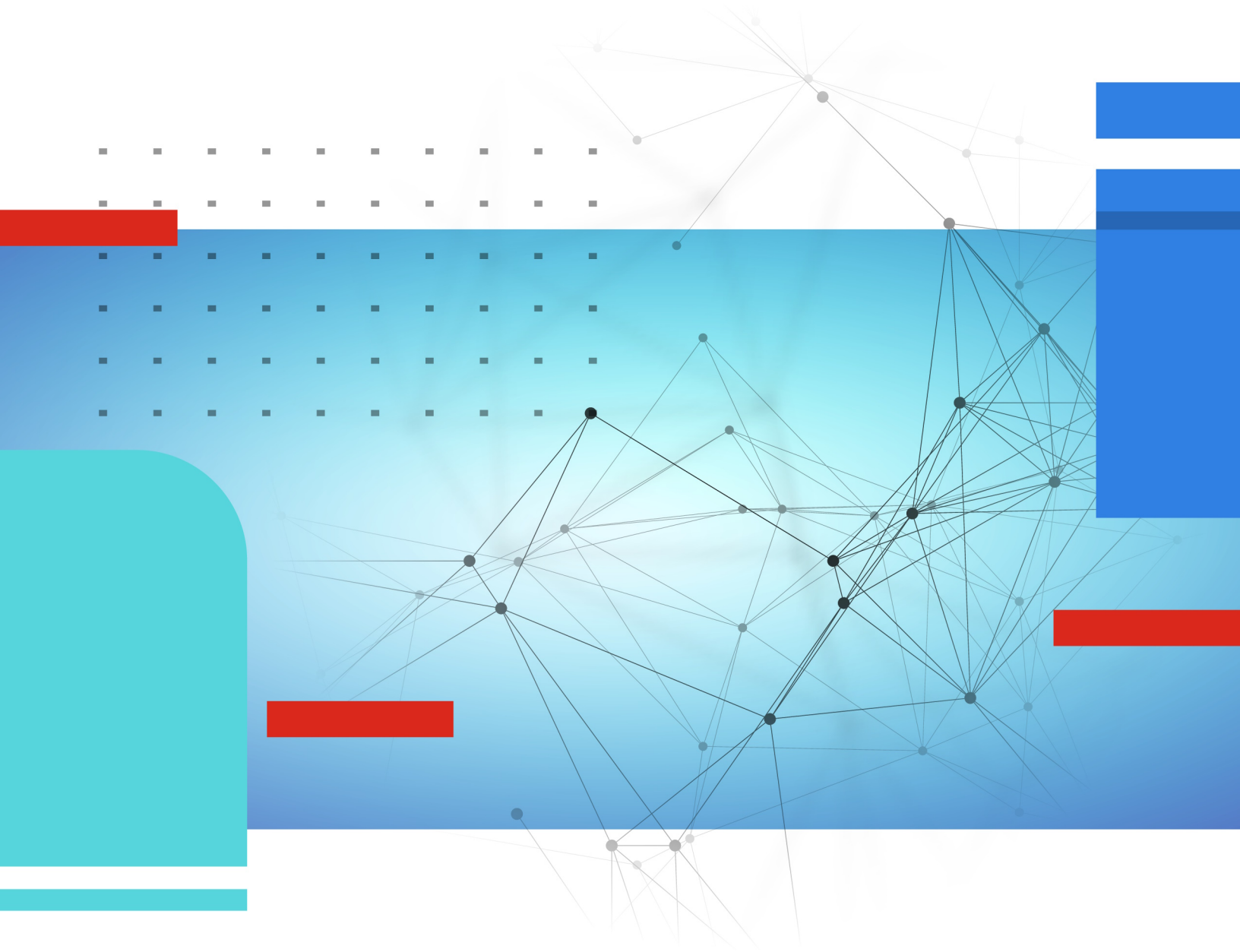
Administration Guide

Container FortiOS 7.2.1



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

July 24, 2024
Container FortiOS 7.2.1 Administration Guide
87-721-982159-20240724

TABLE OF CONTENTS

Change Log

Date	Change Description
2024-07-17	Initial release.

Introduction

Container FortiOS (cFOS) provides enterprise-grade network security from FortiOS, tailored to suit the requirements of container platforms. With its lightweight and modular architecture, network administrators can efficiently deploy only the necessary network security functionality for an application or service, reducing resource requirements and management complexity, thus ensuring the security of the business.

Container FortiOS supports Linux Containers (LXC), Docker, and Kubernetes.

This guide provides information about the administration of Container FortiOS version 7.2.1. Information specific to a particular platform is called out when appropriate.

Getting started

This section provides the following information:

For detailed instructions about deployment of Container FortiOS, see the appropriate Getting Started guide for your platform:

- LXC
- Docker
- Kubernetes

Connecting to the Container FortiOS CLI

Container FortiOS provides access to the FortiOS CLI. It also provides access to the underlying Linux shell.

To connect to the running Container FortiOS container:

1. In the host shell, enter the appropriate command for your platform:
   - LXC: `lxc-console -n <container_name>`
   - Docker: `docker exec -it <container_name> /bin/cli`
   - Kubernetes: `kubectl exec --stdin --tty <container_name> /bin/cli`
2. At the prompt, enter the username and password.
   The default user is `admin` and the password is blank.

The Container FortiOS CLI is similar to the FortiOS CLI. While much of the command syntax is the same, there are a limited number of commands available, reflecting the available Container FortiOS features.

See the Container FortiOS CLI Reference for more information.

Connecting to the REST API

Container FortiOS provides a REST API for configuration and monitoring operations. The API is similar to the FortiOS API.

The API is accessible by default on port 443 at any of the container interfaces. If configured to require a token, all requests must include the API token.

For example, the following examples get the antivirus settings:

```
curl -H "Authorization: Bearer rkMJd3SdLhb8UFBan987CnIrmPBLfaIj"
https://localhost/api/v2/cmdb/antivirus/settings
```

```
curl https://localhost/api/v2/cmdb/antivirus/settings?access_
token=rkMJd3SdLhb8UFBan987CnIrmPBLfaIj
```

Due to the architecture of Kubernetes, the REST API should not be used with Kubernetes deployments.

Configuration should be deployed using `ConfigMap` as described in Deploying configurations to Kubernetes on page 10.

For full details on the available API actions, see the Container FortiOS REST API documentation on FNDN.

Configuring the REST API access port

You may configure HTTP and HTTPS access to the API. By default, HTTP access is disabled.

To configure the REST API HTTP access port:

In the Container FortiOS CLI, run the following command to enable access on port 80:

```
config system global
    set admin-port 80
```

To configure the REST API HTTPS access port:

In the Container FortiOS CLI, run the following command to enable access on port 443:

```
config system global
    set admin-sport 443
```

To disable REST API access:

Disable REST API access for HTTP or HTTPS by setting `admin-port` and `admin-sport`, respectively, to `0`.

API token authentication

By default, an API access token is not needed.

To enable access token authentication, create at least one API user and generate a token.

After an API user has been created, each REST API request requires an API token for authentication.

To create an API user:

In the Container FortiOS CLI, run the following command:

```
config system api-user
    edit "api-user-1"
end
```

To generate an API token:

In the Container FortiOS CLI, run the following command:

```
exec api-user generate-key api-user-1
```

Copy and save the API key as it is only shown once and cannot be retrieved.

Installing a license

For Container FortiOS on Kubernetes, use `ConfigMap` to deploy the Container FortiOS license file.

See Deploying a license in the Container FortiOS Getting Started (Kubernetes) guide.

You must install a valid Container FortiOS license before most features are available.

To upload a license:

1. Open the license file in a text editor and copy the full contents.
2. In the container CLI, enter the following command:

   ```
   exec import-license "<content_of_license_file>"
   ```

   Container FortiOS validates the license.

Deploying configurations to Kubernetes

In Kubernetes, configurations can be applied to Container FortiOS using a `ConfigMap`.

The two types of configuration are as follows:

- Partial configuration: A partial configuration is applied on top of a current configuration in Container FortiOS. A configuration can be split into multiple smaller configurations and applied separately.
- Full configuration: The active configuration will be replaced with the new configuration.

Deploying a partial configuration

The following example ipsec-configmap.yml file shows a partial configuration:

```
apiVersion: v1
kind: ConfigMap
metadata:
    name: foscfg-ipsec
    labels:
        app: fos
        category: config
data:
```

```
    type: partial
    config: |-
        config vpn certificate ca
            edit "ipsec-ca"
                set ca "-----BEGIN CERTIFICATE-----
                MIIDJDCCAgygAwIBAgIJAK6dHv+qKBjJMA0GCSqGSIb3DQEBCwUAMBExDzANBgNV
                BAMMBnRlc3RjYTAeFw0yMjAxMTMxODIxMThaFw0zMjAxMTExODIxMThaMBExDzAN
                BgNVBAMMBnRlc3RjYTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOXE
                ct+WmzZ8YT+rJEQKDGfgqiJu9kzNz+Na0smwPvFEOfcm6XYHqy/li+CdyIGCtlQX
                hDbABD7uQiVBObzO4VzPn3Ik7PMR+hBr0sULqOQ8SkgU/H/pgm5WjSO0oiiPoQon
                LWDQXs294aF0EouNp0KfI9vXkAvzv57RUGeuPfr9tvoLyIgBB1nqWbK98GfMyX1K
                sHB0mp0PCxq1S6hQK9pny3/wvsq3YxggpJAFpCAbDXI97jhk9atMaIRjGErUZNsO
                .....
                .....
                .....
                -----END CERTIFICATE-----"
                next
            end
        config vpn certificate local
            edit "ipsec-cert"
                set password "{{ipsec-certs:ipsec-cert-pass}}"
                set private-key "{{ipsec-certs:ipsec-cert-key}}"
                set certificate "-----BEGIN CERTIFICATE-----
                MIIDYDCCAkigAwIBAgIQAx0NCLIRx9Q5lWcGmS2U+DANBgkqhkiG9w0BAQsFADAR
                MQ8wDQYDVQQDDAZ0ZXN0Y2EwHhcNMjIwMTEzMTkwMDUzWhcNMjQwNDE3MTkwMDUz
                WjAYMRYwFAYDVQQDDA1pcHNlYy1jbGllbnQyMIIBIjANBgkqhkiG9w0BAQEFAAOC
                AQ8AMIIBCgKCAQEAyXXh8OiuEf5Drh+df3FJm2f/ZKNvRONEQba/77cHVRT2pjOV
                07llYQye1mG0JBedUM0SFEkmWkafyYE+KzYzse2r7NSX1bkFizW/TwrNk/VCuLMt
                +HUgClrcmrPAdbDUZYyIKWKN4FwlOyZz0YNA14NuM/gNE+fY1kaaaojxqfpneJCW
                nYcfCTuNgADnyHjzXZMLulj+4Cy1OylKSKX7cAVt9pS2SwzzGF4fGnlDKhfAtxzR
                .....
                .....
                .....
                -----END CERTIFICATE-----"
                next
            end
        config vpn ipsec phase1-interface
            edit "test-p1"
                set interface "eth0"
                set peertype any
                set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-
prfsha384 chacha20poly1305-prfsha256
                set psksecret {{ipsec-psks:psk1}}
                set auto-negotiate disable
                next
            end
        config vpn ipsec phase2-interface
            edit "test-p2"
                set phase1name "test-p1"
                set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
                set dhgrp 14 15 5
                set src-subnet 10.4.96.0 255.255.240.0
                set dst-subnet 10.0.4.0 255.255.255.0
                next
            end
```

Configuration should be created with the following guidelines:

- Labels `app: fos` and `category: config` are required.
- `type: partial` indicates that this is a partial configuration.
- The `config` section holds the actual configuration data as a series of CLI commands.
- In the configuration, there are variables (for example, `{{ipsec-certs:ipsec-cert-pass}}` and `{{ipsec-certs:ipsec-cert-key}}`) that are references to the keys in `Secrets`. Kubenets use `Secrets` to store sensitive data.

  In this example, we save an IPSEC pre-shared key in a `Secret` called `ipsec-certs` with key `ipsec-cert-pass`. In the configuration we can use `{{ipsec-certs:ipsec-cert-pass}}` to refer it.

  The format is `{{<Secret name>:<Key name>}}`.

  The following example command creates this secret:

  ```
  kubectl create secret generic ipsec-certs --from-literal=ipsec-cert-pass=12345678
  ```

  For more information about Kubernets `Secrets`, see https://kubernetes.io/docs/concepts/configuration/secret/.

Deploying a full configuration

Full configuration has the same format with partial configuration except `type` is `full` instead of `partial`.

The following is an example of the commands used to create a `ConfigMap` for a full configuration:

```
kubectl create configmap fos-config --from-file=config=<path to config file> --from-
literal=type=full"
kubectl label configmap fos-config app=fos
kubectl label configmap fos-config category=config
```

Ensure the configuration file contains all required dependencies. For example, if a firewall policy references a web filter profile `block-category-11`, the web filter profile and all of its dependencies must be included in the configuration.

More information

Additional Container FortiOS documentation is available in the Fortinet Documentation Library.

FortiOS documentation

Configuration and administration of Container FortiOS is very similar to FortiOS.

The following FortiOS documents may be helpful:

- FortiOS Administration Guide
- FortiOS CLI Reference
- FortiOS REST API Reference on FNDN

System configuration

This section contains information about configuring your Container FortiOS system after initial deployment.

Configuring system settings

The following system settings can be configured in the CLI or REST API:

Parameter	Description	Type	Size	Default
central-nat	Enable/disable central NAT.	option	-	disable
	Option / **Description** *enable* — Enable central NAT. *disable* — Disable central NAT.			
ngfw-mode	Next Generation Firewall (NGFW) mode.	option	-	profile-based
	Option / **Description** *profile-based* — Application and web-filtering are configured using profiles applied to policy entries. *policy-based* — Application and web-filtering are configured as policy match conditions.			
tcp-session-without-syn	Enable/disable allowing TCP session without SYN flags.	option	-	disable
	Option / **Description** *enable* — Allow TCP session without SYN flags. *disable* — Do not allow TCP session without SYN flags.			

For more information about NGFW mode, see NGFW mode on page 77.

For more information about central NAT, see Central SNAT on page 79.

Configuration

To configure system settings in the CLI:

```
config system settings
    set central-nat [enable|disable]
    set ngfw-mode [profile-based|policy-based]
    set tcp-session-without-syn [enable|disable]
end
```

To configure system settings with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X PUT
    -d '{
        "data": {
            "central-nat": [enable|disable],
            "ngfw-mode": [profile-based|policy-based],
            "tcp-session-without-syn": [enable|disable]
        }
    }'
    http://localhost/api/v2/cmdb/system/settings
```

Configuring global settings

The following global settings can be configured through the CLI or the REST API:

Parameter	Description	Type	Size	Default
admin-port	REST API access port for HTTP. 0 disables HTTP.	integer	Minimum value: 0 Maximum value: 65535	0
admin-server-cert	Server certificate that Container FortiOS uses for HTTPS administrative connections.	string	Maximum length: -	Fortinet_ GUI_Server
admin-sport	REST API access port for HTTPS. 0 disables HTTPS.	integer	Minimum value: 0 Maximum value: 65535	443
cert-chain-max	Maximum number of certificates that can be traversed in a certificate chain.	integer	Minimum value: 1 Maximum value: 2147483647	8

Parameter	Description	Type	Size	Default
remoteauthtimeout	Number of seconds that Container FortiOS waits for responses from remote authentication servers.	integer	Minimum value: 1 Maximum value: 300	5

Configuration

To configure global settings in the CLI:

```
config system global
    set admin-port {integer}
    set admin-server-cert {string}
    set admin-sport {integer}
    set cert-chain-max {integer}
    set remoteauthtimeout {integer}
end
```

To configure global settings with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X PUT
    -d '{
        "data": {
            "admin-port": <integer>,
            "admin-server-cert": <string>,
            "admin-sport": <integer>,
            "cert-chain-max": <integer>,
            "remoteauthtimeout": <integer>
        }
    }'
    http://localhost/api/v2/cmdb/system/global
```

Configuring networks and interfaces

Container FortiOS listens on all interfaces connected to the container.

By default, Container FortiOS forwards all traffic on any interface to the other interfaces. You must configure the host to route traffic to the container interfaces as needed.

- Configuring interfaces on page 16
- Configuring static routing on page 16

Configuring interfaces

While direct configuration of Container FortiOS interfaces is seldom required, you can configure the interfaces through the CLI with the `config system interface` command.

```
config system interface
    edit <name>
        set ip {ipv4-classnet-host}
        set macaddr {mac-address}
    next
end
```

Similarly, use `show system interface` to view the current interface configuration.

Configuring static routing

Static routes can be configured in Container FortiOS.

Generally, most routing is handled by the host. Static routing in Container FortiOS is seldom needed.

Container FortiOS also provides access to the underlying Linux shell of the container for additional routing configuration through the `sysctl sh` command.

There is no REST API option for configuring static routes.

To configure a static route in the CLI:

```
config router static
    edit <seq-num>
        set comment {var-string}
        set device {string}
        set distance {integer}
        set dst {ipv4-classnet}
        set gateway {ipv4-address}
        set src {ipv4-classnet}
        set status [enable|disable]
    next
end
```

Parameter	Description	Type	Size	Default
comment	Optional comments.	var-string	Maximum length: 255	
device	Gateway out interface or tunnel.	string	Maximum length: 35	

Parameter	Description	Type	Size	Default
distance	Administrative distance.	integer	Minimum value: 1 Maximum value: 255	10
dst	Destination IP and mask for this route.	ipv4-classnet	Not Specified	0.0.0.0 0.0.0.0
gateway	Gateway IP for this route.	ipv4-address	Not Specified	0.0.0.0
src	Source prefix for this route.	ipv4-classnet	Not Specified	0.0.0.0 0.0.0.0
status	Enable/disable this static route.	option	-	enable

Option	Description
enable	Enable static route.
disable	Disable static route.

Configuring FortiGuard

FortiGuard services provide signature packages and querying services that provide content, web, and device security, delivered via FortiGuard servers that are part of the FortiGuard Distribution Network (FDN).

FortiGuard services are available on Container FortiOS with appropriate service subscriptions. Container FortiOS must be connected to the Internet in order to automatically connect to the FDN to validate the license and download FDN updates or perform real-time queries.

FortiGuard settings can be configured and viewed through either the CLI or the REST API.

Item	CLI	REST API
FortiGuard settings	`config system fortiguard`	`/api/v2/cmdb /system/fortiguard`
FortiGuard update schedule	`config system autoupdate schedule`	`/api/v2/cmdb /system.autoupdate/schedule`
View database and engine versions	`diagnose autoupdate versions`	

For more information about available CLI options, see the Container FortiOS CLI Reference.

