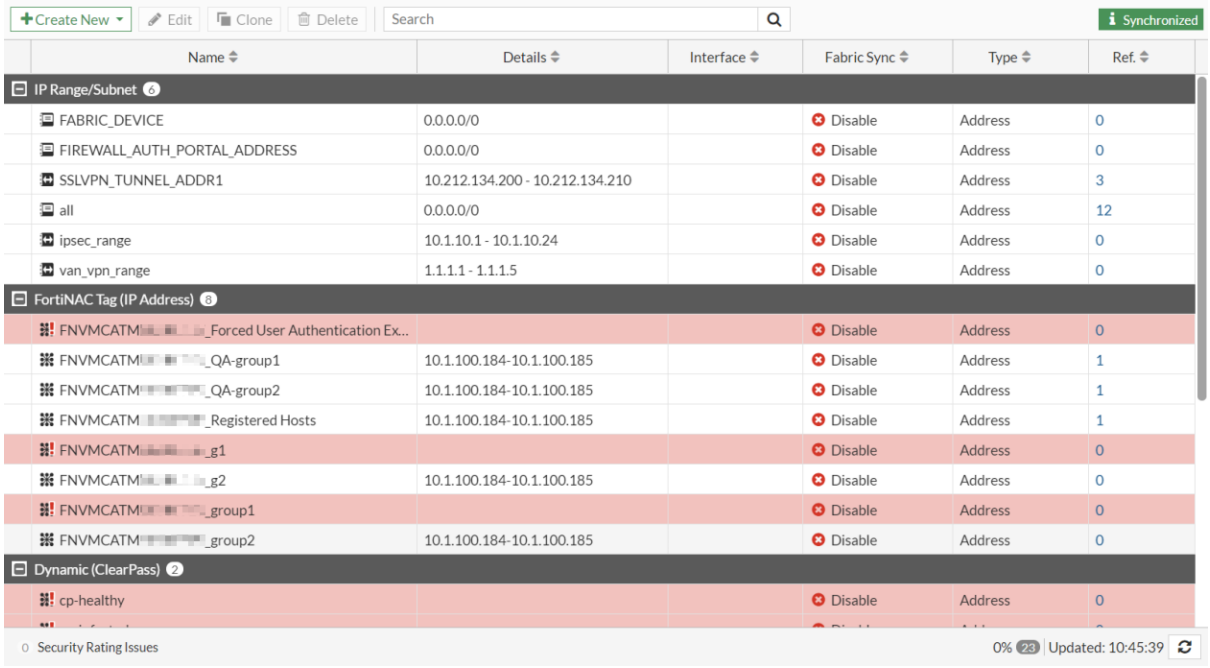


This example assumes the following:

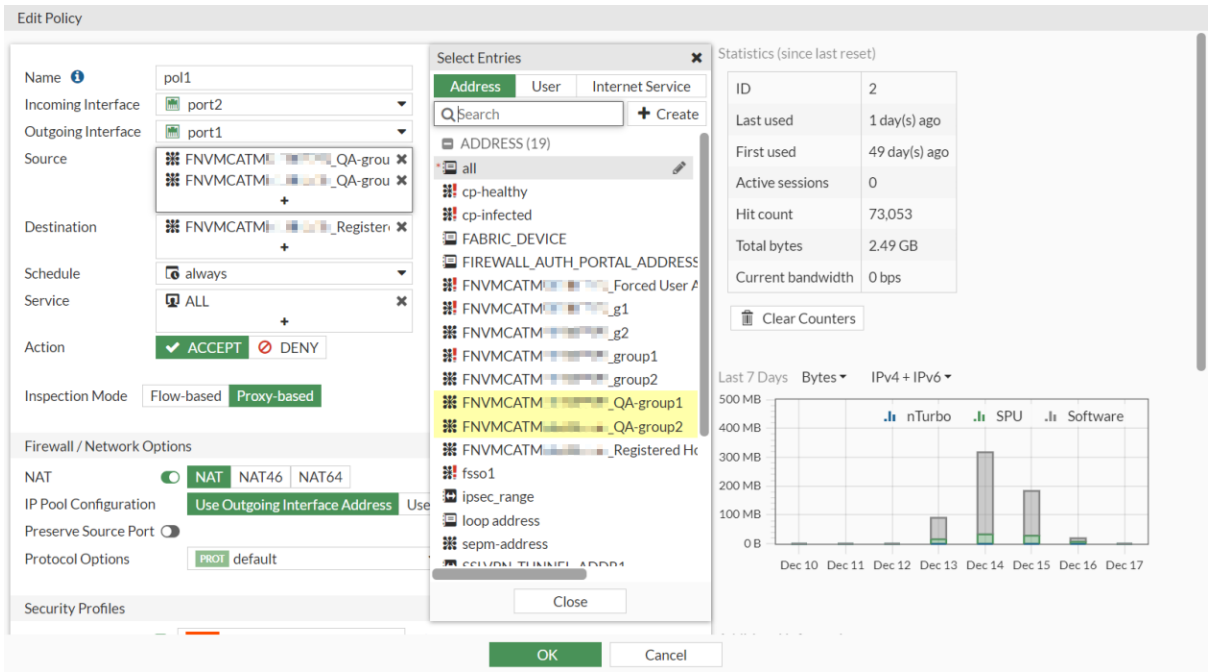
- The FortiGate is the Security Fabric root device (refer to Configuring the root FortiGate and downstream FortiGates on page 3156 for more information).
- The FortiNAC is running version 9.2.2 (or later), and it is connected to the Security Fabric (refer to Configuring FortiNAC on page 3218 for more information).
- Firewall tags and groups have been assigned in FortiNAC to the registered FortiGate (refer to Virtualized Devices for more information). Unlike firewall tags, which are simple labels that can be configured on FortiNAC, firewall groups can be local, built-in, user-defined, or remote user groups imported from a remote server used for user authentication. Only groups that the user of the current logon event belongs to are sent to the FortiGate. Firewall tags are sent for all user authentication.

To use a FortiNAC tag dynamic firewall address in a policy:

1. Trigger two user logon events on the FortiNAC.
2. In FortiOS, go to *Policy & Objects* > *Addresses*, and expand the *FortiNAC Tag (IP Address)* section to view the newly created dynamic firewall address objects. The dynamic firewall addresses matching the current user logon status on FortiNAC have the current IP address of user devices. The addresses without matching user logons are marked with a red exclamation mark (!).

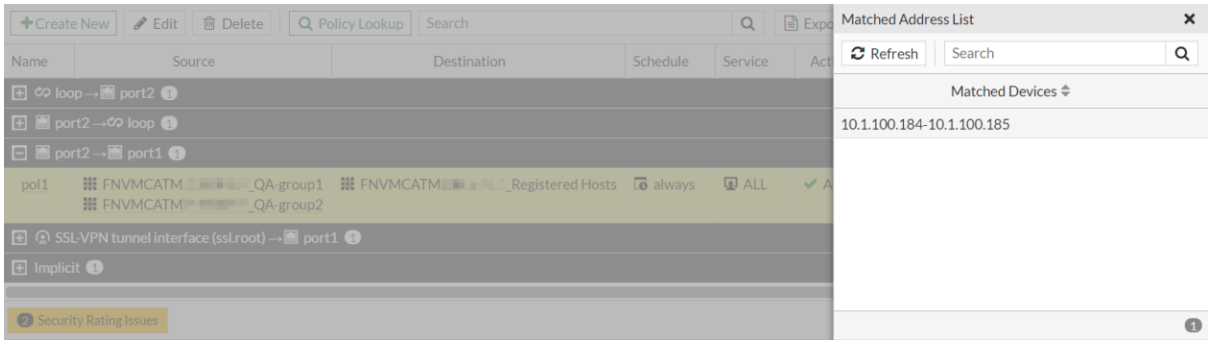


3. Go to *Policy & Objects* > *Firewall Policy* and click *Create New* or edit an existing policy. FortiNAC tag dynamic firewall address an be used as source or destination addresses.

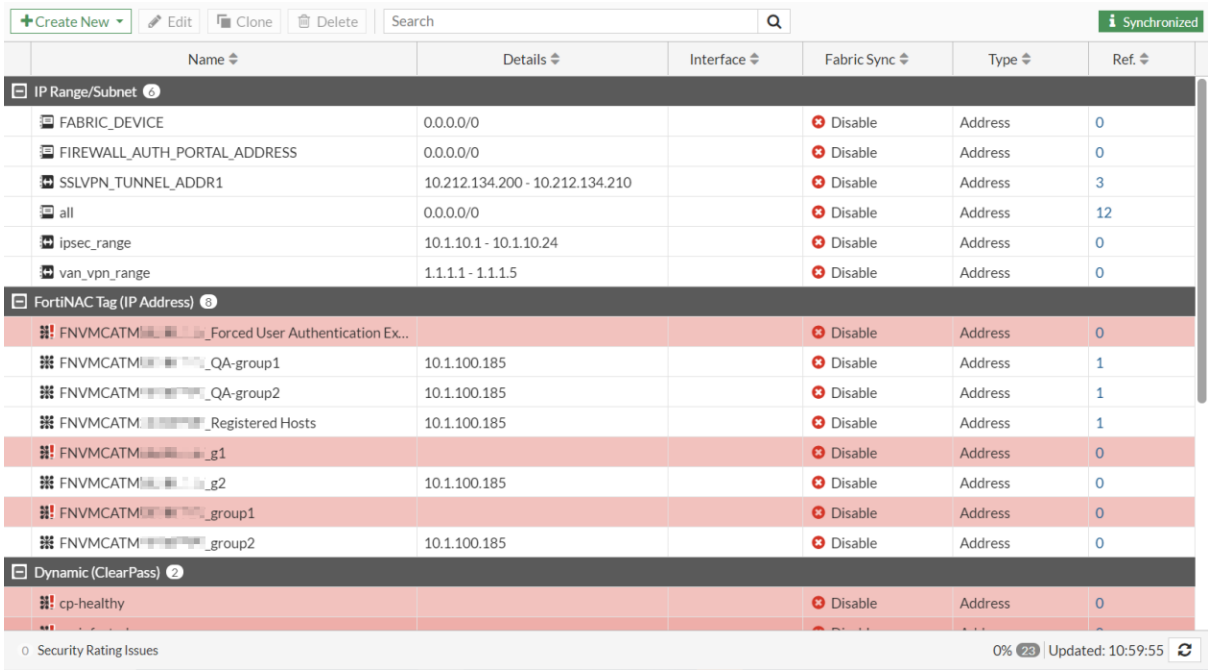


4. Configure the settings as needed, then click *OK*. In this policy, traffic can only pass if it originates from any of the mapped IP addresses (10.1.100.184 and 10.1.100.185); other traffic cannot pass.

5. Hover over the address in the policy, then in the tooltip, click *View Matched Addresses*.

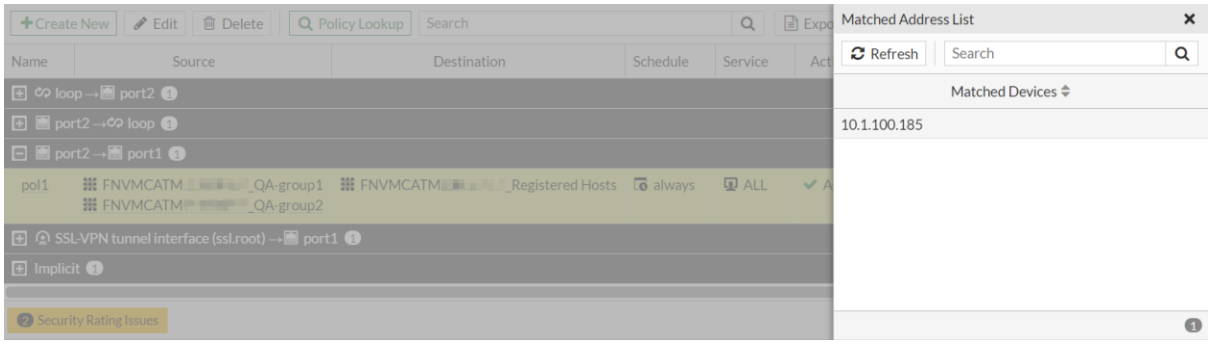


6. Have one of the users log off from the FortiNAC.
7. In FortiOS, go to *Policy & Objects* > *Addresses* and verify the *FortiNAC Tag* addresses. A user logged off from 10.1.100.184, so now only 10.1.100.185 is mapped to the dynamic firewall objects.



All firewall policies using those objects are automatically updated.

8. Go to *Policy & Objects* > *Firewall Policy* . Hover over the address in the policy, then in the tooltip, click *View Matched Addresses*.



The firewall policy was automatically updated so that traffic from 10.1.100.184 can no longer pass, and only traffic from 10.1.100.185 can pass.

FortiVoice tag dynamic address

When a FortiVoice-supplied MAC or IP address is used in a firewall policy, a FortiVoice tag (MAC/IP) dynamic address is automatically created on the FortiGate that contains all the provisioned FortiFones registered with FortiVoice. The dynamic address can be used in firewall policies to restrict rules to authorized FortiFones only. This is useful for large voice deployments that require security and efficiency. See Example of a firewall policy on page 1494.

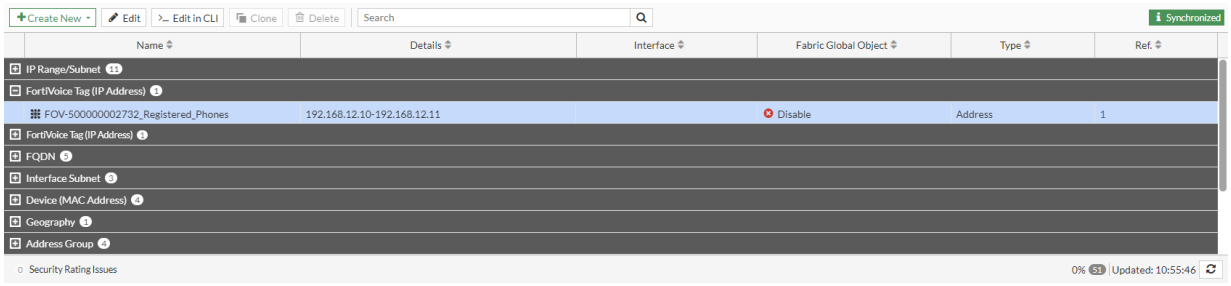
FortiVoice tag dynamic addresses can also be applied to a NAC policy. See Example of a NAC policy NEW on page 1495.

Example of a firewall policy

In this example, two FortiFones are registered to FortiVoice and are assigned names and extension numbers. A FortiVoice Fabric connector has been authorized to join the Security Fabric. The dynamic FortiVoice tags are applied to a firewall policy.

To use a FortiVoice tag dynamic firewall address in a policy:

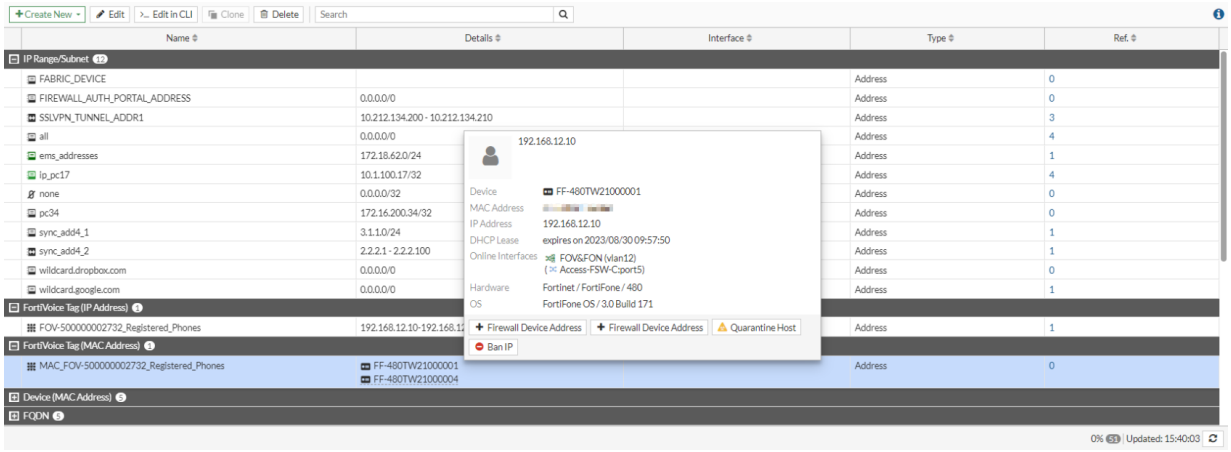
1. Configure and authorize the FortiVoice Fabric connector (see Configuring FortiVoice on page 3225 for more information).
2. Go to *Policy & Objects* > *Addresses* to view the newly created dynamic firewall address objects:
   a. Expand the *FortiVoice Tag (IP Address)* section.



   There is one entry, *FOV-500000002732_Registered_Phones*, which matches 192.168.12.10 to 192.168.12.11.

   b. Expand the *FortiVoice Tag (MAC Address)* section. There is one entry, *MAC_FOV-500000002732_Registered_Phones*, which matches two devices. Hover over the device serial number to view the tooltip that

contains the MAC address and additional information.



3. Go to *Policy & Objects* > *Firewall Policy* and click *Create new* or edit an existing policy.
4. In the *Source* field, click the + and add the *FOV-500000002732_Registered_Phones* and *MAC_FOV-500000002732_Registered_Phones* addresses.
5. In the *Destination* field, click the + and add the *FOV-500000002732_Registered_Phones* address.
6. Configure the other settings as needed.
7. Click *OK*.

Example of a NAC policy - NEW

In this example, a dynamic FortiVoice tag MAC address (MAC_FOV-500000003139_Registered_Phones) is applied to a NAC policy on the FortiGate. Subsequently, the connected FortiSwitch port is moved to vlan12, where traffic can be controlled for registered FortiFones. For more information about NAC policies, see Defining a FortiSwitch NAC policy in the FortiLink Administration Guide. This example assumes that the FortiVoice Fabric connector is authorized to join the Security Fabric and vlan12 is already configured. See Configuring FortiVoice for more information.

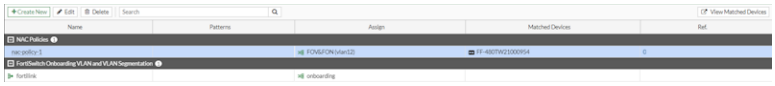
To configure FortiVoice Tag MAC address on NAC policies:

1. Configure the NAC policy:
   a. Go to *WiFi & Switch Controller* > *NAC Policies* and click *Create New*, or edit an existing policy.
   b. In the *Device Patterns* section:
      - Set *Category* to *FortiVoice tag*.
      - Set *FortiVoice tag* to *MAC_FOV-500000003139_Registered_Phones*.
   c. In the *Switch Controller Action* section, enable *Assign VLAN* and select *vlan12*.
   d. Configure the other settings as needed.
   e. Click *OK*.
2. Enable NAC mode on the desired FortiSwitch ports (port6 in this example):
   a. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*.
   b. Select *port6*, then right-click and set the *Mode* to *NAC*.

3. Configure firewall policy that is used to control outbound internet access for FortiFones (vlan12 to wan1):
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create New*.
   c. Name the policy and configure the following parameters:

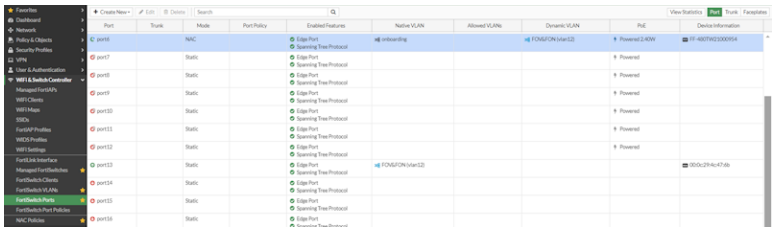
Incoming Interface	vlan12
Outgoing Interface	wan1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   d. Configure the other settings as needed.
   e. Click *OK*.
4. Generate traffic from the FortiFone.
5. Once the NAC policy is matched, go to *WiFi & Switch Controller* > *NAC Policies* to view the device matched to the policy.



FortiFone is also shown on *Dashboards* > *Assets & Identities* in the *Matched NAC Devices* widget.

6. Go to *WiFi & Switch Controller* > *FortiSwitch Ports* and locate the port that the FortiFone is connected to. The port has been dynamically assigned vlan12.



To configure FortiVoice Tag MAC address on NAC policies in the CLI:

1. Configure the NAC policy:

```
config user nac-policy
    edit "nac-policy-1"
        set category fortivoice-tag
        set fortivoice-tag "MAC_FOV-500000003139_Registered_Phones"
        set switch-fortilink "fortilink"
        set switch-mac-policy "mac-policy-
```

```
    next
end
```

2. Configure the VLAN in the MAC policy:

```
config switch-controller mac-policy
    edit "mac-policy-1"
        set fortilink "fortilink"
        set vlan "vlan12"
    next
end
```

3. Enable NAC mode on the desired FortiSwitch ports:

```
config switch-controller managed-switch
    edit "Access-FSW-C"
        config ports
            edit "port6"
                set access-mode nac
            next
        end
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "c_fov_fon"
        set srcintf "vlan12"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

MAC addressed-based policies

MAC addresses can be added to the following IPv4 policies:

- Firewall
- Virtual wire pair
- ACL
- Central SNAT
- DoS

A MAC address is a link layer-based address type and it cannot be forwarded across different IP segments. In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range.

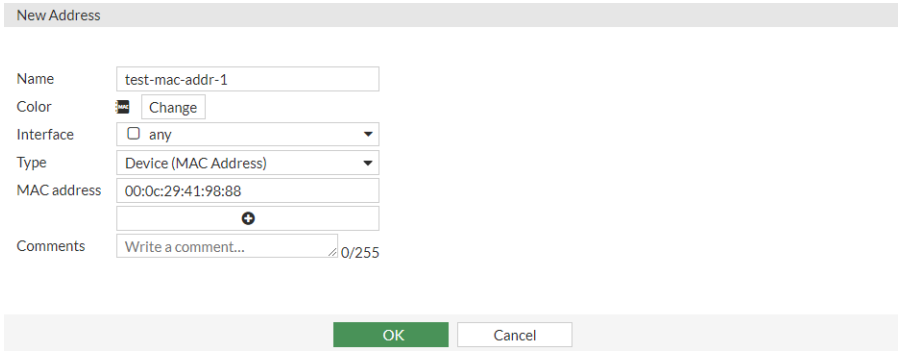
FortiOS only supports the MAC address type as source address for policies in NAT mode VDOM. When you use the MAC address type in a policy as source address in NAT mode VDOM, IP address translation (NAT) is still performed

according to the rules defined in the policy. The MAC address type only works for source address matching. It does not have any association with NAT actions.

For policies in transparent mode or the virtual wire pair interface, you can use the MAC address type as source or destination address.

To configure a MAC address using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter a name.
4. For *Category*, select *Address*.
5. For *Type*, select *Device (MAC Address)*.
6. Enter the MAC address.



7. Click *OK*.
8. Go to *Policy & Objects* > *Firewall Policy* to apply the address type to a policy in NAT mode VDOM:
   a. For *Source*, select the MAC address you just configured.
   b. For *Destination*, select an address.

   In NAT mode VDOM, this address type cannot be used as destination address.

   c. Configure the other settings as needed.
   d. Click *OK*.

To configure a MAC address using the CLI:

1. Create a new MAC address:

```
config firewall address
    edit "test-mac-addr1"
        set type mac
        set macaddr 00:0c:29:41:98:88
    next
end
```

2. Apply the address type to a policy. In transparent mode or the virtual wire pair interface, this address type can be mixed with other address types in the policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "test-mac-addr1" "10-1-100-42"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

Device & OS Identification dynamic address subtype

Another type of MAC address object is the dynamic *Device & OS identification* subtype. This firewall address subtype is an advanced feature that can be used in policies that support dynamic address subtypes, and it relies on device detection configured on the interface connected to user devices to determine device information.

The FortiGate will update the dynamic address used in firewall policies based on the MAC address and other device and OS information for devices matching configured criteria. The criteria could be hardware vendor, hardware model, software OS, software version, or a combination of these parameters.

Only existing devices whose device information has already been detected by the FortiGate and is known can be added to this dynamic address subtype.

Similar to MAC address-based objects, the dynamic address subtype can be used as a source address for firewall policies, proxy policies, and ZTNA rules. The dynamic address subtype can be used as a source or destination address for transparent mode policies or a virtual wire pair policy.

To use the dynamic *Device & OS Identification* subtype, go to *System* > *Feature Visibility* and enable *Dynamic Device & OS Identification*. Once enabled, the dynamic address subtype can be configured on the *Policy & Objects* > *Addresses* page.

ISDB well-known MAC address list

The Internet Service Database (ISDB) includes well-known vendor MAC address range lists. The lists can only be used for source MAC addresses in IPv4 policies, and include the vendor name and the MAC address ranges that the vendor belongs to.

To view the vendor list:

```
# diagnose vendor-mac id
Please input Vendor MAC ID.
ID: 1 name: "Asus"
ID: 2 name: "Acer"
ID: 3 name: "Amazon"
ID: 4 name: "Apple"
ID: 5 name: "Xiaomi"
```

```
ID: 6 name: "BlackBerry"
ID: 7 name: "Canon"
ID: 8 name: "Cisco"
ID: 9 name: "Linksys"
ID: 10 name: "D-Link"
ID: 11 name: "Dell"
ID: 12 name: "Ericsson"
ID: 13 name: "LG"
ID: 14 name: "Fujitsu"
ID: 15 name: "Fitbit"
ID: 16 name: "Fortinet"
ID: 17 name: "OPPO"
ID: 18 name: "Hitachi"
ID: 19 name: "HTC"
ID: 20 name: "Huawei"
ID: 21 name: "HP"
ID: 22 name: "IBM"
ID: 23 name: "Juniper"
ID: 24 name: "Lenovo"
ID: 25 name: "Microsoft"
ID: 26 name: "Motorola"
ID: 27 name: "Netgear"
ID: 28 name: "Nokia"
ID: 29 name: "Nintendo"
ID: 30 name: "PaloAltoNetworks"
ID: 31 name: "Polycom"
ID: 32 name: "Samsung"
ID: 33 name: "Sharp"
ID: 34 name: "Sony"
ID: 35 name: "Toshiba"
ID: 36 name: "VMware"
ID: 37 name: "Vivo"
ID: 38 name: "Zyxel"
ID: 39 name: "ZTE"
```

To view the MAC address ranges for a vendor:

```
# diagnose vendor-mac id 16
Vendor MAC: 16(Fortinet)
Version: 0000700021
Timestamp: 201908081432
Number of MAC ranges: 6
00:09:0f:00:00:00 - 00:09:0f:ff:ff:ff
04:d5:90:00:00:00 - 04:d5:90:ff:ff:ff
08:5b:0e:00:00:00 - 08:5b:0e:ff:ff:ff
70:4c:a5:00:00:00 - 70:4c:a5:ff:ff:ff
90:6c:ac:00:00:00 - 90:6c:ac:ff:ff:ff
e8:1c:ba:00:00:00 - e8:1c:ba:ff:ff:ff
```

To query the vendor of a specific MAC address or range:

```
# diagnose vendor-mac match 00:09:0f:ff:ff:ff 48
Vendor MAC: 16(Fortinet), matched num: 1
```

To use the vendor ID in a firewall policy:

```
config firewall policy
    edit 9
        set name "policy_id_9"
        set uuid 6150cf30-308d-51e9-a7a3-bcbd05d61f93
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set vendor-mac 36 16
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Only packets whose source MAC address belong to Fortinet or VMware are passed by the policy.

