Cisco Cyber Vision Command Reference Guide, Release 5.4.x

First Published: 2025-03-28

Last Modified: 2025-03-28

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

Preface

This preface contains these sections:

Communications, Services, and Additional Information

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html, lists all new and revised Cisco technical documentation.

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Overview

Objective

This document is a practical guide to essential Cisco Cyber Vision commands. It provides clear examples of how to use key commands for managing, monitoring, and troubleshooting your Cyber Vision systems. Commands are organized by theme for easy navigation.

Accessing the CLI

Accessing the Cisco Cyber Vision Center CLI

You can access the Center CLI in several ways:

- **Center hosted on Cisco Unified Computing System (Cisco UCS):**
    - Direct SSH (`ssh cv-admin@<Center IP>`)
    - UCS Cisco Integrated Management Controller (CIMC) virtual KVM
    - Cisco UCS Serial Over LAN
- **Center hosted on a hypervisor:**
    - Direct SSH (`ssh cv-admin@<Center IP>`)
    - Hypervisor virtual machine console

You can use the same credentials for all connections to the Center CLI: username `cv-admin` and the installation password.

For accessing the Cisco Cyber Vision sensor CLI, see Cyber Vision 4.3.0 Onwards.

System Management and Control Commands

sudo

Use the **sudo** command to interact with the shell with root privileges, simulating a root login. It provides full administrative access until you exit the shell.

sudo -i

For more information, see sudo.

setup-center-cli

Use the **setup-center-cli** command to configure Cisco Cyber Vision Center.

setup-center-cli *COMMAND* [*args...*]

Syntax Description

keymap *KEYMAP*	Configures the keyboard mapping for the Cyber Vision Center. **Usage:** `setup-center-cli keymap KEYMAP` `KEYMAP`: Specifies the keyboard mapping to use (for example, us, fr, de, it, es)
network	Generates `systemd-networkd` interface files for network configuration. **Usage:** `setup-center-cli network COMMAND [arg...]` **Subcommands:** • `interface`: Generates network interface files. • `dns`: Generates DNS configuration files. • `single-interface`: Sets the Center to single-interface mode (admin interface that is used for webapp and sensor collection). • `firewall`: Controls network allowed list.
pki *FQDN*	Generates certificates for the Cisco Cyber Vision Center. **Usage:** `setup-center-cli pki FQDN`, where FQDN is the fully qualified domain name of the Center.
renew-certificate	Renews Center or sensor certificates. **Usage:** `setup-center-cli renew-certificate [--center | --sensor=<ID>]` **Options:** • `--center`: Renews the Center certificate. • `--sensor=<ID>`: Renews a specific sensor certificate (where `<ID>` is the sensor ID).
webapp-certificate	Manages the web application's TLS certificate. **Usage:** `setup-center-cli webapp-certificate COMMAND [arg...]` **Subcommands** • `reset`: Resets to the default selfsigned certificate configuration. • `import`: Imports a PKCS#12 certificate file. • `csr`: Uses a Certificate Signing Request (CSR) to configure the web certificate.

authorized-keys

Authorizes public SSH keys for access to the Center. If no arguments are provided, it reads keys from standard input.

Usage: `setup-center-cli authorized-keys [OPTIONS] [FNAME...]`, where FNAME is the filename of the SSH public key to authorize.

Options

- `--user`: Specifies the user for authorized keys (defaults to "cv-admin").

type

`type`: Generates configuration files based on the desired Center type.

Usage: `setup-center-cli type <TYPE>`

Arguments: `<TYPE>`: Specifies the Center type:

- Standalone
- Local Center
- Global Center

Example: `setup-center-cli type Local Center.`

ntp

Generates NTP (Network Time Protocol) configuration files.

Usage: `setup-center-cli ntp [SERVER...]`, where `[SERVER...]` indicates one or more NTP server configurations.

Example: `setup-center-cli ntp ntp.example.com,1,secretkey`

center-id

Changes the Center's unique ID, used for enrollment in a Global Center and for computing component IDs.

Usage: `setup-center-cli center-id [ID]`, where `[ID]` is the new Center ID. If not provided, the DMI system UUID is used.

Example: `setup-center-cli center-id new-center-id`

password

Provides password-related utilities.

Usage: `setup-center-cli password <COMMAND> [arg...]`

Subcommands

- `check`: Checks password strength.
- `setup-user`: Sets the password for the `cv-admin` user.

Example: `setup-center-cli password check`

import

Imports configuration settings from a JSON file.

Usage: `setup-center-cli import <FILE> [-f | --force]`, where `<FILE>` is the path to the JSON configuration file.

Example: `setup-center-cli import config.json --force`

Command History

Release	Modification
4.3.0	This command was introduced.

This example displays how to check the password strength:

```
root@center100:~# setup-center-cli password check

Password must be at least 16 characters long
```

reboot

Use the **reboot** command to restart the Cisco Cyber Vision Center.

reboot

This example displays how to restart the center:

```
root@center100:~# reboot
Connection to 10.2.3.100 closed by remote host.
```

For more information, see reboot.

date

Use the **date** command to check the current date on the Cisco Cyber Vision Center CLI or sensor application.

date

This example displays how to check the current date on center:

```
root@center100:~# date
Wed Jun  5 11:20:36 UTC 2024
```

This example displays how to check the current date on sensor:

```
sh-5.0# date
Wed Jun  5 11:20:54 UTC 2024
```

For more information, see date.

poweroff

Use the **poweroff** command to shut down the Cisco Cyber Vision Center server.

poweroff

This example displays how to shut down the center:

```
root@center100:~# poweroff
Connection to 10.2.3.100 closed by remote host.
```

For more information, see poweroff.

systemctl

Use the **systemctl** command to interact and manage the services running on the Cisco Cyber Vision Center server.

systemctl { **status** } | { **start** } | { **stop** } | { **restart** } | { **--failed** }

Syntax Description

--failed	Lists failed services on the system.
status <*name of the service*>	Checks the status of the specific service.
restart <*name of the service*>	Restarts a specific service or all services.
stop <*name of the service*>	Stops the specific service.