For more information about available REST API options, see the Container FortiOS REST API Reference.

Configuring global IPS settings

Several IPS configuration options can be set globally in Container FortiOS using the CLI.

Extended IPS database

Select `regular` to use the regular IPS database or `extended` to use the extended IPS database.

To enable the extended IPS database:

```
config ips global
    set database extended
end
```

IPS engine-count

Set the number of concurrent IPS engines running.

The recommended and default setting is `0`, which allows Container FortiOS to determine the optimum number of IPS engines.

To specify the number of concurrent IPS engines:

```
config ips global
    set engine-count <int>
end
```

Fail-open

A fail-open scenario is triggered when the IPS raw socket buffer is full. In this case the IPS engine has no space in memory to create more sessions and needs to drop the sessions or bypass the sessions without inspection.

To enable fail-open mode:

```
config ips global
    set fail-open {enable | disable}
end
```

The default setting is `disable`, where sessions are dropped by the IPS engine when the system enters fail-open mode.

When enabled, the IPS engine fails open, and it affects all protocols inspected by FortiOS IPS protocol decoders, including but not limited to HTTP, HTTPS, FTP, SMTP, POP3, IMAP, and so on. When the IPS engine fails open, traffic continues to flow without IPS scanning.

IPS buffer size

If your system enters fail-open mode frequently, it is possible to increase the IPS socket buffer size to allow more data buffering, which reduces the chances of overloading the IPS engine.

To set the socker buffer size:

```
config ips global
    set socket-size <int>
end
```

Session count accuracy

The IPS engine can track the number of open session in two ways: `accurate` or `heuristic`. An accurate count uses more resources than a less accurate heuristic count.

To configure the IPS open session count mode:

```
config ips global
    set session-limit-mode {accurate | heuristic}
end
```

Modifying replacement messages

Container FortiOS has replacement messages that are HTML and text files. These messages can be customized to meet user requirements. The content can be modified, and images can be added.

Replacement messages and images can be configured through the CLI and the REST API.

The following examples show how to modify the *Application Blocked* message. For both examples, update the `buffer` value with the full HTML of the message.

To modify a replacement message in the CLI:

```
edit "appblk-html"
    set buffer "<!DOCTYPE html>
<html lang=\"en\">
    <head>
        <meta charset=\"UTF-8\">
        <meta http-equiv=\"X-UA-Compatible\" content=\"IE=8; IE=EDGE\">
        <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">
        <link href=\"https://fonts.googleapis.com/css?family=Roboto&display=swap\"
rel=\"stylesheet\">
        <style type=\"text/css\">
            {styles...}
        </style>
        <title>Application Control Violation</title>
    </head>
        <body>
```

```
        <div class=\"message-container\">
            <div class=\"logo\"></div>
            <h1>cFOS Application Control</h1>
            <h3>Application Blocked</h3>
            <p>You have attempted to use an application that violates your Internet usage
policy.</p>
            <table><tbody>
                <tr>
                    <td>Application</td>
                    <td>%%APPNAME%%</td>
                </tr>
                <tr>
                    <td>Category</td>
                    <td>%%APPCAT%%</td>
                </tr>
                <tr>
                    <td>URL</td>
                    <td>%%PROTOCOL%%://%%URL%%</td>
                </tr>
                <tr>
                    <td>Username</td>
                    <td>%%USERNAME%%</td>
                </tr>
                <tr>
                    <td>Group Name</td>
                    <td>%%GROUPNAME%%</td>
                </tr>
                <tr>
                    <td>Policy</td>
                    <td>%%POLICY_UUID%%</td>
                </tr>
            </tbody></table>
        </div>
    </body>
</html>"
    set header http
    set format html
    next
end
```

To modify a replacement message with the REST API:

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "header":"http",
    "format": "html",
    "buffer": "<message_content>"
}}' http://localhost/api/v2/cmdb/system.replacemsg/utm/appblk-html
```

Replacement message images

Images can be added to replacement messages.

The supported image formats are GIF, JPEG, TIFF, and PNG. The maximum file size supported is 24 KB.

Add and modify replacement message images through the CLI and the REST API.

Use the added image in a replacement message by including `%%IMAGE:<image name>%%` in the code to add the image.

To add an image in the CLI:

```
config system replacemsg-image
    edit <image_name>
        set image-base64 {var-string}
        set image-type [gif|jpg|...]
    next
end
```

Use the same command to update the image.

To add an image with the REST API:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "image-type": [gif|jpg|...],
    "image-base64": "{var-string}"
}}' http://localhost/api/v2/cmdb/system/replacemsg-image/<image_name>
```

To update this image, send a PUT request in the same format to the same URL with the new data.

To delete the image, send a DELETE request to the same URL.

Using replacement message groups

Replacement message groups allow you to customize replacement messages for UTM settings in individual firewall policies and profiles.

Replacement message groups can be configured through the CLI and the REST API.

The messages added to a group do not need to be customized. The message body content, header type, and format will use the default values if not customized.

The following examples show the creation of a message group that includes custom messages and is assigned to an antivirus profile.

To create a replacement message group in the CLI:

1. Create the replacement message group:

```
config system replacemsg-group
    edit "newutm"
        set comment "UTM message group"
        set group-type utm
        config utm
           edit "new-message"
```

```
                set buffer <html_content>
                set format html
                set header http
            next
        end
    next
end
```

2. Apply the message group to the antivirus filter:

```
config antivirus profile
    edit "new-av-profile"
        set replacemsg-group "newutm"
    next
end
```

3. Apply the antivirus filter and message group to a policy:

```
config firewall policy
    edit 1
        ...
        set av-profile "new-av-profile"
        ...
    next
end
```

To create a replacement message group with the REST API:

1. Create the replacement message group:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "newutm",
    "group-type": "utm",
    "utm: ["alert-notice": {
        "buffer": <html_content>,
        "header": "http",
        "format": "html"
    }]
}}' http://localhost/api/v2/cmdb/system/replacemsg-group
```

2. Apply the message group to the antivirus filter:

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "replacemsg-group": "newutm"
}}' http://localhost/api/v2/cmdb/antivirus/profile/new-av-profile
```

3. Apply the antivirus filter and message group to a policy:

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "av-profile": "new-av-profile"
}}' http://localhost/api/v2/cmdb/firewall/policy/1
```

Backing up and restoring configuration

A backup configuration file can be used to restore system configuration.

The full configuration file must be encrypted with a password in order to be applied to a container with a different data directory.

To back up system configuration:

1. In the Container FortiOS CLI, run the following command:

   ```
   execute config backup <filename> [password]
   ```

   Enter a password to encrypt the file with the provided password.

   The backup file is saved to the configured `data` directory, so you must ensure that you have properly configured a persistent volume for this directory.
2. Copy or move the backup file as needed.

To restore system configuration from a backup file:

1. Copy the backup file to the `data` directory and name it `cfos.conf`.

   You may also restore a partial configuration using a `cfos-partial.conf` file.

   At startup, you may specify either a full configuration file or a partial configuration file, but not both.

   When using an encrypted configuration file at startup, save the backup password in a file named `cfos.key` in the data directory.
2. Start the docker container.

   The configuration file is read and applied to the container when it runs.

   If you are restoring to a new container that does not have a license configured, copy the appropriate Container FortiOS license file to the `data` directory and name it `cfos.lic`. The license must be provided for the configuration to be applied.

   When Container FortiOS applies a configuration file on startup, any provided configuration, license, and key files are deleted after they are applied.

Security profiles

This section contains information about configuring the following types of security profiles:

- Antivirus
- Application control
- IPS
- Web filter

Other FortiOS security profile types are not supported in Container FortiOS.

Container FortiOS operates in flow mode. Proxy mode is not supported.

Antivirus

Container FortiOS offers flow-based antivirus and includes a `default` preloaded antivirus profile.

You can customize this profile, or you can create your own to inspect certain protocols, remove viruses, and apply botnet protection to network traffic. Once configured, you can add the antivirus profile to a firewall policy.

This functionality requires a subscription to FortiGuard Antivirus.

Configuring an antivirus profile

In an antivirus profile, Container FortiOS can be configured to apply antivirus protection to HTTP, FTP, IMAP, POP3, SMTP, CIFS, and NNTP sessions. Antivirus inspection prevents potentially unwanted and malicious files from entering the network. Antivirus profiles include multiple different functions, such as scanning files for virus signatures, scanning for advanced persistent threats, checking external malware hash lists and threat feeds, and others. Malicious files can be blocked or monitored, and can be quarantined. Some antivirus profile options require a license or other Fortinet products.

Antivirus profiles can be configured through the CLI or the REST API.

Options

The following options are available:

Parameter	Description	Type	Size	Default
av-block-log	Enable/disable logging for AntiVirus file blocking.	option	-	enable
	Option / **Description** *enable* — Enable setting. *disable* — Disable setting.			
av-virus-log	Enable/disable AntiVirus logging.	option	-	enable
	Option / **Description** *enable* — Enable setting. *disable* — Disable setting.			
comment	Comment.	var-string	Maximum length: 255	
extended-log	Enable/disable extended logging for antivirus.	option	-	disable
	Option / **Description** *enable* — Enable setting. *disable* — Disable setting.			
name	Profile name.	string	Maximum length: 35	
replacemsg-group	Replacement message group customized for this profile.	string	Maximum length: 35	
scan-mode	Configure scan mode.	option	-	default
	Option / **Description** *default* — On the fly decompression and scanning of certain archive files.			

Protocol options

When applying an antivirus profile to a firewall policy, the protocol options profile defines parameters for handling protocol-specific traffic.

Antivirus scanning is available for the following protocols:

- CIFS
- FTP
- HTTP
- IMAP
- NNTP

- POP3
- SMTP

The following settings are available for each protocol:

Parameter	Description	Type	Size	Default
av-scan	Enable or disable antivirus scanning for this protocol.	option	-	disable

Option	Description
disable	Disable.
block	Block the virus infected files.
monitor	Log the virus infected files.

Parameter	Description	Type	Size	Default
archive-block	Select the archive types to block.	option	-	

Option	Description
encrypted	Block encrypted archives.
corrupted	Block corrupted archives.
partiallycorrupted	Block partially corrupted archives.
multipart	Block multipart archives.
nested	Block nested archives that exceed uncompressed nest limit.
mailbomb	Block mail bomb archives.
fileslimit	Block exceeded archive files limit.
timeout	Block scan timeout.
unhandled	Block archives that FortiOS cannot open.

Parameter	Description	Type	Size	Default
archive-log	Select the archive types to log.	option	-	

Option	Description
encrypted	Log encrypted archives.
corrupted	Log corrupted archives.
partiallycorrupted	Log partially corrupted archives.
multipart	Log multipart archives.
nested	Log nested archives that exceed uncompressed nest limit.
mailbomb	Log mail bomb archives.
fileslimit	Block exceeded archive files limit.
timeout	Log scan timeout.
unhandled	Log archives that FortiOS cannot open.

Parameter	Description	Type	Size	Default
emulator	Enable or disable the virus emulator.	option	-	enable

Option	Description
enable	Enable the virus emulator.
disable	Disable the virus emulator.

Parameter	Description	Type	Size	Default
executables	Enable or disable treatment of Windows executables as viruses. This option is only available for IMAP, POP3, and SMTP protocols.	option	-	default

Option	Description
default	Perform standard antivirus scanning of Windows executable files.
virus	Treat Windows executables as viruses.

Configuration

To configure an antivirus profile in the CLI:

```
config antivirus profile
    edit <name>
        set av-block-log [enable|disable]
        set av-virus-log [enable|disable]
        config cifs
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
        set comment {var-string}
        set extended-log [enable|disable]

        config ftp
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
        config http
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
        config imap
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
```

```
            set emulator [enable|disable]
        end
        config nntp
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
        config pop3
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
        set replacemsg-group {string}
        set scan-mode default
        config smtp
            set av-scan [disable|block|...]
            set archive-block {option1}, {option2}, ...
            set archive-log {option1}, {option2}, ...
            set emulator [enable|disable]
        end
    next
end
```

To configure an antivirus profile with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": <name>,
    "comment": <comment>,
    ...,
    "http": [
        "av-scan":,
        "archive-block": ,
        "archive-log": ,
        "emulator":
    ],
    ...
}}' http://localhost/api/v2/cmdb/antivirus/profile
```

Application control

Container FortiOS can recognize network traffic generated by a large number of applications. Application control sensors specify what action to take with the application traffic. Application control uses IPS protocol decoders that can analyze network traffic to detect application traffic, even if the traffic uses non-standard ports or protocols. Application control supports traffic detection using the HTTP protocol (versions 1.0, 1.1, and 2.0).

Container FortiOS includes one preloaded application sensor, *default*, which monitors all applications.

You can customize this sensor or create your own to log and manage specific applications.

Once configured, you can add the application sensor to firewall policies.

This functionality requires a subscription to FortiGuard Application Control.

Viewing and configuring application signatures

Container FortiOS has many preconfigured application signatures. You may also add and configure customer application signatures.

Application signatures may be viewed and configured through the CLI and the REST API.

To view all application signatures in the CLI:

```
show application name
```

To view all application signatures with the REST API:

```
curl http://localhost/api/v2/cmdb/application/name
```

Configuring application groups

Application groups collect applications into one item through a list of applications or a set of filters.

An application group may be used in the same manner as a single application.

Application groups may be configured through the CLI and the REST API.

Options

The following options are available:

Parameter	Description	Type	Size	Default
application <id>	Application ID list. Application IDs.	integer	Minimum value: 0 Maximum value: 4294967295	
behavior	Application behavior filter.	user	Not Specified	all

Parameter	Description	Type	Size	Default
category <id>	Application category ID list. Category IDs.	integer	Minimum value: 0 Maximum value: 4294967295	
comment	Comments.	var-string	Maximum length: 255	
name	Application group name.	string	Maximum length: 63	
popularity	Application popularity filter.	option	-	1 2 3 4 5

Option	Description
1	Popularity level 1.
2	Popularity level 2.
3	Popularity level 3.
4	Popularity level 4.
5	Popularity level 5.

protocols	Application protocol filter.	user	Not Specified	all
risk <level>	Risk, or impact, of allowing traffic from this application to occur (1 - 5; Low, Elevated, Medium, High, and Critical). Risk, or impact, of allowing traffic from this application to occur (1 - 5; Low, Elevated, Medium, High, and Critical).	integer	Minimum value: 0 Maximum value: 4294967295	
technology	Application technology filter.	user	Not Specified	all
type	Application group type.	option	-	application

Option	Description
application	Application ID.
filter	Application filter.

vendor	Application vendor filter.	user	Not Specified	all

Configuration

To create an application group in the CLI:

```
config application group
    edit <name>
        set application <id1>, <id2>, ...
```

```
        set behavior {user}
        set category <id1>, <id2>, ...
        set comment {var-string}
        set popularity {option1}, {option2}, ...
        set protocols {user}
        set risk <level1>, <level2>, ...
        set technology {user}
        set type [application|filter]
        set vendor {user}
    next
end
```