IPv6 MAC addresses and usage in firewall policies

Users can define IPv6 MAC addresses that can be applied to the following policies:

- Firewall
- Virtual wire pair
- ACL/DoS
- Central NAT
- NAT64
- Local-in

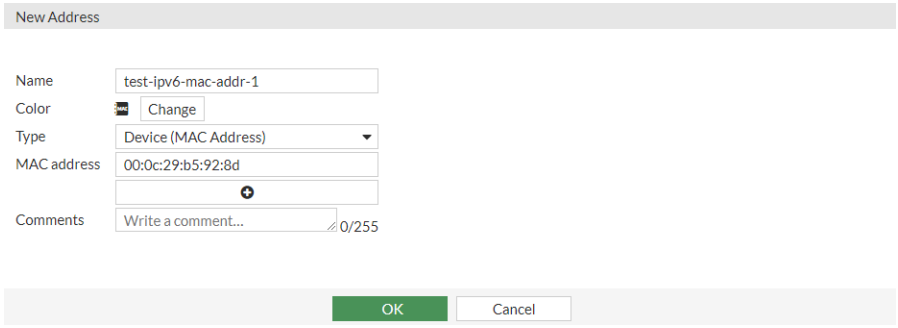
In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range. In this example, a firewall policy is configured in a NAT mode VDOM with the IPv6 MAC address as a source address.

IPv6 MAC addresses cannot be used as destination addresses in VDOMs when in NAT operation mode.

To configure IPv6 MAC addresses in a policy in the GUI:

1. Create the MAC address:
   a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
   b. Click *Create New*.
   c. Enter an address name.
   d. For *Type*, select *Device (MAC Address)*.

e. Enter the the MAC address.



f. Click *OK*.

2. Configure the policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Source*, select the IPv6 MAC address object.
   c. Configure the other settings as needed.
   d. Click *OK*.

To configure IPv6 MAC addresses in a policy in the CLI:

1. Create the MAC address:

```
config firewall address6
    edit "test-ipv6-mac-addr-1"
        set type mac
        set macaddr 00:0c:29:b5:92:8d
    next
end
```

2. Configure the policy:

```
config firewall policy
    edit 2
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "test-ipv6-mac-addr-1" "2000-10-1-100-0"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Protocol options

Firewall policies contain a *Protocol Options* field that defines the parameters for handling protocol-specific traffic. Multiple protocol options profiles can be configured in FortiOS since the requirements may differ between policies. A single protocol options profile is applied per policy, but the profile can be used in multiple policies.

To create a protocol options profile, go to *Policy & Objects* > *Protocol Options*. The following settings can be configured.

Log oversized files

Enable this option to log the occurrence of oversized files being processed. This does not change how they are processed. It only allows the FortiGate to log that they were either blocked or allowed through.

It is common practice to allow larger files through without antivirus processing. Monitor the logs for the frequency of oversized file processing to determine whether or not to alter the settings for treating oversized files. The threshold setting for oversized files and emails is located in the *Common Options* section.

RPC over HTTP

This protocol is used by Microsoft Exchange Servers to perform virus scanning on emails that use RPC over HTTP.

Protocol port mapping

To optimize the FortiGate's resources, the mapping and inspection of the following protocols can be enabled or disabled:

- HTTP
- SMTP
- POP3
- IMAP
- FTP
- NNTP
- MAPI
- DNS
- CIFS

Each protocol has a default TCP port. The ports can be modified to inspect any port with flowing traffic. The packet headers indicate which protocol generated the packet.

Protocol port mapping only works with proxy-based inspection. Flow-based inspection inspects all ports regardless of the protocol port mapping configuration.

Common options

The *Comfort Clients* and *Block Oversized File/Email* options apply to multiple protocols.

Comfort clients

When proxy-based antivirus scanning is enabled, the FortiGate buffers files as they are downloaded. Once the entire file is captured, the FortiGate begins scanning the file. The user must wait during the buffering and scanning procedure.

After the scan is completed and if no infection is found, the file is sent to the next step in the process flow. If the file is large, this part of the process can take some time. In some cases, enough time that some users may get impatient and cancel the download.

The *Comfort Clients* option mitigates this potential issue by feeding a trickle of data while waiting for the scan to complete. The user is aware that processing is taking place, and that there has not been a failure in the transmission. The slow transfer rate continues until the antivirus scan is complete. The transfer will proceed at full speed once the file is scanned successfully and does not contain any viruses.

If there is evidence of an infection, the FortiGate caches the URL and drops the connection. The client does not receive any notification of what happened because the download to the client has already started. Instead, the download stops and the user is left with a partially downloaded file. If the user tries to download the same file again within a short period of time, the cached URL is matched and the download is blocked. A notification is displayed that the download was blocked. The number of URLs in the cache is limited by the size of the cache.

Client comforting is available for HTTP and FTP traffic. If the FortiGate supports SSL content scanning and inspection, client comforting can be configured for HTTPS and FTPS traffic.

Buffering the entire file allows the FortiGate to eliminate the danger of missing an infection due to fragmentation because the file is reassembled before examination. This buffering is performed whenever the *Comfort Clients* option is disabled.

Client comforting can send unscanned and potentially infected content to the client, so only enable this option if you are prepared to accept this risk. Keeping the client comforting interval high and the amount low will reduce the amount of potentially infected data that is downloaded.

Block oversized files and emails

This option is related to antivirus scanning. The FortiGate has a finite amount of resources to buffer and scan a file. If a large file (such as an ISO image or video file) is downloaded, this could overwhelm or exceed the FortiGate's memory, especially if other large files are being downloaded at the same time.

A threshold is assigned to identify an oversize file or email. The default is 10 MB. The range varies per model, and the minimum is 1 MB. Any file or email over this threshold will not be processed by policies applying the antivirus security profile.

If the FortiGate enters conserve mode on a regular basis, lowering the threshold can lessen the impact of processing the files on memory. This can increase risk, even though malware is more likely to be in smaller files.

Web options

The *Chunked Bypass* option applies to traffic containing web protocols.

Chunked bypass

Chunked bypass is a mechanism in HTTP 1.1 that allows a web server to start sending chunks of dynamically generated output in response to a request before actually knowing the actual size of the content. For dynamically generated

content, enabling chunked bypass speeds up the initial response to HTTP requests, but the content is not held in the proxy as an entire file before proceeding.

Email options

The *Allow Fragmented Messages* and *Append Signature (SMTP)* options apply to email protocols.

Allow fragmented messages

The specifications of RFC 2046 allow for the breaking up of emails and sending the fragments in parallel to be rebuilt and read at the other end by the mail server. It was originally designed to increase the performance over slower connections where larger email messages were involved. Feasibility of using this function depends on the mail configuration. Outside of Microsoft Outlook, not many email clients are set up to break up messages like this. The drawback of this feature is that if malware is broken up between multiple fragments of the message, there is a risk that it will not be detected by some antivirus configurations because all the code may not be present at the same time to identify the malware.

Append signature

This option adds a plain text email signature to SMTP email messages as they pass through the FortiGate. The message maximum is 1023 characters.

This feature works best in an environment where there is some standardization of what goes into the senders' personal signatures so that there is no duplication or contradiction of information. For example:

- *This email should not be forwarded without prior approval.*
- *Please consider the environment before printing this email.*
- *For questions regarding purchasing our products, please call ...*

Stripping the X-Forwarded-For value in the HTTP header

The X-Forwarded-For value in the HTTP header can be stripped when the `strip-x-forwarded-for` option is enabled under `firewall profile-protocol-options`. This feature sets the value to empty using the IPS engine.

The following types of traffic support X-Forwarded-For stripping:

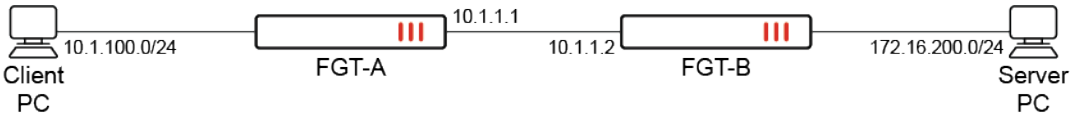
- HTTP/1.1, HTTP/2, and HTTP/3 traffic that matches an NGFW mode security policy with flow-based inspection.
- Plain HTTP/1.1 traffic that matches a firewall policy with proxy-based inspection.

The following types of traffic do not support X-Forwarded-For stripping:

- HTTPS traffic that matches a firewall policy with proxy-based inspection.
- HTTP and HTTPS traffic that matches an explicit web proxy policy.

Example

In this example, FGT-A is configured with `strip-x-forwarded-for` enabled for HTTP. On FGT-B, the IPS sensor is configured to monitor the Eicar.Virus.Test.File signature. The IPS logs on FGT-B are used to verify the traffic sent from FGT-A to FGT-B, namely the `forwardedfor` value in the `rawdata` field.



To configure X-Forwarded-For stripping:

1. Configure FGT-A:
   a. Configure the protocol options for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set ports 80
            unset options
            set strip-x-forwarded-for enable
            unset post-lang
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port5"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol-xff"
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "default"
        set nat enable
    next
end
```

2. Configure FGT-B:
   a. Configure the IPS sensor with extended logging:

```
config ips sensor
    edit "monitor-eicar"
        set extended-log enable
        config entries
            edit 1
                set rule 29844
                set status enable
                set action pass
            next
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 3
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "monitor-eicar"
        set nat enable
    next
end
```

To verify the configuration:

1. Use a cURL request to send HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC:

   ```
   curl -vk -H "X-Forwarded-For: 10.22.22.22" https://172.16.200.52/eicar.com
   ```

2. On FGT-B, verify the corresponding IPS logs.
   a. For HTTP/1.1, the `X-Forwarded-For` value is removed from the `rawdata` field, and the `forwardedfor` value is not included:

      ```
      1: date=2023-09-21 time=14:05:34 eventtime=1695330334919589600 logid="0419016384"
      type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
      severity="info" srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.42
      dstcountry="Reserved" srcintf="port5" srcintfrole="undefined" dstintf="port1"
      dstintfrole="undefined" sessionid=2471 action="detected" proto=6 service="HTTPS"
      policyid=3 poluuid="782b9e86-58a3-51ee-8e0f-79c7682223dd" policytype="policy"
      attack="Eicar.Virus.Test.File" srcport=36018 dstport=443 hostname="172.16.200.42"
      url="/eicar.com" agent="curl/7.61.1" httpmethod="GET" direction="incoming"
      attackid=29844 profile="monitor-eicar" ref="http://www.fortinet.com/ids/VID29844"
      incidentserialno=75497475 msg="file_transfer: Eicar.Virus.Test.File" rawdataid="1/1"
      rawdata="Response-Content-Type=application/x-msdos-program" crscore=5 craction=65536
      crlevel="low"
      ```

   b. For HTTP/2 and HTTP/3, the `X-Forwarded-For` value is removed from the `rawdata` field, and `forwardedfor` is included:

      ```
      1: date=2023-09-21 time=14:05:56 eventtime=1695330356543624871 logid="0419016384"
      type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
      severity="info" srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.52
      dstcountry="Reserved" srcintf="port5" srcintfrole="undefined" dstintf="port1"
      dstintfrole="undefined" sessionid=2474 action="detected" proto=6 service="HTTPS"
      policyid=3 poluuid="782b9e86-58a3-51ee-8e0f-79c7682223dd" policytype="policy"
      attack="Eicar.Virus.Test.File" srcport=37786 dstport=443 hostname="172.16.200.52"
      url="/eicar.com" agent="curl/7.61.1" httpmethod="GET" direction="incoming"
      attackid=29844 profile="monitor-eicar" ref="http://www.fortinet.com/ids/VID29844"
      incidentserialno=75497476 msg="file_transfer: Eicar.Virus.Test.File" rawdataid="1/1"
      forwardedfor="\r\n" rawdata="Response-Content-Type=application/x-msdos-program"
      crscore=5 craction=65536 crlevel="low"
      ```

3. On FGT-A, disable `strip-x-forwarded-for` for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set strip-x-forwarded-for disable
        end
    next
end
```

4. Send the same HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC.
5. On FGT-B, verify the corresponding IPS log, which includes `forwardedfor` and `X-Forwarded-For` values in the `rawdata` field:

```
1: date=2023-09-21 time=16:33:06 eventtime=1695339187144132034 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.52 dstcountry="Reserved"
srcintf="port5" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
sessionid=3776 action="detected" proto=6 service="HTTPS" policyid=3 poluuid="782b9e86-
58a3-51ee-8e0f-79c7682223dd" policytype="policy" attack="Eicar.Virus.Test.File"
srcport=37788 dstport=443 hostname="172.16.200.52" url="/eicar.com" agent="curl/7.61.1"
httpmethod="GET" direction="incoming" attackid=29844 profile="monitor-eicar"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=75497478 msg="file_transfer:
Eicar.Virus.Test.File" rawdataid="1/1" forwardedfor="10.22.22.22" rawdata="Response-
Content-Type=application/x-msdos-program|X-Forwarded-For=10.22.22.22" crscore=5
craction=65536 crlevel="low"
```

Traffic shaping

A FortiGate provides quality of service (QoS) by applying bandwidth limits and prioritization to network traffic. Traffic shaping is one technique used by the FortiGate to provide QoS. A basic approach to traffic shaping is to prioritize higher priority traffic over lower priority traffic during periods of traffic congestion. This provides a stabilizing effect for important traffic while throttling less important traffic.

The FortiGate can be configured to deliver traffic shaping with policing or traffic shaping with queuing. The general difference between the two is as follows:

Technique	Description
Traffic shaping with policing	When traffic exceeds the configured bandwidth limits, traffic is dropped.
Traffic shaping with queuing	When traffic exceeds the configured bandwidth limits, traffic is delayed for transport until bandwidth frees up. Traffic may be dropped if the queues are full.

Policing and queuing can both prioritize traffic and deliver guaranteed bandwidth and maximum bandwidth by setting bandwidth limits. The implementation differs though, since queuing uses queues, and policing does not. In queuing, before a packet egresses an interface, it is first enqueued to a queue using an algorithm such as RED or FIFO. The kernel dequeues the packet based on the HTB algorithm before sending it out. In policing, traffic simply drops if it is over the allocated bandwidth.

The following topics provide information about configuring traffic shaping:

- Traffic shaping policies on page 1510
- Traffic shaping profiles on page 1520
- Traffic shapers on page 1530

Configuration methods

There are different methods to configure traffic shaping on the FortiGate. The following table lists the methods and their capabilities in order of preference. If all three methods are configured, the first will be preferred over the second, which is preferred over the third.

Method	Policing		Queuing
	Traffic prioritization	Guaranteed and maximum bandwidth limits	Traffic queuing
Traffic shaping profile*	Yes	Yes, based on percentage of outbandwidth	Yes
Traffic shaper	Yes	Yes, based on rate	No
Global traffic prioritization	Yes	No	No

* Traffic shaping profiles are configured as either policing or queuing types. Queuing allows for additional options when configuring a shaping class entry.

The features of each method's implementation are slightly different. The following is a brief summary of the traffic policing features and the approach each method takes.

Traffic prioritization

The FortiGate can place packets into different priority levels in order to prioritize certain traffic over others.

Method	Description
Traffic shaping profile	Traffic is placed into classes. A total of 30 classes are available. For each class, traffic can be configured into five priority levels.
Traffic shaper	Traffic can be prioritized into the high (2), medium (3), or low (4) levels. When traffic is below the guaranteed bandwidth of the shaper, the traffic is automatically applied the critical level (1).
Global traffic prioritization	Traffic is prioritized into high (2), medium (3), or low (4) based on ToS (type of service) or DSCP.

Guaranteed and maximum bandwidth limits

The general purpose for configuring guaranteed bandwidth is to allocate a certain proportion of the total outbandwidth to guarantee transport for a certain type of traffic. This is configured and handled differently in each method.

A traffic shaping profile, when applied to an interface's egress shaping profile, can be configured to use up to 100% of the interface's configured bandwidth between all the classes. It does not matter what priority is configured in each class. The guaranteed bandwidth is always honored.

Traffic shapers, however, do not have a hard limit on the guaranteed bandwidth. Administrators need to be aware how much guaranteed bandwidth has been allocated to all their traffic shapers, so that they do not exceed the total outbandwidth of an interface. Traffic under the guaranteed bandwidth of a traffic shaper is given a priority of one. If the total traffic with priority one exceeds the total outbandwidth, traffic can be dropped.

The maximum bandwidth limit caps the maximum bandwidth that can be used. This is configured as a percentage of the outbandwidth in a traffic shaping profile. It is configured as a rate for traffic shapers.

Configuring outbandwidth

Traffic shaping is generally configured for egress traffic leaving the FortiGate. Therefore, it is necessary for the interface outbandwidth to be defined for traffic prioritization to take place in all of the traffic shaping configuration methods. Interface outbandwidth is also needed when defining the guaranteed and maximum bandwidth in a traffic shaping profile.

For traffic shapers, configuring outbandwidth is not necessary to apply maximum bandwidth limits; however, outbandwidth is necessary for guaranteed bandwidth. Traffic under the guaranteed bandwidth limit on a traffic shaper is given priority 1. If outbandwidth is not configured, traffic prioritization does not take place and the priority is meaningless.

Traffic shaping policy

Traffic shaping profiles and traffic shapers are methods of policing traffic. Traffic shaping policies are used to map traffic to a traffic shaper or assign them to a class.

A traffic shaping policy is a rule that matches traffic based on certain IP header fields and/or upper layer criteria. For example, it can match traffic based on source and destination IP, service, application, and URL category. One common use case is to match traffic based on the ToS or DS (differentiated services) field in the IP header. This allows Type of Service or Differentiated Services (DiffServ) tags to be read from traffic from a downstream device and prioritized accordingly on the FortiGate.

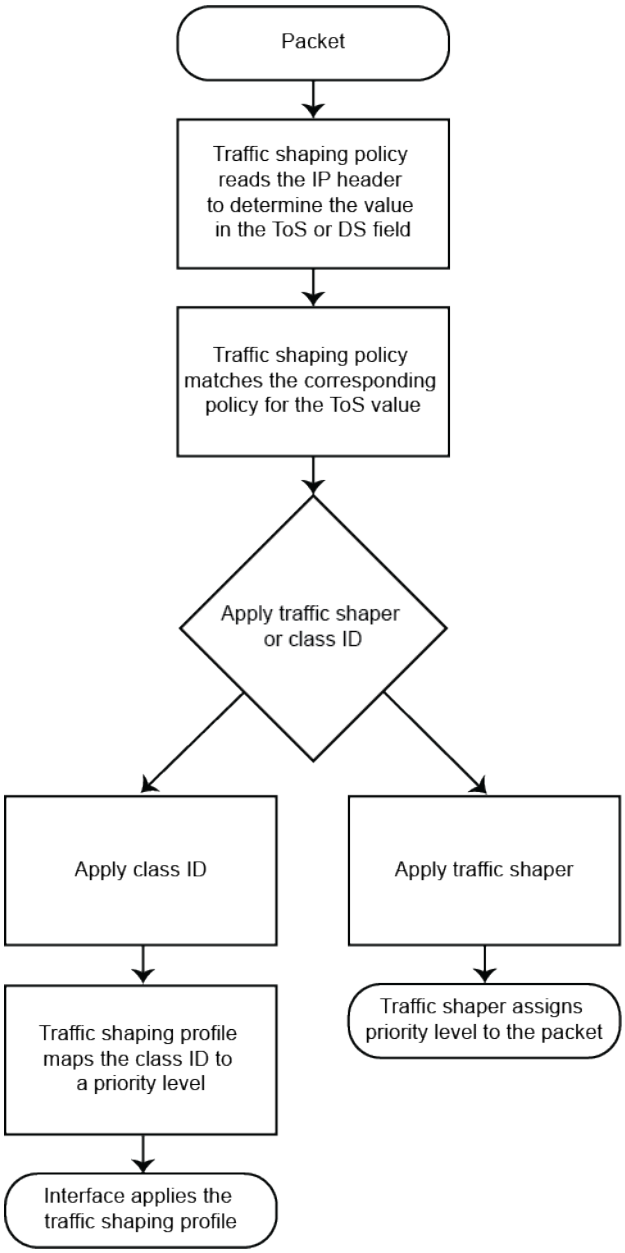
DSCP matching and DSCP marking

DSCP matching and DSCP marking can be performed on a firewall shaping policy and a regular firewall policy. DSCP matching is used to match DSCP tags from ingress traffic, and DSCP marking is used to change the DSCP tag on egress traffic.

In a firewall shaping policy and regular firewall policy, use the `tos` and `tos-mask` fields to perform DSCP matching. Use the `diffserv-forward` and `diffserv-reverse` fields to perform DSCP marking.

Traffic shaping policies

As mentioned in Traffic shaping on page 1508, traffic shaping starts with the traffic shaping policy. Traffic shaping policies are used to map traffic to a traffic shaper or assign them to a class. Traffic is then shaped by the shaper or the shaping profile that is applied on an interface.



Traffic can also be shaped by applying traffic shapers directly on a firewall policy. However, this legacy approach can only be configured from the CLI, and is not a preferred method for applying traffic shaping. As the number of firewall policies increases, managing shaping on each individual policy becomes increasingly difficult. For the same reason, it is also not recommended to mix the legacy approach with traffic shaping policies to avoid the added complexity.

Overview

A traffic shaping policy is a rule that matches traffic based on certain IP header fields and/or upper layer criteria. When traffic hits the firewall, the FortiGate will first look up a firewall policy, and then match a shaping policy. The matching traffic will apply a traffic shaper, class ID, or assign a DSCP DiffServ tag to the outgoing traffic.

The traffic shaping policies must be placed in the correct order in the traffic shaping policy list page to obtain the desired results. Policies are matched from top-down, so the traffic shaping policies should be arranged in a sequence that places the more granular policies above general policies.

The policy can be configured by going to *Policy & Objects* > *Traffic Shaping* and selecting the *Traffic Shaping Policies* tab. If the menu does not display the traffic shaping settings, go to *System* > *Feature Visibility* and enable *Traffic Shaping*.

Configuring traffic shaping policies

A traffic shaping policy can be split into two parts:

- Options used to match the traffic
- Options used to apply actions to the matched traffic

In the GUI, the options are configured in the *If Traffic Matches* and *Then* sections. In the CLI, all options are configured under `config firewall shaping-policy`. Some options can only be configured from the CLI.

The following options can be configured for traffic matching criteria:

GUI option	CLI option	Description
Source		
Address	`set srcaddr <address_object>`	Select the address object to match the source IP.
User	`set users <user_object>`	Select the user object to match the user authenticated for the session.

Internet Service	`set internet-service-src enable`
	`set internet-service-src-name <name>`
	`set internet-service-src-group <group>`
	`set internet-service-src-custom <custom>`
	`set internet-service-src-custom-group <custom_group>`
		Select the internet service to match the source of the incoming traffic. Internet service currently cannot be used with source address.

Destination		
Address	`set dstaddr <address_object>`	Select the address object to match the destination IP.

Internet Service	`set internet-service enable`
	`set internet-service-name <name>`
	`set internet-service-group <group>`
	`set internet-service-custom <custom>`
		Select the internet service to match the destination of the incoming traffic. Internet service currently cannot be used with destination address and service.

GUI option	CLI option	Description
	`set internet-service-custom-group <custom_group>`	
Schedule	`set schedule <schedule>`	Enable to select a schedule (one-time, recurring, or group).
Service	`set service <service>`	Select the service or service group for the traffic.
Application		Application control must be enabled in the related firewall policy to learn the application of the traffic.
Application	`set application <application>`	Select the application to match the application of the traffic.
Category	`set app-category <category>`	Select the application category to match the application of the traffic.
Group	`set app-group <groups>`	Select the application group to match the application of the traffic.
URL Category	`set url-category <category>`	Select the URL category to match the URL of the traffic. A web filter profile must be enabled in the related firewall policy to know the URL of the traffic (see Web filter on page 1659).

n/a	`set tos-mask <hexadecimal_mask>`
	`set tos <value>`
	`set tos-negate {enable | disable}`
		Specify the type of service (ToS) and mask to match.
		These options can only be configured in the CLI.

The following options can be configured for actions to apply to the matched traffic:

GUI option	CLI option	Description
Outgoing interface	`set dstintf <interface>`	Select the destination interface that the traffic shaping applies to (required).
Apply shaper		
Shared shaper	`set traffic-shaper <shaper>`	Select the shared shaper to be applied to traffic in the ingress-to-egress direction. For example, on traffic that egresses on the wan interface, the shaper is applied to upload or outbound traffic.

GUI option	CLI option	Description
Reverse shaper	`set traffic-shaper-reverse <shaper>`	Select the reverse shaper to be applied to traffic in the egress-to-ingress direction. For example, on traffic that egresses on the wan interface, the shaper is applied to download or inbound traffic.
Per-IP shaper	`set per-ip-shaper <shaper>`	Select the per-IP shaper. Per-IP shapers affect downloads and uploads. The allotted bandwidth applies to each individual IP. In a shared shaper, the allotted bandwidth applies to all IPs.
Assign shaping class ID		
Traffic shaping class ID	`set class-id <class>`	Set the class ID to apply the matching traffic. Class IDs are further prioritized within a traffic shaping profile and applied to an interface.

n/a	`set diffserv-forward {enable | disable}`
	`set diffservcode-forward <code>`
	`set diffserv-reverse {enable | disable}`
	`set diffservcode-reverse <code>`
		Specify the settings to apply a DSCP tag to the forward or reverse traffic. The DiffServ code is in 6-bit binary format.
		These options can only be configured in the CLI.

Traffic shapers and class IDs can be applied at the same time when configuring traffic shaping policies. However, to reduce the complexity, it is recommended to use one method over the other.

The following topics include examples with traffic shaping policies:

- Local-in and local-out traffic matching on page 1514
- VLAN CoS matching on a traffic shaping policy on page 1517
- Interface-based traffic shaping profile on page 1556
- Shared traffic shaper on page 1530
- Per-IP traffic shaper on page 1535

Local-in and local-out traffic matching

A FortiGate can apply shaping policies to local traffic entering or leaving the firewall interface based on source and destination IP addresses, ports, protocols, and applications.