This example displays how to check failed services:

```
root@center100:~# systemctl --failed
0 loaded units listed.
```

This example displays how to check the status of the "sbs-backend.service"service:

```
root@center100:~# systemctl status sbs-backend.service
sbs-backend.service - Cisco Cyber Vision Center Backend
     Loaded: loaded (/lib/systemd/system/sbs-backend.service; enabled; vendor preset:
enabled)
     Active: active (running) since Wed 2024-06-05 16:32:32 UTC; 2s ago
   Main PID: 5617 (sbs-backend-sta)
      Tasks: 22 (limit: 77128)
     Memory: 92.4M
```

This example displays how to restart all `sbs` services:

```
root@center100:~# systemctl restart sbs-services.target
```

This example displays how to restart the "sbs-backend.service" service:

```
root@center100:~# systemctl restart sbs-backend.service
```

For more information, see systemctl.

crontab

Crontab is a configuration file that schedules commands or scripts to run automatically at specific intervals.

```
Usage:
 crontab [options] file
 crontab [options]
 crontab -n [hostname]

Options:
 -u <user>  define user
 -e         edit user's crontab
 -l         list user's crontab
 -r         delete user's crontab
 -i         prompt before deleting
 -n <host>  set host in cluster to run users' crontabs
 -c         get host in cluster to run users' crontabs
 -V         print version and exit
 -x <mask>  enable debugging
```

Syntax and Descriptions

Each line in a crontab file follows a specific syntax:

1. **Cron Expression:** The line begins with a cron expression consisting of five fields:
   - Minute (0-59)
   - Hour (0-23)
   - Day of the Month (1-31)
   - Month (1-12 or Jan-Dec)
   - Day of the Week (0-6 or Sun-Sat)

   These fields represent the time and date when the scheduled command should be executed.
2. **Command:** The cron expression is followed by the command or script to be executed.

Note If both the "day of month" and "day of week" fields are restricted (i.e., not "*"), then either or both of these fields must match the current day for the job to be executed.

This is a crontab configuration for purging the components that are inactive for 90 days:

```
5 * * * * sbs-db purge-components --inactive-days 90
```

This is a crontab configuration for deleting the table content every two days at midnight.

```
0 0 */2 * * sudo sbs-db-toolbox exec 'TRUNCATE TABLE dns_request;'
```

For more information, see crontab.

journalctl

Use the **journalctl** command to interact and search through the log entries that are stored in the journal.

journalctl [**-r**] [**--since**] [**-f**] [**-p err**] [**-u** *<servicename>*]

Syntax Description

-r	Displays the latest logs first.
--since	Displays logs within a specified time range.
-f	Displays live logs for live troubleshooting.
-p err	Fecthes only the error logs.
-u *<servicename>*	Displays the logs for a specific service.
--boot=0	Displays the logs from the last system boot.

This example displays how to extract Linux journal for the "sbs-burrow" service:

```
root@center100:~# journalctl -u sbs-burrow
-- Logs begin at Mon 2024-05-13 12:28:06 UTC, end at Thu 2024-06-06 12:51:44 UTC. --
May 14 03:14:31 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:32 center burrow[6748]: burrow -- 1 files handled in 0.020166 seconds
[caller=interfacer.go:71]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.01 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
```

This example displays how to extract the live logs for the "sbs-burrow" service:

```
root@center100:~# journalctl -fu sbs-burrow
-- Logs begin at Mon 2024-05-13 12:28:06 UTC. --
Jun 06 12:52:31 center burrow[147743]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:31 center burrow[147743]: burrow flow table analyzed in 0.01 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:31 center burrow[147743]: burrow -- 3 files handled in 0.049746 seconds
[caller=interfacer.go:71]
Jun 06 12:52:36 center burrow[147743]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:36 center burrow[147743]: burrow -- 1 files handled in 0.013072 seconds
[caller=interfacer.go:71]
Jun 06 12:52:46 center burrow[147743]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:46 center burrow[147743]: burrow -- 1 files handled in 0.009560 seconds
[caller=interfacer.go:71]
Jun 06 12:52:51 center burrow[147743]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:51 center burrow[147743]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
Jun 06 12:52:51 center burrow[147743]: burrow -- 2 files handled in 0.028321 seconds
[caller=interfacer.go:71]
```

This example displays how to extract the logs for the "sbs-burrow" service:

```
root@center100:~# journalctl -u sbs-burrow-- Logs begin at Mon 2024-05-13 12:28:06 UTC, end
 at Thu 2024-06-06 12:51:44 UTC. --
May 14 03:14:31 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:32 center burrow[6748]: burrow -- 1 files handled in 0.020166 seconds
[caller=interfacer.go:71]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.01 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
May 14 03:14:37 center burrow[6748]: burrow flow table analyzed in 0.00 secs
[caller=flowtable_analyzer.go:153]
```

For more information, see journalctl.

unenroll

Use the **unenroll** command to unenroll a local Center from the Cisco Cyber Vision Global Center synchronization. It effectively removes the local Center's data from the global Center and prepares the local Center for future enrollment or replacement.

/opt/sbs/bin/unenroll

Command History

Release	Modification
5.2.0	This command was introduced.

This example displays how to unenroll a local Center:

```
root@center100:~# opt/sbs/bin/unenroll
08/01/2026 18:38:07 +0000 unenroll INFO Center type: standalone
                                                      caller=postgres.go:527
08/01/2026 18:38:07 +0000 unenroll INFO Non-enrolled center, exiting
                                                           caller=main.go:56
root@Center:~#
```

IOS Commands

app-hosting connect

To connect to a specific sensor application, use the **app-hosting connect** command on your router or switch.

app-hosting connect { **appid** *application-id* **session** }

Syntax Description

application-id	Id of the sensor application.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to access a specific sensor application.

```
IE3400ESC01#app-hosting connect appid ccv_sensor_iox_active_discovery_aarch64 session
sh-5.0#
```

For more information, refer to the app-hosting command in the Cisco IOS XE documentation.

app-hosting start

To start the sensor application, use the **app-hosting start** command on your router or switch.

app-hosting start { **appid** *application-id* }

Syntax Description

application-id	Id of the sensor application.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to start the sensor application.

```
IE3400ESC01#app-hosting start appid ccv_sensor_iox_active_discovery_aarch64
sh-5.0#
```

For more information, refer to the app-hosting command in the Cisco IOS XE documentation.

app-hosting stop

To stop the sensor application, use the **app-hosting stop** command on your router or switch.

app-hosting stop { **appid** *application-id* }

Syntax Description

application-id	Id of the sensor application.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to stop a sensor application.

```
IE3400ESC01#app-hosting stop appid ccv_sensor_iox_active_discovery_aarch64
```

For more information, refer to the app-hosting command in the Cisco IOS XE documentation.

show app-hosting

To display application hosting-related information such as the application Id, use the **show app-hosting** command on your router or switch.

show app-hosting { **detail** | **list** }

Syntax Description

detail	Displays detailed information about the application.
list	Displays information about the application or appliance.