To create an application group with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X POST
    -d '{
        "data": {
            "name": <sensor_name>,
            "type": <group_type>,
            ...
        }
    }'
    http://localhost/api/v2/cmdb/application/group
```

Configuring an application sensor

An application sensor includes a list of entries that define the actions to take for matched applications.

Each entry in an application sensor defines the applications you want to control. You can add applications and filters using categories, application overrides, and filter overrides with the actions to take when matched.

A default network service defines the default actions to take for the specified protocols and ports.

Application sensors can be configured through the CLI or the REST API.

Options

The following configuration options are available:

Parameter	Description	Type	Size	Default
app-replacemsg	Enable/disable replacement messages for blocked applications.	option	-	enable

Option	Description
disable	Disable replacement messages for blocked applications.
enable	Enable replacement messages for blocked applications.

Parameter	Description	Type	Size	Default
comment	comments	var-string	Maximum length: 255	
control-default-network-services	Enable/disable enforcement of protocols over selected ports.	option	-	disable

Option	Description
disable	Disable protocol enforcement over selected ports.
enable	Enable protocol enforcement over selected ports.

deep-app-inspection	Enable/disable deep application inspection.	option	-	enable

Option	Description
disable	Disable deep application inspection.
enable	Enable deep application inspection.

enforce-default-app-port	Enable/disable default application port enforcement for allowed applications.	option	-	disable

Option	Description
disable	Disable default application port enforcement.
enable	Enable default application port enforcement.

extended-log	Enable/disable extended logging.	option	-	disable

Option	Description
enable	Enable setting.
disable	Disable setting.

force-inclusion-ssl-di-sigs	Enable/disable forced inclusion of SSL deep inspection signatures.	option	-	disable

Option	Description
disable	Disable forced inclusion of signatures which normally require SSL deep inspection.
enable	Enable forced inclusion of signatures which normally require SSL deep inspection.

Parameter	Description	Type	Size	Default
name	List name.	string	Maximum length: 35	
options	Basic application protocol signatures allowed by default.	option	-	allow-dns

Option	Description
allow-dns	Allow DNS.
allow-icmp	Allow ICMP.
allow-http	Allow generic HTTP web browsing.
allow-ssl	Allow generic SSL communication.
allow-quic	Allow QUIC.

Parameter	Description	Type	Size	Default
other-application-action	Action for other applications.	option	-	pass

Option	Description
pass	Allow sessions matching an application in this application list.
block	Block sessions matching an application in this application list.

Parameter	Description	Type	Size	Default
other-application-log	Enable/disable logging for other applications.	option	-	disable

Option	Description
disable	Disable logging for other applications.
enable	Enable logging for other applications.

Parameter	Description	Type	Size	Default
p2p-block-list	P2P applications to be blocklisted.	option	-	

Option	Description
skype	Skype.
edonkey	Edonkey.
bittorrent	Bit torrent.

Parameter	Description	Type	Size	Default
replacemsg-group	Replacement message group.	string	Maximum length: 35	
unknown-application-action	Pass or block traffic from unknown applications.	option	-	pass

Parameter	Description	Type	Size	Default
	Option / **Description** *pass* / Pass or allow unknown applications. *block* / Drop or block unknown applications.			
unknown-application-log	Enable/disable logging for unknown applications.	option	-	disable
	Option / **Description** *disable* / Disable logging for unknown applications. *enable* / Enable logging for unknown applications.			

config default-network-services

Parameter	Description	Type	Size	Default
id	Entry ID.	integer	Minimum value: 0 Maximum value: 4294967295	0
port	Port number.	integer	Minimum value: 0 Maximum value: 65535	0
services	Network protocols.	option	-	
	Option / **Description** *http* / HTTP. *ssh* / SSH. *telnet* / TELNET. *ftp* / FTP. *dns* / DNS. *smtp* / SMTP. *pop3* / POP3. *imap* / IMAP. *snmp* / SNMP. *nntp* / NNTP. *https* / HTTPS.			

Parameter	Description	Type	Size	Default
violation-action	Action for protocols not in the allowlist for selected port.	option	-	block

Option	Description
pass	Allow protocols not in the allowlist for selected port.
monitor	Monitor protocols not in the allowlist for selected port.
block	Block protocols not in the allowlist for selected port.

config entries

Parameter	Description	Type	Size	Default
action	Pass or block traffic, or reset connection for traffic from this application.	option	-	block

Option	Description
pass	Pass or allow matching traffic.
block	Block or drop matching traffic.
reset	Reset sessions for matching traffic.

application <id>	ID of allowed applications. Application IDs.	integer	Minimum value: 0 Maximum value: 4294967295	
behavior	Application behavior filter.	user	Not Specified	all
category <id>	Category ID list. Application category ID.	integer	Minimum value: 0 Maximum value: 4294967295	
exclusion <id>	ID of excluded applications. Excluded application IDs.	integer	Minimum value: 0 Maximum value: 4294967295	
id	Entry ID.	integer	Minimum value: 0 Maximum value: 4294967295	0

Parameter	Description	Type	Size	Default
log	Enable/disable logging for this application list.	option	-	enable
	Option / **Description** *disable* — Disable logging. *enable* — Enable logging.			
log-packet	Enable/disable packet logging.	option	-	disable
	Option / **Description** *disable* — Disable packet logging. *enable* — Enable packet logging.			
per-ip-shaper	Per-IP traffic shaper.	string	Maximum length: 35	
popularity	Application popularity filter.	option	-	1 2 3 4 5
	Option / **Description** *1* — Popularity level 1. *2* — Popularity level 2. *3* — Popularity level 3. *4* — Popularity level 4. *5* — Popularity level 5.			
protocols	Application protocol filter.	user	Not Specified	all
quarantine	Quarantine method.	option	-	none
	Option / **Description** *none* — Quarantine is disabled. *attacker* — Block all traffic sent from attacker's IP address. The attacker's IP address is also added to the banned user list. The target's address is not affected.			
quarantine-expiry	Duration of quarantine. Requires quarantine set to attacker.	user	Not Specified	5m
quarantine-log	Enable/disable quarantine logging.	option	-	enable
	Option / **Description** *disable* — Disable quarantine logging. *enable* — Enable quarantine logging.			

Parameter	Description	Type	Size	Default
rate-count	Count of the rate.	integer	Minimum value: 0 Maximum value: 65535	0
rate-duration	Duration (sec) of the rate.	integer	Minimum value: 1 Maximum value: 65535	60
rate-mode	Rate limit mode.	option	-	continuous

Option	Description
periodical	Allow configured number of packets every rate-duration.
continuous	Block packets once the rate is reached.

rate-track	Track the packet protocol field.	option	-	none

Option	Description
none	none
src-ip	Source IP.
dest-ip	Destination IP.
dhcp-client-mac	DHCP client.
dns-domain	DNS domain.

risk `<level>`	Risk, or impact, of allowing traffic from this application to occur (1 - 5; Low, Elevated, Medium, High, and Critical). Risk, or impact, of allowing traffic from this application to occur (1 - 5; Low, Elevated, Medium, High, and Critical).	integer	Minimum value: 0 Maximum value: 4294967295	
session-ttl	Session TTL.	integer	Minimum value: 0 Maximum value: 4294967295	0
shaper	Traffic shaper.	string	Maximum length: 35	
shaper-reverse	Reverse traffic shaper.	string	Maximum length: 35	
technology	Application technology filter.	user	Not Specified	all
vendor	Application vendor filter.	user	Not Specified	all

config parameters

Parameter	Description	Type	Size	Default
id	Parameter tuple ID.	integer	Minimum value: 0 Maximum value: 4294967295	0

config members

Parameter	Description	Type	Size	Default
id	Parameter.	integer	Minimum value: 0 Maximum value: 4294967295	0
name	Parameter name.	string	Maximum length: 31	
value	Parameter value.	string	Maximum length: 199	

Configuration

To create an application control sensor in the CLI:

```
config application list
    edit <name>
        set app-replacemsg [disable|enable]
        set comment {var-string}
        set control-default-network-services [disable|enable]
        set deep-app-inspection [disable|enable]
        config default-network-services
            edit <id>
                set port {integer}
                set services {option1}, {option2}, ...
                set violation-action [pass|monitor|...]
            next
        end
        set enforce-default-app-port [disable|enable]
        config entries
            edit <id>
                set risk <level1>, <level2>, ...
                set category <id1>, <id2>, ...
                set application <id1>, <id2>, ...
                set protocols {user}
                set vendor {user}
                set technology {user}
                set behavior {user}
```

```
                set popularity {option1}, {option2}, ...
                set exclusion <id1>, <id2>, ...
                config parameters
                    edit <id>
                        config members
                            edit <id>
                                set name {string}
                                set value {string}
                            next
                        end
                    next
                end
                set action [pass|block|...]
                set log [disable|enable]
                set log-packet [disable|enable]
                set session-ttl {integer}
                set quarantine [none|attacker]
                set quarantine-expiry {user}
                set quarantine-log [disable|enable]
            next
        end
        set extended-log [enable|disable]
        set force-inclusion-ssl-di-sigs [disable|enable]
        set options {option1}, {option2}, ...
        set other-application-action [pass|block]
        set other-application-log [disable|enable]
        set p2p-block-list {option1}, {option2}, ...
        set replacemsg-group {string}
        set unknown-application-action [pass|block]
        set unknown-application-log [disable|enable]
    next
end
```

To create an application control sensor with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X POST
    -d '{
        "data": {
            "name": <sensor_name>,
            ...
            "entries": [
                {
                    "id": <entry_id1>,
                    ...
                },
                {
                    "id": <entry_id2>,
                }
            ],
            "default-network-services": [
                {
                    "id": <service1>,
                    "port": <port>,
                    "services": <services>,
```

```
                    "violation-action": <action>
                },
                {
                    "id": <service2>,
                    "port": <port>,
                    "services": <services>,
                    "violation-action": <action>
                }
            ],
        }
    }'
    http://localhost/api/v2/cmdb/application/list
```

Configuring category filters

Category filters can be configured through the CLI or the REST API.

To configure category filters in the CLI:

```
config application list
    edit <name>
        config entries
            edit <id>
                set category <id1>, <id2>, ...
                set action {pass | block | reset}
                set quarantine {none | attacker}
                set quarantine-expiry <###d##h##m>
                set log {enable | disable}
            next
        end
    next
end
```

To configure category filters with the REST API:

```
curl -H "Content-Type: application/json" -X PUT
    -d '{
        "data":
           {
            "entries": [
                {
                    "id": <id>,
                    "category": <category_list>,
                    "action": <action>,
                    "quarantine": <quarantine>,
                    "quarantine-expiry": <duration>,
                    "log": <log_enable>
                }
            ]
        }
    }'
    http://localhost/api/v2/cmdb/application/list/<name>
```

Available category filters

Category ID	Category
2	P2P
3	VoIP
5	Video/Audio
6	Proxy
7	Remote.Access
8	Game
12	General.Interest
15	Network.Service
17	Update
21	Email
22	Storage.Backup
23	Social.Media
25	Web.Client
26	Industrial
28	Collaboration
29	Business
30	Cloud.IT
31	Mobile
32	Unknown Applications

Configuring overrides

Multiple application signatures can be added for one sensor with a designated action. Filters can be added based on behavior, application category, popularity, protocol, risk, technology, or vendor subtypes.

Overrides can be configured through the CLI or the REST API.

To configure overrides in the CLI:

```
config application list
    edit <name>
        config entries
            edit <id>
                set protocols <integer>
                set risk <integer>
                set vendor <id>
```

```
                set technology <id>
                set behavior <id>
                set popularity <integer>
                set action {pass | block | reset}
                set log {enable | disable}
            next
        end
    next
end
```

To configure overrides with the REST API:

```
curl -H "Content-Type: application/json" -X PUT
    -d '{
        "data":
           {
            "name": <name>,
            "entries": [
                {
                    "id": <id>,
                    "protocols": <protocol_filter>,
                    "risk": <risk_level>,
                    "vendor": <vendor_filter>,
                    "technology": <technology_filter>,
                    "behavior": <behavior_filter>,
                    "popularity": <popularity_level>,
                    "action": <action>,
                    "log": <log_enable>
                }
            ]
        }
    }'
    http://localhost/api/v2/cmdb/application/list/<name>
```

Available filters

Protocol filters

Protocol ID	Protocol
all	all
0	Other
1	TCP
2	UDP
3	ICMP
5	TFN
6	P2P
7	IM

Protocol ID	Protocol
8	BO
9	HTTP
12	TELNET
13	FTP
14	DNS
15	SMTP
16	POP3
17	IMAP
18	SNMP
19	RADIUS
20	LDAP
21	MSSQL
22	RPC
11	SIP
23	H323
24	NBSS
25	DCERPC
10	SSH
26	SSL
27	NNTP
28	RTSP
29	DHCP
30	DNP3
31	SCCP
32	RTP
33	RDT
34	RTCP
37	IEC104
38	MODBUS
39	CAPWAP

Protocol ID	Protocol
40	RAWTCP
41	MISC
42	FTGD
43	HTTPS
44	SMTPS
45	POP3S
46	IMAPS
47	FTPS
51	RLDNP3

Vendor filters

Vendor ID	Vendor
all	all
0	Other
1	IIS
2	Apache
3	Oracle
4	MSSQL
5	MySQL
6	DB2
7	PostgreSQL
8	IE
9	Mozilla
10	Netscape
11	MS_Office
12	MS_Exchange
13	Sendmail
14	MailEnable
15	MediaPlayer
16	Real
17	Winamp

Vendor ID	Vendor
18	Cisco
19	Ipswitch
20	Apple
21	CA
22	Veritas
23	Adobe
24	SAP
25	Sun
26	HP
27	IBM
28	Novell
29	Samba
30	SCADA
31	PHP_app
32	CGI_app
33	ASP_app
34	IM
35	P2P

Technology filters

Technology ID	Technology
all	all
0	Network-Protocol
1	Browser-Based
2	Client-Server
4	Peer-to-Peer

Behavior filters

Behavior ID	Behavior
all	all
2	Botnet

Behavior ID	Behavior
3	Evasive
5	Excessive-Bandwidth
6	Tunneling
9	Cloud

Excluding signatures in application control profiles

In an application control list entry, use the `exclusion` option to specify a list of applications to exclude from the entry. When excluded, the application is no longer processed in this entry, but may match in subsequent entries.

Application signatures may be excluded through the CLI or the REST API.

To configure a signature exclusion in the CLI:

```
config application list
    edit <name>
        config entries
            edit <id>
                set category <id>
                set exclusion <application id>
                set action {pass | block | reset}
            next
        end
    next
end
```

To configure a signature exclusion with the REST API:

```
curl -H "Content-Type: application/json" -X PUT
    -d '{
        "data":
           {
            "name": <name>,
            "entries": [
                {
                    "id": <id>,
                    "category": <category>,
                    "exclusion": <application_id>,
                    "action": <action>,
                }
            ]
        }
    }'
    http://localhost/api/v2/cmdb/application/list/<name>
```

Intrusion prevention

Container FortiOS Intrusion Prevention System (IPS) detects network attacks and prevents threats from compromising the network, including protected devices. IPS utilizes signatures, protocol decoders, heuristics (or behavioral monitoring), threat intelligence (such as FortiGuard Labs), and advanced threat detection in order to prevent exploitation of known and unknown zero-day threats. Container FortiOS IPS is capable of performing deep packet inspection to scan encrypted payloads in order to detect and prevent threats from attackers.

IPS signatures offer a solution to detect and block exploitation of many vulnerabilities before they enter the network.

IPS signatures

FortiGuard Labs uses AI and machine learning to analyze billions of events every day. The FortiGuard Labs research team also proactively performs threat research to discover new vulnerabilities and exploitation, and produces signatures to identify such threats. These IPS signatures are delivered to Container FortiOS daily, so that the IPS engine is armed with the latest databases to match the latest threats.

IPS sensors

An IPS sensor is a collection of IPS signatures and filters that define the scope of what the IPS engine will scan when the IPS sensor is applied. An IPS sensor can have multiple sets of signatures and filters. A set of IPS signatures consists of manually selected signatures, while a set of IPS filters consists of filters based on signature attributes like target, severity, protocol, OS, and application. Each signature has predefined attributes and an action, such as block, allow, pass (monitor), quarantine, and reset. It is also possible to create custom IPS signatures to apply to an IPS sensor.

From the *Security Profiles* > *Intrusion Prevention* pane, you can create new IPS sensors and view a list of predefined sensors.

Container FortiOS includes the following predefined IPS sensors with associated predefined signatures:

Predefined IPS sensors	Description
all_default	Filters all predefined signatures, and sets `action` to the default action of the signature.
all_default_pass	Filters all predefined signatures, and sets `action` to `pass` (monitor).
default	Filters all predefined signatures with `severity` of `critical`, `high`, or `medium` and sets `action` to the default action of the signature.
high_security	Filters all predefined signatures with `severity` of `critical`, `high`, or `medium` and sets `action` to Block. For `low` severity signatures, sets action to the default action of the signature.
protect_client	Protects against client-side vulnerabilities by filtering on `location=client`. Sets `action` to the default action of the signature.
protect_email_server	Protects against email server-side vulnerabilities by filtering on `location=server` and `service=IMAP`, `POP3` or `SMTP`. Sets `action` to the default action of the signature.
protect_http_server	Protects against HTTP server-side vulnerabilities by filtering on `location=server` and `service=HTTP`. Sets `action` to the default action of the signature.

For more information about IPS, see Intrusion Prevention in the FortiOS Administration Guide.

This section contains the following topics:

Configuring an IPS sensor

Configure IPS sensors to be used in security policies.

Each entry defines the signatures or filters to use and actions to take when the rule is matched.

You may also configure IP addresses to exempt from the IPS rule.

IPS sensors can be configured through the CLI or the REST API.

Options

The following configuration options are available:

Parameter	Description	Type	Size	Default
block-malicious-url	Enable/disable malicious URL blocking.	option	-	disable
	Option / **Description** *disable* — Disable malicious URL blocking. *enable* — Enable malicious URL blocking.			
comment	Comment.	var-string	Maximum length: 255	
extended-log	Enable/disable extended logging.	option	-	disable
	Option / **Description** *enable* — Enable setting. *disable* — Disable setting.			
name	Sensor name.	string	Maximum length: 35	
replacemsg-group	Replacement message group.	string	Maximum length: 35	
scan-botnet-connections	Block or monitor connections to Botnet servers, or disable Botnet scanning.	option	-	disable

Parameter	Description	Type	Size	Default
	Option — **Description** *disable* — Do not scan connections to botnet servers. *block* — Block connections to botnet servers. *monitor* — Log connections to botnet servers.			

config entries

Parameter	Description	Type	Size	Default
action	Action taken with traffic in which signatures are detected.	option	-	default
	Option — **Description** *pass* — Pass or allow matching traffic. *block* — Block or drop matching traffic. *reset* — Reset sessions for matching traffic. *default* — Pass or drop matching traffic, depending on the default action of the signature.			
application	Applications to be protected. set application ? lists available applications. all includes all applications. other includes all unlisted applications.	user	Not Specified	all
cve <cve-entry>	List of CVE IDs of the signatures to add to the sensor CVE IDs or CVE wildcards.	string	Maximum length: 19	
id	Rule ID in IPS database.	integer	Minimum value: 0 Maximum value: 4294967295	0
location	Protect client or server traffic.	user	Not Specified	all
log	Enable/disable logging of signatures included in filter.	option	-	enable
	Option — **Description** *disable* — Disable logging of selected rules. *enable* — Enable logging of selected rules.			
log-attack-context	Enable/disable logging of attack context: URL buffer, header buffer, body buffer, packet buffer.	option	-	disable

Parameter	Description	Type	Size	Default
	Option / **Description** *disable* — Disable logging of detailed attack context. *enable* — Enable logging of detailed attack context.			
log-packet	Enable/disable packet logging. Enable to save the packet that triggers the filter. You can download the packets in pcap format for diagnostic use.	option	-	disable
	Option / **Description** *disable* — Disable packet logging of selected rules. *enable* — Enable packet logging of selected rules.			
os	Operating systems to be protected. all includes all operating systems. other includes all unlisted operating systems.	user	Not Specified	all
protocol	Protocols to be examined. set protocol ? lists available protocols. all includes all protocols. other includes all unlisted protocols.	user	Not Specified	all
quarantine	Quarantine method.	option	-	none
	Option / **Description** *none* — Quarantine is disabled. *attacker* — Block all traffic sent from attacker's IP address. The attacker's IP address is also added to the banned user list. The target's address is not affected.			
quarantine-expiry	Duration of quarantine. Requires quarantine set to attacker.	user	Not Specified	5m
quarantine-log	Enable/disable quarantine logging.	option	-	enable
	Option / **Description** *disable* — Disable quarantine logging. *enable* — Enable quarantine logging.			
rate-count	Count of the rate.	integer	Minimum value: 0 Maximum value: 65535	0

Parameter	Description	Type	Size	Default
rate-duration	Duration (sec) of the rate.	integer	Minimum value: 1 Maximum value: 65535	60
rate-mode	Rate limit mode.	option	-	continuous

Option	Description
periodical	Allow configured number of packets every rate-duration.
continuous	Block packets once the rate is reached.

Parameter	Description	Type	Size	Default
rate-track	Track the packet protocol field.	option	-	none

Option	Description
none	none
src-ip	Source IP.
dest-ip	Destination IP.
dhcp-client-mac	DHCP client.
dns-domain	DNS domain.