```
config firewall shaping-policy
    edit <id>
        set traffic-type {forwarding | local-in | local-out}
    next
end
```

This topic contains the following examples:

- Example 1: local-in traffic shaping
- Example 2: local-out traffic shaping

Example 1: local-in traffic shaping

In this example, the traffic shaping policy applies to local-in traffic. The local-in traffic originates from the Linux client and is destined to port1 on the FortiGate.

To configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 2
        set traffic-type local-in
        set service "ALL"
        set schedule "always"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-in traffic to verify that the correct class ID (3) is applied:

```
# diagnose firewall iprope list 100018
policy index=2 uuid_idx=1300 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100018 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 3
```

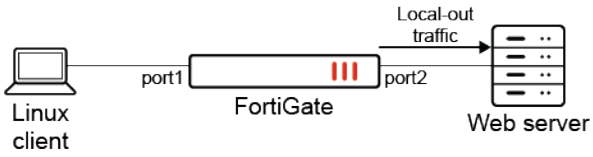
2. Check the session list to verify that the class ID (3) matches the shaping policy ID (2):

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=1195 expire=3574 timeout=3600
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=3 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log local may_dirty
statistic(bytes/packets/allow_err): org=18274/350/1 reply=826037/603/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 1/0
orgin->sink: org pre->in, reply out->post dev=17->34/34->17 gwy=172.16.200.2/0.0.0.0
hook=pre dir=org act=noop 172.16.200.254:55432->172.16.200.2:443(0.0.0.0:0)
```

```
hook=post dir=reply act=noop 172.16.200.2:443->172.16.200.254:55432(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=08:5b:0e:7d:42:db
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
serial=0000009d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

Example 2: local-out traffic shaping

In this example, the traffic shaping policy applies to local-out traffic. The local-out traffic originates from port2 on the FortiGate and is destined to an external web server.



To configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 3
        set traffic-type local-out
        set service "ALL"
        set schedule "always"
        set class-id 2
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-out traffic to verify that the correct class ID (2) is applied:

```
# diagnose firewall iprope list 100019
policy index=3 uuid_idx=1301 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100019 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 2
```

2. Check the session list to verify that the class ID (2) matches the shaping policy ID (3):

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=40 expire=110 timeout=3600 flags=00000000
```

```
    socktype=0 sockport=0 av_idx=0 use=3
    origin-shaper=
    reply-shaper=
    per_ip_shaper=
    class_id=2 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
    state=log local
    statistic(bytes/packets/allow_err): org=3676/14/1 reply=3848/11/1 tuples=2
    tx speed(Bps/kbps): 90/0 rx speed(Bps/kbps): 94/0
    orgin->sink: org out->post, reply pre->in dev=34->17/17->34 gwy=0.0.0.0/172.16.200.2
    hook=out dir=org act=noop 172.16.200.2:19178->140.174.22.68:443(0.0.0.0:0)
    hook=in dir=reply act=noop 140.174.22.68:443->172.16.200.2:19178(0.0.0.0:0)
    pos/(before,after) 0/(0,0), 0/(0,0)
    dst_mac=08:5b:0e:7d:42:db
    misc=0 policy_id=0 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
    serial=00000f1b tos=ff/ff app_list=0 app=0 url_cat=0
    rpdb_link_id=00000000 ngfwid=n/a
    npu_state=00000000
    no_ofld_reason:  local
```

VLAN CoS matching on a traffic shaping policy

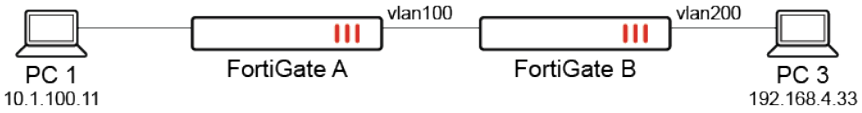
A FortiGate can use the class of service (CoS) value of VLAN packets as a matching criterion for shaping policies. This enables the FortiGate to prioritize traffic based on the CoS value assigned by the switch or router.

```
config firewall shaping-policy
    edit <id>
        set traffic-type {forwarding | local-in | local-out}
        set cos-mask <3-bit_binary>
        set cos <3-bit_binary>
    next
end
```

`traffic-type {forwarding | local-in | local-out}`	Set the traffic type. • `forwarding`: use forwarding traffic (default) • `local-in`: local-in traffic • `local-out`: local-out traffic
`cos-mask <3-bit_binary>`	Set the VLAN CoS evaluated bits, 3-bit binary (000 - 111). This setting is only available for forwarding traffic.
`cos <3-bit_binary>`	Set the VLAN CoS bit pattern, 3-bit binary (000 - 111). This setting is available once `cos-mask` is configured.

Example

In this example, FortiGate A forwards traffic to FortiGate B with VLAN CoS 3, which matches firewall policy 6. When FortiGate B receives traffic, it applies the traffic shaping policy and will prioritize based on the CoS value.



The VLAN CoS range is 000 to 111 (0 - 7), which includes the following values: 000, 001, 010, 011, 100, 101, 110, and 111. The `cos` and `cos-mask` settings can be used to match multiple `vlan_cos` values with a single shaping policy. The following matching logic is used: (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`).

To match all possible `vlan_cos` values, set the `cos-mask` to `000`.

To configure VLAN CoS marking with traffic shaping:

1. Configure the firewall policy on FortiGate A with VLAN CoS forwarding:

```
config firewall policy
    edit 6
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set vlan-cos-fwd 3
    next
end
```

   Traffic marked with CoS 3 will be forwarded to FortiGate B.

2. On FortiGate A, check the session list to verify that CoS 3 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=3/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/10.1.100.11
hook=pre dir=org act=noop 10.1.100.11:28489->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28489->10.1.100.11:0(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=6 pol_uuid_idx=1128 auth_info=0 chk_client_info=0 vd=2
serial=000717ca tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=79/78, ipid=78/79,
vlan=0x0000/0x0064
vlifid=78/79, vtag_in=0x0000/0x0064 in_npu=2/2, out_npu=2/2, fwd_en=0/0, qid=0/1
```

3. Configure the traffic shaping policy to match VLAN CoS 3:

```
config firewall shaping-policy
    edit 1
        set traffic-type forwarding
        set name "vlan-cos-matching"
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set class-id 2
        set cos-mask 111
        set cos 011
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Based on this shaping policy:

- `vlan_cos` = 3, which corresponds to `011`

  `cos-mask` = `111`

  AND both get `011`
- `cos-mask` = `111`

  `cos` = `011`

  AND both get `011`
- (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`), so traffic will pass

The shaping policy will match `vlan_cos3`.

4. Configure the firewall policy on FortiGate B:

```
config firewall policy
    edit 3
        set srcintf "vlan100"
        set dstintf "vlan200"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

5. On FortiGate B, check the session list to verify that the class ID (2) matches the shaping policy ID (1):

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=672 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=56532/673/1 reply=56532/673/1 tuples=2
tx speed(Bps/kbps): 82/0 rx speed(Bps/kbps): 82/0
```

```
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:28735->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28735->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1245 auth_info=0 chk_client_info=0 vd=1
serial=0000160b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
```

If a particular session matches both the `firewall policy` and `firewall shaping-policy`, then anything configured in the `firewall shaping-policy` overrides whatever was configured in the `firewall policy`.

Traffic shaping profiles

As mentioned in Traffic shaping on page 1508, the three main methods of configuring traffic shaping are:

- Traffic shaping profiles
- Traffic shapers
- Global traffic prioritization

A traffic shaping profile allows traffic shaping to be configured with policing or queuing. Up to 30 classes can be defined, with prioritization and bandwidth limits configured for each class. When queuing is enabled, metrics can be configured for traffic queuing in each class.

Traffic shaping with policing

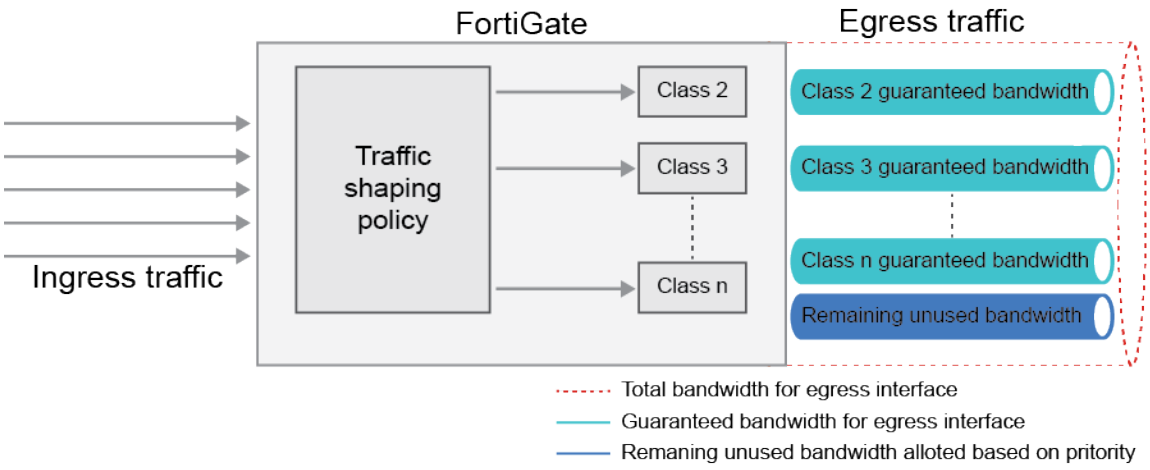
At the most basic level, policing involves traffic prioritization and bandwidth limits. Traffic prioritization helps categorize traffic into different priority levels: low, medium, high, critical, and top. When bandwidth is limited, traffic with higher priority levels will take precedence over lower priority traffic. Traffic with lower priority levels that exceeds available bandwidth will be dropped. These levels are only applicable in the context of traffic shaping profiles and should not be confused with global traffic prioritization levels.

Bandwidth limits define the guaranteed and maximum bandwidth allotted to each traffic class. These limits are configured as a percentage of the outbandwidth, which is the outbound bandwidth configured on an interface.

Guaranteed bandwidth limits guarantee the minimum bandwidth that is allotted to a given class of traffic. The sum of all guaranteed bandwidth of all classes within a traffic shaping profile cannot exceed 100%. However, the sum of all guaranteed bandwidth does not need to add up to 100%. The guaranteed bandwidth is always respected, even if one class has lower priority than another.

Maximum bandwidth limits define the maximum percentage of the outbandwidth that a traffic class can use up. This value often will be 100%, given that when there is no other traffic going through other classes, you would want to fully utilize the bandwidth of the outbound link. Traffic throughput exceeding the maximum bandwidth will be dropped.

The following diagram illustrates ingress traffic and how the FortiGate assigns classes and bandwidth to each class.



When comparing traffic shaping profiles and traffic shapers, it is important to remember that guaranteed and maximum bandwidth in a traffic shaping profile is a percentage of the outbandwidth, while guaranteed and maximum bandwidth in a traffic shaper is a rate (Kbps, Mbps, and so on). As long as the outbandwidth is true to its measurement, the bandwidth usage should not exceed the available bandwidth of a link when using a traffic shaping profile.

Congestion occurs when actual traffic surpasses the outbandwidth limit. At this point, traffic prioritization helps determine which traffic will be prioritized over others. First, the guaranteed bandwidth limit is allocated for each class. The left over bandwidth is allocated to traffic classes based on priority. The traffic classes with the highest priority can use as much of the remaining bandwidth as needed. Then, the remaining bandwidth can be allocated to classes at the next priority level, and so forth.

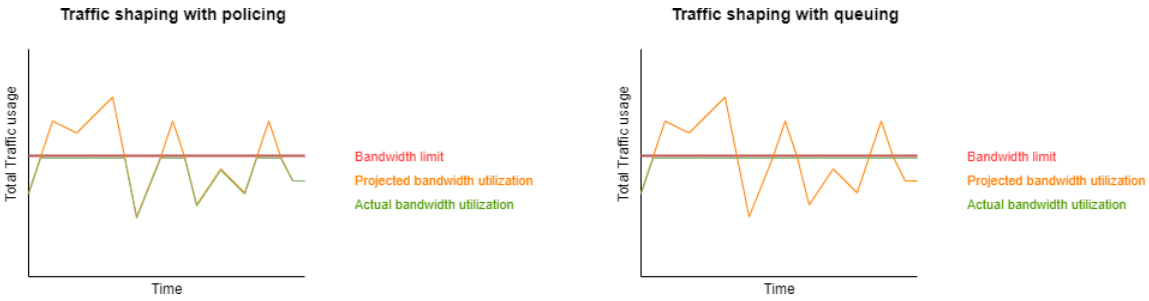
To see examples of applied traffic prioritization and bandwidth limits, see the debugs in Verifying that the traffic is being shaped on page 1524.

Traffic shaping with queuing

When traffic congestion occurs and if there is no queuing, then the excess packets are dropped. With queuing, when traffic exceeds the configured bandwidth limits, the traffic is delayed for transport until bandwidth frees up. Traffic may still be dropped if the queues are full.

In queuing, before a packet egresses an interface, it is first enqueued using an algorithm, such as random early detection (RED) or first in, first out (FIFO). The kernel then dequeues the packet based on the HTB algorithm before sending it out. Queuing can be configured per shaping profile, and it can be customized per class.

The following diagram shows how traffic policing differs from traffic queuing by comparing the bandwidth limit, projected bandwidth utilization, and actual bandwidth utilization.



For more information about traffic shaping with queuing, see Traffic shaping with queuing using a traffic shaping profile on page 1526.

Configuring traffic shaping profiles

The main steps to configure traffic shaping are:

1. Configure the traffic shaping policy, and assign matched traffic to a class (see Traffic shaping policies on page 1510).
2. Configure the traffic shaping profile and apply traffic bandwidth, prioritization and/or queuing per class.
3. Configure the interface outbandwidth and apply an egress shaping profile to the interface.

Configuring the traffic shaping profile

A traffic shaping profile consists of the class ID and the settings per class ID. It also defines the type of traffic shaping to apply (policing or queuing) and the default class ID for traffic that does not match any traffic shaping policies.

A class can be configured in the GUI as part of a traffic shaping profile or policy. In the CLI, a traffic class must be defined before it can be assigned within a traffic shaping profile. Class IDs range from 2 - 31, and they can be reused between different traffic shaping profiles.

When NPU offloading is enabled on the NP6, SoC3, or SoC4 platforms, the class ID limit for egress traffic is 2 - 15. Setting the egress traffic class ID outside of these limits can result in unexpected behavior.

If NPU offloading is disabled, or enabled on the NP7 platform, the class ID limit for egress traffic is 2 - 31.

When configuring a traffic shaping profile, the settings can be defined per class.

The following options can be configured for traffic shaping classes:

GUI option	CLI option	Description
Default	`set default-class-id <class-id>`	Set the default class ID. Each profile must have one default class ID. The default class ID can be changed at any time.
Traffic shaping class ID	`set class-id <integer>`	Set the class ID (2 - 31).
Guaranteed bandwidth	`set guaranteed-bandwidth-percentage <integer>`	Set the percentage of the outbandwidth that will be guaranteed for the class ID.
Maximum bandwidth	`set maximum-bandwidth-percentage <integer>`	Set the percentage of the outbandwidth that will be the maximum bandwidth for the class ID.
Priority	`set priority {top | critical | high | medium | low}`	Select the priority level for the class ID.

To configure a traffic shaping profile in the GUI:

1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Profiles* tab, and click *Create New*.
2. Enter the profile name, and optionally enter a comment.
3. In the *Traffic Shaping Classes* section, click *Create New*.
4. Configure the traffic shaping class ID settings (*Traffic shaping class ID*, *Guaranteed bandwidth*, *Maximum bandwidth*, and *Priority*).
5. Click *OK*.
6. Create more shaping classes as needed (the total guaranteed bandwidth of all classes cannot exceed 100%).
7. Click *OK*.

To configure a traffic shaping profile in the CLI:

1. Configure the shaping class:

```
config firewall traffic-class
    edit <integer>
        set class-name <string>
    next
end
```

2. Configure the shaping profile:

```
config firewall shaping-profile
    edit <name>
        set type {policing | queuing}
        set default-class-id <class-id>
        config shaping-entries
            edit <id>
                set class-id <integer>
                set priority {top | critical | high | medium | low}
                set guaranteed-bandwidth-percentage <integer>
                set maximum-bandwidth-percentage <integer>
            next
        end
    next
end
```

Configuring the interface outbandwidth

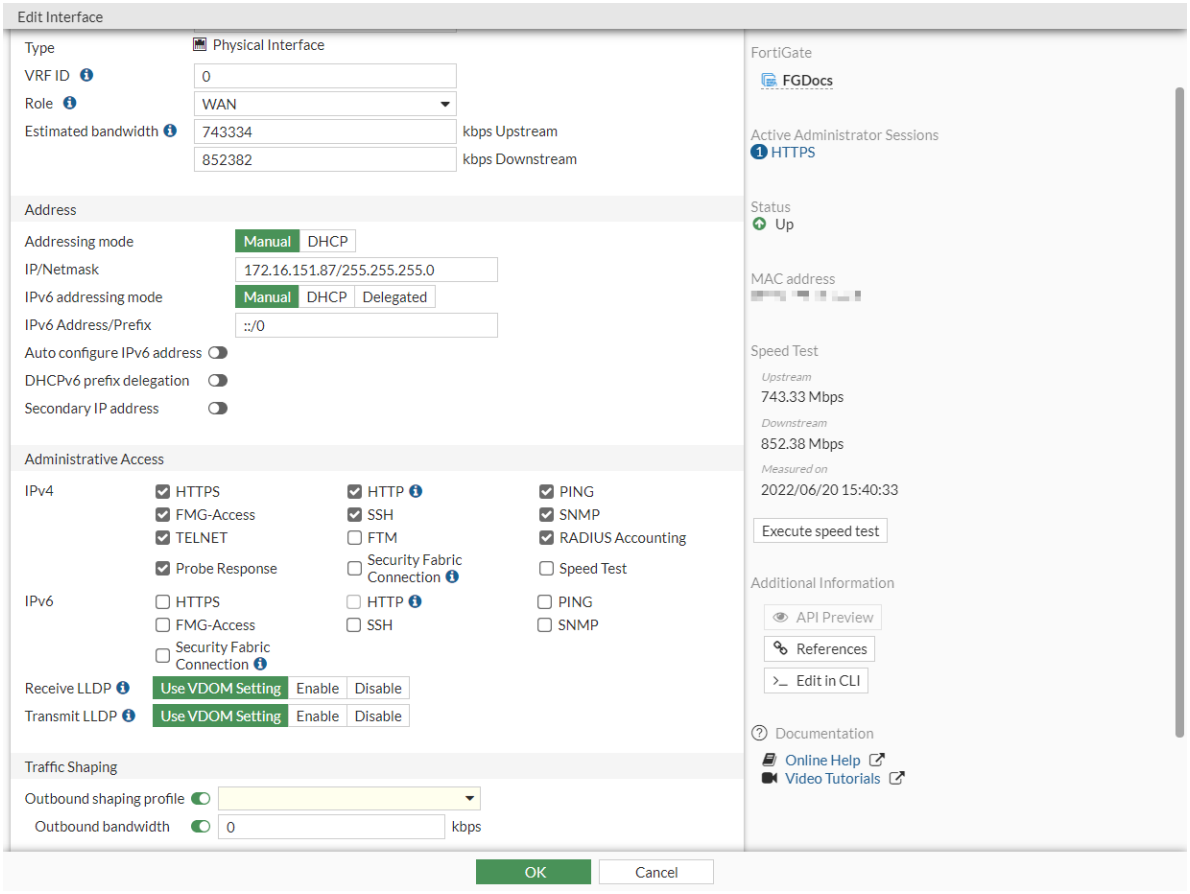
There are two settings that must be configured on an interface that has traffic shaping applied to egressing traffic: a traffic shaping profile must be assigned, and the outbound bandwidth must be configured.

Since traffic shaping is often configured on the WAN interface for egressing traffic, the outbound bandwidth is effectively the upstream bandwidth allowed by your ISP. On the FortiGate, it is possible to perform a speed test on interfaces are assigned a WAN role assigned (see GUI speed test on page 1147). The speed test performs measurements against public cloud servers, and provides an accurate measurement of the upstream bandwidth. After the test is complete, the results can be used to populate the *Outbound bandwidth* field.

To configure traffic shaping on an interface:

1. Go to *Network* > *Interfaces* and double-click an interface to edit it.
2. For interfaces assigned a *WAN* role, in the right-side of the screen, click *Execute speed test*.

3. When the test completes, click *OK* in the *Confirm* pane to apply the results to the estimated bandwidth. The speed test results are populated in the *Estimated bandwidth* fields for *kbps Upstream* and *kbps Downstream*.



4. In the *Traffic Shaping* section, enable *Outbound shaping profile* and select a profile.
5. Enable *Outbound bandwidth* and copy the *kbps Upstream* value from the speed test, or enter a custom value.
6. Click *OK*.

Verifying that the traffic is being shaped

In this example, three traffic classes are defined in the traffic shaping profile assigned to port1. The outbandwidth configured on port1 is 1000 Kbps. Each class has an `allocated-bandwidth`, `guaranteed-bandwidth`, `max-bandwidth`, and `current-bandwidth` value.

- The `guaranteed-bandwidth` and `max-bandwidth` are rates that are converted from the percentage of outbandwidth configured for each class. For example, `class-id` 2 has 10% `guaranteed-bandwidth`, equivalent to 100 Kbps, and 100% `max-bandwidth` equivalent to 1000 Kbps.
- The `allocated-bandwidth` displays the real-time bandwidth allocation for the traffic class based on all available factors. This value changes as traffic demand changes.
- The `current-bandwidth` displays the real-time bandwidth usage detected for the traffic class.

To verify that traffic is being shaped by the traffic shaping profile:

1. Enable debug flow to view the live traffic as it matches a traffic shaping policy:

```
# diagnose debug flow show function-name enable
# diagnose debug flow show iprope enable
# diagnose debug flow filter <filters>
# diagnose debug flow trace start <repeat_number>
# diagnose debug enable
```

The `iprope_shaping_check` function outputs the shaping policy matched for any given traffic:

```
...
id=20085 trace_id=21 func=iprope_shaping_check line=934 msg="in-[port3], out-[port1],
skb_flags-02000000, vid-0"
id=20085 trace_id=21 func=__iprope_check line=2277 msg="gnum-100015, check-
ffffffffa002a8fe"
id=20085 trace_id=21 func=__iprope_check_one_policy line=2029 msg="checked gnum-100015
policy-3, ret-matched, act-accept"
id=20085 trace_id=21 func=__iprope_check_one_policy line=2247 msg="policy-3 is matched,
act-accept"
id=20085 trace_id=21 func=__iprope_check line=2294 msg="gnum-100015 check result: ret-
matched, act-accept, flag-00000000, flag2-00000000"
```

2. Display the session list:

```
# diagnose sys session filter <filters>
# diagnose sys session list
```

Sessions that match a shaping policy will display `class_id` and `shaping_policy_id` fields:

```
...
session info: proto=6 proto_state=05 duration=32 expire=0 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=4 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
```

3. Display the interface statistics:

```
# diagnose netlink interface list port1
if=port1 family=00 type=1 index=3 mtu=1500 link=0 master=0
ref=95 state=start present fw_flags=2001b800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:7e:af:a6 broadcast_addr=ff:ff:ff:ff:ff:ff
inbandwidth=10000(kbps)         total_bytes=2098887K    drop_bytes=7854K
egress traffic control:
        bandwidth=1000(kbps) lock_hit=241 default_class=3 n_active_class=3
        class-id=2      allocated-bandwidth=140(kbps)   guaranteed-bandwidth=100(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=147(kbps)
                        priority=low    forwarded_bytes=8161K
                        dropped_packets=2032    dropped_bytes=3074K
        class-id=3      allocated-bandwidth=30(kbps)    guaranteed-bandwidth=300(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=10(kbps)
                        priority=medium         forwarded_bytes=501K
                        dropped_packets=1       dropped_bytes=1195
        class-id=4      allocated-bandwidth=830(kbps)   guaranteed-bandwidth=500(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=810(kbps)
                        priority=high   forwarded_bytes=1393K
                        dropped_packets=379     dropped_bytes=572K
```

```
stat: rxp=8349728 txp=11101735 rxb=2216101183 txb=1394077978 rxe=0 txe=0 rxd=0 txd=0
mc=0 collision=0 @ time=1654202868
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=95
```

If the debug output does not display egress traffic control by class and displays them by priority, it is likely that global traffic prioritization is configured. The global traffic prioritization settings must be disabled to view the preceding debug output (see Global traffic prioritization on page 1546).

Traffic shaping with queuing using a traffic shaping profile

You can use the weighted random early detection (WRED) queuing function within traffic shaping.

This topic includes two parts:

- Traffic shaping with queuing on page 1526
- Burst control in queuing mode on page 1527

You cannot configure or view WRED in the GUI; you must use the CLI.

WRED is not supported when traffic is offloaded to an NPU.

Traffic shaping with queuing

Traffic shaping has a queuing option. Use this option to fine-tune the queue by setting the profile queue size or performing random early drop (RED) according to queue usage.

This example shows setting the profile queue size limit to 5 so that the queue can contain a maximum of five packets and more packets are dropped.

To set the profile queue size limit:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 10
                set limit 5  <range from 5 to 10000; default: 1000>
            next
        end
    next
end
```

This example shows performing RED according to queue usage by setting `red-probability`, `min`, and `max`. Setting `red-probability` to `10` means start to drop packets when queue usage reaches the `min` setting. When queue usage reaches the `max` setting, drop 10% of the packets.

- Level 1: when queue is less than `min` packets, drop 0% of packets.
- Level 2: when queue reaches `min` packets, start to drop packets.
- Level 3: when queue usage is between `min` and `max` packets, drop 0–10% of packets by proportion.
- Level 4: when queue (average queue size) is more than `max` packets, drop 100% of packets.

To set RED according to queue usage:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 10
                set red-probability 10 <range from 0 to 20; default: 0 no drop>
                set min 100 <range from 3 to 3000>
                set max 300 <range from 3 to 3000>
            next
        end
    next
end
```

To troubleshoot this function, use the following diagnose commands:

```
diagnose netlink intf-class list <intf>
diagnose netlink intf-qdisc list <intf>
```

Burst control in queuing mode

In a hierarchical token bucket (HTB) algorithm, each traffic class has buckets to allow a burst of traffic. The maximum burst is determined by the bucket size `burst` (for guaranteed bandwidth) and `cburst` (for maximum bandwidth). The shaping profile has `burst-in-msec` and `cburst-in-msec` parameters for each shaping entry (`class id`) to control the bucket size.