Command History

Release	Modification
Release 3.0.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

This example displays how to identify the application Id (appid):

```
Device#show app-hosting list
App id State
---------------------------------------------------------
ccv_sensor_iox_active_discovery_aarch64 RUNNING
```

The following is sample output from the **show app-hosting detail** command:

```
Device# show app-hosting detail

App id                 : perfsonar
Owner                  : iox
State                  : RUNNING
Application
  Type                 : lxc
  Name                 : perfsonar-lxc
  Version              : 1.0.0
  Description          : PerfSONAR 4.1 Cisco IOx LXC
Activated profile name : custom

Resource reservation
  Memory               : 2048 MB
  Disk                 : 10 MB
  CPU                  : 4000 units

Attached devices
  Type              Name               Alias
  ---------------------------------------------
  serial/shell     iox_console_shell   serial0
  serial/aux       iox_console_aux     serial1
  serial/syslog    iox_syslog          serial2
  serial/trace     iox_trace           serial3

Network interfaces
   ---------------------------------------
eth0:
   MAC address         : 52:54:dd:38:a3:da
```

Sbs Commands

sbs-backup

Use the **sbs-backup** command to back up and restore the configuration of the Cisco Cyber Vision Center.

sbs-backup *command* [*args..*]

Syntax Description

export	Creates a backup of the Cisco Cyber Vision Center. The backup includes the configuration, data, and sensor management information.
import *file*	Restores a Cisco Cyber Vision Center from the specified backup archive. **Note** All existing data and configuration on the Center will be erased and replaced with the data from the backup.

Command History

Release	Modification
4.0.0	This command was introduced.

This example displays how to back up Cisco Cyber Vision Center data:

```
root@center100:~# sbs-backup export
Usbs-backup export
Please note that license information is also backed up and will be restored if you restore
 the backup on the same system from which the backup was taken.
If you restore the backup on a different system, first return the license reservation to
Cisco Smart Software Licensing so you can set it up again after the restoration on the new
 system.
***************** Taking backup of file system     *****************
***************** Taking backup of database        *****************
***************** Taking backup of RMQ definitions *****************
***************** Taking backup of center version  *****************
***************** Taking backup of symlinks        *****************
***************** Taking backup of extension       *****************
Created center archive at
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
```

This example displays how to restore a configuration from an archive file:

```
root@center100:~# sbs-backup import
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
Usbs-backup import
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
***************** Restoring file system     *****************
***************** Restoring database        *****************
***************** Restoring RMQ definitions *****************
***************** Restoring symlinks        *****************
***************** Restoring extension       *****************
Restore completed, please reboot to finalise the system configuration. After reboot, please
 install the Reports extension compatible with the center version.
```

sbs-closest-sensor-mode

Use the **sbs-closest-sensor-mode** to control the display of sensor and PCAP data sources associated with assets. By default, the **sbs-closest-sensor-mode** option is disabled.

sbs-closest-sensor-mode *action*

Syntax Description

enable	Displays the **Seen By** column in the **Assets seen in current active view** under the **Asset Visiblity** section of the Cisco Cyber Vision User Interface.
disable	Displays the **Data Sources** column in the **Assets seen in current active view** under the **Asset Visiblity** section of the Cisco Cyber Vision User Interface.

Command History

Release	Modification
5.2.0	This command was introduced.

To enable the closest sensor mode, run this command:

```
root@center100:~# sbs-closest-sensor-mode enable


Enabling closest sensor mode...
Successfully enabled!
```

To disable the mode, run this command:

```
root@center100:~# sbs-closest-sensor-mode disable


Disabling closest sensor mode...
Successfully disabled!
```

sbs-db

Use the **sbs-db** command to manage and interact with a database. It provides a wide range of functionalities for database administration, maintenance, data manipulation, and troubleshooting.

sbs-update *commands* [*args..*]

Syntax Description

`aggregate-flows`	Enables or disables the aggregation of flows based on client port.
`cleanup`	Cleans up the database.
`connect`	Opens a psql shell (PostgreSQL interactive terminal).
`count`	Counts rows in all tables.
`count-short`	Counts rows in relevant tables.
`create-extensions`	Creates database extensions like `hstore` and `pg_stat_statements`
`destroy`	Drops the database and all its data.
`drop-matviews`	Deletes all warehouse materialized views.
`dump`	Dumps all database content. **Note** By default, the database dump is stored in the `/data/tmp` folder. The filename is in `sbs-data-dump-<FQDN of the center>-<centertype>-<center version>-<timestamp>..sql.gz.` For example, `sbs-db-dump-centerdoc165labautomccvlocal-standalone-5.2.0`
`dump-tables`	Dumps specific tables.
`execute`	Executes a SQL query.
`exec-pretty`	Executes a SQL query and formats the output for readability.
`find-schema-files`	Lists the SQL files that are loaded during initialization.
`force-expiration`	Forces immediate data expiration.
`import-snort`	Imports Snort rules and categories
`indexes-size`	Displays the size of all indexes.
`init`	Creates the database user and database.
`init-load`	Creates the database user and database, and loads data from a provided file.
`list-custom-networks`	Lists all custom networks.
`list-extensions`	Lists all installed extensions.

`list-migrations`	Lists database migrations by time.
`list-schemas`	Lists all database schemas.
`list-storage-settings`	Lists all storage settings.
`list-tables`	Displays table names by schema.
`list-triggers`	Displays triggers by schema.
`load`	Loads an SQL command file into the database.
`migrate`	Migrates the database.
`optimize`	Optimizes the database using `VACUUM` if a flag is defined.
`port-scan-detection`	Enables or disables port scan detection.
`purge-components`	Removes components and associated data.
`purge-credentials-until`	Removes all credentials until a specified date.
`purge-events`	Removes events between dates with a specific metadata ID.
`purge-events-since`	Removes all events since a specified date.
`purge-events-until`	Removes all events until a specified date.
`purge-external-communications`	Removes components and associated data.
`purge-flows`	Removes all flows by tag.
`purge-flows-since`	Removes all flows since a specified date.
`purge-flows-until`	Removes all flows until a specified date.
`purge-orphan-components`	Removes all orphan components.
`purge-since`	Removes flows, events, and variables since a specified date.
`purge-until`	Removes flows, events, and variables until a specified date.
`purge-variables-since`	Removes all variables since a specified date.
`purge-variables-until`	Removes all variables until a specified date.
`remote-access-protocol`	Adds or removes remote access protocols.
`remote-domain-regex`	Adds or removes remote access domain regular expressions.
`reset-data`	Repacks tables.
`reset-group-impact`	Updates group criticality.
`reset-password`	Resets a user's password.
`reset-users`	Removes all users.

restore	Restores a database dump.