Parameter	Description	Type	Size	Default
rule `<id>`	Identifies the predefined or custom IPS signatures to add to the sensor. Rule IPS.	integer	Minimum value: 0 Maximum value: 4294967295	
severity	Relative severity of the signature, from info to critical. Log messages generated by the signature include the severity.	user	Not Specified	all
status	Status of the signatures included in filter. default enables the filter and only use filters with default status of enable. Filters with default status of disable will not be used.	option	-	default

Option	Description
disable	Disable status of selected rules.
enable	Enable status of selected rules.
default	Default.

config exempt-ip

Parameter	Description	Type	Size	Default
dst-ip	Destination IP address and netmask.	ipv4-classnet	Not Specified	0.0.0.0 0.0.0.0
id	Exempt IP ID.	integer	Minimum value: 0 Maximum value: 4294967295	0
src-ip	Source IP address and netmask.	ipv4-classnet	Not Specified	0.0.0.0 0.0.0.0

Configuration

To create an IPS sensor in the CLI:

```
config ips sensor
    edit <name>
        set block-malicious-url [disable|enable]
        set comment {var-string}
        config entries
            Description: IPS sensor filter.
            edit <id>
                set action [pass|block|...]
                set application {user}
                set cve <cve-entry1>, <cve-entry2>, ...
                config exempt-ip
                    Description: Traffic from selected source or destination IP addresses is
exempt from this signature.
                    edit <id>
                        set dst-ip {ipv4-classnet}
                        set src-ip {ipv4-classnet}
                    next
                end
                set location {user}
                set log [disable|enable]
                set log-attack-context [disable|enable]
                set log-packet [disable|enable]
                set os {user}
                set protocol {user}
                set quarantine [none|attacker]
                set quarantine-expiry {user}
                set quarantine-log [disable|enable]
                set rate-count {integer}
                set rate-duration {integer}
                set rate-mode [periodical|continuous]
                set rate-track [none|src-ip|...]
                set rule <id1>, <id2>, ...
                set severity {user}
                set status [disable|enable|...]
```

```
            next
        end
        set extended-log [enable|disable]
        set replacemsg-group {string}
        set scan-botnet-connections [disable|block|...]
    next
end
```

To create an IPS sensor with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X POST
    -d '{
        "data": {
            "name": <sensor_name>,
            ...
            "entries": [
                {
                    "id": <entry_id1>,
                    ...
                },
                {
                    "id": <entry_id2>,
                }
            ]
        }
    }'
    http://localhost/api/v2/cmdb/ips/sensor
```

IPS global configuration options

You can configure IPS options globally for all sensors through the CLI or the REST API.

Options

The following configuration options are available:

Parameter	Description	Type	Size	Default
anomaly-mode	Global blocking mode for rate-based anomalies.	option	-	continuous

Option	Description
periodical	After an anomaly is detected, allow the number of packets per second according to the anomaly configuration.
continuous	Block packets once an anomaly is detected. Overrides individual anomaly settings.

Parameter	Description	Type	Size	Default
database	Regular or extended IPS database. Regular protects against the latest common and in-the-wild attacks. Extended includes protection from legacy attacks.	option	-	regular
	Option / **Description** *regular* — IPS regular database package. *extended* — IPS extended database package.			
deep-app-insp-db-limit	Limit on number of entries in deep application inspection database	integer	Minimum value: 0 Maximum value: 2147483647	0
deep-app-insp-timeout	Timeout for Deep application inspection.	integer	Minimum value: 0 Maximum value: 2147483647	0
engine-count	Number of IPS engines running. If set to the default value of 0, FortiOS sets the number to optimize performance depending on the number of CPU cores.	integer	Minimum value: 0 Maximum value: 255	0
exclude-signatures	Excluded signatures.	option	-	industrial
	Option / **Description** *none* — No signatures excluded. *industrial* — Exclude industrial signatures.			
fail-open	Enable to allow traffic if the IPS process crashes. Default is disable and IPS traffic is blocked when the IPS process crashes.	option	-	disable
	Option / **Description** *enable* — Enable IPS fail open. *disable* — Disable IPS fail open.			

Parameter	Description	Type	Size	Default
ngfw-max-scan-range	NGFW policy-mode app detection threshold.	integer	Minimum value: 0 Maximum value: 4294967295	4096
packet-log-queue-depth	Packet/pcap log queue depth per IPS engine.	integer	Minimum value: 128 Maximum value: 4096	128
session-limit-mode	Method of counting concurrent sessions used by session limit anomalies. Choose between greater accuracy (accurate) or improved performance (heuristics).	option	-	heuristic

Option	Description
accurate	Accurately count concurrent sessions, demands more resources.
heuristic	Use heuristics to estimate the number of concurrent sessions. Acceptable in most cases.

Parameter	Description	Type	Size	Default
socket-size	IPS socket buffer size. Max and default value depend on available memory. Can be changed to tune performance.	integer	Minimum value: 0 Maximum value: 512	256
sync-session-ttl	Enable/disable use of kernel session TTL for IPS sessions.	option	-	enable

Option	Description
enable	Enable use of kernel session TTL for IPS sessions.
disable	Disable use of kernel session TTL for IPS sessions.

Parameter	Description	Type	Size	Default
traffic-submit	Enable/disable submitting attack data found by this Container FortiOS to FortiGuard.	option	-	disable

Option	Description
enable	Enable traffic submit.
disable	Disable traffic submit.

config tls-active-probe

Parameter	Description	Type	Size	Default
interface	Specify outgoing interface to reach server.	string	Maximum length: 15	
interface-select-method	Specify how to select outgoing interface to reach server.	option	-	auto
	Option / **Description** *auto* — Set outgoing interface automatically. *sdwan* — Set outgoing interface by SD-WAN or policy routing rules. *specify* — Set outgoing interface manually.			
source-ip	Source IP address used for TLS active probe.	ipv4-address	Not Specified	0.0.0.0
source-ip6	Source IPv6 address used for TLS active probe.	ipv6-address	Not Specified	::

Configuration

To configure IPS global options in the CLI:

```
config ips global
    set anomaly-mode [periodical|continuous]
    set database [regular|extended]
    set deep-app-insp-db-limit {integer}
    set deep-app-insp-timeout {integer}
    set engine-count {integer}
    set exclude-signatures [none|industrial]
    set fail-open [enable|disable]
    set ngfw-max-scan-range {integer}
    set packet-log-queue-depth {integer}
    set session-limit-mode [accurate|heuristic]
    set socket-size {integer}
    set sync-session-ttl [enable|disable]
    config tls-active-probe
        Description: TLS active probe configuration.
        set interface {string}
        set interface-select-method [auto|sdwan|...]
        set source-ip {ipv4-address}
        set source-ip6 {ipv6-address}
    end
    set traffic-submit [enable|disable]
end
```

To configure IPS global options with the REST API:

The configuration options are the same as the CLI.

```
curl H "Content-Type: application/json" -X POST
    -d '{
        "data": {
            "fail-open":[enable|diable]
            ...
    }'
    http://localhost/api/v2/cmdb/ips/global
```

Web filter

Web filtering restricts or controls user access to web resources and can be applied to firewall policies using either policy-based or profile-based NGFW mode.

In Container FortiOS, there are three main components of web filtering:

- Web content filter: blocks web pages containing words or patterns that you specify.
- URL filter: uses URLs and URL patterns to block or exempt web pages from specific sources, or block malicious URLs discovered by FortiSandbox.
- FortiGuard Web Filtering service: provides many additional categories you can use to filter web traffic.

These components interact with each other to provide maximum control over what users on your network can view and protect your network from many internet content threats.

Web filters are applied in the following order:

1. URL filter
2. FortiGuard Web Filtering
3. Web content filter
4. Web script filter

Container FortiOS includes a *default* web filter profile.

You can customize this profile or create your own to manage network user access.

All web filter configuration can be done through the CLI or the REST API.

The following topics provide more information about web filters:

- URL filter on page 57
- FortiGuard filter on page 59
- Web content filter on page 60

For more advanced information about web filters, see the Web filter chapter in the FortiOS Administration Guide.

URL filter

URL filters match URLs with patterns and process the traffic according to the defined action (exempt, block, allow, monitor) for web pages that match the criteria. Once a URL filter is configured, it can be applied to a firewall policy.

URL filters in Container FortiOS operate in the same manner as FortiOS. For more details, see URL filter in the FortiOS Administration Guide.

The following examples show the process for creating and using a URL filter in the CLI and the REST API.

To create a URL filter for Facebook in the CLI:

1. Create the URL filter:

```
config webfilter urlfilter
    edit 1
        set name "filter-facebook"
        config entries
            edit 1
                set url "*facebook.com"
                set type wildcard
                set action block
            next
        end
    next
end
```

2. Apply the URL filter to a web filter profile:

```
config webfilter profile
    edit "new-webfilter"
        config web
            set urlfilter-table 1
        end
        config ftgd-wf
            ...
        end
    next
end
```

3. Apply the web filter profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "WF"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "ALL"
        set utm-status enable
        set logtraffic all
        set webfilter-profile "new-webfilter"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

To configure a URL filter for Facebook with the REST API:

1. Create the URL filter:

```
curl -H "Content-Type: application/json" -X POST
    -d '{
        "data": {
           "id": 1,
            "name": "facebook-filter",
            "entries": [
               {
                   "id": 1,
                    "url": "*facebook.com",
                           "type": "wildcard",
                    "action": "block"
                }
            ]
        }
    }'
    http://localhost/api/v2/cmdb/webfilter/urlfilter
```

2. Apply the URL filter to an existing web filter profile:

```
curl -H "Content-Type: application/json" -X PUT
    -d '{
        "data": {
           "name": "new-webfilter",
            "web": {
                "urlfilter-table": 1
            }
       }
    }'
    http://localhost/api/v2/cmdb/webfilter/profile/new-webfilter
```

3. Apply the web filter profile to an existing firewall policy:

```
curl -H "Content-Type: application/json" -X PUT
    -d '{
        "data": {
            "policyid": 1,
            "webfilter-profile": "new-webfilter"
       }
    }'
    http://localhost/api/v2/cmdb/firewall/policy/1
```

FortiGuard filter

The FortiGuard filter enhances the web filter features by sorting billions of web pages into a wide range of categories that users can allow or block.

FortiGuard filters in Container FortiOS operate in the same manner as FortiOS. For more details, see FortiGuard filter in the FortiOS Administration Guide.

To use this service, you must have a valid FortiGuard license.

Web content filter

Control access to web content by blocking webpages containing specific words or patterns.

Content filters in Container FortiOS operate in the same manner as FortiOS. For more details, see Content filter in the FortiOS Administration Guide.

VPN

Container FortiOS supports the configuration of IPsec VPN.

SSL VPN is not supported.

VPN can be configured through the CLI or the REST API.

The following sections provide configuration instructions and examples:

- Phase 1 configuration on page 61
- Phase 2 configuration on page 67
- Configuring site-to-site VPN with pre-shared key on page 72

For more information about IPsec VPN, see the FortiOS Administration Guide.

Phase 1 configuration

Phase 1 configuration primarily defines the parameters used in IKE (Internet Key Exchange) negotiation between the ends of the IPsec tunnel. The local end is the Container FortiOS interface that initiates the IKE negotiations. The remote end is the remote gateway that responds and exchanges messages with the initiator.

The purpose of phase 1 is to secure a tunnel with one bi-directional IKE SA (security association) for negotiating IKE phase 2 parameters.

Options

The following options are available:

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td>add-route</td><td>Enable/disable control addition of a route to peer destination selector.</td><td>option</td><td>-</td><td>enable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>disable</td><td>Do not add a route to destination of peer selector.</td></tr><tr><td>enable</td><td>Add route to destination of peer selector.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>authmethod</td><td>Authentication method.</td><td>option</td><td>-</td><td>psk</td></tr>
</table>

Parameter	Description	Type	Size	Default
	Option / **Description** *psk* — PSK authentication method. *signature* — Signature authentication method.			
auto-negotiate	Enable/disable automatic initiation of IKE SA negotiation.	option	-	enable
	Option / **Description** *enable* — Enable automatic initiation of IKE SA negotiation. *disable* — Disable automatic initiation of IKE SA negotiation.			
dhgrp	DH group.	option	-	14
	Option / **Description** *1* — DH Group 1. *2* — DH Group 2. *5* — DH Group 5. *14* — DH Group 14. *15* — DH Group 15. *16* — DH Group 16. *17* — DH Group 17. *18* — DH Group 18. *19* — DH Group 19. *20* — DH Group 20. *21* — DH Group 21. *27* — DH Group 27. *28* — DH Group 28. *29* — DH Group 29. *30* — DH Group 30. *31* — DH Group 31. *32* — DH Group 32.			
dpd	Dead Peer Detection mode.	option	-	on-demand

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>disable</td><td>Disable Dead Peer Detection.</td></tr><tr><td>on-idle</td><td>Trigger Dead Peer Detection when IPsec is idle.</td></tr><tr><td>on-demand</td><td>Trigger Dead Peer Detection when IPsec traffic is sent but no reply is received from the peer.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>dpd-retrycount</td><td>Number of DPD retry attempts.</td><td>integer</td><td>Minimum value: 0 Maximum value: 10</td><td>3</td></tr>
<tr><td>dpd-retryinterval</td><td>DPD retry interval.</td><td>user</td><td>Not Specified</td><td></td></tr>
<tr><td>fragmentation</td><td>Enable/disable fragment IKE message on re-transmission.</td><td>option</td><td>-</td><td>enable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>enable</td><td>Enable intra-IKE fragmentation support on re-transmission.</td></tr><tr><td>disable</td><td>Disable intra-IKE fragmentation support.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>fragmentation-mtu</td><td>IKE fragmentation MTU.</td><td>integer</td><td>Minimum value: 500 Maximum value: 16000</td><td>1200</td></tr>
<tr><td>interface</td><td>Local physical, aggregate, or VLAN outgoing interface.</td><td>string</td><td>Maximum length: 35</td><td></td></tr>
<tr><td>keepalive</td><td>NAT-T keep alive interval.</td><td>integer</td><td>Minimum value: 10 Maximum value: 900</td><td>10</td></tr>
<tr><td>keylife</td><td>Time to wait in seconds before phase 1 encryption key expires.</td><td>integer</td><td>Minimum value: 120 Maximum value: 172800</td><td>86400</td></tr>
<tr><td>localid</td><td>Local ID.</td><td>string</td><td>Maximum length: 63</td><td></td></tr>
<tr><td>localid-type</td><td>Local ID type.</td><td>option</td><td>-</td><td>auto</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>auto</td><td>Select ID type automatically.</td></tr><tr><td>fqdn</td><td>Use fully qualified domain name.</td></tr><tr><td>user-fqdn</td><td>Use user fully qualified domain name.</td></tr><tr><td>keyid</td><td>Use key-id string.</td></tr><tr><td>address</td><td>Use local IP address.</td></tr><tr><td>asn1dn</td><td>Use ASN.1 distinguished name.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>mode-cfg</td><td>Enable/disable configuration method.</td><td>option</td><td>-</td><td>disable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>disable</td><td>Disable Configuration Method.</td></tr><tr><td>enable</td><td>Enable Configuration Method.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>name</td><td>IPsec remote gateway name.</td><td>string</td><td>Maximum length: 15</td><td></td></tr>
<tr><td>peertype</td><td>Accept this peer type.</td><td>option</td><td>-</td><td>peer</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>any</td><td>Accept any peer ID.</td></tr><tr><td>one</td><td>Accept this peer ID.</td></tr><tr><td>dialup</td><td>Accept peer ID in dialup group.</td></tr><tr><td>peer</td><td>Accept this peer certificate.</td></tr><tr><td>peergrp</td><td>Accept this peer certificate group.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>proposal</td><td>Phase1 proposal.</td><td>option</td><td>-</td><td></td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>des-md5</td><td>des-md5</td></tr><tr><td>des-sha1</td><td>des-sha1</td></tr><tr><td>des-sha256</td><td>des-sha256</td></tr><tr><td>des-sha384</td><td>des-sha384</td></tr><tr><td>des-sha512</td><td>des-sha512</td></tr><tr><td>3des-md5</td><td>3des-md5</td></tr><tr><td>3des-sha1</td><td>3des-sha1</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

Parameter	Description	Type	Size	Default

Option	Description
3des-sha256	3des-sha256
3des-sha384	3des-sha384
3des-sha512	3des-sha512
aes128-md5	aes128-md5
aes128-sha1	aes128-sha1
aes128-sha256	aes128-sha256
aes128-sha384	aes128-sha384
aes128-sha512	aes128-sha512
aes128gcm-prfsha1	aes128gcm-prfsha1
aes128gcm-prfsha256	aes128gcm-prfsha256
aes128gcm-prfsha384	aes128gcm-prfsha384
aes128gcm-prfsha512	aes128gcm-prfsha512
aes192-md5	aes192-md5
aes192-sha1	aes192-sha1
aes192-sha256	aes192-sha256
aes192-sha384	aes192-sha384
aes192-sha512	aes192-sha512
aes256-md5	aes256-md5
aes256-sha1	aes256-sha1
aes256-sha256	aes256-sha256
aes256-sha384	aes256-sha384
aes256-sha512	aes256-sha512
aes256gcm-prfsha1	aes256gcm-prfsha1
aes256gcm-prfsha256	aes256gcm-prfsha256
aes256gcm-prfsha384	aes256gcm-prfsha384
aes256gcm-prfsha512	aes256gcm-prfsha512
chacha20poly1305-prfsha1	chacha20poly1305-prfsha1
chacha20poly1305-prfsha256	chacha20poly1305-prfsha256

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td></td><td>
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>chacha20poly1305-prfsha384</td><td>chacha20poly1305-prfsha384</td></tr>
<tr><td>chacha20poly1305-prfsha512</td><td>chacha20poly1305-prfsha512</td></tr>
<tr><td>aria128-md5</td><td>aria128-md5</td></tr>
<tr><td>aria128-sha1</td><td>aria128-sha1</td></tr>
<tr><td>aria128-sha256</td><td>aria128-sha256</td></tr>
<tr><td>aria128-sha384</td><td>aria128-sha384</td></tr>
<tr><td>aria128-sha512</td><td>aria128-sha512</td></tr>
<tr><td>aria192-md5</td><td>aria192-md5</td></tr>
<tr><td>aria192-sha1</td><td>aria192-sha1</td></tr>
<tr><td>aria192-sha256</td><td>aria192-sha256</td></tr>
<tr><td>aria192-sha384</td><td>aria192-sha384</td></tr>
<tr><td>aria192-sha512</td><td>aria192-sha512</td></tr>
<tr><td>aria256-md5</td><td>aria256-md5</td></tr>
<tr><td>aria256-sha1</td><td>aria256-sha1</td></tr>
<tr><td>aria256-sha256</td><td>aria256-sha256</td></tr>
<tr><td>aria256-sha384</td><td>aria256-sha384</td></tr>
<tr><td>aria256-sha512</td><td>aria256-sha512</td></tr>
<tr><td>seed-md5</td><td>seed-md5</td></tr>
<tr><td>seed-sha1</td><td>seed-sha1</td></tr>
<tr><td>seed-sha256</td><td>seed-sha256</td></tr>
<tr><td>seed-sha384</td><td>seed-sha384</td></tr>
<tr><td>seed-sha512</td><td>seed-sha512</td></tr>
</table>
</td><td></td><td></td><td></td></tr>
<tr><td>psksecret</td><td>Pre-shared secret for PSK authentication (ASCII string or hexadecimal encoded with a leading 0x).</td><td>password-3</td><td>Not Specified</td><td></td></tr>
<tr><td>reauth</td><td>Enable/disable re-authentication upon IKE SA lifetime expiration.</td><td>option</td><td>-</td><td>disable</td></tr>
<tr><td></td><td>
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>disable</td><td>Disable IKE SA re-authentication.</td></tr>
<tr><td>enable</td><td>Enable IKE SA re-authentication.</td></tr>
</table>
</td><td></td><td></td><td></td></tr>
</table>

Parameter	Description	Type	Size	Default
rekey	Enable/disable phase1 rekey.	option	-	enable
	Option / **Description** *enable* — Enable phase1 rekey. *disable* — Disable phase1 rekey.			
remote-gw	IPv4 address of the remote gateway's external interface.	ipv4-address	Not Specified	0.0.0.0

Phase 2 configuration

After phase 1 negotiations end successfully, phase 2 begins. In Phase 2, the VPN peer or client and Container FortiOS exchange keys again to establish a secure communication channel. The phase 2 proposal parameters select the encryption and authentication algorithms needed to generate keys for protecting the implementation details of security associations (SAs). The keys are generated automatically using a Diffie-Hellman algorithm.

The basic phase 2 settings associate IPsec phase 2 parameters with the phase 1 configuration that specifies the remote end point of the VPN tunnel. In most cases, you need to configure only basic Phase 2 settings, such as name, source, and destination.

Options

The following options are available:

Parameter	Description	Type	Size	Default
dhgrp	Phase2 DH group.	option	-	14
	Option / **Description** *1* — DH Group 1. *2* — DH Group 2. *5* — DH Group 5. *14* — DH Group 14. *15* — DH Group 15. *16* — DH Group 16. *17* — DH Group 17. *18* — DH Group 18.			

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>19</td><td>DH Group 19.</td></tr><tr><td>20</td><td>DH Group 20.</td></tr><tr><td>21</td><td>DH Group 21.</td></tr><tr><td>27</td><td>DH Group 27.</td></tr><tr><td>28</td><td>DH Group 28.</td></tr><tr><td>29</td><td>DH Group 29.</td></tr><tr><td>30</td><td>DH Group 30.</td></tr><tr><td>31</td><td>DH Group 31.</td></tr><tr><td>32</td><td>DH Group 32.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>dst-addr-type</td><td>Remote proxy ID type.</td><td>option</td><td>-</td><td>subnet</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>subnet</td><td>IPv4 subnet.</td></tr><tr><td>range</td><td>IPv4 range.</td></tr><tr><td>ip</td><td>IPv4 IP.</td></tr><tr><td>name</td><td>IPv4 firewall address or group name.</td></tr><tr><td>subnet6</td><td>IPv6 subnet.</td></tr><tr><td>range6</td><td>IPv6 range.</td></tr><tr><td>ip6</td><td>IPv6 IP.</td></tr><tr><td>name6</td><td>IPv6 firewall address or group name.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>dst-end-ip</td><td>Remote proxy ID IPv4 end.</td><td>ipv4-address-any</td><td>Not Specified</td><td>0.0.0.0</td></tr>
<tr><td>dst-name</td><td>Remote proxy ID name.</td><td>string</td><td>Maximum length: 79</td><td></td></tr>
<tr><td>dst-start-ip</td><td>Remote proxy ID IPv4 start.</td><td>ipv4-address-any</td><td>Not Specified</td><td>0.0.0.0</td></tr>
<tr><td>dst-subnet</td><td>Remote proxy ID IPv4 subnet.</td><td>ipv4-classnet-any</td><td>Not Specified</td><td>0.0.0.0 0.0.0.0</td></tr>
</table>

Parameter	Description	Type	Size	Default
keylifeseconds	Phase2 key life in time in seconds.	integer	Minimum value: 120 Maximum value: 172800	43200
name	IPsec tunnel name.	string	Maximum length: 35	
phase1name	Phase 1 determines the options required for phase 2.	string	Maximum length: 15	
proposal	Phase2 proposal.	option	-	

Option	Description
null-md5	null-md5
null-sha1	null-sha1
null-sha256	null-sha256
null-sha384	null-sha384
null-sha512	null-sha512
des-null	des-null
des-md5	des-md5
des-sha1	des-sha1
des-sha256	des-sha256
des-sha384	des-sha384
des-sha512	des-sha512
3des-null	3des-null
3des-md5	3des-md5
3des-sha1	3des-sha1
3des-sha256	3des-sha256
3des-sha384	3des-sha384
3des-sha512	3des-sha512
aes128-null	aes128-null
aes128-md5	aes128-md5
aes128-sha1	aes128-sha1

Parameter	Description	Type	Size	Default

Option	Description
aes128-sha256	aes128-sha256
aes128-sha384	aes128-sha384
aes128-sha512	aes128-sha512
aes128gcm	aes128gcm
aes192-null	aes192-null
aes192-md5	aes192-md5
aes192-sha1	aes192-sha1
aes192-sha256	aes192-sha256
aes192-sha384	aes192-sha384
aes192-sha512	aes192-sha512
aes256-null	aes256-null
aes256-md5	aes256-md5
aes256-sha1	aes256-sha1
aes256-sha256	aes256-sha256
aes256-sha384	aes256-sha384
aes256-sha512	aes256-sha512
aes256gcm	aes256gcm
chacha20poly1305	chacha20poly1305
aria128-null	aria128-null
aria128-md5	aria128-md5
aria128-sha1	aria128-sha1
aria128-sha256	aria128-sha256
aria128-sha384	aria128-sha384
aria128-sha512	aria128-sha512
aria192-null	aria192-null
aria192-md5	aria192-md5
aria192-sha1	aria192-sha1
aria192-sha256	aria192-sha256

Parameter	Description	Type	Size	Default
	Option / **Description** *aria192-sha384* / aria192-sha384 *aria192-sha512* / aria192-sha512 *aria256-null* / aria256-null *aria256-md5* / aria256-md5 *aria256-sha1* / aria256-sha1 *aria256-sha256* / aria256-sha256 *aria256-sha384* / aria256-sha384 *aria256-sha512* / aria256-sha512 *seed-null* / seed-null *seed-md5* / seed-md5 *seed-sha1* / seed-sha1 *seed-sha256* / seed-sha256 *seed-sha384* / seed-sha384 *seed-sha512* / seed-sha512			
src-addr-type	Local proxy ID type.	option	-	subnet
	Option / **Description** *subnet* / IPv4 subnet. *range* / IPv4 range. *ip* / IPv4 IP. *name* / IPv4 firewall address or group name. *subnet6* / IPv6 subnet. *range6* / IPv6 range. *ip6* / IPv6 IP. *name6* / IPv6 firewall address or group name.			
src-end-ip	Local proxy ID end.	ipv4-address-any	Not Specified	0.0.0.0
src-name	Local proxy ID name.	string	Maximum length: 79	
src-start-ip	Local proxy ID start.	ipv4-address-any	Not Specified	0.0.0.0

Parameter	Description	Type	Size	Default
src-subnet	Local proxy ID subnet.	ipv4-classnet-any	Not Specified	0.0.0.0 0.0.0.0

Configuring site-to-site VPN with pre-shared key

In this example, an IPsec VPN connects Container FortiOS (*cFOS*) to a remote FortiGate (*FGT*) peer authenticating with a pre-shared key.

To configure IPsec VPN authenticating a remote FortiGate peer with a pre-shared key in the CLI:

1. Configure the WAN interface and the default route. The WAN interface is the interface connected to the ISP. The IPsec tunnel is established over the WAN interface.
   a. Configure *cFOS*:

```
config system interface
    edit "eth0"
        set ip 172.16.200.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.3
        set device "eth0"
    next
end
```

   b. Configure the *FGT*:

```
config system interface
    edit "port25"
        set vdom "root"
        set ip 172.16.202.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.202.2
        set device "port25"
    next
end
```

2. Configure the protected internal interface that connects to the corporate internal network. Traffic from this interface routes out through the IPsec VPN tunnel.
   a. Configure *cFOS*:

```
config system interface
    edit "eth1"
        set ip 10.1.100.1 255.255.255.0
```

```
    next
end
```

b. Configure *FGT*:

```
config system interface
    edit "port9"
        set vdom "root"
        set ip 172.16.101.1 255.255.255.0
    next
end
```

3. Configure the phase 1 interface:

a. Configure *cFOS*:

```
config vpn ipsec phase1-interface
    edit "to_FGT"
        set interface "eth0"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample
    next
end
```

b. Configure *FGT*:

```
config vpn ipsec phase1-interface
    edit "to_cFOS"
        set interface "port25"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.200.1
        set psksecret  sample
    next
end
```

4. Configure the IPsec phase 2 interface:

a. Configure *cFOS*:

```
config vpn ipsec phase2-interface
    edit "to_FGT"
        set phase1name "to_FGT"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

b. Configure *FGT*:

```
config vpn ipsec phase2-interface
    edit "to_cFOS"
        set phase1name "to_cFOS"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
```

```
    next
end
```

5. Configure a static route to reach the remote protected subnet.
   a. Configure *cFOS*:

```
config router static
    edit 2
        set dst 172.16.101.0 255.255.255.0
        set device "to_FGT"
    next
end
```

   b. Configure *FGT*:

```
config router static
    edit 2
        set dst 10.1.100.0 255.255.255.0
        set device "to_cFOS"
    next
end
```

6. Configure two policies to allow bidirectional IPsec traffic flow through the IPsec VPN tunnel:
   a. Configure *cFOS*:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "to_FGT"
        set dstintf "eth1"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "eth1"
        set dstintf "to_FGT"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set service "ALL"
    next
end
```

   b. Configure *FGT*:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "to_cFOS"
        set dstintf "port9"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
        next
        edit 2
            set name "outbound"
            set srcintf "port9"
            set dstintf "to_cFOS"
            set srcaddr "172.16.101.0"
            set dstaddr "10.1.100.0"
            set action accept
            set schedule "always"
            set service "ALL"
        next
    end
```

Policies and objects

This section contains topics about configuring policies and objects:

Policy and object differences from FortiOS

Container FortiOS supports a limited set of FortiOS policy and object types. The following provides a brief overview of the major differences.

Policy support

The policy types that are supported in Container FortiOS are the following:

- Policy
- Security Policy (in policy-based NGFW mode)
- Central SNAT map (in policy-based NGFW mode)

Local-in and DoS policies are not supported.

Container FortiOS does not support `ipsec` action in firewall policies.

Object support

- Internet service database is not supported.
- IP pools are not supported.
- Virtual servers are not supported.
- Health check is not supported.
- Traffic shaping is not supported.
- Virtual IP: Container FortiOS only supports static NAT and access proxy VIP types.
- Firewall address objects only support subnet and IP range types.

Policies

Container FortiOS policies operate in the same manner as FortiOS.

Policies can be configured through the CLI and the REST API.

This section contains the following information:

See also Policy and object differences from FortiOS on page 77.

For more detailed information about policies, see Policies in the FortiOS Administration Guide.

Policy and object differences from FortiOS

Container FortiOS supports a limited set of FortiOS policy and object types. The following provides a brief overview of the major differences.

Policy support

The policy types that are supported in Container FortiOS are the following:

- Policy
- Security Policy (in policy-based NGFW mode)
- Central SNAT map (in policy-based NGFW mode)

Local-in and DoS policies are not supported.

Container FortiOS does not support `ipsec` action in firewall policies.

Object support

- Internet service database is not supported.
- IP pools are not supported.
- Virtual servers are not supported.
- Health check is not supported.
- Traffic shaping is not supported.
- Virtual IP: Container FortiOS only supports static NAT and access proxy VIP types.
- Firewall address objects only support subnet and IP range types.

NGFW mode

Container FortiOS supports both NGFW modes:

- **Profile-based**: The traditional mode. Create security profiles (antivirus, web filter, and so on) and apply them to a policy.
- **Policy-based**: Use applications and URL categories directly in policies. Security profiles are not required, although web filter URL categories and groups may be used.

In policy-based mode:

- Central NAT is always enabled. If no Central SNAT policy exists, you must create one. See Central SNAT on page 79 for more information.

- Pre-match rules are defined separately from security policies, and define broader rules, such as SSL inspection and user authentication.

If Container FortiOS operates in central NAT mode, rather than enabling source NAT in individual policies, create source NAT policies that apply to all matching traffic. In many cases, you may only need one SNAT policy for each interface pair. See Central SNAT on page 79.

For the full set of supported configuration options, see the Container FortiOS CLI Guide or the Container FortiOS REST API documentation on FNDN.

For more detailed information about policy-based NGFW mode, see NGFW policy in the FortiOS Administration Guide.

Configuring a security policy

In this example, policies are configured to grant access to Facebook and Gmail.

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

1. Configure a policy with SSL deep inspection:

```
config firewall policy
    edit 1
        set name "Policy-1"
        set srcintf "eth1"
        set dstintf "eth0"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "deep-inspection"
    next
end
```

2. Configure security policies:

```
config firewall security-policy
    edit 2
        set name "allow-QA-Facebook"
        set srcintf "eth1"
        set dstintf "eth0"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set application 15832
    next
    edit 4
        set name "allow-QA-Email"
        set srcintf "eth1"
        set dstintf "eth0"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set url-category 23
    next
end
```

Central SNAT

Central SNAT in Container FortiOS functions the same as in FortiOS.

It is disabled by default. If you set NGFW mode to policy-based, central SNAT is enabled implicitly. You may also enable it explicitly. When it is enabled, you must create a central SNAT policy.

Central SNAT can be configured through the CLI or the REST API.

To enable central SNAT, see Configuring system settings on page 13.

For the full set of supported configuration options, see the Container FortiOS CLI Guide or the Container FortiOS REST API documentation on FNDN.

For more detailed information about central SNAT, see Central SNAT in the FortiOS Administration Guide.

Configuring a central SNAT policy

In this example, central SNAT is configured to apply SNAT to all traffic from *eth1* to *eth0*.

To create a central SNAT policy in the CLI:

```
config firewall central-snat-map
    edit 1
        set srcintf "eth1"
        set dstintf "eth0"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

To create central SNAT policy with the REST API:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "1",
    "status": "enable",
    "srcintf": [{"name": "eth1"}],
    "dstintf": [{"name": "eth0"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "all" }],
}}' http://192.168.1.1/api/v2/cmdb/firewall/central-snat-map
```

Policy configuration examples

This section includes the following policy configuration examples:

- Configuring a 3-tuple policy on page 80
- Configuring a policy with an IPS sensor on page 82

See Configuring site-to-site VPN with pre-shared key on page 72 for an example of policy configuration for IPsec VPN.

See NGFW mode on page 77 for an example of a policy-based NGFW mode security policy.

See Central SNAT on page 79 for an example of a central SNAT map policy.

Configuring a 3-tuple policy

A 3-tuple firewall policy forwards traffic based on source IP, destination IP, and service. In this example, HTTPS traffic from subnet 172.19.0.0/16 is forwarded to a server at IP address 192.168.1.100.

Prerequisites

- External to Container FortiOS, configure your network static routes to direct traffic to 192.168.1.100 to the container *eth0* IP address.
- Configure the server at 192.168.1.100 to respond to HTTPS requests.

Examples

- CLI example on page 80
- REST API example on page 81

CLI example

The following process details the configuration of a 3-tuple policy in the CLI.

To configure a 3-tuple policy in the CLI:

1. Configure a source address object:

```
config firewall address
    edit "OfficeNetwork"
        set type ipmask
        set subnet 172.19.0.0 255.255.0.0
       next
end
```

2. Configure a destination address object:

```
config firewall address
    edit "ProtectedServer"
        set type ipmask
        set subnet 192.168.1.100 255.255.255.255
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set status enable
        set name "HTTPS to protected server"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "OfficeNetwork"
        set dstaddr "ProtectedServer"
        set service "HTTPS"
        set logtraffic "all"
        set action accept
```

```
    next
end
```

4. Configure a firewall policy to block all other traffic:

```
config firewall policy
    edit 2
        set status enable
        set name "Deny all inbound"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set logtraffic "all"
        set action deny
    next
end
```

5. In a web browser in subnet 172.19.0.0/16, go to https://192.168.1.100. The configured web page displays.
6. In a web browser from another subnet, go to https://192.168.1.100.

REST API example

The following process details the configuration of a 3-tuple policy with the REST API.

The REST API in this example is configured to listen on *eth1* address `192.168.1.1`.

To configure a 3-tuple policy with the REST API:

1. Configure a source address object:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "OfficeNetwork",
    "type": "ipmask",
    "subnet": "172.19.0.0 255.255.0.0"
}}' http://192.168.1.1/api/v2/cmdb/firewall/address
```

2. Configure a destination address object:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "ProtectedServer",
    "type": "ipmask",
    "subnet": "192.168.1.100 255.255.255.255"
}}' http://192.168.1.1/api/v2/cmdb/firewall/address
```

3. Configure the firewall policy:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "1",
    "name": "HTTPS to protected server",
    "status": "enable",
    "srcintf": [{"name": "eth0"}],
    "dstintf": [{"name": "eth1"}],
    "srcaddr": [{"name": "OfficeNetwork"}],
    "dstaddr": [{ "name": "ProtectedServer" }],
    "service": [{"name":"HTTPS"}],
    "action": "accept",
```

```
    "logtraffic": "all"
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

4. Configure a firewall policy to block all other traffic:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "2",
    "name": "Deny all inbound",
    "status": "enable",
    "srcintf": [{"name": "eth0"}],
    "dstintf": [{"name": "eth1"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "all" }],
    "service": [{"name":"ALL"}],
    "action": "deny",
    "logtraffic": "all"
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

5. In a web browser in subnet 172.19.0.0/16, go to https://192.168.1.100. The configured web page displays.
6. In a web browser from another subnet with a route to 192.168.1.100, go to https://192.168.1.100. The configured web page does not display.

Configuring a policy with an IPS sensor

In this example, all traffic passing through the container is scanned using an IPS sensor.

Examples

- CLI example on page 82
- REST API example on page 83

CLI example

The following process details the configuration of a policy with an IPS sensor in the CLI.