This example uses the outbandwidth of the interface as 1 Mbps and the maximum bandwidth of class is 50%.

burst = burst-in-msec * guaranteed bandwidth = 100 ms × 1 Mbps x 50% = 50000 b = 6250 B

cburst = cburst-in-msec * maximum bandwidth = 200 ms × 1 Mbps x 50% = 100000 b = 12500 B

The following example sets `burst-in-msec` to 100 and `cburst-in-msec` to 200.

To set burst control in queuing mode:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
```

```
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 50
                set burst-in-msec 100 <range from 0 to 2000>
                set cburst-in-msec 200 <range from 0 to 2000>
            next
        end
    next
end
```

Example

Enabling RED for FTP traffic from QA

This example shows how to enable RED for FTP traffic from QA. This example sets a maximum of 10% of the packets to be dropped when queue usage reaches the maximum value.

To configure the firewall address:

```
config firewall address
    edit QA_team
        set subnet 10.1.100.0/24
    next
end
```

To set the shaping policy to classify traffic into different class IDs:

```
config firewall shaping-policy
    edit 1
        set service HTTPS HTTP
        set dstintf port1
        set srcaddr QA_team
        set dstaddr all
        set class-id 10
    next
    edit 2
        set service FTP
        set dstintf port1
        set srcaddr QA_team
        set dstaddr all
        set class-id 20
    next
end
```

To set the shaping policy to define the speed of each class ID:

```
config firewall shaping-profile
    edit QA_team_profile
       set type queuing
       set default-class-id 30
            config shaping-entries
                edit 1
                    set class-id 10
```

```
                    set guaranteed-bandwidth-percentage 50
                    set maximum-bandwidth-percentage 100
                next
                edit 2
                    set class-id 20
                    set guaranteed-bandwidth-percentage 30
                    set maximum-bandwidth-percentage 60
                    set red-probability 10
                next
                edit 3
                    set class-id 30
                    set guaranteed-bandwidth-percentage 20
                    set maximum-bandwidth-percentage 50
                next
            end
    next
end
```

To apply the shaping policy to the interface:

```
config sys interface
    edit port1
        set outbandwidth 10000
        set egress-shaping-profile QA_team_profile
    next
end
```

To use diagnose commands to troubleshoot:

```
# diagnose netlink intf-class list port1
class htb 1:1 root rate 1250000Bps ceil 1250000Bps burst 1600B/8 mpu 0B overhead 0B cburst
1600B/8 mpu 0B overhead 0B level 7 buffer [00004e20] cbuffer [00004e20]
 Sent 11709 bytes 69 pkt (dropped 0, overlimits 0 requeues 0)
 rate 226Bps 2pps backlog 0B 0p
 lended: 3 borrowed: 0 giants: 0
 tokens: 18500 ctokens: 18500
class htb 1:10 parent 1:1 leaf 10: prio 1 quantum 62500 rate 625000Bps ceil 1250000Bps burst
1600B/8 mpu 0B overhead 0B cburst 1600B/8 mpu 0B overhead 0B level 0 buffer [00009c40]
cbuffer [00004e20]
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0Bps 0pps backlog 0B 0p
 lended: 0 borrowed: 0 giants: 0
 tokens: 40000 ctokens: 20000
class htb 1:20 parent 1:1 leaf 20: prio 1 quantum 37500 rate 375000Bps ceil 750000Bps burst
1599B/8 mpu 0B overhead 0B cburst 1599B/8 mpu 0B overhead 0B level 0 buffer [0001046a]
cbuffer [00008235]
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0Bps 0pps backlog 0B 0p
 lended: 0 borrowed: 0 giants: 0
 tokens: 66666 ctokens: 33333
class htb 1:30 parent 1:1 leaf 30: prio 1 quantum 25000 rate 250000Bps ceil 625000Bps burst
1600B/8 mpu 0B overhead 0B cburst 1600B/8 mpu 0B overhead 0B level 0 buffer [000186a0]
cbuffer [00009c40]
 Sent 11709 bytes 69 pkt (dropped 0, overlimits 0 requeues 0)
 rate 226Bps 2pps backlog 0B 0p
 lended: 66 borrowed: 3 giants: 0
```

```
 tokens: 92500 ctokens: 37000
class red 20:1 parent 20:0

 # diagnose netlink intf-qdisc list port1
qdisc htb 1: root refcnt 5 r2q 10 default 30 direct_packets_stat 0 ver 3.17
 Sent 18874 bytes 109 pkt (dropped 0, overlimits 5 requeues 0)
 backlog 0B 0p
qdisc pfifo 10: parent 1:10 refcnt 1 limit 1000p
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
qdisc red 20: parent 1:20 refcnt 1 limit 4000000B min 300000B max 1000000B ewma 9 Plog 23
Scell_log 20 flags 0
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
  marked 0 early 0 pdrop 0 other 0
qdisc pfifo 30: parent 1:30 refcnt 1 limit 1000p
 Sent 18874 bytes 109 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
```

Traffic shapers

The following topics provide more information about traffic shapers:

- Shared traffic shaper on page 1530
- Per-IP traffic shaper on page 1535
- Changing traffic shaper bandwidth unit of measurement on page 1538
- Multi-stage DSCP marking and class ID in traffic shapers on page 1538
- Multi-stage VLAN CoS marking on page 1540
- Adding traffic shapers to multicast policies on page 1544

Shared traffic shaper

Shared traffic shaper is used in a firewall shaping policy to indicate the priority and guaranteed and maximum bandwidth for a specified type of traffic use.

The maximum bandwidth indicates the largest amount of traffic allowed when using the policy. You can set the maximum bandwidth to a value between 1 and 16776000 Kbps. The GUI displays an error if any value outside this range is used. If you want to allow unlimited bandwidth, use the CLI to enter a value of 0.

The guaranteed bandwidth ensures that there is a consistent reserved bandwidth available. When setting the guaranteed bandwidth, ensure that the value is significantly less than the interface's bandwidth capacity. Otherwise, the interface will allow very little or no other traffic to pass through, potentially causing unwanted latency.

In a shared traffic shaper, the administrator can prioritize certain traffic as high, medium, or low. FortiOS provides bandwidth to low priority connections only when high priority connections do not need the bandwidth. For example, you should assign a high traffic priority to a policy for connecting a secure web server that needs to support e-commerce traffic. You should assign less important services a low priority.

When you configure a shared traffic shaper, you can apply bandwidth shaping per policy or for all policies. By default, a shared traffic shaper applies traffic shaping evenly to all policies that use the shared traffic shaper.

When configuring a per-policy traffic shaper, FortiOS applies the traffic shaping rules defined for each security policy individually. For example, if a per-policy traffic shaper is configured with a maximum bandwidth of 1000 Kbps, any security policies that have that traffic shaper enabled get 1000 Kbps of bandwidth each.

If a traffic shaper for all policies is configured with a maximum bandwidth of 1000 Kbps, all policies share the 1000 Kbps on a first-come, first-served basis.

The configuration is as follows:

```
config firewall shaper traffic-shaper
   edit "traffic_shaper_name"
      set per-policy enable
   next
end
```

The shared traffic shaper selected in the traffic shaping policy affects traffic in the direction defined in the policy. For example, if the source port is LAN and the destination is WAN1, the traffic shaping affects the flow in this direction only, affecting the outbound traffic's upload speed. You can define the traffic shaper for the policy in the opposite direction (reverse shaper) to affect the inbound traffic's download speed. In this example, that would be from WAN1 to LAN.

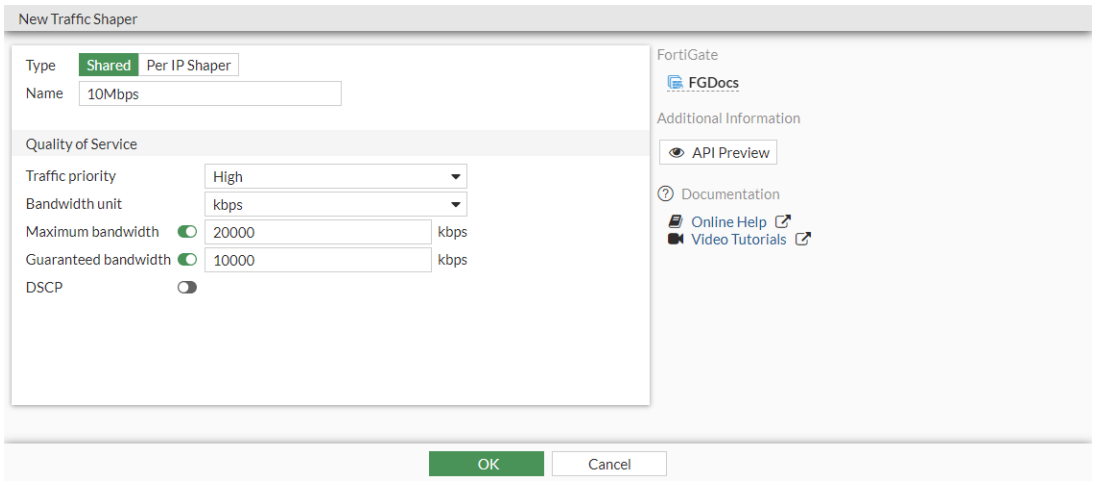
Only traffic through forward traffic shapers will be included in FortiView; reverse and per-IP shapers are not included.

Traffic shapers can be added to a multicast policy when multicast routing is enabled.

The following example shows how to apply different speeds to different types of service. The example configures two shared traffic shapers to use in two firewall shaping policies. One policy guarantees a speed of 10 Mbps for VoIP traffic. The other policy guarantees a speed of 1 Mbps for other traffic. In the example, FortiOS communicates with a PC using port10 and the Internet using port9.

To configure shared traffic shapers in the GUI:

1. Create a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set the *Name* to *Internet Access*.
   c. Set the *Incoming Interface* to *port10*.
   d. Set the *Outgoing Interface* to *port9*.
   e. Set the *Source* and *Destination* to *all*.
   f. Set the *Schedule* to *always*.
   g. Set the *Service* to *ALL*.
   h. Click *OK*.
2. Create the shared traffic shapers:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shapers* tab, and click *Create New*.
   b. Set the *Name* to *10Mbps*. This shaper is for VoIP traffic.
   c. Set the *Traffic Priority* to *High*.
   d. Enable *Max Bandwidth* and enter *20000*.
   e. Enable *Guaranteed Bandwidth* and enter *10000*.



f. Click *OK*.

g. Repeat the above steps to create another traffic shaper named *1Mbps* with the *Traffic Priority* set to *Low*, the *Max Bandwidth* set to *10000*, and the *Guaranteed Bandwidth* set to *1000*.

3. Create a firewall shaping policy:

a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.

b. Set the *Name* to *VoIP_10Mbps_High*. This policy is for VoIP traffic.

c. Set the *Source* and *Destination* to *all*.

d. Set the *Service* to all VoIP services.

e. Set the *Outgoing Interface* to *port9*.

f. Enable *Shared shaper* and select *10Mbps*.

g. Enable *Reverse shaper* and select *10Mbps*.

h. Click *OK*.

i. Repeat the above steps to create another firewall shaping policy named *Other_1Mbps_Low* for other traffic, with the *Source* and *Destination* set to *all*, *Service* set to *ALL*, *Outgoing Interface* set to *port9*, and *Shared shaper* and *Reverse shaper* set to *1Mbps*.

To configure shared traffic shapers in the CLI:

1. Create a firewall policy:

```
config firewall policy
   edit 1
      set name "Internet Access"
      set srcintf "port10"
      set dstintf "port9"
      set srcaddr "all"
      set dstaddr "all"
      set action accept
      set schedule "always"
      set service "ALL"
      set fsso disable
      set nat enable
   next
end
```

2. Create the shared traffic shapers:

```
config firewall shaper traffic-shaper
```

```
    edit "10Mbps"
      set guaranteed-bandwidth 10000
      set maximum-bandwidth 20000
    next
    edit "1Mbps"
      set guaranteed-bandwidth 1000
      set maximum-bandwidth 10000
      set priority low
    next
end
```

3. Create a firewall shaping policy:

```
config firewall shaping-policy
    edit 1
      set name "VOIP_10Mbps_High"
      set service "H323" "IRC" "MS-SQL" "MYSQL" "RTSP" "SCCP" "SIP" "SIP-MSNmessenger"
      set dstintf "port9"
      set traffic-shaper "10Mbps"
      set traffic-shaper-reverse "10Mbps"
      set srcaddr "all"
      set dstaddr "all"
    next
    edit 2
      set name "Other_1Mbps_Low"
      set service "ALL"
      set dstintf "port9"
      set traffic-shaper "1Mbps"
      set traffic-shaper-reverse "1Mbps"
      set srcaddr "all"
      set dstaddr "all"
    next
end
```

To troubleshoot shared traffic shapers:

1. Check if specific traffic is attached to the correct traffic shaper. The example output shows the traffic attached to the 10Mbps and 1Mbps shapers:

```
# diagnose firewall iprope list 100015
policy index=1 uuid_idx=0 action=accept
flag (0):
shapers: orig=10Mbps(2/1280000/2560000)
cos_fwd=0 cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(1): 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
service(15):
   [6:0x0:0/(1,65535)->(1720,1720)] helper:auto
   [6:0x0:0/(1,65535)->(1503,1503)] helper:auto
   [17:0x0:0/(1,65535)->(1719,1719)] helper:auto
   [6:0x0:0/(1,65535)->(6660,6669)] helper:auto
   [6:0x0:0/(1,65535)->(1433,1433)] helper:auto
   [6:0x0:0/(1,65535)->(1434,1434)] helper:auto
   [6:0x0:0/(1,65535)->(3306,3306)] helper:auto
   [6:0x0:0/(1,65535)->(554,554)] helper:auto
```

```
   [6:0x0:0/(1,65535)->(7070,7070)] helper:auto
   [6:0x0:0/(1,65535)->(8554,8554)] helper:auto
   [17:0x0:0/(1,65535)->(554,554)] helper:auto
   [6:0x0:0/(1,65535)->(2000,2000)] helper:auto
   [6:0x0:0/(1,65535)->(5060,5060)] helper:auto
   [17:0x0:0/(1,65535)->(5060,5060)] helper:auto
   [6:0x0:0/(1,65535)->(1863,1863)] helper:auto

policy index=2 uuid_idx=0 action=accept
flag (0):
shapers: orig=1Mbps(4/128000/1280000)
cos_fwd=0 cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(1): 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
service(1):
   [0:0x0:0/(0,0)->(0,0)] helper:auto
```

2. Check if the correct traffic shaper is applied to the session. The example output shows that the 1Mbps shaper is applied to the session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=11 expire=3599 timeout=3600 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=5
origin-shaper=1Mbps prio=4 guarantee 128000Bps max 1280000Bps traffic 1050Bps drops 0B
reply-shaper=
per_ip_shaper=
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty npu npd os mif route_preserve
statistic(bytes/packets/allow_err): org=868/15/1 reply=752/10/1 tuples=2
tx speed(Bps/kbps): 76/0 rx speed(Bps/kbps): 66/0
orgin->sink: org pre->post, reply pre->post dev=39->38/38->39 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:58241->172.16.200.55:21(172.16.200.1:58241)
hook=pre dir=reply act=dnat 172.16.200.55:21->172.16.200.1:58241(10.1.100.11:58241)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=4
serial=0003255f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
     vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason: offload-denied helper
total session 1
```

3. Check the statuses of shared traffic shapers:

```
# diagnose firewall shaper traffic-shaper list
name 10Mbps
maximum-bandwidth 2500 KB/sec
guaranteed-bandwidth 1250 KB/sec
current-bandwidth 0 B/sec
priority 2
tos ff
packets dropped 0
```

```
bytes dropped 0

name 1Mbps
maximum-bandwidth 1250 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4
tos ff
packets dropped 0
bytes dropped 0
```

Per-IP traffic shaper

With per-IP traffic shaping, you can limit each IP address's behavior to avoid a situation where one user uses all of the available bandwidth. In addition to controlling the maximum bandwidth used per IP address, you can also define the maximum number of concurrent sessions for an IP address. For example, if you apply a per-IP shaper of 1 Mbps to your entire network, FortiOS allocates each user/IP address 1 Mbps of bandwidth. Even if the network consists of a single user, FortiOS allocates them 1 Mbps. If there are ten users, each user gets 1 Mbps of bandwidth, totaling 10 Mbps of outgoing traffic.

For shared shapers, all users share the set guaranteed and maximum bandwidths. For example, if you set a shared shaper for all PCs using an FTP service to 10 Mbps, all users uploading to the FTP server share the 10 Mbps.

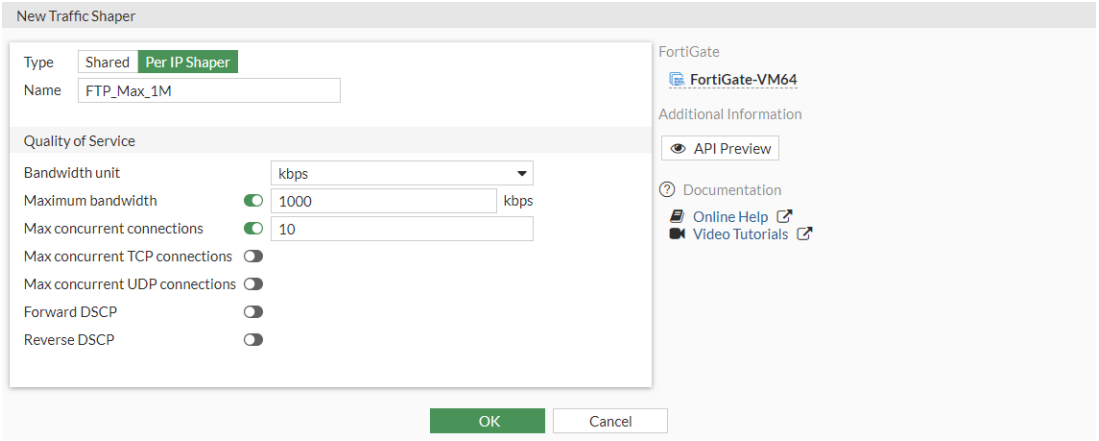
Shared shapers affect upload speed. If you want to limit the download speed from the FTP server in the example, you must configure the shared shaper as a reverse shaper. Per-IP shapers apply the speed limit on both upload and download operations. Only traffic through forward traffic shapers will be included in FortiView; reverse and per-IP shapers are not included.

The following example shows how to apply a per-IP shaper to a traffic shaping policy. This shaper assigns each user a maximum bandwidth of 1 Mbps and allows each user to have a maximum of ten concurrent connections to the FTP server. In the example, FortiOS communicates with users using port10 and the FTP server using port9.

To configure a per-IP traffic shaper in the GUI:

1. Create a firewall policy:
   a. Go to *Policy & Objects* > *IPv4 Policy* and click *Create New*.
   b. Set the *Name* to *FTP Access*.
   c. Set the *Incoming Interface* to *port10*.
   d. Set the *Outgoing Interface* to *port9*.
   e. Set the *Source* to *all*.
   f. Set the *Destination* to *FTP_Server*.
   g. Set the *Schedule* to *always*.
   h. Set the *Service* to *ALL*.
   i. Click *OK*.
2. Create the per-IP traffic shaper:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shapers* tab, and click *Create New*.
   b. Set *Type* to *Per IP Shaper*.
   c. Enter the *Name* (*FTP_Max_1M*). This shaper is for VoIP traffic.
   d. Enable *Max Bandwidth* and enter *1000*.

e. Enable *Max Concurrent Connections* and enter *10*. This means that each user can have up to ten concurrent connections to the FTP server.



f. Click *OK*.

3. Create a firewall shaping policy:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.
   b. Enter the *Name* (*FTP speed 1M*).
   c. Set the *Source* to the addresses and users that require access to the FTP server.
   d. Set the *Destination* to *FTP_Server*.
   e. Set the *Service* to *ALL*.
   f. Set the *Outgoing Interface* to *port9*.
   g. Enable *Per-IP shaper* and select *FTP_Max_1M*.
   h. Click *OK*.

To configure a per-IP traffic shaper in the CLI:

1. Create a firewall policy:

```
config firewall policy
   edit 1
      set name "FTP Access"
      set srcintf "port10"
      set dstintf "port9"
      set srcaddr "all"
      set dstaddr "FTP_Server"
      set action accept
      set schedule "always"
      set service "ALL"
      set fsso disable
      set nat enable
   next
end
```

2. Create the per-IP traffic shaper:

```
config firewall shaper per-ip-shaper
   edit "FTP_Max_1M"
      set max-bandwidth 1000
      set max-concurrent-session 10
   next
end
```

3. Create a firewall shaping policy:

```
config firewall shaping-policy
   edit 1
      set name "FTP speed 1M"
      set service "ALL"
      set dstintf "port9"
      set per-ip-shaper "FTP_Max_1M"
      set srcaddr "PC1" "WinPC" "PC2"
      set dstaddr "FTP_Server"
   next
end
```

To troubleshoot per-IP traffic shapers:

1. Check if specific traffic is attached to the correct traffic shaper. The example output shows the traffic attached to the FTP_Max_1M shaper:

```
# diagnose firewall iprope list 100015
policy index=3 uuid_idx=0 action=accept
flag (0):
shapers: per-ip=FTP_Max_1M
cos_fwd=0 cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(1): 38
source(3): 10.1.100.11-10.1.100.11, uuid_idx=30, 10.1.100.143-10.1.100.143, uuid_idx=32,
     10.1.100.22-10.1.100.22, uuid_idx=31,
dest(1): 172.16.200.55-172.16.200.55, uuid_idx=89,
service(1):
   [0:0x0:0/(0,65535)->(0,65535)] helper:auto
```

2. Check if the correct traffic shaper is applied to the session. The example output shows that the FTP_Max_1M shaper is applied to the session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=36 expire=3567 timeout=3600 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=FTP_Max_1M
class_id=0 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty per_ip npu npd mif route_preserve
statistic(bytes/packets/allow_err): org=506/9/1 reply=416/6/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=39->38/38->39 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:58275->172.16.200.55:21(172.16.200.1:58275)
hook=pre dir=reply act=dnat 172.16.200.55:21->172.16.200.1:58275(10.1.100.11:58275)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=2
serial=0000211a tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
     vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
```

```
no_ofld_reason: offload-denied helper
```

3. Check the statuses of per-IP traffic shapers. The output should resemble the following:

```
# diagnose firewall shaper per-ip-shaper list
name FTP_Max_1M
maximum-bandwidth 125 KB/sec
maximum-concurrent-session 10
tos ff/ff
packets dropped 0
bytes dropped 0
addr=10.1.100.11 status: bps=0 ses=3
```

Changing traffic shaper bandwidth unit of measurement

Bandwidth speeds are measured in kilobits per second (Kbps), and bytes that are sent and received are measured in megabytes (MB). In some cases, this can cause confusion depending on whether your ISP uses kilobits per second (Kbps), kilobytes per second (KBps), megabits per second (Mbps), or gigabits per second (Gbps).

You can change the unit of measurement for traffic shapers in the CLI.

To change the bandwidth unit of measurement for a shared traffic shaper:

```
config firewall shaper traffic-shaper
    edit <traffic_shaper_name>
        set bandwidth-unit {kbps | mbps | gbps}
    next
end
```

To change the bandwidth unit of measurement for a per-IP traffic shaper:

```
config firewall shaper per-ip-shaper
    edit <traffic_shaper_name>
        set bandwidth-unit {kbps | mbps | gbps}
    next
end
```

Multi-stage DSCP marking and class ID in traffic shapers

Traffic shapers have a multi-stage method so that packets are marked with a different differentiated services code point (DSCP) and `class id` at different traffic speeds. Marking packets with a different DSCP code is for the next hop to classify the packets. The FortiGate benefits by marking packets with a different `class id`. Combined with the egress interface shaping profile, the FortiGate can handle the traffic differently according to its `class id`.

Rule	DSCP code	Class ID
speed < guarantee bandwidth	`diffservcode`	`class id` in shaping policy
guarantee bandwidth < speed < exceed bandwidth	`exceed-dscp`	`exceed-class-id`
exceed bandwidth < speed	`maximum-dscp`	`exceed-class-id`

This example sets the following parameters:

- When the current bandwidth is less than 50 Kbps, mark packets with `diffservcode` 100000 and set `class id` to 10.
- When the current bandwidth is between 50 Kbps and 100 Kbps, mark packets with `exceed-dscp` 111000 and set `exceed-class-id` to 20.
- When the current bandwidth is more than 100 Kbps, mark packets with `maximum-dscp` 111111 and set `exceed-class-id` to 20.

To set multi-stage DSCP marking and class ID in a traffic shaper:

```
config firewall shaper traffic-shaper
    edit "50k-100k-150k"
        set guaranteed-bandwidth 50
        set maximum-bandwidth 150
        set diffserv enable
        set dscp-marking-method multi-stage
        set exceed-bandwidth 100
        set exceed-dscp 111000
        set exceed-class-id 20
        set maximum-dscp 111111
        set diffservcode 100000
    next
end
```

```
config firewall shaping-policy
     edit 1
         set service "ALL"
         set dstintf PORT2
         set srcaddr "all"
         set dstaddr "all"
         set class-id 10
     next
end
```

Traffic shapers also have an `overhead` option that defines the per-packet size overhead used in rate computation.

To set the traffic shaper overhead option:

```
config firewall shaper traffic-shaper
    edit "testing"
        set guaranteed-bandwidth 50
        set maximum-bandwidth 150
        set overhead 14 <range from 0 to 100>
    next
end
```

Example

This example shows how to mark QA traffic with a different DSCP according to real-time traffic speed.

To configure the firewall address:

```
config firewall address
    edit QA_team
```

```
        set subnet 10.1.100.0/24
    next
end
```

To configure the firewall shaper traffic shaper:

```
config firewall shaper traffic-shaper
    edit "500k-1000k-1500k"
        set guaranteed-bandwidth 500
        set maximum-bandwidth 1500
        set diffserv enable
        set dscp-marking-method multi-stage
        set exceed-bandwidth 1000
        set exceed-dscp 111000
        set maximum-dscp 111111
        set diffservcode 100000
    next
end
```

```
config firewall shaping-policy
    edit QA_team
        set service "ALL"
        set dstintf port1
        set traffic-shaper "500k-1000k-1500k"
        set traffic-shaper-reverse "500k-1000k-1500k"
        set srcaddr "QA_team"
        set dstaddr "all"
    next
end
```

Multi-stage VLAN CoS marking

A FortiGate can configure the traffic shaper to dynamically change the CoS value of outgoing VLAN packets based on the shaper profile. This allows the FortiGate to mark traffic with different CoS values at different stages of the shaping process.