Command History

Release	Modification
4.0.0	This command was introduced.

This example displays how to dump all the content of a database:

```
root@center100:~# sbs-db dump
dump database to:
/data/tmp/sbs-db-dump-centerdoc165labautomccvlocal-standalone-5.2.0-20250423042226.sql.gz
```

This example displays how to reset the password of a database user:

```
root@center100:~# sbs-db reset-password user@cisco.com
User password successfully reset. You can now set a new password by login to the GUI using
 this temporary password: iO********l26fqc
```

This example displays how to connect to the database using psql:

```
root@center100:~# sbs-db connect
```

This example displays how to purge all flows since a specific date:

```
root@center100:~# sbs-db purge-flows-since "2024-01-01"
```

sbs-extension

Use the **sbs-extension** command to install, upgrade, and list the Cisco Cyber Vision sensor management extensions.

sbs-extension list [**--json**]
sbs-extension install [**--background**] [**--remove-source-file**] [**--run**] *file*
sbs-extension upgrade [**--background**] [**--remove-source-file**] [**--no-version-check**] *file*
sbs-extension remove [**--background**] *id*
sbs-extension run [**--no-version-check**] *id*
sbs-extension run-all [**--no-version-check**]
sbs-extension cmd *id* [*parameters*]

Table 1: Syntax description

list [--json]	List all installed extensions; use --json for JSON output.	Plain text output by default
install [--background] [--remove-source-file] [--run] <file>	Install an extension from <file>. **--background**: Run install in background. **--remove-source-file**: Delete source file after install. **--run**: Start extension after install.	<file> is required
upgrade [--background] [--remove-source-file] [--run] [--no-version-check] <file>	Upgrade an extension. **--background**: Run upgrade in background. **--remove-source-file**: Delete source file after upgrade. **--run**: Start extension after upgrade. **--no-version-check**: Skip version compatibility check.	<file> is required
remove [--background] <id>	Remove extension by ID. **--background**: Run removal in background.	<id> is required
run [--no-version-check] <id>	Start extension by ID. **--no-version-check**: Skip version compatability check.	<id> is required
run-all [--no-version-check]	Start all installed extensions. **--no-version-check**: Skip version check for all.	N/A
cmd <id> [parameters]	Execute a command on a specified extension. **[parameters]**: Command-specific arguments.	<id> is required

Table 2: Command history

Release	Modification
2.0.0	This command was introduced.

This example displays how to list the available extensions:

```
root@Center:~# sbs-extension list
+--------------------+---------------------------------+---------+-------------+--------------------+
| ID                 | Name                            | Version | CCV Version | CCV Version
 locked |
+--------------------+---------------------------------+---------+-------------+--------------------+
| reports-management | Cyber Vision Reports Management | 5.4.0   | 5.4.0       | Yes
          |
| sensor-management  | Cyber Vision sensor management  | 5.5.0   | 5.5.0       | Yes
          |
+--------------------+---------------------------------+---------+-------------+--------------------+
root@Center:~#
```

sbs-device-engine

Use the **sbs-device-engine** command to group various components within a system into logical devices. It interacts with a database and uses a configuration file to control its behavior.

sbs-device-engine [*options*]

Syntax Description

`-center-id string`	Overrides the default Center ID. `string`: The Center ID to use.
`-db_host string`	Specifies the database hostname. `string`: The hostname or IP address of the database server.
`-db_name string`	Specifies the database name. `string`: The name of the database.
`-db_password string`	Specifies the database user password. `string`: The password for the database user.
`-db_port int`	Specifies the database port. `int`: The port number on which the database server is listening.
`db_user string`	Specifies the database username. `string`: The username for accessing the database.
`-f string`	Specifies the configuration filename. Default: `"/data/etc/sbs/device-engine.conf"` `string`: The path to the configuration file
`-loglevel string`	Specifies the logging verbosity level. `string`: The logging level (for example, debug, info, warning, error).
`-logoutput string`	Specifies the logging output. `string`: The location where logs should be written (for example, a file path or "stdout").

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to group components into devices:

```
root@center100:~# sbs-device-engine
06/06/2024 12:32:48 +0000 undefined INFO CenterID provided by: /data/etc/sbs/center-id
```

```
                            caller=config.go:236
06/06/2024 12:32:48 +0000 undefined INFO center ID: 3ea90f42-3830-ac99-b12e-efd0c64fda7d
                            caller=config.go:261
06/06/2024 12:32:48 +0000 device-engine INFO Center type: standalone
                          caller=postgres.go:587
06/06/2024 12:32:48 +0000 device-engine INFO Connected to postgres on /var/run/postgresql:5432
 with user: ics on dbname: ics caller=app.go:35
06/06/2024 12:32:48 +0000 device-engine INFO RabbitMQ available
                         caller=connection.go:29
06/06/2024 12:32:48 +0000 device-engine INFO Using default activity tags period: 1 month
                            caller=engine.go:144
06/06/2024 12:32:48 +0000 device-engine INFO Number of components taken into account: 13
                            caller=engine.go:135
06/06/2024 12:32:48 +0000 device-engine INFO Rule SwitchAggregation is disabled
                             caller=rules.go:179
06/06/2024 12:32:48 +0000 device-engine INFO Creating 6 devices covering 10 components
                       caller=save_devices.go:31
06/06/2024 12:32:48 +0000 device-engine INFO exiting...
                               caller=main.go:42
```

To use a custom configuration file:

```
root@center100:~# sbs-device-engine -f /path/to/myconfig.conf
```

sbs-diag

Use the **sbs-diag** command to extract the diagnostic files from the Cisco Cyber Vision Center.