To configure a policy with IPS in the CLI:

1. Configure the firewall policy:

```
config firewall policy
    edit 1
        set status enable
        set utm-status enable
        set name "IPS policy"
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "deep-inspection"
        set ips-sensor "high_security"
        set action accept
        set logtraffic all
```

```
    next
end
```

2. Verify that the policy is scanning the traffic:
   a. In a connected device, run the following command, substituting `ip_address` with the IP address of one of the container ports:

      ```
      nmap <ip_address>
      ```

   b. In the container CLI, view the `utm-ips` log:

      ```
      execute log filter category utm-ips
      execute log display
      ```

      The log includes entries similar to the following:

      ```
      date=2024-04-01 time=16:00:01 eventtime=1711987201 tz="+0000" logid="0419016384"
      type="utm" subtype="ips" eventtype="signature" level="alert" severity="low"
      srcip=192.168.154.200 dstip=192.168.154.50 srcintf="eth1" dstintf="intf-0"
      sessionid=4846 action="detected" proto=6 service="TCP" policyid=1
      attack="Port.Scanning" srcport=41474 dstport=17 direction="outgoing" attackid=43814
      profile="high_security" incidentserialno=238026758 msg="applications: Port.Scanning"
      ```

REST API example

The following process details the configuration of a policy with an IPS sensor with the REST API.

The REST API in this example is configured to listen on *eth1* address `192.168.1.1`.

To configure a policy with IPS with the REST API:

1. Configure the firewall policy:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "1",
    "name": "IPS policy",
    "status": "enable",
    "srcintf": [{"name": "any"}],
    "dstintf": [{"name": "any"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "all" }],
    "service": [{"name":"ALL"}],
    "utm-status": "enable",
    "ssl-ssh-profile": "deep-inspection",
    "ips-sensor": "high_security",
    "action": "accept",
    "logtraffic": "all"
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

2. Verify that the policy is scanning the traffic:
   a. In a connected device, run the following command, substituting `ip_address` with the IP address of one of the container ports:

      ```
      nmap <ip_address>
      ```

   b. In the container CLI, view the `utm-ips` log:

      ```
      execute log filter category utm-ips
      ```

```
execute log display
```

The log includes entries similar to the following:

```
date=2024-04-01 time=16:00:01 eventtime=1711987201 tz="+0000" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" severity="low"
srcip=192.168.154.200 dstip=192.168.154.50 srcintf="eth1" dstintf="intf-0"
sessionid=4846 action="detected" proto=6 service="TCP" policyid=1
attack="Port.Scanning" srcport=41474 dstport=17 direction="outgoing" attackid=43814
profile="high_security" incidentserialno=238026758 msg="applications: Port.Scanning"
```

Objects

The following types of objects are available:

Addresses

Use address objects to define sources and destinations of network traffic.

The following types of address objects are supported in Container FortiOS:

- Subnet: `ipmask`
- IP range: `iprange`

Use address groups to collect multiple addresses into a single object.

For more information about addresses and address groups, see Address objects in the FortiOS Administration Guide.

Examples

To create an address in the CLI:

```
config firewall address
    edit "new-address"
        set type ipmask
        set subnet "192.168.100.100 255.255.255.255"
    next
end
```

To create an address with the REST API:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "new-address",
```

```
    "type": "ipmask",
    "subnet": "192.168.100.100 255.255.255.255"
}}' http://localhost/api/v2/cmdb/config/firewall/address
```

Protocol options

Use protocol options profiles to define the parameters for handling protocol-specific traffic.

As in FortiOS, you may configure multiple protocol options profiles since the requirements may differ between policies. You are limited to a single protocol options profile per policy, but you may use the same profile in multiple policies.

Container FortiOS supports the same set of protocols as FortiOS.

Unlike FortiOS, you may edit the `default` protocol options profile.

For the full set of supported configuration options, see the Container FortiOS CLI Guide or the Container FortiOS REST API documentation on FNDN.

For more detailed information about protocol option profiles, see Protocol options in the FortiOS Administration Guide.

Schedules

A schedule is used to determine when a policy is active.

Container FortiOS supports the following schedule types:

- `onetime`: A one-time schedule has a start date and time, an end date and time, and does not repeat.
- `recurring`: A recurring schedule repeats on the specified day of the week, with a start and end time.

Container FortiOS includes the following predefined schedules:

- `always`: This schedule always runs. It is defined as a recurring schedule that starts at `00:00`, ends at `00:00`, and repeats every day of the week.
- `default-darrp-optimize`: This schedule starts at `01:00`, ends at `01:30`, and repeats every day of the week.
- `none`: This schedule never runs.

You may also collect schedules into a schedule group to apply to a policy as a single item. A group can be used to create more complex schedules that include multiple ranges.

Schedules may be configured through the CLI and the REST API.

For the full set of supported configuration options, see the Container FortiOS CLI Guide or the Container FortiOS REST API documentation on FNDN.

Configuration

The following example shows the configuration of a recurring schedule that repeats every weekday from 9:00 AM to 5:00 PM.

To configure a repeating schedule in the CLI:

```
config firewall schedule recurring
    edit "OfficeHours"
        set start "09:00"
```

```
        set end "17:00"
        set day monday tuesday wednesday thursday friday
    next
end
```

To configure a repeating schedule with the REST API:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "OfficeHours",
    "start": "09:00",
    "end": "17:00",
    "day": "monday tuesday wednesday thursday friday"
}}' http://192.168.1.1/api/v2/cmdb/firewall.schedule/recurring
```

Services

Services define one or more protocols and port numbers associated with each service.

Container FortiOS includes many predefined services. You may also configure your own custom services.

Services may be configured through the CLI and the REST API.

For the full set of supported configuration options, see the Container FortiOS CLI Guide or the Container FortiOS REST API documentation on FNDN.

Configuration

The following example shows the configuration of a custom service matching TCP traffic on port 8080 which can then be used in a virtual IP or policy.

To configure a custom service in the CLI:

```
config firewall service custom
    edit "TCP_8080"
        set category "Web Access"
        set iprange "0.0.0.0"
        set tcp-portrange "8080"
    next
end
```

To configure a custom service with the REST API:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "TCP_8080",
    "category": "Web Access",
    "iprange": "0.0.0.0",
    "tcp-portrange": "8080"
    "protocol": "TCP/UDP/SCTP"
}}' http://localhost/api/v2/cmdb/firewall.service/custom
```

Virtual IPs

Use static virtual IPs (VIPs) to map external IP addresses to internal IP addresses.

Static VIPs are commonly used to map public IP addresses to resources behind Container FortiOS that use private IP addresses. You may map single ports or a range of ports to forward. You may also specify services to map to a port.

Virtual IPs are applied to policies in the same manner as address objects.

Container FortiOS supports the following VIP types:

- Static NAT
- Access proxy

Static NAT VIP is not supported on Kubernetes.

VIPs may be configured through the CLI and the REST API.

Options

The following configuration options are available:

Parameter	Description	Type	Size	Default
comment	Comment.	var-string	Maximum length: 255	
extintf	Interface connected to the source network that receives the packets that will be forwarded to the destination network.	string	Maximum length: 35	
extip	IP address or address range on the external interface that you want to map to an address or address range on the destination network.	user	Not Specified	
extport	Incoming port number range that you want to map to a port number range on the destination network.	user	Not Specified	
id	Custom defined ID.	integer	Minimum value: 0 Maximum value: 65535	0
mappedip `<range>`	IP address or address range on the destination network to which the external IP address is mapped. Mapped IP range.	string	Maximum length: 79	
mappedport	Port number range on the destination network to which the external port number range is mapped.	user	Not Specified	

Parameter	Description	Type	Size	Default
name	Virtual IP name.	string	Maximum length: 79	
portforward	Enable/disable port forwarding.	option	-	disable
	Option / **Description** *disable* — Disable port forward. *enable* — Enable port forward.			
portmapping-type	Port mapping type.	option	-	1-to-1
	Option / **Description** *1-to-1* — One to one. *m-to-n* — Many to many.			
protocol	Protocol to use when forwarding packets.	option	-	tcp
	Option / **Description** *tcp* — TCP. *udp* — UDP. *sctp* — SCTP. *icmp* — ICMP.			
service <name>	Service name. Service name.	string	Maximum length: 79	
type	Configure a static NAT or access proxy.	option	-	static-nat
	Option / **Description** *static-nat* — Static NAT. *access-proxy* — Access proxy.			

Configuration

To create a VIP in the CLI:

```
config firewall vip
    edit <name>
        set comment {var-string}
        set extintf {string}
        set extip {user}
        set id {integer}
        set mappedip <range1>, <range2>, ...
```

```
        set portforward [disable|enable]
        set service <name1>, <name2>, ...
        set type [static-nat|access-proxy]
    next
end
```

To create a VIP with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": <name>,
    "type": [static-nat|access-proxy],
    ...
}}' http://localhost/api/v2/cmdb/config/firewall/vip
```

Static NAT VIP examples

This section contains the following examples of VIP configuration:

- Configuring a one-to-one virtual IP on page 89
- Configuring a port forwarding virtual IP on page 90
- Configuring a virtual IP with services on page 91

Configuring a one-to-one virtual IP

In this example, a virtual IP (VIP)is configured to forward all traffic from external IP 10.1.100.199 to internal IP 172.16.200.55. The virtual IP is then applied to a policy.

To configure and use a V IP in the CLI:

1. Create a new virtual IP:

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

2. Apply the virtual IP to a policy:

```
config firewall policy
    edit 8
        set name "Example_Virtual_IP_in_Policy"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "all"
        set dstaddr "Internal_WebServer"
        set action accept
        set service "ALL"
```

```
    next
end
```

To configure and use a VIP with the REST API:

1. Create a new virtual IP:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "Internal_WebServer",
    "extip": "10.1.100.199",
    "extinf": "any",
    "mappedip": "172.16.200.55"
}}' http://localhost/api/v2/cmdb/firewall/vip
```

2. Apply the virtual IP to a policy:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "8",
    "name": "Example_Virtual_IP_in_Policy",
    "status": "enable",
    "srcintf": [{"name": "eth0"}],
    "dstintf": [{"name": "eth1"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "Internal_WebServer" }],
    "service": [{"name":"ALL"}],
    "action": "accept",
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

Configuring a port forwarding virtual IP

In this example, a virtual IP is configured to forward traffic from external IP 10.1.100.199 on port 8080 to port 80 on internal IP 172.16.200.55. The virtual IP is then applied to a policy.

To configure and use a virtual IP in the CLI:

1. Create a new virtual IP:

```
config firewall vip
    edit "Internal_WebServer"
        set extip 10.1.100.199
        set extintf "any"
        set mappedip "172.16.200.55"
        set portforward enable
        set protocol tcp
        set extport 8080
        set mappedport 80
    next
end
```

2. Apply the virtual IP to a policy:

```
config firewall policy
    edit 8
        set name "Example_Virtual_IP_in_Policy"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "all"
```

```
        set dstaddr "Internal_WebServer"
        set action accept
        set service "ALL"
    next
end
```

To configure and use a virtual IP with the REST API:

1. Create a new virtual IP:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "Internal_WebServer",
    "extip": "10.1.100.199",
    "extinf": "any",
    "mappedip": "172.16.200.55",
    "portforward": "enable",
    "protocol": "tcp",
    "extport": "8080",
    "mappedport": "80"
}}' http://localhost/api/v2/cmdb/config/firewall/vip
```

2. Apply the virtual IP to a policy:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "8",
    "name": "Example_Virtual_IP_in_Policy",
    "status": "enable",
    "srcintf": [{"name": "eth0"}],
    "dstintf": [{"name": "eth1"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "Internal_WebServer" }],
    "service": [{"name":"ALL"}],
    "action": "accept",
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

Configuring a virtual IP with services

In this example, a virtual IP is configured to forward traffic from external IP 10.1.100.199 on TCP ports 8080, 8081, and 8082 to port 80 on internal IP 172.16.200.55. The virtual IP is then applied to a policy.

To configure and use a virtual IP in the CLI:

1. Create custom service objects for TCP ports 8080, 8081, and 8082:

```
config firewall service custom
    edit "TCP_8080"
        set category "Web Access"
        set iprange "0.0.0.0"
        set tcp-portrange "8080"
    next
    edit "TCP_8081"
        set category "Web Access"
        set iprange "0.0.0.0"
        set tcp-portrange "8081"
    next
    edit "TCP_8082"
```

```
        set category "Web Access"
        set iprange "0.0.0.0"
        set tcp-portrange "8082"
    next
end
```

2. Add these services to a service group:

```
config firewall service group
    edit "Internal_WebAccess"
        set member "TCP_8080" "TCP_8081" "TCP_8082"
    next
end
```

3. Create a new virtual IP:

```
config firewall vip
    edit "Internal_WebServer"
        set extip 10.1.100.199
        set extintf "any"
        set mappedip "172.16.200.55"
        set portforward enable
        set service "Internal_WebAccess"
        set mappedport 80
    next
end
```

4. Apply the virtual IP to a policy:

```
config firewall policy
    edit 8
        set name "Example_Virtual_IP_in_Policy"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "all"
        set dstaddr "Internal_WebServer"
        set action accept
        set service "ALL"
    next
end
```

To configure and use a virtual IP with the REST API:

1. Create custom service objects for TCP ports 8080, 8081, and 8082:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "TCP_8080",
    "category": "Web Access",
    "iprange": "0.0.0.0",
    "tcp-portrange": "8080"
    "protocol": "TCP/UDP/SCTP"
}}' http://localhost/api/v2/cmdb/firewall.service/custom
```

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "TCP_8081",
    "category": "Web Access",
    "iprange": "0.0.0.0",
    "tcp-portrange": "8081"
```

```
    "protocol": "TCP/UDP/SCTP"
}}' http://localhost/api/v2/cmdb/firewall.service/custom

curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "TCP_8082",
    "category": "Web Access",
    "iprange": "0.0.0.0",
    "tcp-portrange": "8082"
    "protocol": "TCP/UDP/SCTP"
}}' http://localhost/api/v2/cmdb/firewall.service/custom
```

2. Add these services to a service group:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "Internal_WebAccess",
    "member": [
        {"name": "TCP_8080"},
        {"name": "TCP_8081"},
        {"name": "TCP_8082"}
    ]
}}' http://localhost/api/v2/cmdb/firewall.service/group
```

3. Create a new virtual IP:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "name": "Internal_WebServer",
    "extip": "10.1.100.199",
    "extinf": "any",
    "mappedip": "172.16.200.55",
    "portforward": "enable",
    "service": "Internal_WebAccess",
    "mappedport": "80"
}}' http://localhost/api/v2/cmdb/firewall/vip
```

4. Apply the virtual IP to a policy:

```
curl -H "Content-Type: application/json" -X POST -d '{ "data":{
    "policyid": "8",
    "name": "Example_Virtual_IP_in_Policy",
    "status": "enable",
    "srcintf": [{"name": "eth0"}],
    "dstintf": [{"name": "eth1"}],
    "srcaddr": [{"name": "all"}],
    "dstaddr": [{ "name": "Internal_WebServer" }],
    "service": [{"name":"ALL"}],
    "action": "accept",
}}' http://192.168.1.1/api/v2/cmdb/firewall/policy
```

Logging and reporting

Container FortiOS can write logs to the following destinations:

- Disk
- Memory
- Syslog server

Logging to memory is enabled by default.

Up to four syslog servers can be enabled.

Viewing logs

Logs can be viewed through the CLI or REST API.

The following log categories are available:

- traffic
- event
- utm-virus
- utm-webfilter
- utm-ips
- utm-app-ctrl
- utm-ssl

Only one category may be selected at a time.

`traffic` is the default.

To view the currently selected filters in the CLI:

```
execute log filter dump
```

To view logs in the CLI:

1. Specify the logging device:

   ```
   execute log filter device disk
   ```

2. Alternatively, specify the log category to display:

   ```
   execute log filter category traffic
   ```

3. View the logs:

   ```
   execute log display
   ```

To view logs with the REST API:

Configuring logging

Log settings determine what information is recorded in logs, where the logs are stored, and how often storage occurs.

Logging can be configured through the CLI and the REST API.

Log settings are divided into the following areas:

- General log settings
- Memory log settings
- Disk log settings
- Syslog device log settings

Options

The following configuration options are available:

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td>brief-traffic-format</td><td>Enable/disable brief format traffic logging.</td><td>option</td><td>-</td><td>disable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>enable</td><td>Enable brief format traffic logging.</td></tr><tr><td>disable</td><td>Disable brief format traffic logging.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>custom-log-fields <code><field-id></code></td><td>Custom fields to append to all log messages.
Custom log field.</td><td>string</td><td>Maximum length: 35</td><td></td></tr>
<tr><td>daemon-log</td><td>Enable/disable daemon logging.</td><td>option</td><td>-</td><td>disable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>enable</td><td>Enable daemon logging.</td></tr><tr><td>disable</td><td>Disable daemon logging.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>expolicy-implicit-log</td><td>Enable/disable explicit proxy firewall implicit policy logging.</td><td>option</td><td>-</td><td>disable</td></tr>
</table>

Parameter	Description	Type	Size	Default
	Option **Description** *enable* Enable explicit proxy firewall implicit policy logging. *disable* Disable explicit proxy firewall implicit policy logging.			
faz-override	Enable/disable override FortiAnalyzer settings.	option	-	disable
	Option **Description** *enable* Enable override FortiAnalyzer settings. *disable* Disable override FortiAnalyzer settings.			
fwpolicy-implicit-log	Enable/disable implicit firewall policy logging.	option	-	disable
	Option **Description** *enable* Enable implicit firewall policy logging. *disable* Disable implicit firewall policy logging.			
fwpolicy6-implicit-log	Enable/disable implicit firewall policy6 logging.	option	-	disable
	Option **Description** *enable* Enable implicit firewall policy6 logging. *disable* Disable implicit firewall policy6 logging.			
local-in-allow	Enable/disable local-in-allow logging.	option	-	disable
	Option **Description** *enable* Enable local-in-allow logging. *disable* Disable local-in-allow logging.			
local-in-deny-broadcast	Enable/disable local-in-deny-broadcast logging.	option	-	disable
	Option **Description** *enable* Enable local-in-deny-broadcast logging. *disable* Disable local-in-deny-broadcast logging.			
local-in-deny-unicast	Enable/disable local-in-deny-unicast logging.	option	-	disable

Parameter	Description	Type	Size	Default
	Option **Description** *enable* Enable local-in-deny-unicast logging. *disable* Disable local-in-deny-unicast logging.			
local-out	Enable/disable local-out logging.	option	-	enable
	Option **Description** *enable* Enable local-out logging. *disable* Disable local-out logging.			
log-invalid-packet	Enable/disable invalid packet traffic logging.	option	-	disable
	Option **Description** *enable* Enable invalid packet traffic logging. *disable* Disable invalid packet traffic logging.			
log-policy-comment	Enable/disable inserting policy comments into traffic logs.	option	-	disable
	Option **Description** *enable* Enable inserting policy comments into traffic logs. *disable* Disable inserting policy comments into traffic logs.			
log-user-in-upper	Enable/disable logs with user-in-upper.	option	-	disable
	Option **Description** *enable* Enable logs with user-in-upper. *disable* Disable logs with user-in-upper.			
neighbor-event	Enable/disable neighbor event logging.	option	-	disable
	Option **Description** *enable* Enable neighbor event logging. *disable* Disable neighbor event logging.			
resolve-ip	Enable/disable adding resolved domain names to traffic logs if possible.	option	-	disable

Parameter	Description	Type	Size	Default
	Option **Description** *enable* Enable adding resolved domain names to traffic logs. *disable* Disable adding resolved domain names to traffic logs.			
resolve-port	Enable/disable adding resolved service names to traffic logs.	option	-	enable
	Option **Description** *enable* Enable adding resolved service names to traffic logs. *disable* Disable adding resolved service names to traffic logs.			
syslog-override	Enable/disable override Syslog settings.	option	-	disable
	Option **Description** *enable* Enable override Syslog settings. *disable* Disable override Syslog settings.			
user-anonymize	Enable/disable anonymizing user names in log messages.	option	-	disable
	Option **Description** *enable* Enable anonymizing user names in log messages. *disable* Disable anonymizing user names in log messages.			