```
config firewall shaper traffic-shaper
    edit <name>
        set bandwidth-unit {kbps | mbps | gbps}
        set guaranteed-bandwidth <integer>
        set maximum-bandwidth <integer>
        set cos-marking {enable | disable}
        set cos-marking-method {static | multi-stage}
        set cos <3-bit_binary>
        set exceed-cos <3-bit_binary>
        set maximum-cos <3-bit_binary>
        set exceed-bandwidth <integer>
    next
end
```

`cos-marking {enable | disable}`	Enable/disable VLAN CoS marking (default = `disable`).
`cos-marking-method {static | multi-stage}`	Set the VLAN CoS marking method. • `static`: use static VLAN CoS marking (default)

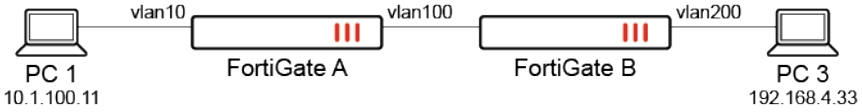
	• `multi-stage`: multi-stage VLAN CoS marking
`cos <3-bit_binary>`	Set the VLAN CoS mark, 3-bit binary (000 - 111).
`exceed-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `guaranteed-bandwidth` and `exceed-bandwidth`, 3-bit binary (000 - 111).
`maximum-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `exceed-bandwidth` and `maximum-bandwidth`, 3-bit binary (000 - 111).
`exceed-bandwidth <integer>`	Set the exceed bandwidth used for DSCP or VLAN CoS multi-stage marking. The integer value range depends on the `bandwidth-unit` setting. This setting is only available for CoS multi-stage marking.

Example

In this example, mutli-stage VLAN CoS marking is configured using traffic shapers on FortiGate A and FortiGate B. FortiGate A applies multi-stage CoS marking with the following traffic shaper settings:

- Traffic below the guaranteed bandwidth will apply CoS 6.
- Traffic greater than the guaranteed bandwidth will apply CoS 6 and 5.
- Traffic greater than the exceed bandwidth will apply CoS 6, 5, and 4.

A traffic shaper and shaping policy are configured on FortiGate B. When traffic comes from FortiGate A with CoS 6, the traffic shaping policy will be applied because the CoS matches.



Multi-stage VLAN CoS marking is not supported on NP models. Traffic is not offloaded when it is enabled.

To configure mutli-stage VLAN CoS marking on FortiGate A:

1. Configure the firewall policy:

```
config firewall policy
    edit 7
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set traffic-shaper "multi-stage-cos-fgta"
        set traffic-shaper-reverse "multi-stage-cos-fgta"
```

```
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgta"
        set guaranteed-bandwidth 1000
        set maximum-bandwidth 4000
        set per-policy enable
        set exceed-bandwidth 2000
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 110
        set exceed-cos 101
        set maximum-cos 100
    next
end
```

3. Check the session list to verify that CoS 6 is marked:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=6 expire=180 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic
504900Bps drops 163905268B
reply-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic
504900Bps drops 0B
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=6/6
state=log may_dirty npu npd os rs f00
statistic(bytes/packets/allow_err): org=3804176/292/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 583462/4667 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:37586->192.168.4.33:5001(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:5001->10.1.100.11:37586(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=7 pol_uuid_idx=1129 auth_info=0 chk_client_info=0 vd=2
serial=0006613c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied
```

To configure mutli-stage VLAN CoS marking on FortiGate B:

1. Configure the firewall policy:

```
config firewall policy
    edit 4
        set srcintf "vlan100"
        set dstintf "vlan200"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
```

```
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgtb"
        set guaranteed-bandwidth 250
        set maximum-bandwidth 1000
        set per-policy enable
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 100
        set exceed-cos 101
        set maximum-cos 110
        set exceed-bandwidth 500
    next
end
```

Based on this traffic shaper, the following CoS marking rules will be applied:

- If all traffic is less than the guaranteed bandwidth, then the traffic will be marked with CoS 4.
- If all traffic is greater than the guaranteed bandwidth but less than the exceed bandwidth, then 50% of the traffic will be marked as CoS 4 and 50% as CoS 5.
- If traffic is greater than the guaranteed bandwidth but less than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6; CoS 4 and 5 will have another 50%.
- If traffic is greater than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6, 25% will be marked as CoS 4, and 25% will be marked as CoS 5. Packet drops will be visible in the debug output.

3. Configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 1
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set traffic-shaper "multi-stage-cos-fgtb"
        set traffic-shaper-reverse "multi-stage-cos-fgtb"
        set class-id 2
        set cos-mask 111
        set cos 110
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

4. Check the session list to verify that the shaping ID (1) applied and CoS 4 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic
236Bps drops 0B
reply-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic 236Bps
```

```
drops 0B
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=4/4
state=log may_dirty os rs f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 120/0 rx speed(Bps/kbps): 120/0
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:29899->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:29899->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1377 auth_info=0 chk_client_info=0 vd=4
serial=00024329 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
total session 1
```

Adding traffic shapers to multicast policies

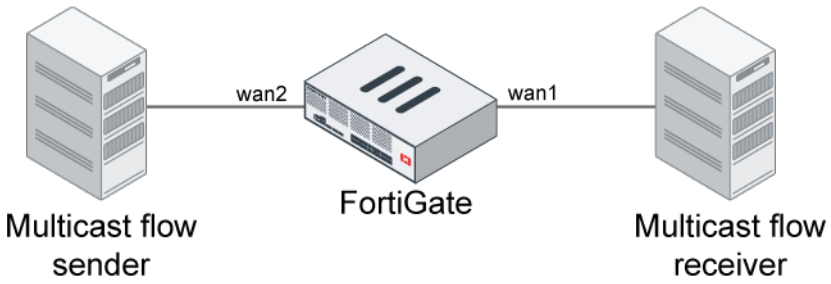
When multicast routing is enabled, a traffic shaper can be added to a multicast policy.

Only a shared traffic shaper with the `per-policy` option disabled can be used. This is the default state of the `per-policy` option. The `auto-asic-offload` option must also be disabled on the multicast policy.

This feature is currently not supported on IPv6 multicast policies or on transparent mode VDOMs.

Example

In this example, a traffic shaper is applied to the multicast policy. A multicast flow sender sends the multicast data stream. The shaper attached to the multicast session is checked, and the shaping of the data stream is confirmed in the multicast session.



To apply traffic shaping to a multicast policy:

1. Enable multicast routing on the VDOM:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        config rp-address
```

```
            edit 1
                set ip-address 10.1.100.10
            next
        end
    end
    config interface
        edit "wan2"
            set pim-mode sparse-mode
        next
        edit "wan1"
            set pim-mode sparse-mode
        next
    end
end
```

2. Create a traffic shaper:

```
config firewall shaper traffic-shaper
    edit "shaper128kbps-high"
        set guaranteed-bandwidth 128
        set maximum-bandwidth 128
        set per-policy disable
        set diffserv enable
        set diffservcode 010101
    next
end
```

3. Apply the traffic shaper to the multicast policy and disable NPU offloading:

```
config firewall multicast-policy
    edit 1
        set name "test_multicast-policy"
        set logtraffic enable
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set snat enable
        set auto-asic-offload disable
        set traffic-shaper "shaper128kbps-high"
    next
end
```

4. Check the shaper and DSCP in the multicast session:

```
# diagnose sys mcast-session list
    session info: id=26 vf=0 proto=17 10.1.100.41.35537->230.0.0.1.7878
    used=2 path=1 duration=118 expire=179 indev=18 pkts=119 bytes=64260
    state=00000000:
    session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0
fwd_map=0x00000000
    path: log snat npu-deny nsaddr=172.16.200.10 policy=1, outdev=17, tos=0x15
            origin-shaper=shaper128kbps-high prio=2 tos=0x15 guarantee 16000Bps max
16000Bps traffic 620Bps drops 0pkt/0B
    Total 1 sessions
```

Global traffic prioritization

Global traffic prioritization allows your traffic to be prioritized as high (2), medium (3), or low (4) based on ToS (type of service) or DSCP. When using ToS-based priority, integers 0 to 15 can be used, which correspond to the definitions of the ToS field values in RFC 1349. When using DSCP, values 0 to 63 can be used, which correspond to the six bits in the DSCP value.

The outbandwidth must be defined in order for global prioritization to take effect. When the outbandwidth is defined on an interface without an applied `egress-shaping-profile`, the interface has a total of five priority levels:

Priority level	Description
0	Top
1	Critical
2	High
3	Medium
4	Low

Priority level 0 is reserved for administrative and local out traffic. Priority level 1 is used for traffic that is below guaranteed bandwidth when using a traffic shaper.

Traffic shaper and traffic shaping profile configurations take precedence over global traffic prioritization.

CLI commands

The following commands are used to configure the prioritization either by ToS or DSCP.

To configure the traffic prioritization type and level:

```
config system global
    set traffic-priority {tos | dscp}
    set traffic-priority-level {high | medium | low}
end
```

To configure the ToS-based priority table:

```
config system tos-based-priority
    edit <id>
        set tos <0-15>
        set priority (high | medium | low)
    next
end
```

To configure the DSCP-based priority table:

```
config system dscp-based-priority
    edit <id>
        set ds <0-63>
        set priority (high | medium | low)
    next
end
```

To configure the interface outbandwidth:

```
config system interface
    edit <name>
        set outbandwidth <bandwidth in kbps>
    next
end
```

Example

In the following configuration, packets with DSCP markings of 1 are prioritized as high, and packets with DSCP markings of 2 are prioritized as medium. All the other traffic is prioritized as low. The outbandwidth on interface port3 is set to 1000 kbps.

To configure DSCP-based traffic prioritization:

1. Configure DSCP-based prioritization in the global settings:

```
config system global
    set traffic-priority dscp
    set traffic-priority-level low
end
```

2. Configure the DSCP-based priority table:

```
config system dscp-based-priority
    edit 1
        set ds 1
        set priority high
    next
    edit 2
        set ds 2
        set priority medium
    next
end
```

3. Configure the outbandwidth on port3:

```
config system interface
    edit "port3"
        set outbandwidth 1000
    next
end
```

Verifying the traffic prioritization

When traffic exceeds the outbandwidth of 1000 kbps, traffic prioritization will take effect. Since the form of traffic shaping applied here is policing, excess packets above the outbandwidth are dropped.

In scenario 1, approximately 300 kbps of high priority traffic and 300 kbps of medium priority traffic passes through the FortiGate on port3.

To debug the bandwidth allocation:

```
# diagnose netlink interface list port3
if=port3 family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=35 state=start present fw_flags=3800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:fb:81:0c broadcast_addr=ff:ff:ff:ff:ff:ff
outbandwidth=1000(kbps)
        priority=0      allocated-bandwidth=0(kbps)     total_bytes=9311K       drop_
bytes=197K
        priority=1      allocated-bandwidth=0(kbps)     total_bytes=0   drop_bytes=0
        priority=2      allocated-bandwidth=354(kbps)   total_bytes=20407K      drop_
bytes=48K
        priority=3      allocated-bandwidth=354(kbps)   total_bytes=7093K       drop_
bytes=1262K
        priority=4      allocated-bandwidth=290(kbps)   total_bytes=266018K     drop_
bytes=7743K
stat: rxp=15450901 txp=25933756 rxb=5456860515 txb=17257309292 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1629439926
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=35
```

High priority (2) traffic is allocated 354 kbps of bandwidth. Medium priority (3) traffic is also allocated 354 kbps of bandwidth. The remaining bandwidth is allocated to low priority (4) traffic.

In scenario 2, approximately 400 kbps of high priority traffic and 800 kbps of medium priority traffic passes through the FortiGate on port3.

To debug the bandwidth allocation:

```
# diagnose netlink interface list port3
if=port3 family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=3800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:fb:81:0c broadcast_addr=ff:ff:ff:ff:ff:ff
outbandwidth=1000(kbps)
        priority=0      allocated-bandwidth=7(kbps)     total_bytes=9981K       drop_
bytes=240K
        priority=1      allocated-bandwidth=0(kbps)     total_bytes=0   drop_bytes=0
        priority=2      allocated-bandwidth=425(kbps)   total_bytes=31478K      drop_
bytes=101K
        priority=3      allocated-bandwidth=567(kbps)   total_bytes=12056K      drop_
bytes=1984K
        priority=4      allocated-bandwidth=0(kbps)     total_bytes=266795K     drop_
bytes=7771K
stat: rxp=15461740 txp=25950805 rxb=5459688950 txb=17273940560 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1629440553
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
```

```
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36
```

High priority (2) traffic is allocated 425 kbps of bandwidth. Medium priority (3) traffic is allocated 567 kbps of bandwidth. Since the total bandwidth required exceeds 1000 kbps, the remaining medium priority (3) traffic is dropped. In comparing the successive debug outputs, the `drop_bytes` counter for medium priority (3) traffic gets bigger.

DSCP matching and DSCP marking

This section includes:

- DSCP matching in firewall policies
- DSCP matching in firewall shaping policies
- DSCP marking in firewall shaping policies
- DSCP marking for self-generated traffic NEW on page 1552

DSCP matching in firewall policies

Traffic is allowed or blocked according to the Differentiated Services Code Point (DSCP) values in the incoming packets.

The following CLI variables are available in the `config firewall policy` command:

`tos-mask <mask_value>`	Non-zero bit positions are used for comparison. Zero bit positions are ignored (default = 0x00). This variable replaces the `dscp-match` variable.
`tos <tos_value>`	Type of Service (ToC) value that is used for comparison (default = 0x00). This variable is only available when `tos-mask` is not zero. This variable replaces the `dscp-value` variable.
`tos-negate {enable | disable}`	Enable/disable negated ToS match (default = disable). This variable is only available when `tos-mask` is not zero. This variable replaces the `dscp-negate` variable.

DSCP matching in firewall shaping policies

Shaping is applied to the session or not according to the DSCP values in the incoming packets. The same logic and commands as in firewall policies are used.

DSCP marking in firewall shaping policies

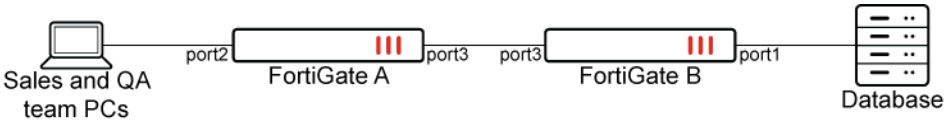
Traffic is allowed or blocked according to the DSCP values in the incoming packets. DSCP marking in firewall shaping policies uses the same logic and commands as in firewall policy and traffic-shaper.

When DSCP marking on `firewall shaper traffic-shaper`, `firewall shaping-policy`, and `firewall policy` all apply to the same session, `shaping-policy` overrides `policy`, and `shaper traffic-shaper` overrides both `shaping-policy` and `policy`.

The following CLI variables in `config firewall policy` are used to mark the packets:

`diffserv-forward {enable | disable}`	Enable/disable changing a packet's DiffServ values to the value specified in `diffservcode-forward` (default = disable).
`diffservcode-forward <dscp_value>`	The value that packet's DiffServ is set to (default = 000000). This variable is only available when `diffserv-forward` is enabled.
`diffserv-reverse {enable | disable}`	Enable/disable changing a packet's reverse (reply) DiffServ values to the value specified in `diffservcode-rev` (default = disable).
`diffservcode-rev <dscp_value>`	The value that packet's reverse (reply) DiffServ is set to (default = 000000). This variable is only available when `diffserv-rev` is enabled.

The following topology is used in the examples:



Example 1

FortiGate A marks traffic from the sales and QA teams with different DSCP values. FortiGate B does DSCP matching, allowing only the sales team to access the database.

1. Configure FortiGate A:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "QA"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 110000
        set nat enable
    next
    edit 5
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "Sales"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 111011
        set nat enable
    next
end
```

2. Configure FortiGate B:

```
config firewall policy
    edit 2
```

```
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "Database"
        set action accept
        set schedule "always"
        set service "ALL"
        set tos-mask 0xf0
        set tos 0xe0
        set fsso disable
        set nat enable
    next
end
```

Example 2

FortiGate A marks traffic from the sales and QA teams with different DSCP values. FortiGate B uses a firewall shaping policy to do the DSCP matching, limiting the connection speed of the sales team to the database to 10MB/s.

1. Configure FortiGate A:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "QA"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 110000
        set nat enable
    next
    edit 5
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "Sales"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 111011
        set nat enable
    next
end
```

2. Configure FortiGate B:

```
config firewall policy
    edit 2
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
```

```
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
config firewall shaper traffic-shaper
    edit "10MB/s"
        set guaranteed-bandwidth 60000
        set maximum-bandwidth 80000
    next
end
config firewall shaping-policy
    edit 1
        set service "ALL"
        set dstintf "port1"
        set tos-mask 0xf0
        set tos 0xe0
        set traffic-shaper "10MB/s"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Example 3

FortiGate A has a traffic shaping policy to mark traffic from the QA team with a DSCP value of 100000, while reverse traffic is marked with 000011.

1. Configure FortiGate A:

```
config firewall shaping-policy
    edit 1
        set name "QA Team 50MB"
        set service "ALL"
        set dstintf "port3"
        set traffic-shaper "50MB/s"
        set traffic-shaper-reverse "50MB/s"
        set diffserv-forward enable
        set diffserv-reverse enable
        set srcaddr "QA"
        set dstaddr "all"
        set diffservcode-forward 100000
        set diffservcode-rev 000011
    next
end
```

DSCP marking for self-generated traffic - NEW

FortiOS supports DSCP and VLAN CoS marking for both local-in and local-out traffic.

Most network providers often require that both application traffic and FortiGate self-generated traffic must be marked with specific DSCP values to ensure efficient traffic management and quality of service (QoS). FortiOS DSCP marking ensures that self-generated traffic complies with the network's standards. This enables the FortiGate to operate as a fully functional Customer Premises Equipment (CPE) that is capable of directly connecting to the provider's network without a CPE router.

To configure DSCP and VLAN CoS for local-in traffic:

1. Configure the traffic shaper with bandwidth settings and the DSCP and VLAN CoS mark:

```
config firewall shaper traffic-shaper
    edit "test-shaper-300kbps"
        set guaranteed-bandwidth 30
        set maximum-bandwidth 300
        set per-policy enable
        set diffserv enable
        set cos-marking enable
        set cos 001
        set diffservcode 000001
    next
end
```

2. Configure the shaping policy for local-in traffic:

```
config firewall shaping-policy
    edit 2
        set traffic-type local-in
        set service "ALL"
        set traffic-shaper-reverse "test-shaper-300kbps"
        set class-id 2
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Verify that the shaper was successfully applied to the shaping policy:

```
# diagnose firewall iprope list 100018
policy index=2 uuid_idx=926 action=accept
flag (0):
schedule(always)
shapers: reply=test-shaper-300kbps(2/3750/37500)
cos_fwd=255  cos_rev=255
group=00100018 av=00000000 au=00000000 split=00000000
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 2
```

4. Test local-in traffic from the PC to the FortiGate.
   a. Check the session list:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=9 expire=179 timeout=0 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=test-shaper-300kbps prio=2 guarantee 3750Bps max 37500Bps traffic
7881Bps drops 651B
per_ip_shaper=
class_id=2 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/1
```

```
state=local may_dirty rs
statistic(bytes/packets/allow_err): org=337599/4717/1 reply=342414/4708/1 tuples=2
tx speed(Bps/kbps): 34948/279 rx speed(Bps/kbps): 35446/283
orgin->sink: org pre->in, reply out->post dev=7->48/48->7 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.55:58382->172.16.200.2:161(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.2:161->172.16.200.55:58382(0.0.0.0:0)
src_mac=00:0c:29:d6:12:20
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=0000249b tos=ff/01 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

b. Check the shaper information to verify the DSCP mark and bandwidth limitation:

```
# diagnose firewall shaper traffic-shaper list | grep test- -A 10
name test-shaper-300kbps
maximum-bandwidth 37 KB/sec
guaranteed-bandwidth 3 KB/sec
current-bandwidth 37 KB/sec
priority 2
policy 2
overhead 0
tos 01
packets dropped 10
bytes dropped 725
```

To configure DSCP and VLAN CoS for local-out traffic:

1. Configure the traffic shaper with bandwidth settings and the DSCP and VLAN CoS mark:

```
config firewall shaper traffic-shaper
    edit "test-shaper-600kbps"
        set guaranteed-bandwidth 60
        set maximum-bandwidth 600
        set per-policy enable
        set diffserv enable
        set cos-marking enable
        set cos 110
        set diffservcode 110000
    next
end
```

2. Configure the shaping policy for local-out traffic:

```
config firewall shaping-policy
    edit 5
        set traffic-type local-out
        set service "ALL"
        set traffic-shaper "test-shaper-600kbps"
        set class-id 5
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Verify that the shaper was successfully applied to the shaping policy:

```
# diagnose firewall iprope list 100019
policy index=5 uuid_idx=928 action=accept
flag (0):
schedule()
shapers: orig=test-shaper-600kbps(2/7500/75000)
cos_fwd=255  cos_rev=255
group=00100019 av=00000000 au=00000000 split=00000000
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 5
```

4. Test local-in traffic from the FortiGate to the remote PC.
   a. Check the session list:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=4 expire=3599 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=test-shaper-600kbps prio=2 guarantee 7500Bps max 75000Bps traffic
73557Bps drops 70500B
reply-shaper=
per_ip_shaper=
class_id=5 shaping_policy_id=5 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=6/255
state=log local os
statistic(bytes/packets/allow_err): org=85701/60/1 reply=2140/41/1 tuples=2
tx speed(Bps/kbps): 19172/153 rx speed(Bps/kbps): 478/3
orgin->sink: org out->post, reply pre->in dev=48->7/7->48 gwy=0.0.0.0/0.0.0.0
hook=out dir=org act=noop 172.16.200.2:23964->209.52.38.114:5201(0.0.0.0:0)
hook=in dir=reply act=noop 209.52.38.114:5201->172.16.200.2:23964(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5d:ed:fe
misc=0 policy_id=0 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000152f5 tos=30/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

   b. Check the shaper information to verify the DSCP mark and bandwidth limitation:

```
#  diagnose firewall shaper traffic-shaper list | grep test- -A 10
name test-shaper-600kbps
maximum-bandwidth 75 KB/sec
guaranteed-bandwidth 7 KB/sec
current-bandwidth 65 KB/sec
priority 2
policy 5
overhead 0
tos 30
packets dropped 5086
bytes dropped 1148949
```

Examples

This section includes the following traffic shaping configuration examples:

Interface-based traffic shaping profile

A traffic shaping policy can be used for interface-based traffic shaping by organizing traffic into 30 class IDs. The shaping profile defines the percentage of the interface bandwidth that is allocated to each class. Each traffic class ID is shaped to the assigned speed according to the outgoing bandwidth limit configured to the interface.

Traffic classification

A shaping policy classifies traffic and organizes it into different class IDs, based on matching criteria. For traffic matching a criteria, you can choose to put it into 30 different shaping classes, identified by class ID 2 - 31.

When NPU offloading is enabled on the NP6, SoC3, or SoC4 platforms, the class ID limit for egress traffic is 2 - 15. Setting the egress traffic class ID outside of these limits can result in unexpected behavior.

If NPU offloading is disabled, or enabled on the NP7 platform, the class ID limit for egress traffic is 2 - 31.

You must select an outgoing interface for the traffic. The shaping policy is only applied when the traffic goes to one of the selected outgoing interfaces.

Criterion	Description
Source	• Address: match the source address of the traffic to the selected address or address group. • User: use the user credentials of the traffic to match the selected user or user group. At least one address, address group, or internet service must also be selected. • Internet service: match the traffic to the selected internet service. Internet services cannot be used if addresses or address or groups are used.
Destination	• Address: match the destination address of the traffic to the selected address or address group. • Internet service: match the traffic to the selected internet service. Internet services cannot be used if addresses or address or groups are used.
Schedule	Match the current date and time to the selected schedule. You can select a one-time schedule, recurring schedule, or schedule group. This setting is optional.
Service	Match the service of the traffic to the selected service or service group.

Criterion	Description
Application	Match the application of the traffic to the selected application, application category, or application group. Application control must be enabled in the related firewall policy to know the application of the traffic. See Application control on page 1756 for more information.
URL category	Match the URL of the traffic to the selected URL category. Web filter must be enabled in the related firewall policy to know the URL of the traffic. See Web filter on page 1659 for more information.

When multiple items are selected in one criterion, it is considered a match when traffic matches any one of them.

Traffic prioritization

Shaping profiles define how different shaping classes of traffic are prioritized. For each class, you can define three prioritization strategies: guaranteed bandwidth, maximum bandwidth, and priority.

For each shaping profile, a default shaping class must be defined. Traffic is prioritized based on the default shaping group in the following two circumstances:

- All traffic to the outgoing interface that does not match to any shaping policy
- Traffic with a shaping group that is not defined in a shaping profile

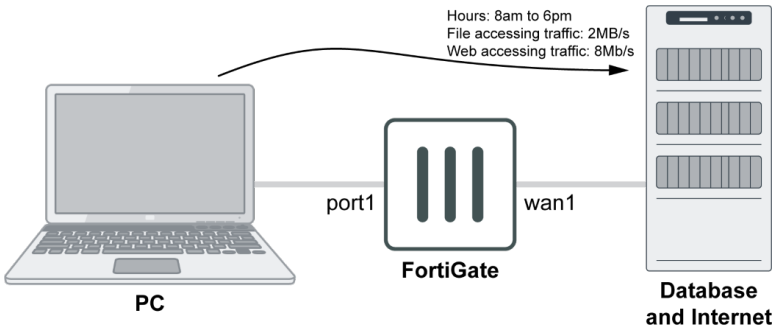
Prioritization strategy	Description
Guaranteed bandwidth	The percentage of the link speed that is reserved for the shaping group. The total guaranteed bandwidth for all shaping groups cannot exceed 100%.
Maximum bandwidth	The maximum percentage of the link speed that the shaping group can use.
Priority	The shaping class priority: top, critical, high, medium, or low. When groups are competing for bandwidth on the interface, the group with the higher priority wins.

Applying a shaping profile to an interface

Traffic shaping is accomplished by configuring the outgoing bandwidth and outgoing shaping profile on an interface. The shaping profile uses the outgoing bandwidth of the interface as the maximum link speed, and it only works when the outgoing bandwidth is configured.

This example shows how to apply interface-based traffic shaping to web and file accessing traffic according to a schedule:

- The link speed of the wan1 interface is 10 Mb/s.
- File access can use up to 2 Mb/s from 8:00 AM to 6:00 PM.
- Web access can use 8 Mb/s from 8:00 AM to 6:00 PM.



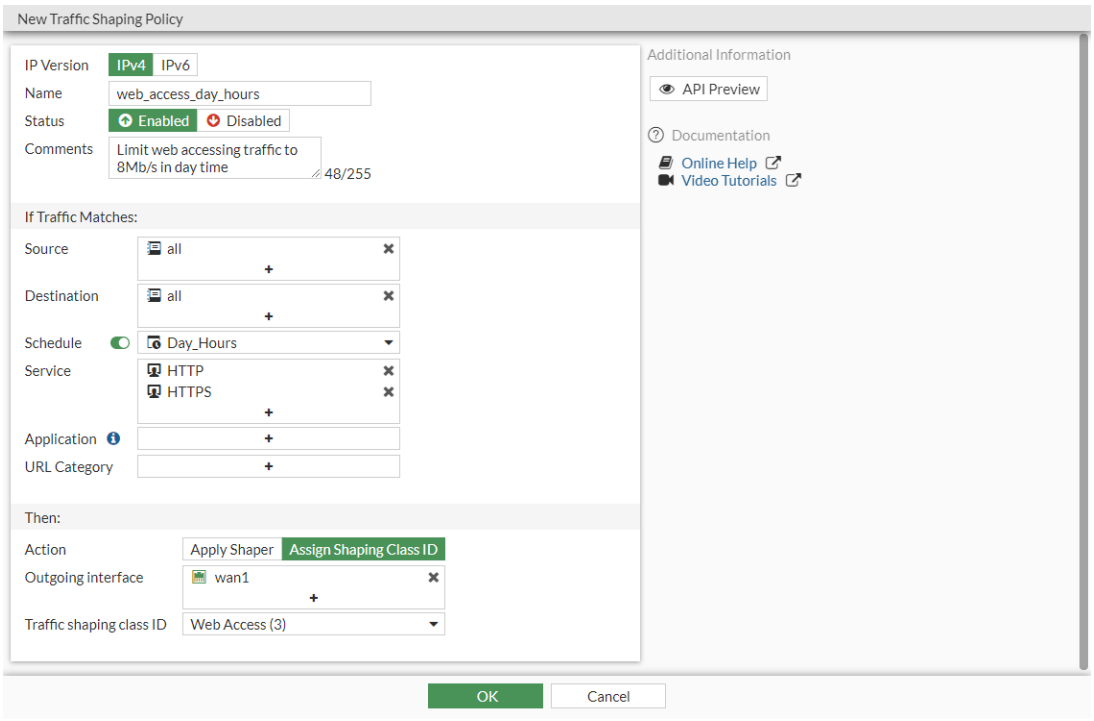
Putting the traffic into shaping classes

To create a recurring schedule in the GUI:

1. Go to *Policy & Objects* > *Schedules* and navigate to the *Recurring Schedule* tab.
2. Click *Create New*.
3. Configure a recurring schedule called *Day_Hours* for everyday from 8:00 AM to 6:00 PM.
4. Click *OK*.

To create a traffic shaping policy and class ID for the web accessing traffic in the GUI:

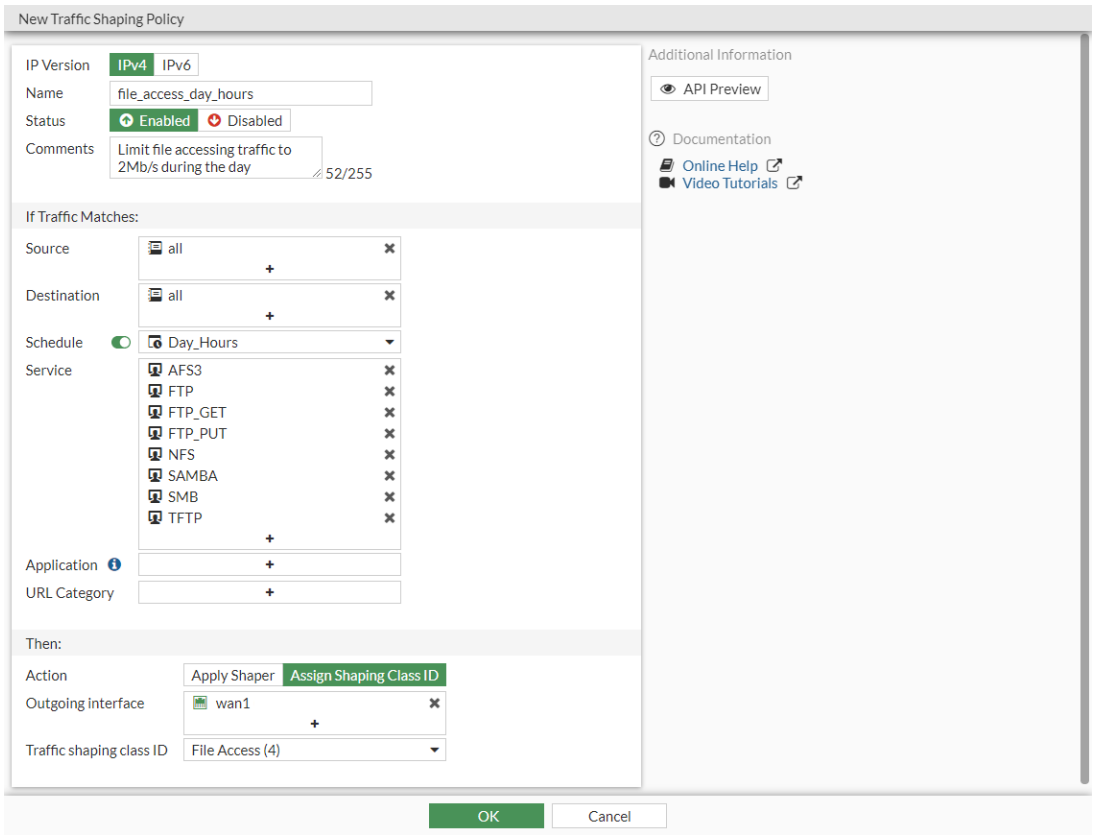
1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.
2. Enter a name for the policy, such as *web_access_day_hours*.
3. Enable *Schedule* and select the schedule you just created.
4. Set *Service* to web accessing services, such as *HTTP* and *HTTPS*.
5. Set *Action* to *Assign Shaping Class ID*, and *Outgoing interface* to *wan1*.
6. Click the *Traffic shaping class ID* drop down then click *Create*.
7. Enter an integer value for the *ID* (*3*) and a description for the *Name*, such as *Web Access*.
8. Click *OK*.
9. Select the class ID you just created for *Traffic shaping class ID*.



10. Configure the remaining settings as required.
11. Click *OK*.

To create a traffic shaping policy and class ID for the file accessing traffic in the GUI:

1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.
2. Enter a name for the policy, such as *file_access_day_hours*.
3. Enable *Schedule* and select the schedule you just created.
4. Set *Service* to file accessing services, such as *ASF3*, *FTP* and *SMB*.
5. Set *Action* to *Assign Shaping Class ID*, and *Outgoing interface* to *wan1*.
6. Click the *Traffic shaping class ID* drop down then click *Create*.
7. Enter an integer value for the *ID* (*4*) and a description for the *Name*, such as *File Access*.
8. Click *OK*.
9. Select the class ID you just created for *Traffic shaping class ID*.



10. Configure the remaining settings as required.
11. Click *OK*.

To put the traffic into shaping classes in the CLI:

1. Create a recurring schedule:

```
config firewall schedule recurring
    edit "Day_Hours"
        set start 08:00
        set end 18:00
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

2. Create the traffic class IDs:

```
config firewall traffic-class
    edit 3
        set class-name "Web Access"
    next
    edit 4
        set class-name "File Access"
    next
end
```

3. Create the web and file accessing traffic shaping policies:

```
config firewall shaping-policy
    edit 2
        set name "web_access_day_hours"
        set comment "Limit web accessing traffic to 8Mb/s in day time"
        set service "HTTP" "HTTPS"
        set schedule "Day_Hours"
        set dstintf "wan1"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
    edit 3
        set name "file_access_day_hours"
        set comment "Limit file accessing traffic to 2Mb/s during the day"
        set service "AFS3" "FTP" "FTP_GET" "FTP_PUT" "NFS" "SAMBA" "SMB" "TFTP"
        set schedule "Day_Hours"
        set dstintf "wan1"
        set class-id 4
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Allocating bandwidth to the shaping classes

A traffic shaping profile defines the guaranteed and maximum bandwidths each class receives. In this example, file access can use up to 2 Mb/s and web access can use 8 Mb/s from 8:00 AM to 6:00 PM.

To create a traffic shaping profile using the GUI:

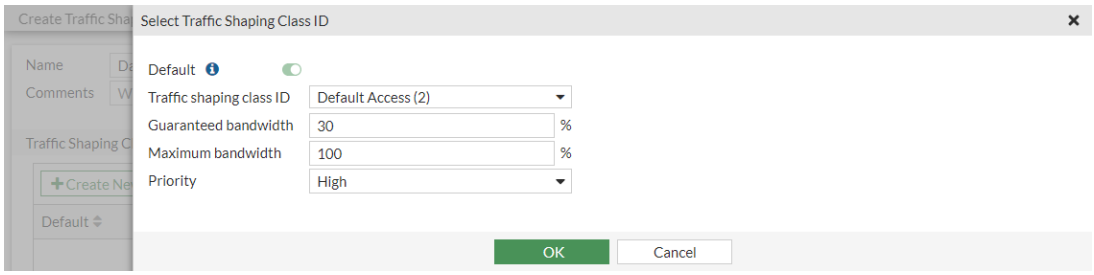
1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Profiles* tab, and click *Create New*.
2. Enter a name for the profile, such as *Day_Hours_Profile*.
3. Configure a default traffic shaping class:

   This class has a high priority, meaning that when the other classes have reached their guaranteed bandwidths, this default class will use the rest of the available bandwidth.

   a. In the *Traffic Shaping Classes* table click *Create New*.
   b. Click the *Traffic shaping class ID* drop down then click *Create*.
   c. Enter a name for the class, such as *Default Access*.
   d. Click *OK*.
   e. Select the class ID you just created for *Traffic shaping class ID*.

f. Configure the following settings, then click *OK*:

Guaranteed bandwidth	30
Maximum bandwidth	100
Priority	High



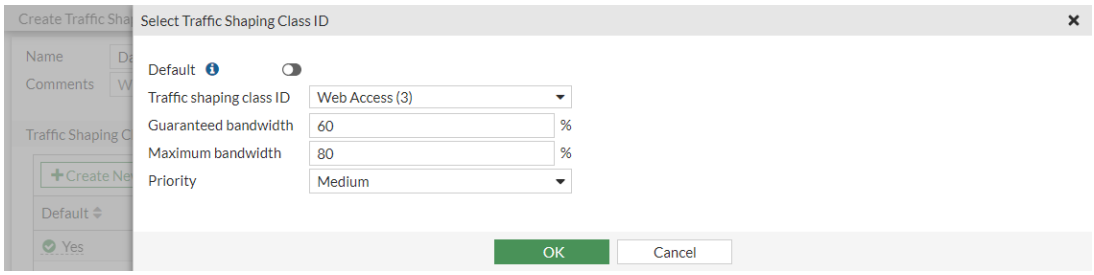
4. Configure a web accessing traffic shaping class:

   When other types of traffic are competing for bandwidth, this class is guaranteed to 6 Mb/s, or 60% of the bandwidth.

   a. In the *Traffic Shaping Classes* table click *Create New*.

   b. Configure the following settings, then click *OK*:

Traffic shaping class ID	Web Access
Guaranteed bandwidth	60
Maximum bandwidth	80
Priority	Medium



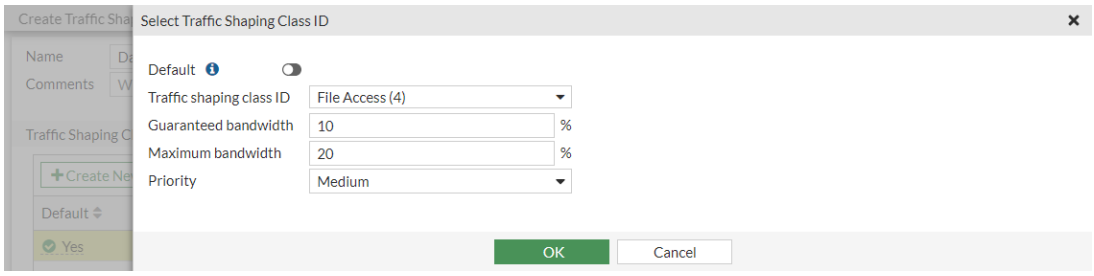
5. Configure a file accessing traffic shaping class:

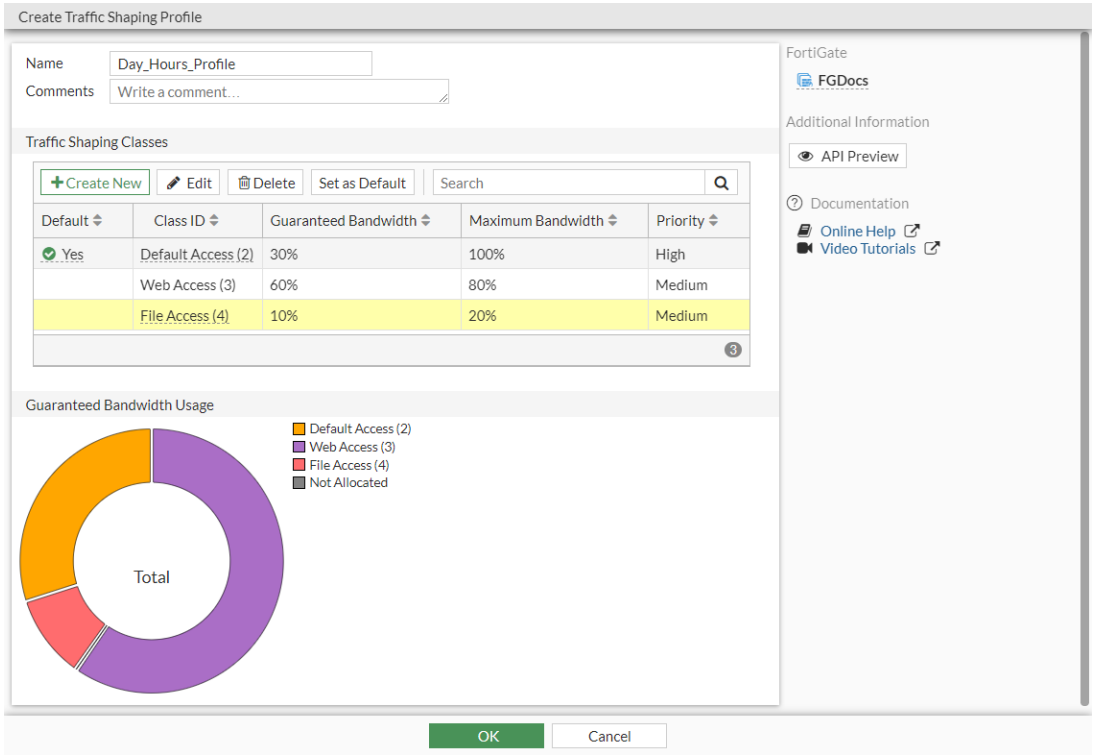
   When other types of traffic are competing for bandwidth, this group is guaranteed to 1 Mb/s, or 10% of the bandwidth.

   a. In the *Traffic Shaping Classes* table click *Create New*.

   b. Configure the following settings, then click *OK*:

Traffic shaping class ID	File Access
Guaranteed bandwidth	10
Maximum bandwidth	20
Priority	Medium





6. Click *OK*.

To create a traffic shaping profile using the CLI:

```
config firewall shaping-profile
    edit "Day_Hours_Profile"
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 100
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 60
                set maximum-bandwidth-percentage 80
            next
            edit 3
                set class-id 4
```

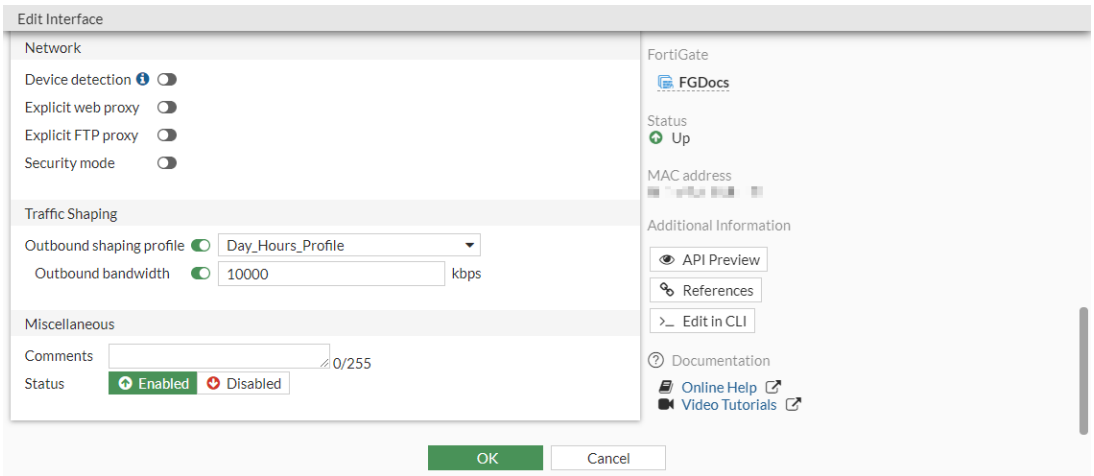
```
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 20
            next
        end
    next
end
```

Defining the available bandwidth on an interface

In this example, the link speed of the wan1 interface is 10 Mb/s.

To set the bandwidth of the wan1 interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit the wan1 interface.
3. Under Traffic Shaping, enable *Outbound shaping profile* and select the profile that you just created, *Day_Hours_Profile*.
4. Enable *Outbound Bandwidth* and set it to *10000* Kbps.



5. Click *OK*.

To set the bandwidth of the wan1 interface in the CLI:

```
config system interface
    edit "wan1"
        set egress-shaping-profile "Day_Hours_Profile"
        set outbandwidth 10000
    next
end
```

Diagnose commands

To check that the specific traffic is put into the correct shaping group or class ID:

```
# diagnose firewall iprope list 100015
```

To check the speed limit for each class ID on an interface:

```
# diagnose netlink interface list wan1
```

Interface-based traffic shaping with NP acceleration

Interface-based traffic shaping with NP acceleration is supported on some devices.

An administrator configures the WAN interface's maximum outbound bandwidth and, based on that, creates a traffic shaping profile with a percentage based shaper. This allows for proper QoS and traffic shaping. VLAN interfaces are not supported.

This feature is supported on FortiGate 600E, 500E, 400E, and 300E models.

To configure interface-based traffic shaping:

1. Enable NPU offloading when doing interface-based traffic shaping according to the egress-shaping-profile:

```
config system npu
    set intf-shaping-offload enable
end
```

2. Configure shaping profiles:

```
config firewall shaping-profile
    edit "sdwan"
        set default-class-id 4
        config shaping-entries
            edit 1
                set class-id 4
                set guaranteed-bandwidth-percentage 3
                set maximum-bandwidth-percentage 5
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 50
                set maximum-bandwidth-percentage 100
            next
            edit 3
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 1
                set maximum-bandwidth-percentage 5
            next
        end
    next
end
```

   The class number is limited to 16.

3. Configure a traffic shaper and shaping policy:

```
config firewall shaper traffic-shaper
    edit "Transactional"
        set priority medium
    next
end

config firewall shaping-policy
    edit 1
        set service "ALL"
        set dstintf "any"
        set traffic-shaper "Transactional"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

4. Apply the egress shaping profile on the interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.1.100.23 255.255.255.0
        set allowaccess ping
        set type physical
        set outbandwidth 500
        set egress-shaping-profile "sdwan"
        set snmp-index 4
    next
end
```

5. Configure a firewall policy:

```
config firewall policy
    edit 3
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

QoS assignment and rate limiting for FortiSwitch quarantined VLANs

When devices are quarantined, they are isolated from the rest of the network. However, they can still impact the network if not controlled beyond isolation. A quarantined host, which offers heavy traffic, could congest the network and create a DOS-style reduction in service to authorized hosts.

Within the quarantined VLAN, two restrictions are available within the network:

- Traffic policing (also known as rate limiting)
- QoS (Quality of Service) assignment (also known as priority assignment)

Each quarantined host's traffic can be subject to rate limiting and priority adjustment. This reduces the impact that any quarantined host can have on authorized traffic on the network.

To configure QoS assignment and rate limiting for quarantined VLANs:

1. Configure a traffic policy, or use the default "quarantine" policy:

```
config switch-controller traffic-policy
    edit "quarantine"
        set description "Rate control for quarantined traffic"
        set guaranteed-bandwidth 163840
        set guaranteed-burst 8192
        set maximum-burst 163840
        set cos-queue 0
    next
end
```

2. Configure an interface:

```
config system interface
    edit "qtn.aggr1"
        set vdom "root"
        set ip 10.254.254.254 255.255.255.0
        set description "Quarantine VLAN"
        set security-mode captive-portal
        set replacemsg-override-group "auth-intf-qtn.aggr1"
        set device-identification enable
        set snmp-index 30
        set switch-controller-access-vlan enable
        set switch-controller-traffic-policy "quarantine"
        set color 6
        set interface "aggr1"
        set vlanid 4093
    next
end
```

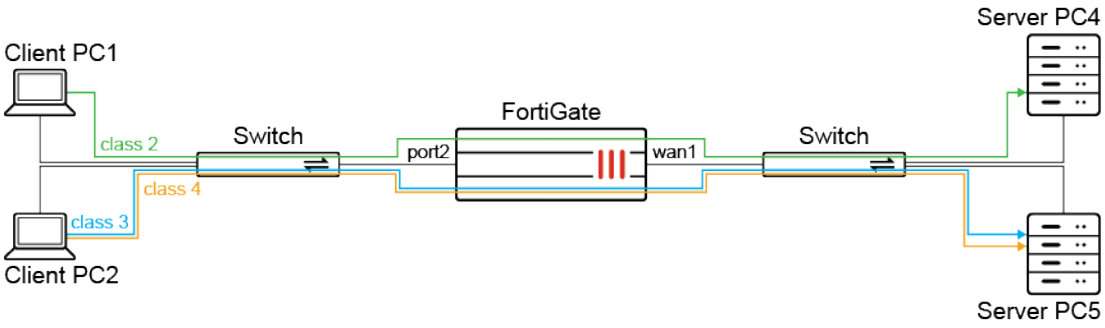
   By default, `switch-controller-traffic-policy` is empty. You need to apply the necessary traffic policy (not only limited to "quarantine").

Ingress traffic shaping profile

A traffic shaping profile can be applied to an interface for traffic in the ingress direction. Similar to an egress traffic shaping profile, the guaranteed bandwidth and priority of the profile will be respected when an interface receives inbound traffic. When congestion occurs, any remaining bandwidth will be allotted to classes based on priority.

Example

In this example, the port2 interface has a total inbound bandwidth of 100 Mbps. Traffic from certain clients to certain servers are assigned different classes.



IPv6 traffic from any client PCs to server PCs is assigned class 5.

For each class, the priority, guaranteed bandwidth, and maximum bandwidth are as follows:

Class	Priority	Guaranteed bandwidth	Maximum bandwidth
2	Low	10%	60%
3	High	20%	100%
4	High	30%	100%
5	Medium	10%	50%

Bandwidth will first be allotted to each class according to its guaranteed bandwidth. Then remaining available bandwidth will be allotted to class 3 and 4 first based on their priority. The allocation will be proportional to their guaranteed bandwidth ratio.

To configure ingress traffic shaping:

1. Configure the client and server addresses:

```
config firewall address
    edit "pc1"
        set subnet 10.1.100.11 255.255.255.255
    next
    edit "pc2"
        set subnet 10.1.100.22 255.255.255.255
    next
    edit "pc4"
        set subnet 172.16.200.44 255.255.255.255
    next
    edit "pc5"
        set subnet 172.16.200.55 255.255.255.255
    next
end
```

2. Configure the class IDs:

```
config firewall traffic-class
    edit 2
        set class-name "class2"
    next
    edit 3
        set class-name "class3"
    next
```

```
    edit 4
        set class-name "class4"
    next
    edit 4
        set class-name "class5"
    next
end
```

3. Configure traffic shaping policies to assign classes to each group of traffic.
   a. Configure a policy to assign traffic from PC1 to PC4 in class 2:

```
config firewall shaping-policy
    edit 1
        set name "shaping policy 1"
        set service "ALL"
        set dstintf "wan1"
        set class-id 2
        set srcaddr "pc1"
        set dstaddr "pc4"
    next
end
```

   b. Configure a policy to assign traffic from PC2 to PC4 in class 3:

```
config firewall shaping-policy
    edit 2
        set name "shaping policy 2"
        set service "ALL"
        set dstintf "wan1"
        set class-id 3
        set srcaddr "pc2"
        set dstaddr "pc4"
    next
end
```

   c. Configure a policy to assign traffic from PC2 to PC5 in class 4:

```
config firewall shaping-policy
    edit 3
        set name "shaping policy 3"
        set service "ALL"
        set dstintf "wan1"
        set class-id 4
        set srcaddr "pc2"
        set dstaddr "pc5"
    next
end
```

   d. Configure a policy to assign all IPv6 traffic to class 5:

```
config firewall shaping-policy
    edit 4
        set name "shaping policy 4"
        set ip-version 6
        set service "ALL"
        set dstintf "wan1"
        set class-id 5
        set srcaddr6 "all"
```

```
            set dstaddr6 "all"
        next
    end
```

4. Configure a shaping profile to set the priority, and the guaranteed and maximum bandwidth percentages for each class:

```
config firewall shaping-profile
    edit "ingShapeProfile"
        set default-class-id 2
        config shaping-entries
            edit 2
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 60
            next
            edit 3
                set class-id 3
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 100
            next
            edit 4
                set class-id 4
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 100
            next
            edit 5
                set class-id 5
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 50
            next
        end
    next
end
```

5. Configure the inbandwidth and apply the ingress shaping profile on port2:

```
config system interface
    edit "port2"
        set ip 10.1.100.1 255.255.255.0
        set inbandwidth 100000
        set ingress-shaping-profile "ingShapeProfile"
        config ipv6
            set ip6-address 2000:10:1:100::1/64
        end
    next
end
```

Inbandwidth must be configured for traffic shaping to take effect.