The diagnostic file will be copied to the `/data/tmp` folder with the `sbs-diag-export-<CENTERTYPE>-<CENTERNAME>-<DATETIME>.tgz` name.

sbs-diag [**OPTIONS**]

Syntax Description

-h	To generate the command help.
-o	To specific the path where the generated diagnostics are saved. If you do not specify the path, the file will be copied to the `/data/tmp` folder.
-v	To use the verbose mode.
-n	To generate diagnostics without accessing the database.
-l	To generate a reduced version of the diagnostics.
-b	To generate additional benchmarking by adding more information such as disk performances in the diagnostics. **Note** This is a CPU-intensive activity, and several processes including the user interface will be unavailable for a few minutes during benchmarking.

Command History

Release	Modification
3.0	This command was introduced.

This example displays how to change the network configuration,:

```
root@center100:~# sbs-diag
[2024-06-06 12:43:10.550568354] [sbs-diag:126221] Exporting diagnostics data...
[2024-06-06 12:43:10.563196830] [sbs-diag:126221] - Gathering system data
[2024-06-06 12:43:10.574734602] [sbs-diag:126221] - Gathering hardware data
[2024-06-06 12:43:19.307577251] [sbs-diag:126221] - Journal error+warning
[…]
[2024-06-06 12:43:21.266439032] [sbs-diag:126221] - Configs
[…]
[2024-06-06 12:43:26.591608618] [sbs-diag:126221] - pg_stats
[…]
[2024-06-06 12:43:35.061380378] [sbs-diag:126221] - Compressing data
[2024-06-06 12:43:36.172173931] [sbs-diag:126221] - Deleting temporary data
[2024-06-06 12:43:36.192001612] [sbs-diag:126221] Archive
/data/tmp/sbs-diag-export-standalone-center100-202406061243.tgz is ready.
```

sbs-erase

Use the **sbs-erase** command to factory reset the Cisco Cyber Vision Center configuration.

Note Running this command will remove all data from the system.

sbs-erase

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to reset the Center configuration:

```
root@center100:~# sbs-erase
This will reboot and destroy all data on this system. Type yes in capital letters to confirm:
 YES
Connection to 10.2.3.100 closed by remote host.
```

sbs-netconf

Use the **sbs-netconf** command to reconfigure or add network routes to the Ethernet interfaces on your Cisco Cyber Vision Center.

sbs-netconf

Command History

Release	Modification
4.0.0	This command was introduced.
5.4.x	This command permits you to configure IPv4 or IPv6, or both.

This example displays how to change the network configuration:

```
root@center100:~# sbs-netconf
```

On running this command, a configuration window opens. You can follow the options there.

sbs-passwd

Use the **sbs-passwd** command to change the password of the cv-admin user of the Cisco Cyber Vision Center.

Note After the fresh installation, you must change the cv-admin password.

sbs-passwd

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to change the password:

```
root@center100:~# sbs-passwd
Password must be at least 16 characters long.
Password must contain characters from at least 3 of the following characters class:
    lowercase, capitals, numbers or punctuation.
Enter password:
Confirm Password:
```

sbs-system-fqdn

Use the **sbs-system-fqdn** command to check the fully qualified domain name (FQDN) of the Cisco Cyber Vision Center.

sbs-system-fqdn

Command History

Release	Modification
Release 4.0.0	This command was introduced.

This example displays how to check your center's FQDN:

```
Device# /opt/sbs/bin/sbs-system-fqdn
center100.sentryo.local
```

sbs-timeconf

Use the **sbs-timeconf** command to change the Network Time Protocol (NTP) parameters.

sbs-timeconf [**-h**] [**-a** *sensor_serial_number*] [**-r** *sensor_serial_number*] [**-p** *dest_directory_path center_ip_address sensor_serial_number*] [**-g**] [**-n** *network_address network_mask*] [**-m** *network_address network_mask*] [**-s** *server_ip_address* [*key_id AES128CMAC_key_value...*]] [**-t** *server_ip_address*]

Syntax Description

-a *sensor_serial_number*	Adds a sensor with the specified serial number to the configuration. This allows the system to communicate with that sensor.
-r *sensor_serial_number*	Removes a sensor with the specified serial number from the configuration. This stops the system from communicating with that sensor.
-p *dest_directory_path center_ip_address sensor_serial_number*	Generates provisioning files for a given sensor. These files likely contain configuration data that is needed for the sensor to operate.
-g	Generates base configuration files for the `ntpd` daemon, which is the NTP daemon that are used for time synchronization.
-n *network_address network_mask*	Allows machines on the specified network to communicate with the Cisco Cyber Vision Center.
-m *network_address network_mask*	Revokes communication access for machines on the specified network.
-s *server_ip_address* [*key_id AES128CMAC_key_value ...*]	Adds an NTP server with the specified IP address to the configuration. It also supports authentication using AES128CMAC with a key ID and key value.
-t *server_ip_address*	Removes an NTP server with the specified IP address from the configuration.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to add an NTP parameter:

```
root@center100:~# sbs-timeconf -s time1.google.com
```

This example displays how to remove an NTP parameter:

```
root@center100:~# sbs-timeconf -t time1.google.com
```

sbs-update

Use the **sbs-update** command to perform various operations related to software update of the Cisco Cyber Vision Center application including integrity checks, upgrade, and rollback.

sbs-update *commands* [*options*]

Syntax Description

check *file*	Checks the integrity of the specified update file.
prepare-install *file* *dir* [**allow-rollback**]	Performs the integrity check and prepares the update for installation after the next reboot. The **allow-rollback** flag enables you to roll back the update if needed.
install *file*	Installs the update from the specified file. The update is only installed if the version is newer than the currently installed version and has a valid signature.
install-with-rollback *file*	Installs the update and allows rollbacks. This means you can revert to the previous version if needed.
-undo *network_address* *network_mask*	Reverts to the last installed update.
update-script *file* *dir* [**allow-rollback**]	Updates the `sbs-update.sh` script itself. The **allow-rollback** flag enables you to roll back the update if needed.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to upgrade the Cisco Cyber Vision Center:

```
root@center100:~# sbs-update install /data/tmp/CiscoCyberVision-update-center-5.0.0.dat
Extracted archive directory /data/tmp/sbs-update.Mzb0PA/files
Installed version 4.4.0+202405071704
Updated version 5.0.0+202405241346
Preparing /data/tmp/CiscoCyberVision-update-center-5.0.0.dat for setup on next reboot.
WARNING: rebooting...
```

sbs-version

Use the **sbs-version** command to obtain the Cisco Cyber Vision Center version. This command outputs the current version of the Cisco Cyber Vision Center software installed on the system. It is useful for verifying the software version directly from the CLI.

sbs-version

Command History

Release	Modification
5.2.0	This command was introduced.