Custom log field options

Parameter	Description	Type	Size	Default
id	Field ID string.	string	Maximum length: 35	
name	Field name (max: 15 characters).	string	Maximum length: 15	
value	Field value (max: 63 characters).	string	Maximum length: 63	

Configuration

To configure general log settings in the CLI:

```
config log setting
    set brief-traffic-format [enable|disable]
    set custom-log-fields <field-id1>, <field-id2>, ...
    set daemon-log [enable|disable]
    set expolicy-implicit-log [enable|disable]
    set faz-override [enable|disable]
    set fwpolicy-implicit-log [enable|disable]
    set fwpolicy6-implicit-log [enable|disable]
    set local-in-allow [enable|disable]
    set local-in-deny-broadcast [enable|disable]
    set local-in-deny-unicast [enable|disable]
    set local-out [enable|disable]
    set log-invalid-packet [enable|disable]
    set log-policy-comment [enable|disable]
    set log-user-in-upper [enable|disable]
    set neighbor-event [enable|disable]
    set resolve-ip [enable|disable]
    set resolve-port [enable|disable]
    set syslog-override [enable|disable]
    set user-anonymize [enable|disable]
end
```

To configure general log settings with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "<setting>": <value>,
    ...,
    "custom-log-fields": [
        {'field-id': <field_id1>},
        {'field-id': <field_id2>},
        ...
    ]
}}' http://localhost/api/v2/cmdb/log/setting
```

Configuring memory logging

Container FortiOS by defaults stores logs locally in system memory.

Memory logging can be configured through the CLI or the REST API.

Options

Memory logging global settings

Parameter	Description	Type	Size	Default
full-final-warning-threshold	Log full final warning threshold as a percent.	integer	Minimum value: 3 Maximum value: 100	95
full-first-warning-threshold	Log full first warning threshold as a percent.	integer	Minimum value: 1 Maximum value: 98	75
full-second-warning-threshold	Log full second warning threshold as a percent.	integer	Minimum value: 2 Maximum value: 99	90
max-size	Maximum amount of memory that can be used for memory logging in bytes.	integer	Minimum value: 0 Maximum value: 4294967295	168439644 **

Memory logging settings

Parameter	Description	Type	Size	Default
status	Enable/disable logging to the FortiGate's memory.	option	-	enable **

Option	Description
enable	Enable logging to memory.
disable	Disable logging to memory.

Enabling memory logging

To enable memory logging in the CLI:

```
config log memory setting
    set status [enable|disable]
end
```

To enable memory logging with the REST API:

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "status": "enable",
```

```
    ...
}}' http://localhost/api/v2/cmdb/log.memory/setting
```

Configuration

To configure memory logging in the CLI:

```
config log memory global-setting
    set full-final-warning-threshold {integer}
    set full-first-warning-threshold {integer}
    set full-second-warning-threshold {integer}
    set max-size {integer}
end
```

To configure memory logging with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "full-final-warning-threshold": <integer>,
    "full-first-warning-threshold": <integer>,
    "full-second-warning-threshold": <integer>,
    "max-size": <integer>
}}' http://localhost/api/v2/cmdb/log.memory/global-setting
```

Configuring disk logging

You may enable logging to local disk. The destination disk is the data disk you defined at deployment.

Unlike FortiOS, Container FortiOS logs to memory by default, rather than to disk.

Disk logging can be configured through the CLI or the REST API.

Because Security Fabric features are not supported, Container FortiOS does not fall back to remote logging if disk logging is disabled.

Options

The following disk log settings are available:

Parameter	Description	Type	Size	Default
diskfull	Action to take when disk is full. The system can overwrite the oldest log messages or stop logging when the disk is full.	option	-	overwrite

Parameter	Description	Type	Size	Default
	Option / **Description** *overwrite* — Overwrite the oldest logs when the log disk is full. *nolog* — Stop logging when the log disk is full.			
dlp-archive-quota	DLP archive quota (MB).	integer	Minimum value: 0 Maximum value: 4294967295	0
full-final-warning-threshold	Log full final warning threshold as a percent.	integer	Minimum value: 3 Maximum value: 100	95
full-first-warning-threshold	Log full first warning threshold as a percent.	integer	Minimum value: 1 Maximum value: 98	75
full-second-warning-threshold	Log full second warning threshold as a percent.	integer	Minimum value: 2 Maximum value: 99	90
interface	Specify outgoing interface to reach server.	string	Maximum length: 15	
interface-select-method	Specify how to select outgoing interface to reach server.	option	-	auto
	Option / **Description** *auto* — Set outgoing interface automatically. *sdwan* — Set outgoing interface by SD-WAN or policy routing rules. *specify* — Set outgoing interface manually.			
ips-archive	Enable/disable IPS packet archiving to the local disk.	option	-	enable
	Option / **Description** *enable* — Enable IPS packet archiving. *disable* — Disable IPS packet archiving.			

Parameter	Description	Type	Size	Default
log-quota	Disk log quota (MB).	integer	Minimum value: 0 Maximum value: 4294967295	0
max-log-file-size	Maximum log file size before rolling.	integer	Minimum value: 1 Maximum value: 100	20
max-policy-packet-capture-size	Maximum size of policy sniffer in MB (0 means unlimited).	integer	Minimum value: 0 Maximum value: 4294967295	100
maximum-log-age	Delete log files older than (days).	integer	Minimum value: 0 Maximum value: 3650	7
report-quota	Report db quota (MB).	integer	Minimum value: 0 Maximum value: 4294967295	0
roll-day	Day of week on which to roll log file.	option	-	sunday
	Option / **Description** *sunday* — Sunday *monday* — Monday *tuesday* — Tuesday *wednesday* — Wednesday *thursday* — Thursday *friday* — Friday *saturday* — Saturday			
roll-schedule	Frequency to check log file for rolling.	option	-	daily
	Option / **Description** *daily* — Check the log file once a day. *weekly* — Check the log file once a week.			

Parameter	Description	Type	Size	Default
roll-time	Time of day to roll the log file (hh:mm).	user	Not Specified	
source-ip	Source IP address to use for uploading disk log files.	ipv4-address	Not Specified	0.0.0.0
status	Enable/disable local disk logging.	option	-	disable
	Option / **Description** *enable* — Log to local disk. *disable* — Do not log to local disk.			
upload	Enable/disable uploading log files when they are rolled.	option	-	disable
	Option / **Description** *enable* — Enable uploading log files when they are rolled. *disable* — Disable uploading log files when they are rolled.			
upload-delete-files	Delete log files after uploading.	option	-	enable
	Option / **Description** *enable* — Delete log files after uploading. *disable* — Do not delete log files after uploading.			
upload-destination	The type of server to upload log files to. Only FTP is currently supported.	option	-	ftp-server
	Option / **Description** *ftp-server* — Upload rolled log files to an FTP server.			
upload-ssl-conn	Enable/disable encrypted FTPS communication to upload log files.	option	-	default
	Option / **Description** *default* — FTPS with high and medium encryption algorithms. *high* — FTPS with high encryption algorithms. *low* — FTPS with low encryption algorithms. *disable* — Disable FTPS communication.			
uploaddir	The remote directory on the FTP server to upload log files to.	string	Maximum length: 63	

Parameter	Description	Type	Size	Default
uploadip	IP address of the FTP server to upload log files to.	ipv4-address	Not Specified	0.0.0.0
uploadpass	Password required to log into the FTP server to upload disk log files.	password	Not Specified	
uploadport	TCP port to use for communicating with the FTP server.	integer	Minimum value: 0 Maximum value: 65535	21
uploadsched	Set the schedule for uploading log files to the FTP server.	option	-	disable
	Option / **Description** *disable* — Upload when rolling. *enable* — Scheduled upload.			
uploadtime	Time of day at which log files are uploaded if uploadsched is enabled (hh:mm or hh).	user	Not Specified	
uploadtype	Types of log files to upload. Separate multiple entries with a space.	option	-	traffic event virus webfilter IPS emailfilter dlp-archive anomaly voip dlp app-ctrl waf dns ssh ssl
	Option / **Description** *traffic* — Upload traffic log. *event* — Upload event log. *virus* — Upload anti-virus log. *webfilter* — Upload web filter log. *IPS* — Upload IPS log. *emailfilter* — Upload spam filter log. *dlp-archive* — Upload DLP archive. *anomaly* — Upload anomaly log.			

Parameter	Description	Type	Size	Default
	<table><tr><th>Option</th><th>Description</th></tr><tr><td>voip</td><td>Upload VoIP log.</td></tr><tr><td>dlp</td><td>Upload DLP log.</td></tr><tr><td>app-ctrl</td><td>Upload application control log.</td></tr><tr><td>waf</td><td>Upload web application firewall log.</td></tr><tr><td>dns</td><td>Upload DNS log.</td></tr><tr><td>ssh</td><td>Upload SSH log.</td></tr><tr><td>ssl</td><td>Upload SSL log.</td></tr><tr><td>file-filter</td><td>Upload file-filter log.</td></tr><tr><td>icap</td><td>Upload ICAP log.</td></tr></table>			
uploaduser	Username required to log into the FTP server to upload disk log files.	string	Maximum length: 35	

Configuration

To configure disk logging in the CLI:

```
config log disk setting
    set diskfull [overwrite|nolog]
    set dlp-archive-quota {integer}
    set full-final-warning-threshold {integer}
    set full-first-warning-threshold {integer}
    set full-second-warning-threshold {integer}
    set interface {string}
    set interface-select-method [auto|sdwan|...]
    set ips-archive [enable|disable]
    set log-quota {integer}
    set max-log-file-size {integer}
    set max-policy-packet-capture-size {integer}
    set maximum-log-age {integer}
    set report-quota {integer}
    set roll-day {option1}, {option2}, ...
    set roll-schedule [daily|weekly]
    set roll-time {user}
    set source-ip {ipv4-address}
    set status [enable|disable]
    set upload [enable|disable]
    set upload-delete-files [enable|disable]
    set upload-destination {option}
    set upload-ssl-conn [default|high|...]
    set uploaddir {string}
    set uploadip {ipv4-address}
    set uploadpass {password}
    set uploadport {integer}
```

```
    set uploadsched [disable|enable]
    set uploadtime {user}
    set uploadtype {option1}, {option2}, ...
    set uploaduser {string}
end
```

To configure disk logging with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "<setting>": <value>,
    ...
}}' http://localhost/api/v2/cmdb/log.disk/setting
```

Configuring logging to syslog servers

You can configure Container FortiOS to send logs to up to four external syslog servers:

- syslogd
- syslogd2
- syslogd3
- syslogd4

Update the commands outlined below with the appropriate syslog server. For example, `config log` **`syslogd3`** `setting`.

Syslog server logging can be configured through the CLI or the REST API.

Options

The following configuration options are available:

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td>certificate</td><td>Certificate used to communicate with Syslog server.</td><td>string</td><td>Maximum length: 35</td><td></td></tr>
<tr><td>enc-algorithm</td><td>Enable/disable reliable syslogging with TLS encryption.</td><td>option</td><td>-</td><td>disable</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>high-medium</td><td>SSL communication with high and medium encryption algorithms.</td></tr><tr><td>high</td><td>SSL communication with high encryption algorithms.</td></tr><tr><td>low</td><td>SSL communication with low encryption algorithms.</td></tr><tr><td>disable</td><td>Disable SSL communication.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>facility</td><td>Remote syslog facility.</td><td>option</td><td>-</td><td>local7</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th><th>Type</th><th>Size</th><th>Default</th></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>kernel</td><td>Kernel messages.</td></tr><tr><td>user</td><td>Random user-level messages.</td></tr><tr><td>mail</td><td>Mail system.</td></tr><tr><td>daemon</td><td>System daemons.</td></tr><tr><td>auth</td><td>Security/authorization messages.</td></tr><tr><td>syslog</td><td>Messages generated internally by syslog.</td></tr><tr><td>lpr</td><td>Line printer subsystem.</td></tr><tr><td>news</td><td>Network news subsystem.</td></tr><tr><td>uucp</td><td>Network news subsystem.</td></tr><tr><td>cron</td><td>Clock daemon.</td></tr><tr><td>authpriv</td><td>Security/authorization messages (private).</td></tr><tr><td>ftp</td><td>FTP daemon.</td></tr><tr><td>ntp</td><td>NTP daemon.</td></tr><tr><td>audit</td><td>Log audit.</td></tr><tr><td>alert</td><td>Log alert.</td></tr><tr><td>clock</td><td>Clock daemon.</td></tr><tr><td>local0</td><td>Reserved for local use.</td></tr><tr><td>local1</td><td>Reserved for local use.</td></tr><tr><td>local2</td><td>Reserved for local use.</td></tr><tr><td>local3</td><td>Reserved for local use.</td></tr><tr><td>local4</td><td>Reserved for local use.</td></tr><tr><td>local5</td><td>Reserved for local use.</td></tr><tr><td>local6</td><td>Reserved for local use.</td></tr><tr><td>local7</td><td>Reserved for local use.</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>format</td><td>Log format.</td><td>option</td><td>-</td><td>default</td></tr>
<tr><td></td><td><table><tr><th>Option</th><th>Description</th></tr><tr><td>default</td><td>Syslog format.</td></tr><tr><td>csv</td><td>CSV (Comma Separated Values) format.</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

Parameter	Description	Type	Size	Default
	Option / **Description** *cef* — CEF (Common Event Format) format. *rfc5424* — Syslog RFC5424 format.			
interface	Specify outgoing interface to reach server.	string	Maximum length: 15	
interface-select-method	Specify how to select outgoing interface to reach server.	option	-	auto
	Option / **Description** *auto* — Set outgoing interface automatically. *sdwan* — Set outgoing interface by SD-WAN or policy routing rules. *specify* — Set outgoing interface manually.			
max-log-rate	Syslog maximum log rate in MBps (0 = unlimited).	integer	Minimum value: 0 Maximum value: 100000	0
mode	Remote syslog logging over UDP/Reliable TCP.	option	-	udp
	Option / **Description** *udp* — Enable syslogging over UDP. *legacy-reliable* — Enable legacy reliable syslogging by RFC3195 (Reliable Delivery for Syslog). *reliable* — Enable reliable syslogging by RFC6587 (Transmission of Syslog Messages over TCP).			
port	Server listen port.	integer	Minimum value: 0 Maximum value: 65535	514
priority	Set log transmission priority.	option	-	default
	Option / **Description** *default* — Set Syslog transmission priority to default. *low* — Set Syslog transmission priority to low.			
server	Address of remote syslog server.	string	Maximum length: 127	

Parameter	Description	Type	Size	Default
source-ip	Source IP address of syslog.	string	Maximum length: 63	
ssl-min-proto-version	Minimum supported protocol version for SSL/TLS connections.	option	-	default
	Option / **Description** *default* — Follow system global setting. *SSLv3* — SSLv3. *TLSv1* — TLSv1. *TLSv1-1* — TLSv1.1. *TLSv1-2* — TLSv1.2.			
status	Enable/disable remote syslog logging.	option	-	disable
	Option / **Description** *enable* — Log to remote syslog server. *disable* — Do not log to remote syslog server.			

Configuration

To configure logging to a syslog server in the CLI:

```
config log syslogd setting
    set certificate {string}
    config custom-field-name
        edit <id>
            set name {string}
            set custom {string}
        next
    end
    set enc-algorithm [high-medium|high|...]
    set facility [kernel|user|...]
    set format [default|csv|...]
    set interface {string}
    set interface-select-method [auto|sdwan|...]
    set max-log-rate {integer}
    set mode [udp|legacy-reliable|...]
    set port {integer}
    set priority [default|low]
    set server {string}
    set source-ip {string}
    set ssl-min-proto-version [default|SSLv3|...]
    set status [enable|disable]
end
```

To configure logging to a syslog server with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "status": "enable",
    "server": <ip_address>,
    "format": <format>,
    ...
}}' http://localhost/api/v2/cmdb/log.syslogd/setting
```

Logging to FortiAnalyzer

Container FortiOS does not natively support FortiAnalyzer as a log destination. To log to FortiAnalyzer, configure the FortiAnalyzer device as a syslogd destination.

To enable logging to FortiAnalyzer in the CLI:

```
config log syslogd setting
    set status enable
    set server <FAZ_IP>
    set format default
end
```

To configure logging to FortiAnalyzer with the REST API:

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "status": "enable",
    "server": <FAZ_IP>,
    "format": "default",
}}' http://localhost/api/v2/cmdb/log.syslogd/setting
```

Configuring log filters

Log filters can be configured to determine which logs are sent to the syslog servers. This allows certain logging levels and types of logs to be directed to specific log devices.

Each syslog server has an associated filter, which is referenced using the server ID. For example, `config log **syslogd3** filter`.

Options

The following configuration options are available:

Parameter	Description	Type	Size	Default
forward-traffic	Enable/disable forward traffic logging.	option	-	enable

Option	Description
enable	Enable forward traffic logging.