6. Configure a firewall policy to allow traffic to go through. Since traffic shaping is for inbound traffic on port2, the policy is defined from port2 to wan1:

```
config firewall policy
    edit 2
        set srcintf "port2"
```

```
            set dstintf "wan1"
            set action accept
            set srcaddr "all"
            set dstaddr "all"
            set srcaddr6 "all"
            set dstaddr6 "all"
            set schedule "always"
            set service "ALL"
            set logtraffic all
            set auto-asic-offload enable
            set nat enable
        next
    end
```

Ingress traffic shaping supports NPU offloading and it is enabled by default. Set `auto-asic-offload` to `disable` to disable it.

Verifying that the traffic is being shaped

In each of the following cases, the server PCs (PC4 and PC5) are configured as iPerf servers. The client PCs (PC1 and PC2) are configured as iPerf clients. The client sends traffic to the server from the client to server direction, triggering inbound traffic shaping on the port2 interface. The inbound bandwidth on port2 is 100 Mbps.

Case 1: single stream, PC1 to PC4

Traffic is sent from PC1 to PC4. There is no other traffic. Traffic is marked with class ID 2 and allocated the maximum bandwidth 60 Mbps (60%).

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=25 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=50 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=60000(kbps)         guaranteed-bandwidth=10000
(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=60002(kbps)
                        priority=low    forwarded_bytes=58157K
                        dropped_packets=94K     dropped_bytes=125385K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=15000(kbps)         guaranteed-bandwidth=20000
(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=24000(kbps)         guaranteed-bandwidth=30000
(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
stat: rxp=173465879 txp=2430534 rxb=194665548609 txb=2767375732 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628814469
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
```

```
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=25
```

Case 2: dual stream, PC1 to PC4, PC2 to PC4

Traffic is sent from both PC1 and PC2 to PC4. PC1 to PC4 traffic is marked with class ID 2 and low priority, and PC2 to PC4 traffic is marked with class ID 3 and high priority. Both class 2 and 3 will be allocated their guaranteed bandwidth first, using up 10% and 20% respectively. The remaining available bandwidth is used by class 3 since it has a higher priority. Class 2 uses around 10 Mbps, and class 3 uses around 90 Mbps.

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=181 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=10000(kbps)         guaranteed-bandwidth=10000
(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=10001(kbps)
                        priority=low    forwarded_bytes=1799482K
                        dropped_packets=5998K   dropped_bytes=7965553K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=88000(kbps)         guaranteed-bandwidth=20000
(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=88000(kbps)
                        priority=high   forwarded_bytes=345039K
                        dropped_packets=324K    dropped_bytes=430862K
        class-id=4      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=30000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
stat: rxp=181269891 txp=2433428 rxb=205136511596 txb=2771214402 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628815849
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36
```

Case 3: multiple streams

Multiple streams of traffic are sent at the same time:

- PC1 to PC4 traffic is assigned class 2 with low priority, and a guaranteed bandwidth of 10 Mbps.
- PC2 to PC4 traffic is assigned class 3 with high priority, and a guaranteed bandwidth of 20 Mbps.
- PC2 to PC5 traffic is assigned class 4 with high priority, and a guaranteed bandwidth of 30 Mbps.

All classes will be allocated their guaranteed bandwidth first, using up 10 Mbps, 20 Mbps, and 30 Mbps respectively. The remaining available bandwidth (40 Mbps) is shared by class 3 and class 4 based on their guaranteed bandwidth ratio of 20:30.

- Class 3’s share of the remaining 40 Mbps traffic = 40 × 20/(20 + 30) =16 Mpbs
- Class 4’s share of the remaining 40 Mbps traffic = 40 × 30/(20 + 30) =24 Mpbs

Each class is allocated roughly the following bandwidth:

- Class 2: 10 Mbps
- Class 3: 20 Mbps + 16 Mbps = 36 Mbps
- Class 4: 30 Mbps + 24 Mbps = 54 Mbps

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=27 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=148731 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=10000(kbps)         guaranteed-bandwidth=10000
(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=10004(kbps)
                        priority=low    forwarded_bytes=2267956K
                        dropped_packets=10389K  dropped_bytes=13796469K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=35000(kbps)         guaranteed-bandwidth=20000
(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=35729(kbps)
                        priority=high   forwarded_bytes=2119502K
                        dropped_packets=6020K   dropped_bytes=7994926K
        class-id=4      allocated-bandwidth=54000(kbps)         guaranteed-bandwidth=30000
(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=53907(kbps)
                        priority=high   forwarded_bytes=902415K
                        dropped_packets=4141K   dropped_bytes=5499248K
stat: rxp=197827723 txp=2433885 rxb=227356779526 txb=2771602657 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628816440
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=27
```

Internet Services

The following topics provide instructions on configuring policies with Internet Service:

- Internet Service Database on-demand mode on page 1593
- Enabling the ISDB cache in the FortiOS kernel on page 1596

Using Internet Service in a policy

This topic shows how to apply a predefined Internet Service entry into a policy.

The Internet Service Database is a comprehensive public IP address database that combines IP address range, IP owner, service port number, and IP security credibility. The data comes from the FortiGuard service system. Information is regularly added to this database, for example, geographic location, IP reputation, popularity & DNS, and so on. All this information helps users define Internet security more effectively. You can use the contents of the database as criteria for inclusion or exclusion in a policy.

From FortiOS version 5.6, Internet Service is included in the firewall policy. It can be applied to a policy only as a destination object. From version 6.0, Internet Service can be applied both as source and destination objects in a policy. You can also apply Internet Services to shaping policy.

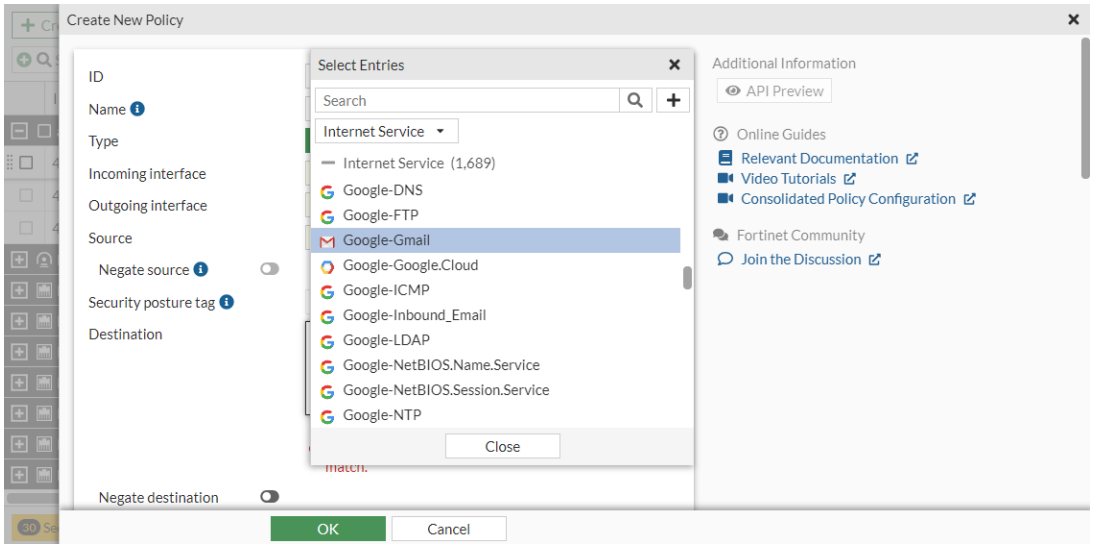
There are three types of Internet Services you can apply to a firewall policy:

- Predefined Internet Services
- Custom Internet Services
- Extension Internet Services

Sample IPv4 configuration

To apply a predefined Internet Service entry to a policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Click in the *Destination* field.
3. In the *Select Entries* pane, select *Internet Service* from the dropdown list and select *Google-Gmail*.



4. Configure the remaining fields as needed.
5. Click *OK*.

To apply a predefined Internet Service entry to a policy in the CLI:

In the CLI, enable the `internet-service` first and then use its ID to apply the policy.

This example uses Google Gmail and its ID is 65646. Each Internet Service has a unique ID.

```
config firewall policy
    edit 9
        set name "Internet Service in Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

To diagnose an Internet Service entry in the CLI:

```
# diagnose internet-service id-summary 65646
Version: 0000600096
Timestamp: 201902111802
Total number of IP ranges: 444727
Number of Groups: 7
Group(0), Singularity(20), Number of IP ranges(142740)
Group(1), Singularity(19), Number of IP ranges(1210)
Group(2), Singularity(16), Number of IP ranges(241)
Group(3), Singularity(15), Number of IP ranges(38723)
Group(4), Singularity(10), Number of IP ranges(142586)
Group(5), Singularity(8), Number of IP ranges(5336)
Group(6), Singularity(6), Number of IP ranges(113891)
Internet Service: 65646(Google.Gmail)
Number of IP range: 60
Number of IP numbers: 322845
Singularity: 15
Reputation: 5(Known and verified safe sites such as Gmail, Amazon, eBay, etc.)
Icon Id: 510
Second Level Domain: 53(gmail.com)
Direction: dst
Data source: isdb
```

Result

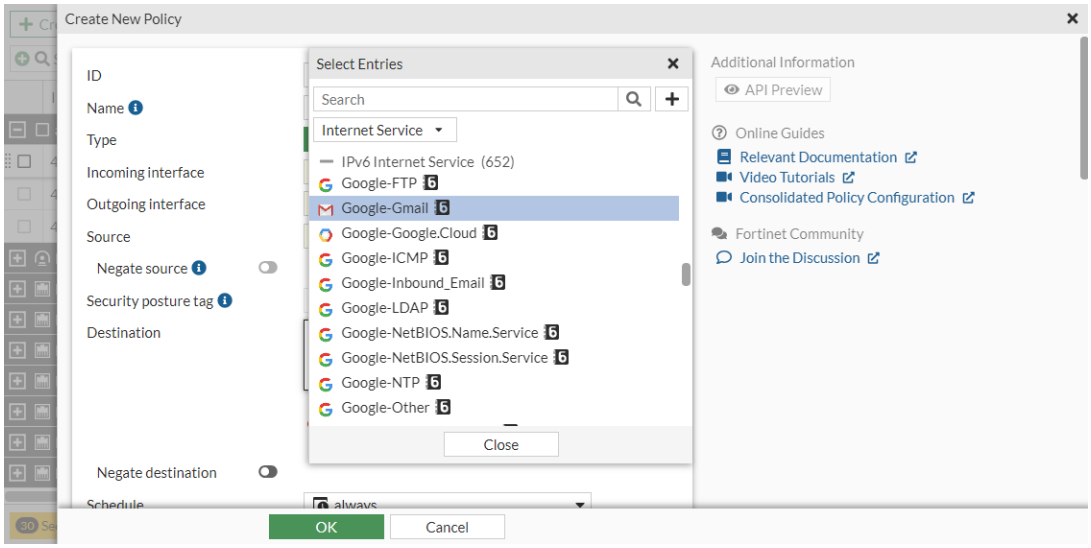
Because the IP and services related to Google Gmail on the Internet are included in this Internet Service (65646), all traffic to Google Gmail is forwarded by this policy.

Sample IPv6 configuration

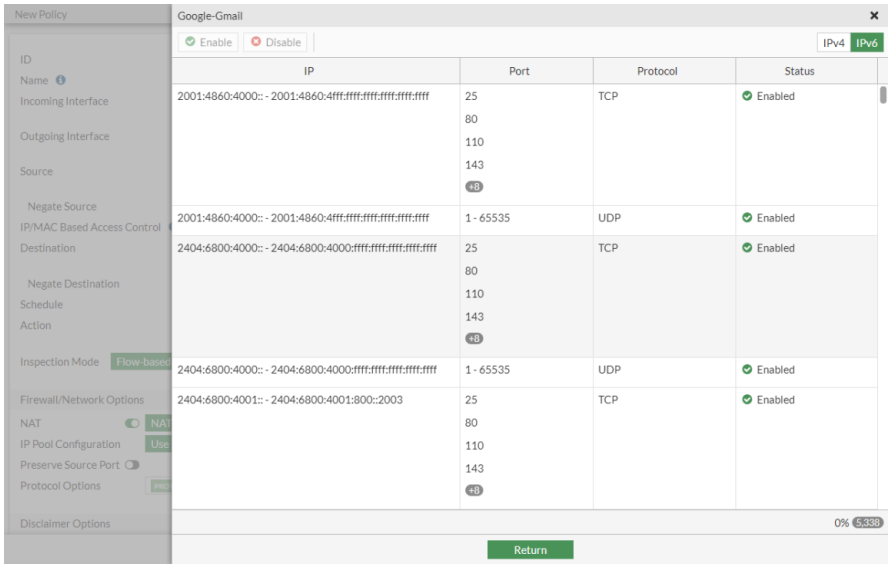
In this example, the Google Gmail IPv6 ISDB address (ID 65646) is used as a destination in a firewall policy.

To apply a predefined IPv6 Internet Service entry to a policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. In the *Destination* field, select *Internet Service* from the dropdown list.
3. In the *IPv6 Internet Service* section, select *Google-Gmail*.



4. Optionally, hover over the *Google Gmail* and click *View/Edit Entries*. A pane appears that displays the IPv6 address ranges for this Internet Service.



5. Click *Return* to close the pane.
6. Configure the other settings as needed.
7. Click *OK*.

To apply a predefined IPv6 Internet Service entry to a policy using the CLI:

```
config firewall policy
    edit 4
        set name "Internet Service6 policy"
```

```
        set srcintf "vlan100"
        set dstintf "wan1"
        set action accept
        set srcaddr6 "all"
        set internet-service6 enable
        set internet-service6-name "Google-Gmail"
        set schedule "always"
        set nat enable
    next
end
```

To diagnose an IPv6 Internet Service entry in the CLI:

```
# diagnose internet-service6 id-summary 65646

Version: 00007.02907
Timestamp: 202212161345
Total number of IP ranges: 36878
Number of Groups: 12
Group(0), Singularity(20), Number of IP ranges(60)
Group(1), Singularity(18), Number of IP ranges(12)
Group(2), Singularity(17), Number of IP ranges(2728)
Group(3), Singularity(16), Number of IP ranges(2812)
Group(4), Singularity(15), Number of IP ranges(4011)
Group(5), Singularity(10), Number of IP ranges(2345)
Group(6), Singularity(9), Number of IP ranges(14)
Group(7), Singularity(8), Number of IP ranges(1555)
Group(8), Singularity(7), Number of IP ranges(2704)
Group(9), Singularity(6), Number of IP ranges(7300)
Group(10), Singularity(5), Number of IP ranges(3154)
Group(11), Singularity(4), Number of IP ranges(10183)
Internet Service: 65646(Google-Gmail)
Number of IP ranges: 482
Singularity: 15
Icon Id: 510
Direction: both
Data source: isdb
Country: 32 36 56 76 124 152 158 203 208 246 250 276 344 348 356 372 376 380 392 404 458 484
        528 616 634 643 682 702 710 724 752 756 784 826 840
Region: 65535
City: 65535
```

Result

Because the IP and services related to Google Gmail on the Internet are included in this Internet Service (65646), all traffic to Google Gmail is forwarded by this policy.

Using custom Internet Service in policy

Custom Internet Services can be created and used in firewall policies.

When creating a custom Internet Service, you must set following elements:

- IP or IP ranges
- Protocol number
- Port or port ranges
- Reputation

You must use CLI to create a custom Internet Service, except for geographic based services (see Allow creation of ISDB objects with regional information on page 1589).

CLI syntax

```
config firewall internet-service-custom
    edit <name>
       set comment <comment>
       set reputation {1 | 2 | 3 | 4 | 5}
       config entry
           edit <ID>
               set protocol <protocol #>
               set dst <object_name>
               config port-range
                   edit <ID>
                       set start-port <port #>
                       set end-port <port #>
                   next
               end
           next
       end
    end
end
```

Sample configuration

To configure a custom Internet Service:

```
config firewall internet-service-custom
   edit "test-isdb-1"
       set comment "Test Custom Internet Service"
       set reputation 4
       config entry
          edit 1
              set protocol 6
              config port-range
                  edit 1
                      set start-port 80
                      set end-port 443
                  next
              end
              set dst "10-1-100-0"
          next
          edit 2
              set protocol 6
              config port-range
                  edit 1
                      set start-port 80
```

```
                    set end-port 80
                next
            end
            set dst "172-16-200-0"
        next
    end
next
end
```

To apply a custom Internet Service into a policy:

```
config firewall policy
    edit 1
        set name "Internet Service in Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
        set internet-service-custom "test-isdb-1"
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Result

In addition to the IP address, IP address ranges, and services allowed by Google.Gmail, this policy also allows the traffic which access to 10.1.100.0/24 and TCP/80-443 and 172.16.200.0/24 and TCP/80.

Using extension Internet Service in policy

Extension Internet Service lets you add custom or remove existing IP address and port ranges to an existing predefined Internet Service entries. Using an extension type Internet Service is actually editing a predefined type Internet Service entry and adding IP address and port ranges to it.

When creating an extension Internet Service and adding custom ranges, you must set following elements:

- IP or IP ranges
- Protocol number
- Port or port ranges

You must use CLI to add custom IP address and port entries into a predefined Internet Service.

You must use GUI to remove entries from a predefined Internet Service.

Custom extension Internet Service CLI syntax

```
config firewall internet-service-extension
    edit <ID #>
        set comment <comment>
        config entry
            edit <ID #>
                set protocol <number #>
                set dst <object_name>
                config port-range
                    edit <ID #>
                         set start-port <number #>
                         set end-port <number #>
                    next
                end
            next
        end
    end
end
```

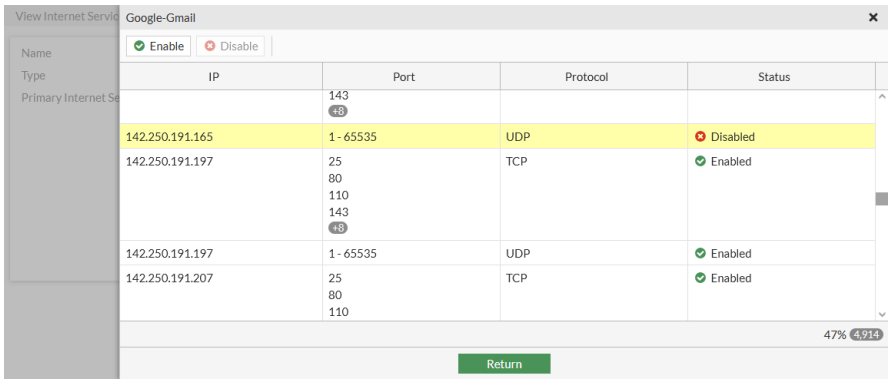
Sample configuration

To configure an extension Internet Service in the CLI:

```
config firewall internet-service-extension
    edit 65646
        set comment "Test Extension Internet Service 65646"
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 80
                        set end-port 443
                    next
                end
                set dst "172-16-200-0"
            next
            edit 2
                set protocol 17
                config port-range
                    edit 1
                        set start-port 53
                        set end-port 53
                    next
                end
                set dst "10-1-100-0"
            next
        end
    next
end
```

To remove IP address and port entries from an existing Internet Service in the GUI:

1. Go to *Policy & Objects* > *Internet Service Database*.
2. Search for *Google-Gmail*.
3. Select *Google-Gmail* and click *Edit*.
4. In the gutter, click *View/Edit Entries*.
5. Select the *IP* entry that you need to remove and click *Disable*.



6. Click *Return* twice.

To remove IP address and port entries from an existing Internet Service in the CLI:

```
config firewall internet-service-extension
    edit 65646
        config disable-entry
            edit 1
                set protocol 17
                config port-range
                    edit 1
                    next
                end
                config ip-range
                    edit 1
                        set start-ip 142.250.191.165
                        set end-ip 142.250.191.165
                    next
                end
            next
        end
    next
end
```

To apply an extension Internet Service into policy in the CLI:

```
config firewall policy
    edit 9
        set name "Internet Service in Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
```

```
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Result

In addition to the IP addresses, IP address ranges, and services allowed by Google.Gmail, this policy also allows the traffic which accesses 10.1.100.0/24 and UDP/53 and 172.16.200.0/24 and TCP/80-443. At the same time, the traffic that accesses 2.20.183.160 is dropped because this IP address and port is disabled from Google.Gmail.

Global IP address information database

The Internet Service and IP Reputation databases download details about public IP address, including: ownership, known services, geographic location, blocklisting information, and more. The details are available in drilldown information, tooltips, and other mechanisms in the FortiView and other pages.

The global IP address database is an integrated database containing all public IP addresses, and is implemented in the Internet Service Database.

To view the owner of the IP address:

```
(global) # get firewall internet-service-owner ?
    id    Internet Service owner ID.
    1  Google
    2  Facebook
    3  Apple
    4  Yahoo
    5  Microsoft
    ......
    115  Cybozu
    116  VNC
```

To check for any known service running on an IP address:

```
(global) # diagnose internet-service info FG-traffic 6 80 8.8.8.8
    Internet Service: 65537(Google.Web)
```

To check GeoIP location and blocklist information:

```
(global) # diagnose internet-service id 65537 | grep 8.8.8.8
    8.8.8.8-8.8.8.8 geo_id(11337) block list(0x0) proto(6) port(80 443)
    8.8.8.8-8.8.8.8 geo_id(11337) block list(0x0) proto(17) port(443)
```

To check a known malicious server:

```
(global) # diagnose internet-service id-summary 3080383
    Version: 0000600096
    Timestamp: 201902111802
```

```
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 3080383(Botnet.C&C.Server)
    Number of IP range: 111486
    Number of IP numbers: 111486
    Singularity: 20
    Reputation: 1(Known malicious sites related to botnet servers, phishing sites, etc.)
    Icon Id: 591
    Second Level Domain: 1(other)
    Direction: dst
    Data source: irdb
```

To check questionable usage:

```
(global) # diagnose internet-service id-summary 2818238
    Version: 0000600096
    Timestamp: 201902111802
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 2818238(Tor.Relay.Node)
    Number of IP range: 13718
    Number of IP numbers: 13718
    Singularity: 20
    Reputation: 2(Sites providing high risk services such as TOR, proxy, P2P, etc.)
    Icon Id: 43
    Second Level Domain: 1(other)
    Direction: dst
    Data source: irdb

(global) # diagnose internet-service id-summary 2818243
    Version: 0000600096
    Timestamp: 201902111802
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 2818243(Tor.Exit.Node)
```

```
Number of IP range: 1210
Number of IP numbers: 1210
Singularity: 19
Reputation: 2(Sites providing high risk services such as TOR, proxy, P2P, etc.)
Icon Id: 43
Second Level Domain: 1(other)
Direction: src
Data source: irdb
```

IP reputation filtering

There are currently five reputation levels in the Internet Service Database (ISDB), and custom reputation levels can be defined in a custom internet service. You can configure firewall policies to filter traffic according to the desired reputation level. If the reputation level of either the source or destination IP address is equal to or greater than the level set in the policy, then the packet is forwarded, otherwise, the packet is dropped.

The five default reputation levels are:

1	Known malicious sites, such as phishing sites or sites related to botnet servers
2	High risk services sites, such as TOR, proxy, and P2P
3	Unverified sites
4	Reputable social media sites, such as Facebook and Twitter
5	Known and verified safe sites, such as Gmail, Amazon, and eBay

The default minimum reputation level in a policy is zero, meaning that the reputation filter is disabled.

For IP addresses that are not included in the ISDB, the default reputation level is three.

The default reputation direction is `destination`.

Example 1

Packets from the source IP address with reputation levels three, four, or five will be forwarded by this policy.