This example displays how to check the version information of the Center:

```
root@center100:~# cat /etc/sbs-version
SBS_MAJOR_VER="5"
SBS_MINOR_VER="4"
SBS_INCR_VER="0"
SBS_BUILD_VER="202512191120"
root@Center:~#
```

Application Management Commands

rabbitmqadmin

Use the **rabbitmqadmin** command to check the queue status for the **RabbitAdmin** service running on the Cisco Cyber Vision Center.

rabbitmqadmin [*options*] *subcommands*

Syntax Description

`-C CONFIG`, `--config=CONFIG`	Specifies the configuration file to use. Default: `~/.rabbitmqadmin.conf`
`-N NODE`, `--node=NODE`	Specifies the node described in the configuration file. Default: `'default'` (only if a configuration file is specified).
`-H HOST`, `--host=HOST`	Specifies the hostname of the RabbitMQ server. Default: `localhost`
`-P PORT`, `--port=PORT`	Specifies the port number for the RabbitMQ HTTP API. Default: `15672`
`--path-prefix=PATH_PREFIX`	Specifies a custom URI path prefix for the RabbitMQ HTTP API. `api` and the operation path will be appended to it. Default: blank string.
`-V VHOST`, `--vhost=VHOST`	Specifies the virtual host (vhost) to connect to. Default: All vhosts for listing commands and `/` for declaration commands.
`-u USERNAME`, `--username=USERNAME`	Specifies the username for authentication. Default: `guest`
`-p PASSWORD`, `--password=PASSWORD`	Specifies the password for authentication. Default: `guest`
`-U URI`, `--base-uri=URI`	Specifies a base HTTP API URI. `/api` and the operation path will be appended to it. Overrides `--host`, `--port`, and `--path-prefix`. Separately provide `-vhost`.
`-q`, `--quiet`	Suppresses status messages. Default: `True`
`-s`, `--ssl`	Enables SSL/TLS connection. Default: `False`
`--ssl-key-file=SSL_KEY_FILE`	Specifies the path to the PEM-formatted private key file for SSL/TLS.

`--ssl-cert-file=SSL_CERT_FILE`	Specifies the path to the PEM-formatted certificate file for SSL/TLS.
`-ssl-ca-cert-file=SSL_CA_CERT_FILE`	Specifies the path to the PEM-formatted CA certificate file for SSL/TLS.
`--ssl-disable-hostname-verification`	Disables hostname verification for SSL/TLS connections.
`-k`, `--ssl-insecure`	Disables all SSL/TLS validation (similar to `curl -k`).
`-t REQUEST_TIMEOUT`, `--request-timeout=REQUEST_TIMEOUT`	Sets the HTTP request timeout in seconds. Default: `120`
`-f FORMAT`, `--format=FORMAT`	Specifies the output format.
`S SORT`, `--sort=SORT`	Specifies the sort key for listing commands.
`-R`, `--sort-reverse`	Reverses the sort order.
`-d DEPTH`, `--depth=DEPTH`	Specifies the maximum recursion depth for listing tables. Default: 1

Command History

Release	Modification
4.0	This command was introduced.

This example displays how to check RabbitMQ queues:

```
root@center100:~# rabbitmqadmin list queues
+----------------------------------------------------------------------------+----------+
|                                    name                                    | messages |
+----------------------------------------------------------------------------+----------+
| ccv.queue.activediscovery_scanrequest.049e52db-0178-4df6-9a63-05a04cb60736 | 0        |
| ccv.queue.activediscovery_scanresult                                       | 0        |
| ccv.queue.authorized_sensors                                               | 0        |
| ccv.queue.authorized_sensors.burrow                                        | 0        |
| ccv.queue.authorized_sensors.republisher                                   | 0        |
| ccv.queue.burrow_results                                                   | 0        |
| ccv.queue.decode_errors                                                    | 0        |
| ccv.queue.device                                                           | 0        |
| ccv.queue.dpad                                                             | 0        |
| ccv.queue.events                                                           | 0        |
| ccv.queue.flow_properties_stats                                            | 0        |
| ccv.queue.flowtables.049e52db-0178-4df6-9a63-05a04cb60736                  | 0        |
| ccv.queue.flowtables.049e52db-0178-4df6-9a63-05a04cb60736.burrow           | 0        |
| ccv.queue.flowtables.expired                                               | 0        |
| ccv.queue.flowtables.unknown                                               | 0        |
| ccv.queue.jobs.049e52db-0178-4df6-9a63-05a04cb60736                        | 0        |
| ccv.queue.jobs_results.aspic                                               | 0        |
| ccv.queue.jobs_results.backend                                             | 0        |
| ccv.queue.jobs_results.extensionapid                                       | 0        |
| ccv.queue.jobs_results.inputd                                              | 0        |
| ccv.queue.jobs_results.worker                                              | 0        |
| ccv.queue.snort_events                                                     | 0        |
| ccv.queue.sysinfo                                                          | 0        |
| kraken.initial_modifications                                                | 0          |
| kraken.modifications                                                        | 0          |
| kraken.queue.lc.certificate_renewal                                         | 0          |
+-----------------------------------------------------------------------------+----------+
```

smartagentctl

Use the **smartagentctl** command to manage the license information.

smartagentctl [**COMMAND**]

Syntax Description

`--register <registration-token-file-path>`	Registers the agent using the registration token file provided.
`--reregister <registration-token-file-path>`	Reregisters the agent using the registration token file provided.
`--deregister`	Deregisters the agent.
`-request-entitlement <entitlement-tag> <component-count>`	Requests an entitlement with the specified tag and component count.
`--set-transport-mode <mode> [<host>] [<port>] [<username>] [<password>]`	Sets the transport mode and optionally provides connection details (host, port, username, password).
`--get-transport-mode`	Retrieves the current transport mode.
`--get-license-summary`	Retrieves a summary of the license information.
`--enable-reservation`	Enables license reservation.
`--disable-reservation`	Disables license reservation.
`--get-reservation-request-code`	Retrieves the license reservation request code.
`--cancel-reservation-request`	Cancels the license reservation request.
`--get-reservation-confirmation-code`	Retrieves the license reservation confirmation code.
`--get-reservation-return-code`	Retrieves the license reservation return code.
`--install-reservation <authorization-code-file-path>`	Installs a license reservation using the authorization code file provided.
`-return-reservation-authorization-code`	Returns the license reservation authorization code.
`--utility-mode-is-enabled`	Checks if utility mode is enabled.
`--enable-utility-mode`	Enables utility mode.
`--disable-utility-mode`	Disables utility mode.
`-set-customer-info <info-type> <info-data>`	Sets customer information of the specified type.
`--get-customer-info`	Retrieves customer information.
`--get-tech-support`	Retrieves technical support information.