Parameter	Description	Type	Size	Default
	Option / **Description** *disable* — Disable forward traffic logging.			
local-traffic	Enable/disable local in or out traffic logging.	option	-	enable
	Option / **Description** *enable* — Enable local in or out traffic logging. *disable* — Disable local in or out traffic logging.			
multicast-traffic	Enable/disable multicast traffic logging.	option	-	enable
	Option / **Description** *enable* — Enable multicast traffic logging. *disable* — Disable multicast traffic logging.			
severity	Lowest severity level to log.	option	-	information
	Option / **Description** *emergency* — Emergency level. *alert* — Alert level. *critical* — Critical level. *error* — Error level. *warning* — Warning level. *notification* — Notification level. *information* — Information level. *debug* — Debug level.			

config free-style

Parameter	Description	Type	Size	Default
category	Log category.	option	-	traffic
	Option / **Description** *traffic* — Traffic log. *event* — Event log.			

Parameter	Description	Type	Size	Default
	Option / **Description** *virus* — Antivirus log. *webfilter* — Web filter log. *attack* — Attack log. *app-ctrl* — Application control log. *ssl* — SSL log.			
filter	Free style filter string.	string	Maximum length: 1023	
filter-type	Include/exclude logs that match the filter.	option	-	include
	Option / **Description** *include* — Include logs that match the filter. *exclude* — Exclude logs that match the filter.			
id	Entry ID.	integer	Minimum value: 0 Maximum value: 4294967295	0

Configuration

To configure a syslog filter in the CLI:

```
config log syslogd filter
    set forward-traffic [enable|disable]
    config free-style
        Description: Free Style Filters
        edit <id>
            set category [traffic|event|...]
            set filter {string}
            set filter-type [include|exclude]
        next
    end
    set local-traffic [enable|disable]
    set multicast-traffic [enable|disable]
    set severity [emergency|alert|...]
end
```

To configure a syslog filter with the REST API:

The configuration options are the same as the CLI.

```
curl -H "Content-Type: application/json" -X PUT -d '{ "data":{
    "severity": "notification",
```

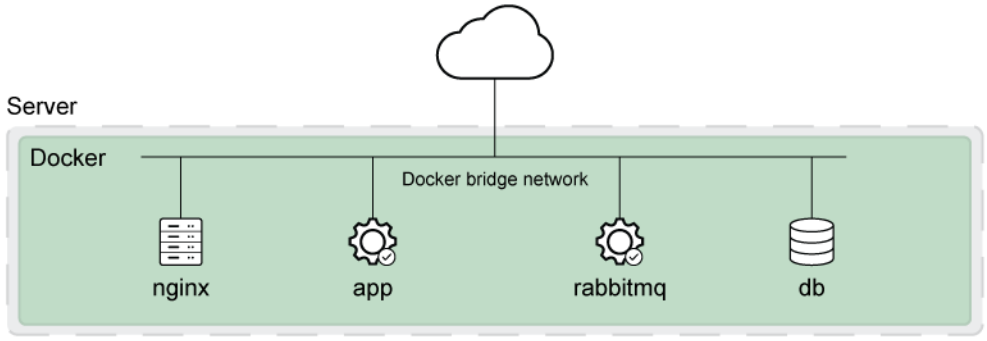
```
    "forward-traffic": "enable",
    "local-traffic": "disable",
    "multicast-traffic": "disable",
}}' http://localhost/api/v2/cmdb/log.syslogd/filter
```

Appendix A - Docker deployment - example

In this example, Container FortiOS is deployed to protect a web application available via HTTP in Docker.

In the following `docker-compose.yml` file, NGINX proxies requests on port 80 to a Django application. Port 80 on the host is forwarded by the Docker default bridge network to the `nginx` container.

The details of the initial web application configuration are beyond the scope of this example.




```
services:
  db:
    image: postgres
    volumes:
      - ./data/db:/var/lib/postgresql/data
    environment:
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}

  rabbitmq:
    image: rabbitmq:latest
    environment:
      - RABBITMQ_DEFAULT_USER=${RABBITMQ_USER}
      - RABBITMQ_DEFAULT_PASS=${RABBITMQ_PASSWORD}
      - RABBITMQ_DEFAULT_VHOST=${RABBITMQ_VHOST}
      - RABBITMQ_ERLANG_COOKIE=${RABBITMQ_COOKIE}
    ports:
      - 5672:5672
      - 15672:15672
    volumes:
      - "./rabbitmq-data:/var/lib/rabbitmq/mnesia"
    healthcheck:
      test: ["CMD", "wget", "http://rabbitmq:15672/", "-O", "index.html"]
      interval: 10s
      timeout: 2s
      retries: 20

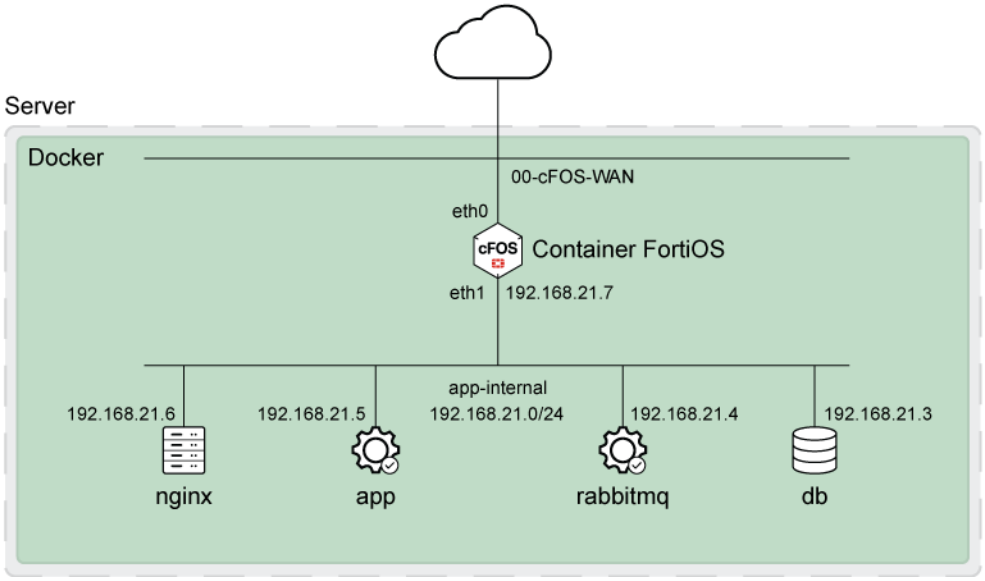
  app:
    build: .
    volumes:
      - ./src:/code
```

```
      - static_volume:/code/static
      - media_volume:/code/media
    expose:
      - 8000
    environment:
      - DJANGO_SUPERUSER_USERNAME=${DJANGO_SUPERUSER_USERNAME}
      - DJANGO_SUPERUSER_EMAIL=${DJANGO_SUPERUSER_EMAIL}
      - DJANGO_SUPERUSER_PASSWORD=${DJANGO_SUPERUSER_PASSWORD}
      - EMAIL_USER=${EMAIL_USER}
      - EMAIL_PASSWORD=${EMAIL_PASSWORD}
      - ENVIRONMENT=PROD
      - POSTGRES_NAME=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    healthcheck:
      test: ["CMD", "wget", "http://app:8080/", "-O", "index.html"]
      interval: 10s
      timeout: 2s
      retries: 20
    depends_on:
      - db
      - rabbitmq

  nginx:
    build: ./nginx
    volumes:
      - static_volume:/code/static
      - media_volume:/code/media
    ports:
      - "80:80"
    depends_on:
      - app

volumes:
  static_volume:
  media_volume:
```

To add Container FortiOS as a firewall for this web application:



1. Import the Docker image tarball using the following command:

   ```
   sudo docker load -i FOS_X64_DOCKER-v7-build255-FORTINET.tar.gz
   ```

2. Define internal and external networks in the compose file:

   ```
   networks:
     app-internal:
       driver: bridge
       ipam:
         driver: default
         config:
           - subnet: 192.168.21.0/24
     00-cFOS-WAN:
       driver: bridge
   ```

   The `app-internal` network will be the protected internal network on the Container FortiOS `eth1` interface.

   The `00-cFOS-WAN` network is a bridge network to the host interface.

3. Specify an IP address for each of the containers on the `app-internal` network:

   - ```
     db:
       ...
       networks:
         app-internal:
           ipv4_address: 192.168.21.3
     ```
   - ```
     rabbitmq:
       networks:
         app-internal:
           ipv4_address: 192.168.21.4
     ```
   - ```
     app:
       networks:
         app-internal:
           ipv4_address: 192.168.21.5
     ```
     ```

- nginx:
    networks:
      app-internal:
        ipv4_address: 192.168.21.6

4. Update the nginx container to expose port 80 internally instead of bridging to the host:
   Replace:

```
ports:
  - "80:80"
```

   with:

```
expose:
  - 80
```

5. In the compose file services section, define a Container FortiOS service:

```
cfos1:
  container_name: "cfos1"
  image: fos:latest
  ports:
    - "80:80"
  environment:
    TZ: 'America/Vancouver'
  volumes:
    - './cfos_data:/data'
  cap_add:
    - NET_ADMIN
    - SYS_ADMIN
  restart: unless-stopped
  networks:
    app-internal:
      ipv4_address: 192.168.21.7
    00-cFOS-WAN:
```

6. Prepare the Container FortiOS license:
   a. In the cfos_data directory, save the Container FortiOS license file as cfos.lic.
7. Prepare the Container FortiOS configuration file:
   a. In the cfos_data directory, create a cfos-partial.conf file.
   b. In the cfos-partial.conf file, define a virtual IP to forward traffic from the eth0 interface to the nginx container IP:

```
config firewall vip
    edit "DemoApp"
        set type access-proxy
        set mappedip "192.168.21.6"
        set extintf "eth0"
        set portforward enable
        set extport "80"
        set mappedport "80"
    next
end
```

   c. In the cfos-partial.conf file, define a firewall policy using this virtual IP:

```
config firewall policy
    edit 1
        set name "DemoApp_forwarding"
        set srcintf "eth0"
        set dstintf "eth1"
        set srcaddr "all"
        set dstaddr "DemoApp"
        set service "ALL"
        set logtraffic all
    next
end
```

8. Start the containers:

   ```
   sudo docker-compose -f docker-compose.yml up -d
   ```

9. In a web browser, access the public IP of the container host.
   The web application displays.

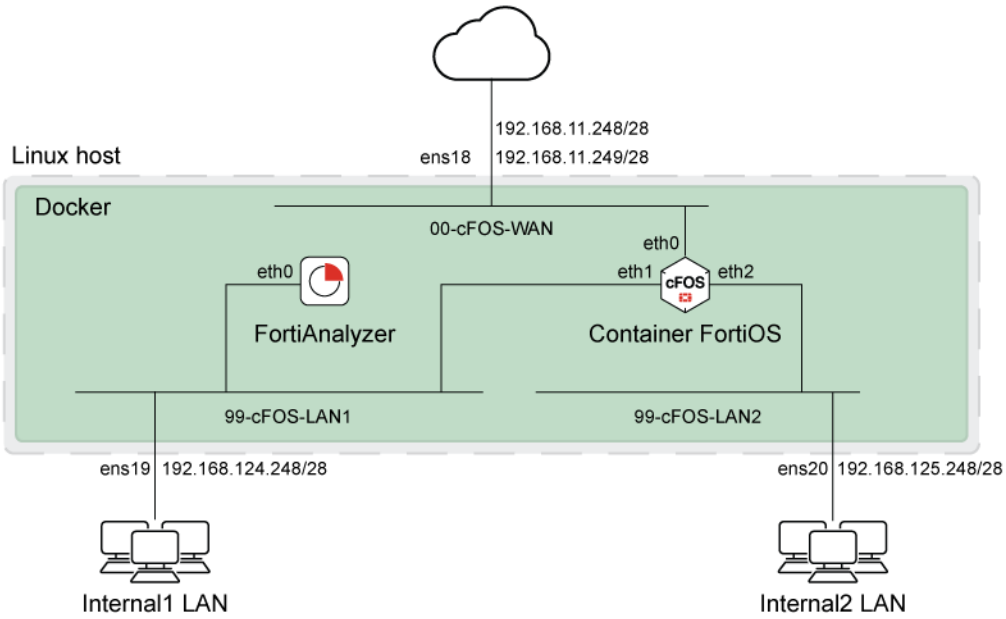
Appendix B - Networked deployment - example

In this example, Container FortiOS is deployed in Docker on a Debian Linux host to police traffic to and from two internal networks and the internet. Container FortiOS sends traffic logs to FortiAnalyzer deployed in Docker on the same Linux host.

The Linux host could be a standalone device or a virtual machine.

Full configuration of FortiAnalyzer is beyond the scope of this example.

Components



- **Linux host**: The Linux host has the following network interfaces:
  - `ens18`: WAN interface listening on 192.168.11.248/28 (host) and 192.168.11.249/28 (Container FortiOS)
  - `ens19`: LAN interface listening on 192.168.124.248/28.
  - `ens20`: LAN interface listening on 192.168.125.248/28.
- **Container FortiOS**: Container FortiOS runs in a Docker container on the Linux host and listens on the following interfaces:
  - `eth0`: WAN interface connected to host interface `ens18` using a bridge network `00-cFOS-WAN`.
  - `eth1`: LAN interface connected to host interface `ens19` using a Macvlan network `99-cFOS-LAN1`.
  - `eth2`: LAN interface connected to host interface `ens20` using a Macvlan network `99-cFOS-LAN2`.
- **FortiAnalyzer**: FortiAnalyzer runs in a Docker container on the same Linux host as Container FortiOS and has one interface:
  - `eth0`: Connected to host interface `ens19` using Macvlan network `99-cFOS-LAN1`.
- **Internal1** and **Internal2**: Internal network protected by Container FortiOS.

Prerequisites

Before proceeding with the following example, ensure that you have done the following:

- Installed Docker engine on the Linux host.
- Obtained licenses for Container FortiOS and FortiAnalyzer.
-

Deployment procedures

1. Configure the Docker networks.
2. Create the Container FortiOS container.
3. Create the FortiAnalyzer container.
4. Connect the Container FortiOS to the Docker internal networks.
5. Import the Container FortiOS license.
6. Configure logging to FortiAnalyzer.
7. Configure Container FortiOS policies.

Configuring the Docker networks

To configure the WAN bridge network:

On the Linux host, run the following command:

```
docker network create \
    --driver=bridge \
    --subnet=10.222.222.0/24 \
    --ip-range=10.222.222.0.24 \
    --gateway=10.222.222.1 \
    00-cFOS-WAN
```

Note that this network will not be seen externally.

To configure the internal Macvlan networks:

On the Linux host, run the following commands:

```
docker network create \
    --driver=macvlan\
    --subnet=192.168.124.240/28 \
    --gateway=192.168.124.241 \
    --aux-address="clinux=192.168.124.243" \
    --aux-address="this-host=192.168.124.248" \
    -o parent=ens19 \
    99-cFOS-LAN1

docker network create \
    --driver=macvlan\
```

```
--subnet=192.168.125.240/28 \
--gateway=192.168.125.241 \
--aux-address="clinux=192.168.125.243" \
--aux-address="this-host=192.168.125.248" \
-o parent=ens20 \
99-cFOS-LAN2
```

Creating the Container FortiOS container

To create the Container FortiOS container:

1. Import the Docker image tarball using the following command:

   ```
   sudo docker load -i FOS_X64_DOCKER-v7-build255-FORTINET.tar.gz
   ```

2. On the Linux host, run the following command:

   ```
   docker container create \
       --network 00-cFOS-WAN \
       --ip=10.222.222.254 \
       -p 443:443 \
       -p 4022:4022 \
       --cap-add=NET_ADMIN \
       --cap-add=SYS_ADMIN \
       --security-opt apparmor:unconfined \
       --name cfos1 \
       -v/demo/cfos/cfos1:/data \
       --dns 9.9.9.9 \
       -it fos
   ```

   This creates a container with TCP ports 443 and 4022 exposed on the WAN interface of the host.

Creating the FortiAnalyzer container

To create the FortiAnalyzer container:

1. On the Linux host, run the following command:

   ```
   docker container create \
       --network 99-cFOS-LAN1 \
       --ip=192.168.124.252 \
       --cap-add=ALL \
       --security-opt apparmor:unconfined \
       --name faz1 \
       -v/demo/faz/faz1/var:/var \
       -v/demo/faz/faz1/data:/data \
       --dns 9.9.9.9 \
       -it fortinet/fortianalyzer:latest
   ```

2. Get the PID of the running FortiAnalyzer container:
   docker inspect -f '{{.State.Pid}}' faz1
3. Configure routing for the FortiAnalyzer container:

   ```
   /usr/bin/nsenter -t <PID> -n ip route del default
   ```

```
/usr/bin/nsenter -t <PID> -n ip route add default via 192.168.124.154
```

The FortiAnalyzer is now accessed through the Container FortiOS container.

`nsenter` runs a command in the namespace of the given process ID, here configuring the default route on the FortiAnalyzer container rather than the Linux host. **Note that this change does not persist after a reboot.**

For more information about deploying FortiAnalyzer on Docker, see the FortiAnalyzer Docker Administration Guide.

Connecting Container FortiOS to the Docker internal networks

To connect Container FortiOS to the Docker internal networks:

In the Linux host, run the following commands:

```
docker network connect --ip 192.168.124.254 99-cFOS-LAN1 cfos1
docker network connect --ip 192.168.125.254 99-cFOS-LAN2 cfos1
```

Importing the Container FortiOS license

To import the Container FortiOSlicense:

1. Open the license file in a text editor and copy the full contents.
2. In the container CLI, enter the following command:

```
exec import-license "<content_of_license_file>"
```

3. Verify your license status:

```
diagnose sys license
```

Configuring logging to FortiAnalyzer

To configure logging to FortiAnalyzer:

In the Container FortiOS CLI, run the following command:

```
config log syslogd setting
     set status enable
     set server "192.168.124.252"
     set mode udp
     set port 514
     set facility local7
     set source-ip "192.168.124.254"
     set format default
     set priority default
     set max-log-rate 0
     set enc-algorithm disable
     set ssl-min-proto-version default
     set certificate ""
     set interface-select-method auto
     set interface ""
end
```

Configuring policies to and from internal networks

Configure policies as needed to monitor traffic to and from internal networks.

For example, to allow and log all traffic from Internal1 to the internet:

```
config firewall policy
    edit 1
        set status enable
        set utm-status disable
        set name "Internal1_egress_allow"
        set comments ""
        set srcintf "eth1"
        set dstintf "eth0"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6
        set dstaddr6
        set service "ALL"
        set ssl-ssh-profile "no-inspection"
        set action accept
        set nat disable
        set logtraffic all
    next
end
```

www.fortinet.com