To set the reputation level and direction in a policy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "port1"
        set srcaddr “all”
        set dstaddr "all"
        set reputation-minimum 3
        set reputation-direction source
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
```

```
        set nat enable
    next
end
```

Packets from the source IP address with reputation levels three, four, or five will be forwarded by this policy.

Example 2

This policy allows only outbound FTP traffic, if the destination server has a minimum reputation of 4.

To set the reputation level and direction in a policy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "wan2"
        set srcaddr "all"
        set dstaddr "all"
        set reputation-minimum 4
        set reputation-direction destination
        set action accept
        set schedule "always"
        set service "FTP"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Internet service groups in policies

This feature provides support for Internet Service Groups in traffic shaping and firewall policies. Service groups can be used as the source and destination of the policy. Internet Service Groups are used as criteria to match traffic; the shaper will be applied when the traffic matches.

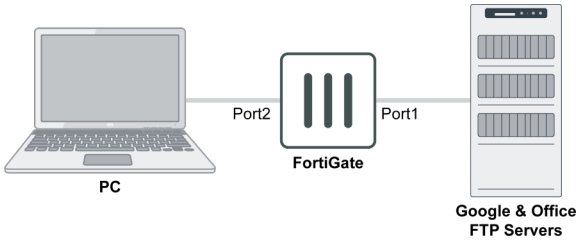
To use a group as a destination, `internet-service` must be enabled. To use a group as a source, `internet-service-src` must be enabled.

The following CLI variables are available in the `firewall policy` and `firewall shaping-policy` commands:

Variable	Description
`internet-service-group <string>`	Internet Service group name.
`internet-service-custom-group <string>`	Custom Internet Service group name.
`internet-service-src-group <string>`	Internet Service source group name.
`internet-service-src-custom-group <string>`	Custom Internet Service source group name.

Examples

The following examples use the below topology.



Example 1

In this example, the PC is allowed to access Google, so all Google services are put into an Internet Service Group.

To configure access to Google services using an Internet Service Group using the CLI:

1. Create a Service Group:

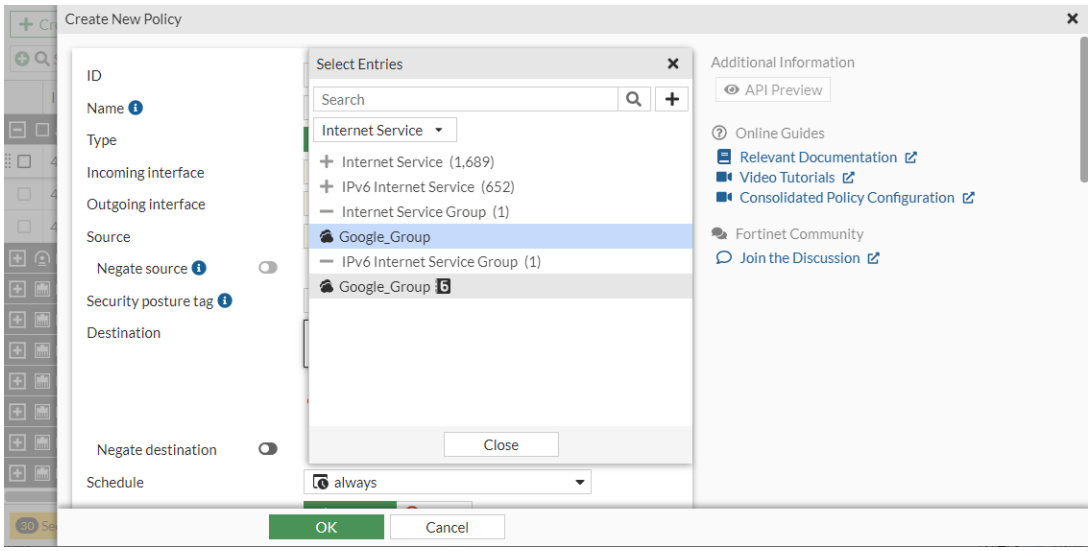
```
config firewall internet-service-group
    edit "Google_Group"
        set direction destination
        set member Google-Other Google-Web Google-ICMP Google-DNS Google-Outbound_Email
Google-SSH Google-FTP Google-NTP Google-Inbound_Email Google-LDAP Google-
NetBIOS.Session.Service Google-RTMP Google-NetBIOS.Name.Service Google-Google.Cloud
Google-Gmail
    next
end
```

2. Create a firewall policy to allow access to all Google Services from the PC:

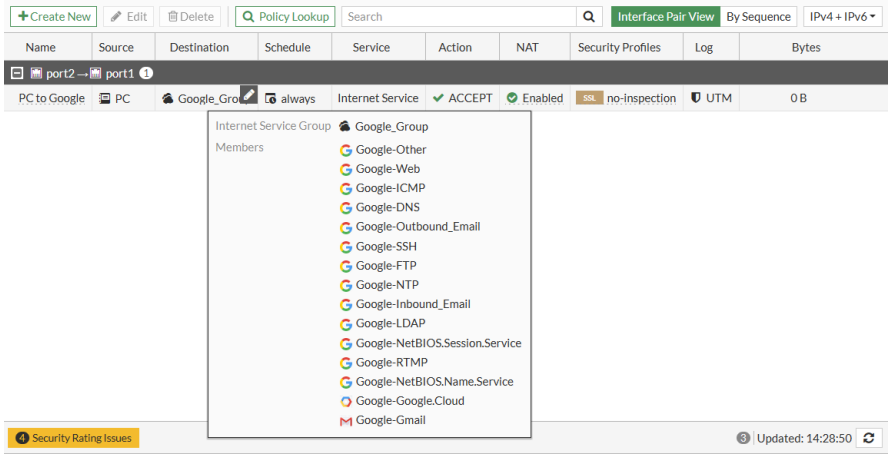
```
config firewall policy
    edit 1
        set name "PC to Google"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-group "Google_Group"
        set action accept
        set schedule "always"
        set fsso disable
        set nat enable
    next
end
```

To configure access to Google services using an Internet Service Group in the GUI:

1. On the FortiGate, create a Service Group using the CLI.
2. Go to *Policy & Objects* > *Firewall Policy*, and create a new policy.
3. Set the *Destination* as the just created Internet Service Group.



4. Configure the remaining options, then click *OK*.
5. Go to *Policy & Objects* > *Firewall Policy* and hover over the group to view a list of its members.



Example 2

In this example, two office FTP servers are put into an Internet Custom Service Group, and the PC connection to the FTP servers is limited to 1Mbps.

To put two FTP servers into a custom service group and limit the PC connection speed to them in the CLI:

1. Create custom internet services for the internal FTP servers:

```
config firewall internet-service-custom
    edit "FTP_PM"
        config entry
            edit 1
                config port-range
                    edit 1
                        set start-port 21
                        set end-port 21
```

```
                    next
                end
                set dst "PM_Server"
            next
        end
    next
    edit "FTP_QA"
        config entry
            edit 1
                config port-range
                    edit 1
                        set start-port 21
                        set end-port 21
                    next
                end
                set dst "QA_Server"
            next
        end
    next
end
```

2. Create a custom internet server group and add the just created custom internet services to it:

```
config firewall internet-service-custom-group
    edit "Internal_FTP"
        set member "FTP_QA" "FTP_PM"
    next
end
```

3. Create a traffic shaper to limit the maximum bandwidth:

```
config firewall shaper traffic-shaper
    edit "Internal_FTP_Limit_1Mbps"
        set guaranteed-bandwidth 500
        set maximum-bandwidth 1000
        set priority medium
    next
end
```

4. Create a firewall shaping policy to limit the speed from the PC to the internal FTP servers:

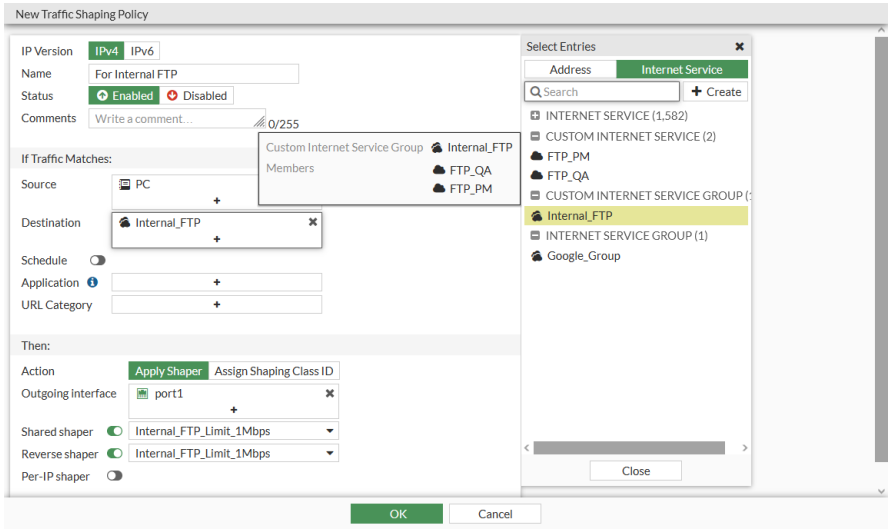
```
config firewall shaping-policy
    edit 1
        set name "For Internal FTP"
        set internet-service enable
        set internet-service-custom-group "Internal_FTP"
        set dstintf "port1"
        set traffic-shaper "Internal_FTP_Limit_1Mbps"
        set traffic-shaper-reverse "Internal_FTP_Limit_1Mbps"
        set srcaddr "PC"
    next
end
```

To put two FTP servers into a custom service group and limit the PC connection speed to the in the GUI:

1. Create custom internet services for the internal FTP servers using the CLI.
2. Create a custom internet server group and add the just created custom internet services to it using the CLI.
3. Create a traffic shaper to limit the maximum bandwidth:

a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shapers* tab, and click *Create New*.
b. Enter a *Name* for the shaper, such as *Internal_FTP_Limit_1Mbps*.
c. Set the *Traffic Priority* to *Medium*.
d. Enable *Max Bandwidth* and set it to *1000*.
e. Enable *Guaranteed Bandwidth* and set it to *500*.
f. Click *OK*.

4. Create a firewall shaping policy to limit the speed from the PC to the internal FTP servers:
a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
b. Set the *Destination* to the just created custom internet service group, and apply the just create traffic shaper.



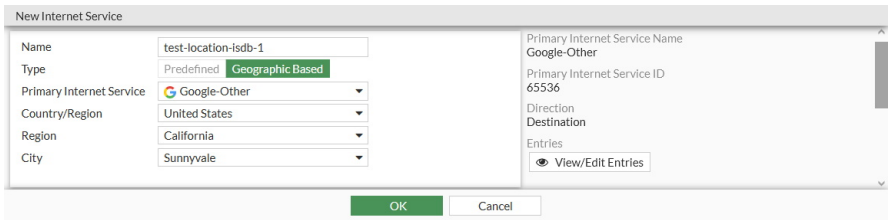
c. Configure the remaining options as shown, then click *OK*.

Allow creation of ISDB objects with regional information

Geographic-based Internet Service Database (ISDB) objects allow users to define a country, region, and city. These objects can be used in firewall policies for more granular control over the location of the parent ISDB object. ISDB objects are now referenced in policies by name instead of ID.

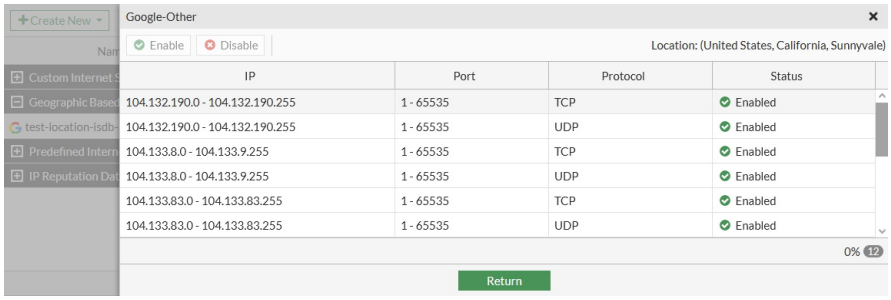
To apply a location-based ISDB object to a policy in the GUI:

1. Create the ISDB object:
a. Go to *Policy & Objects* > *Internet Service Database* and click *Create New* > *Geographic Based Internet Service*.
b. Configure the settings as required.

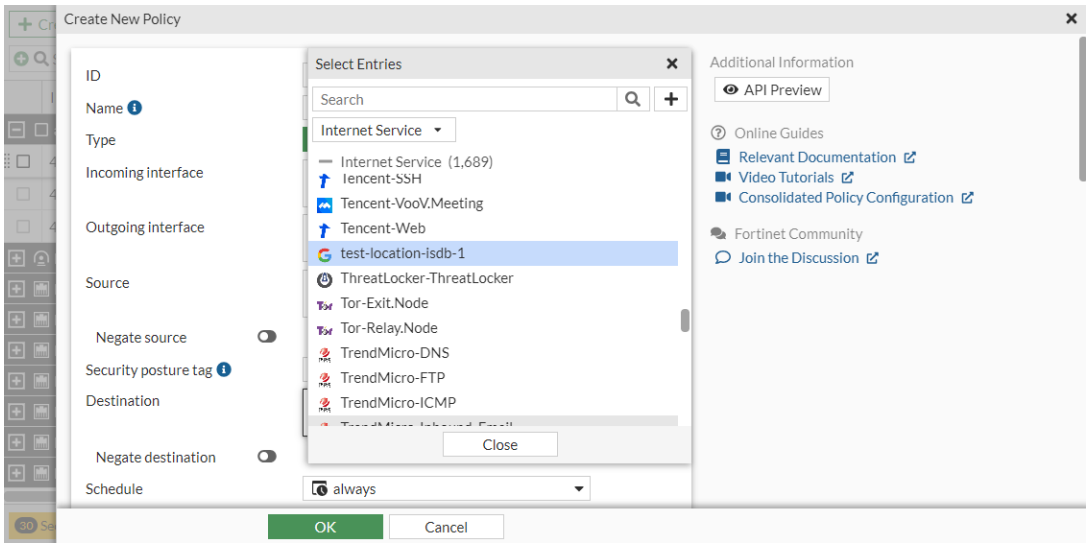


c. Click *OK*.

2. View the IP ranges in the location-based internet service:
   a. Go to *Policy & Objects* > *Internet Service Database* .
   b. In the table, hover over the object created in step 1 and click *View/Edit Entries*. The list of IPs is displayed:



   c. Click *Return*.
3. Add the ISDB object to a policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and create a new policy or edit an existing one.
   b. For *Destination*, select *Internet Service* from the dropdown list and select the ISDB object created in step 1.
   c. Configure the other settings as needed.



   d. Click *OK*.

To apply a location-based ISDB object to a policy in the CLI:

1. Create the ISDB object:

```
config firewall internet-service-name
    edit "test-location-isdb-1"
        set type location
        set internet-service-id 65536
        set country-id 840
        set region-id 283
        set city-id 23352
    next
end
```

2. View the IP ranges in the location-based internet service:

```
# diagnose internet-service id 65536 | grep "country(840) region(283) city(23352)"
96.45.33.73-96.45.33.73 country(840) region(283) city(23352) blocklist(0x0) reputation
(4), domain(5) popularity(0) botnet(0) proto(6) port(1-65535)
96.45.33.73-96.45.33.73 country(840) region(283) city(23352) blocklist(0x0) reputation
(4), domain(5) popularity(0) botnet(0) proto(17) port(1-65535)
198.94.221.56-198.94.221.56 country(840) region(283) city(23352) blocklist(0x0)
reputation(4), domain(5) popularity(4) botnet(0) proto(6) port(1-65535)
198.94.221.56-198.94.221.56 country(840) region(283) city(23352) blocklist(0x0)
reputation(4), domain(5) popularity(4) botnet(0) proto(17) port(1-65535)
```

3. Add the ISDB object to a policy:

```
config firewall policy
    edit 3
        set name "PC to Google"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "PC"
        set internet-service enable
        set internet-service-name "test-location-isdb-1"
        set action accept
        set schedule "always"
        set logtraffic all
        set logtraffic-start enable
        set auto-asic-offload disable
        set nat enable
    next
end
```

Internet service customization

Internet Service Database (ISDB) entries can be tuned for their environments by adding custom ports and port ranges, as well as port mapping.

If you are in multi-VDOM mode, Internet service customization can only occur at the Global level and not in a VDOM. See VDOM overview on page 2810 for more information.

To add a custom port range:

```
config firewall internet-service-addition
    edit 65646
        set comment "Add custom port-range:tcp/8080-8090 into 65646"
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 8080
                        set end-port 8090
                    next
                end
```

```
            next
        end
    next
end
Warning: Configuration will only be applied after rebooting or using the 'execute internet-
service refresh' command.
```

To verify that the change was applied:

```
 # diagnose internet-service info FG-traffic 6 8080 2.20.183.160
 Internet Service: 65646(Google.Gmail)
```

To configure additional port mapping:

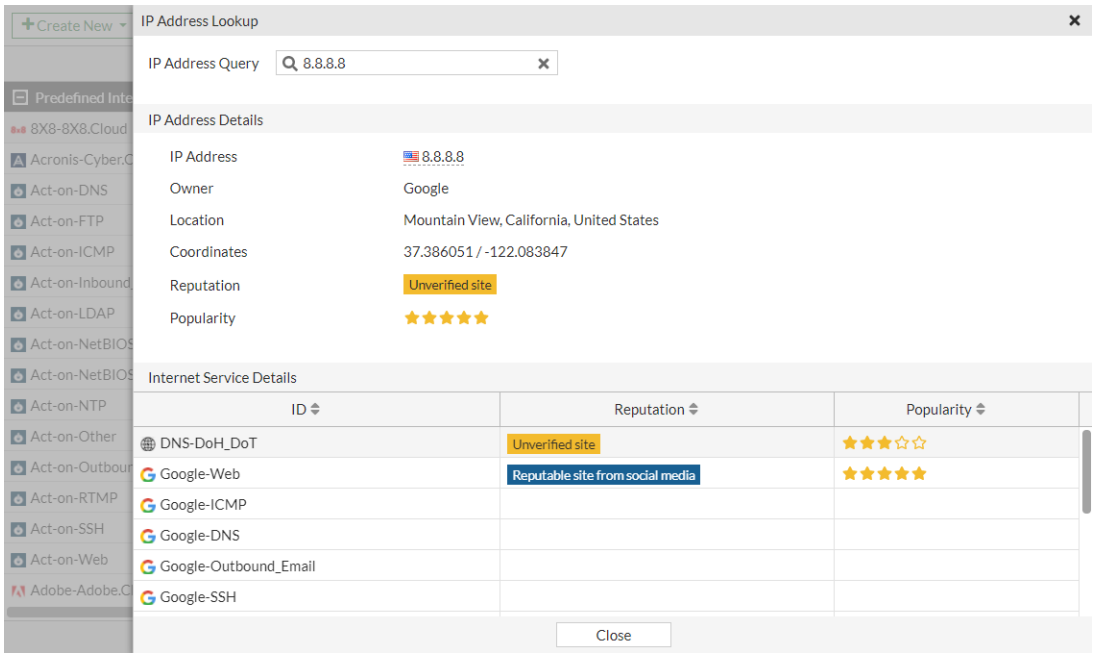
```
config firewall internet-service-append
    set match-port 10
    set append-port 20
end
Warning: Configuration will only be applied after rebooting or using the 'execute internet-
service refresh' command.
```

Look up IP address information from the Internet Service Database page

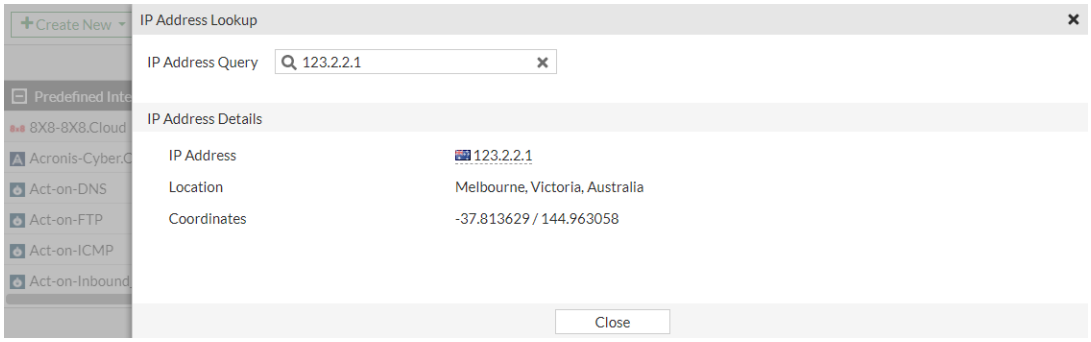
The *IP Address Lookup* button allows users to look up IP address information from the Internet Service Database and GeoIP Database. Returned IP address information includes the reverse IP address/domain lookup, location, reputation, and other internet service information.

To look up IP address information:

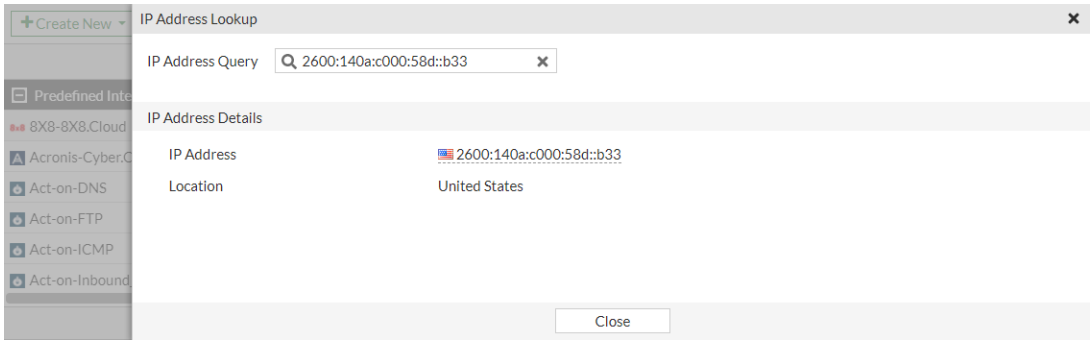
1. Go to *Policy & Objects* > *Internet Service Database*.
2. Click *IP Address Lookup*. The *IP Address Lookup* pane opens.
3. In the *IP Address Query* field, enter the IP address and press `Enter`.
   Results of an IP address from the Internet Service Database:



Results of an IP address from the GeoIP Database:



Results of an IPv6 address from the GeoIP Database:



4. Click *Close*.

Internet Service Database on-demand mode

Internet Service Database (ISDB) on-demand mode replaces the full-sized ISDB file with a much smaller file that is downloaded onto the flash drive. This file contains only the essential entries for Internet Services. When a service is

used in a firewall policy, the FortiGate queries FortiGuard to download the IP addresses and stores them on the flash drive. The FortiGate also queries the local MAC Database (MADB) for corresponding MAC information. The content of the ISDB entries used in firewall policies persists through reboots.

To enable ISDB (FFDB) on-demand mode:

1. Configure the global setting:

```
config system global
    set internet-service-database on-demand
end
```

   All FFDB files are erased.

2. Verify that there are no ISDB (FFDB) files:

```
# diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
---------
Version: 0.00000
Contract Expiry Date: n/a
Last Updated using manual update on Mon Jan  1 00:00:00 2001
Last Update Attempt: n/a
Result: Updates Installed
```

   Shortly after, the ISDB (FFDB) data structure is downloaded on the FortiGate. The following message appears in the debug messages:

```
do_ffsr_update[1567]-Starting  Update FFDB ondemand:(not final retry)
```

3. Run diagnostics again to verify that the ISDB (FFDB) files are saved on the FortiGate flash drive:

```
# diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
---------
Version: 7.02950
Contract Expiry Date: n/a
Last Updated using manual update on Fri Jan  6 06:45:00 2023
Last Update Attempt: n/a
Result: Updates Installed
```

4. Since no services have been applied to a policy, the IP range and IP address values are blank in the the summary details. For example, check the summary details for ID 1245187, Fortinet DNS:

```
# diagnose internet-service id-summary 1245187
Version: 00007.02950
Timestamp: 202301060645
Total number of IP ranges: 3085
Number of Groups: 1
Group(0), Singularity(90), Number of IP ranges(3085)
Internet Service: 1245187(Fortinet-DNS)
Number of IP ranges: 0
Number of IP addresses: 0
Singularity: 0
Icon Id: 19
Direction: dst
Data source: isdb
Country:
```

```
Region:
City:
```

5. Apply the Fortinet DNS service in a firewall policy:

```
config firewall policy
    edit 1
        set name "FDNS"
        set srcintf "port1"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Fortinet-DNS"
        set schedule "always"
        set nat enable
    next
end
```

6. Verify the summary details again for ID 1245187 (Fortinet DNS). There is now data for the IP range and IP address values:

```
# diagnose internet-service id-summary 1245187
Version: 00007.02951
Timestamp: 202301061144
Total number of IP ranges: 3558
Number of Groups: 2
Group(0), Singularity(90), Number of IP ranges(3078)
Group(1), Singularity(10), Number of IP ranges(480)
Internet Service: 1245187(Fortinet-DNS)
Number of IP ranges: 480
Number of IP addresses: 55242
Singularity: 10
Icon Id: 19
Direction: dst
Data source: isdb
Country: 12 32 36 40 56 124 158 170 203 222 250 276 320 332 344 356 360 372 380 392 458
484
        528 591 600 604 642 643 702 764 784 807 826 840
Region: 55 132 159 169 251 261 283 444 501 509 529 565 596 634 697 709 721 742 744 758
776 860
        1002 1056 1073 1151 1180 1190 1195 1216 1264 1280 1283 1284 1287 1290 1315 1319
1348 1363 1373 1380 1387
        1437 1457 1509 1536 1539 1660 1699 1740 1752 1776 1777 1826 1833 1874 1906 1965
2014 2028 2039 2060 2063
        2147 2206 65535
City: 615 679 818 1001 1106 1117 1180 1207 1330 1668 1986 2139 2812 2868 3380 3438 3485
3670 4276 4588 4622 4904
        5334 5549 5654 5827 6322 6325 6330 6355 6652 7844 9055 10199 10333 11420 12930
13426 13685 13769 14107 14813 15121
        15220 15507 15670 16347 16561 16564 16567 16631 17646 17746 17885 17975 17995
18071 18476 19066 19285 20784 21065 21092 21136
        21146 21266 21337 21779 21993 22292 22414 22912 23352 23367 23487 23574 23635
23871 23963 24076 24203 24298 24611 24955 25050
        25332 26854 27192 27350 28825 28866 65535
```

To verify MAC vendor information:

```
# diagnose vendor-mac id 1
Vendor MAC: 1(ASUS)
Version: 0000100146
Timestamp: 202301031100
Number of MAC ranges: 85
00:04:0f:00:00:00 - 00:04:0f:ff:ff:ff
00:0c:6e:00:00:00 - 00:0c:6e:ff:ff:ff
00:0e:a6:00:00:00 - 00:0e:a6:ff:ff:ff
...
```

Enabling the ISDB cache in the FortiOS kernel

A software ISDB cache can be enabled in the FortiOS kernel. This ISDB cache can be used to enhance lookup performance by circumventing the ISDB lookup penalty when revisiting the same resources.

The ISDB cache can be enabled using the following command:

```
config system settings
    set internet-service-database-cache {enable | disable}
end
```

Example

In the following example, after enabling the software ISDB cache, traffic will be generated twice to the same resource. Since the ISDB cache is enabled, no new query will occur in the ISDB. Instead, the ISDB lookup in performed in the cache table.

To enable the software ISDB cache:

1. Enable the ISDB cache:

   ```
   config system settings
       set internet-service-database-cache enable
   end
   ```

2. Create an ISDB firewall policy:

   ```
   config firewall policy
       edit 1
           set internet-service enable
           set internet-service-name "Google-DNS" "Google-Other" "Google-Web"
           set internet-service6 enable
           set internet-service6-name "Google-DNS" "Google-Other" "Google-Web"
       next
   end
   ```

3. Generate traffic to access the resource which matches the ISDB ID in the firewall policy.
4. Check the Internet Service cache lists:

   ```
   # diagnose firewall internet-service-cache list
   List Internet Service (IPV4) Cache in Kernel:
   MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=2 isdb_cache_hit_count=0 isdb_query_
   ```

```
count=2
proto=6 port=443 IP=10.151.118.105 id=1245185 country_id=840 region_id=283 city_id=21065
reputation=5 insert_timestamp=4302579542 cache_hit_count=0
proto=6 port=443 IP=10.8.8.8 id=65537 country_id=840 region_id=283 city_id=15905
reputation=5 insert_timestamp=4302579760 cache_hit_count=0

# diagnose firewall internet-service6-cache list
List Internet Service (IPV6) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=1 isdb_cache_hit_count=0 isdb_query_
count=1
proto=6 port=443 IP=2600:140a:1000:196::b33 id=7929993 country_id=124 region_id=65535
city_id=65535 reputation=4 insert_timestamp=4302580009 cache_hit_count=0
```

5. Generate traffic to access the same resource again.
6. Check the Internet Service cache lists:

```
# diagnose firewall internet-service-cache list
List Internet Service (IPV4) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=2 isdb_cache_hit_count=1 isdb_query_
count=2
proto=6 port=443 IP=10.151.118.105 id=1245185 country_id=840 region_id=283 city_id=21065
reputation=5 insert_timestamp=4302579542 cache_hit_count=0
proto=6 port=443 IP=10.8.8.8 id=65537 country_id=840 region_id=283 city_id=15905
reputation=5 insert_timestamp=4302579760 cache_hit_count=1

# diagnose firewall internet-service6-cache list
List Internet Service (IPV6) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=1 isdb_cache_hit_count=1 isdb_query_
count=1
proto=6 port=443 IP=2600:140a:1000:196::b33 id=7929993 country_id=124 region_id=65535
city_id=65535 reputation=4 insert_timestamp=4302580009 cache_hit_count=1
```

The ISDB lookup is performed in the cache table so there is no new query in the full ISDB.