Command History

Release	Modification
4.0	This command was introduced.

This example displays how to disable a license:

```
root@center100:~# smartagentctl --disable-reservation
{"status":"SUCCESS","message":"Operation Success."}
```

This example displays how to enable a license service:

```
root@center100:~# smartagentctl --enable-reservation
{"status":"SUCCESS","message":"Success"}
```

This example displays how to restart the smart agent service:

```
root@center100:~# systemctl restart sbs-backend.service
```

Network Diagnostics and Configuration Commands

ping

Use the **ping** command to check if a host is reachable.

ping [*options*] [*destination*]

This example checks if the host 1.1.1.1 is reachable:

```
root@center100:~# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: icmp_seq=0 ttl=54 time=21.439 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=54 time=12.201 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=54 time=19.945 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=54 time=19.250 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=54 time=20.691 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=54 time=13.313 ms
64 bytes from 1.1.1.1: icmp_seq=6 ttl=54 time=19.154 ms
64 bytes from 1.1.1.1: icmp_seq=7 ttl=54 time=19.491 ms
64 bytes from 1.1.1.1: icmp_seq=8 ttl=54 time=13.291 ms
```

For more information, see ping.

flowctl

Use the **flowctl** command to manage and troubleshoot deep packet inspection (DPI) on Cisco Cyber Vision sensors.

flowctl `[` *options* `]` *command* `[` *args..* `]`

Syntax Description

Options `--port`	Specifies the flow HTTP port. Default: `6666`
`--syncd-sock`	Specifies the sensorsyncd HTTP socket. Default: `/tmp/sensorsyncd/sensorsyncd.sock`
`--scand-port`	Specifies the scan HTTP port. Default: `6668`
Commands `add-dpad-variable-export-periods`	Adds variable export periods for DPAD.
`buffer-ratio`	Sets the flow buffer ratio (RAM percentage).
`count-files`	Counts the number of files in the flow directory.
`disk`	Displays disk usage.
`dump-pcap`	Writes a PCAP file containing the latest received packets.
`environment`	Displays IOX environment variables.
`flushtcp`,	Asks flow to flush all TCP connections (use with caution).
`forget`,	Forgets pending data and reloads services. Default: `False`
`list-dpad-variable-export-periods`	Lists current variable export periods for DPAD.
`meminfo`	Displays memory information.
`network-interfaces`	Displays network interface information.
`pause`	Pauses flow processing.
`pids`	Returns flow PIDs.
`ping`	Checks flow status.
`print-conf`	Prints the flow configuration file.
`processes`	Displays information about running processes.
`read-capture-file`,	Uploads, and analyzes a PCAP file (modified timestamps).

`read-capture-file-raw`,	Uploads, and analyzes a PCAP file (original timestamps).
`reload`	Asks flow to reload the configuration file.
`remove-filter`	Removes BPF filter from flow configuration.
`set-dpad-variable-export-periods`	Replaces variable export periods for DPAD.
`set-sensor-id`	Updates the sensor ID in the flow configuration.
`since-last-captured-packet`	Displays the duration since the last captured packet (in milliseconds).
`start-recording`	Starts recording packets.
`stats`	Prints flow run time statistics.
`stop-recording`	Stops recording packets.
`sub-dpad-variable-export-periods`	Removes variable export periods for DPAD.
`syncd-stats`	Prints sensorsyncd run-time statistics.
`unpause`	Unpauses flow processing.
`update-filter`	Updates BPF filter in flow configuration.

Command History

Release	Modification
4.0	This command was introduced.

This example displays how to print the flow run-time statistics:

```
sh-5.0# flowctl stats --human
{
  "flow_dumper_active": 0,
  "flow_internal_buffer_length": 0,
  "flow_internal_buffer_memory": 12582912,
  "flow_internal_buffer_use_percent": 0,
  "flow_nb_afpacket_captured_packets_eth1": 572724,
  "flow_nb_afpacket_dropped_packets_eth1": 0,
  "flow_nb_erspan_decapsulation_error": 0,
  "flow_nb_erspan_fragemented_packets": 0,
  "flow_nb_erspan_ip4": 0,
  "flow_nb_flows_in_flowtable": 4,
  "flow_nb_hsrp_lru_errors": 0,
  "flow_nb_iface_dropped_packets_eth1": 0,
  "flow_nb_ipv4_defrag_errors": 0,
  "flow_nb_no_flow_cleaned_up": 0,
  "flow_nb_packets_deduplicated": 497837,
  "flow_nb_packets_per_interface_eth1": 572724,
  "flow_nb_panics_recovered": 0,
  "flow_nb_s7plus_subscriptions": 0,
  "flow_nb_s7plus_subscriptions_dropped": 0,
  "flow_nb_scan_detected_sources": 0,
  "flow_nb_too_many_flows": 0,
  "flow_nb_tracked_packets": 74887,
  "flow_nb_tracked_packets_per_layer_icmp": 10,
  "flow_nb_tracked_packets_per_layer_tcp": 354,
  "flow_nb_tracked_packets_per_protocol_arp": 4,
  "flow_nb_tracked_packets_per_protocol_cisco_discovery": 26396,
  "flow_nb_tracked_packets_per_protocol_icmp": 10,
  "flow_nb_tracked_packets_per_protocol_lldp": 95152,
  "flow_nb_tracked_packets_per_protocol_smb": 222,
  "flow_nb_tracked_packets_per_protocol_snap": 26396,
  "flow_paused": 0,
  "flow_since_last_captured_packet_ms": 461,
  "flow_sum_capture_length_eth1": 82221916,
  "flow_sum_length_eth1": 82221916,
  "flow_sum_tracked_capture_length": 26207377,
  "flow_sum_tracked_length": 26207377
}
```

route

Use the **route** command to view the routing table.

route -n

This example displays how to check the routing table:

```
root@center100:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.2.3.254      0.0.0.0         UG    0      0        0 eth0
10.2.0.0        0.0.0.0         255.255.252.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.255.248 U     0      0        0 brrsyslogd
169.254.0.8     0.0.0.0         255.255.255.252 U     0      0        0 brntpd
169.254.0.16    0.0.0.0         255.255.255.248 U     0      0        0 brburrow
169.254.0.32    0.0.0.0         255.255.255.248 U     0      0        0 brbackend
169.254.0.40    0.0.0.0         255.255.255.252 U     0      0        0 brhaproxyadmin
169.254.0.48    0.0.0.0         255.255.255.252 U     0      0        0 brhaproxyacq
169.254.0.56    0.0.0.0         255.255.255.248 U     0      0        0 brhaproxylog
169.254.0.64    0.0.0.0         255.255.255.248 U     0      0        0 bralfred
169.254.0.72    0.0.0.0         255.255.255.248 U     0      0        0 brsysinfodh
169.254.0.80    0.0.0.0         255.255.255.248 U     0      0        0 brsensorinputd
169.254.0.88    0.0.0.0         255.255.255.248 U     0      0        0 brpxgridagent
169.254.0.96    0.0.0.0         255.255.255.248 U     0      0        0 brext-apid
169.254.0.120   0.0.0.0         255.255.255.248 U     0      0        0 brsyncd
169.254.0.128   0.0.0.0         255.255.255.248 U     0      0        0 braspic
169.254.0.136   0.0.0.0         255.255.255.248 U     0      0        0 brnodeexporter
169.254.0.144   0.0.0.0         255.255.255.248 U     0      0        0 brpgexporter
169.254.0.152   0.0.0.0         255.255.255.248 U     0      0        0 brmarmotd
169.254.0.160   0.0.0.0         255.255.255.248 U     0      0        0 brrefreshviews
169.254.0.168   0.0.0.0         255.255.255.248 U     0      0        0 brsnmp
169.254.0.224   0.0.0.0         255.255.255.224 U     0      0        0 brrmq
192.168.69.0    0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

For more information, see route.

ssh

Use the **ssh** command to securely log in to a remote machine in a network.

ssh

This example displays how to log in to a remote host.

```
Device# ssh cv-admin@center100
```

For more information, see ssh.

tcpdump

Use the **tcpdump** command on the sensor application CLI to create PCAP files, which may be required for troubleshooting or issue reporting.

tcpdump *--options*

Syntax Description

-i *interface*	Creates dumps for the specified interface.
-G *rotate_seconds*	Rotates the dump file after the specified duration.
-z *postrotate_command*	Used with **-G** option. Creates a zip file using the gunzip utility.
--W*filecount*	Used with **-G** option. Limits the number of rotated dump files to the specified value.
-w*file*	Writes the PCAP data to the specified file.

Command History

Release	Modification
Release 4.0.0	This command was introduced.

This example creates 5 zipped PCAP files for 120 seconds of traffic and saves the files to the `/iox_data/appdata/` folder

```
sh-5.0# tcpdump -i eth1 -G 120 -z gzip -W 5 -w /iox_data/appdata/capture-%H-%M-%S.pcaptcpdump:
 listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
Maximum file limit reached: 5
648161 packets captured
861891 packets received by filter
212901 packets dropped by kernel
```

To verify the creation of the files at the specified location, run this command:

```
sh-5.0# ls -lh /iox_data/appdata/-rw-r--r--    1 root     root        3.8M Jun 21 11:40
capture-11-37-51.pcap.gz
-rw-r--r--    1 root     root        4.6M Jun 21 11:42 capture-11-39-51.pcap.gz
-rw-r--r--    1 root     root        3.8M Jun 21 11:44 capture-11-41-58.pcap.gz
-rw-r--r--    1 root     root        4.4M Jun 21 11:46 capture-11-44-15.pcap.gz
-rw-r--r--    1 root     root        5.4M Jun 21 11:49 capture-11-46-15.pcap.gz
```

This example checks if any data is received or sent on the **eth0** interface on the Cisco Cyber Vision Center:

```
root@center100:~# tcpdump -i eth0 -wroot@center100:~# tcpdump -i eth0 -w
/data/tmp/tcpdumpeth0.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C262 packets captured
264 packets received by filter
0 packets dropped by kernel
```

For more information, see tcpdump.

ip address

Use the **ip address** command to check the network interface IP address and the status of the interface. You can use this command for troubleshooting when the Cisco Cyber Vision Center is not reachable.

ip address [**show** [**dev** *IFNAME*]]

Syntax Description

IFNAME	Interface name such as eth0, eth1, and so on.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to check the network interface IP address and the status of the "eth0" interface:

```
root@center100:~# ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 00:50:56:8f:9d:16 brd ff:ff:ff:ff:ff:ff
    inet 10.2.3.102/22 brd 10.2.3.255 scope global eth0
       valid_lft forever preferred_lft forever
```

For more information, see ip address.

iptables

Use the **iptables** command to list the packet filter rules.

iptables -L [*chain*]

Syntax Description

chain	To list rules in the specified chain.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to check the rules of the IP table:

```
root@center100:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
DROP       tcp  --  anywhere             anywhere             tcp flags:!FIN,SYN,RST,ACK/SYN
 state NEW
DROP       all  --  anywhere             anywhere             state INVALID
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
NFLOG      all  --  anywhere             anywhere             ! match-set
center_admin_networks src nflog-prefix  "DropUnAuthNetwork:" nflog-group 1
DROP       all  --  anywhere             anywhere             ! match-set
center_admin_networks src
ACCEPT     icmp --  anywhere             anywhere             icmp destination-unreachable
ACCEPT     icmp --  anywhere             anywhere             icmp source-quench
ACCEPT     icmp --  anywhere             anywhere             icmp time-exceeded
ACCEPT     icmp --  anywhere             anywhere             icmp parameter-problem
ACCEPT     icmp --  anywhere             anywhere             icmp echo-request
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh state NEW
ACCEPT     udp  --  anywhere             anywhere             udp dpt:bootps state NEW
ACCEPT     udp  --  anywhere             anywhere             udp dpt:bootpc state NEW
```

For more information, see iptables.

nslookup

Use the **nslookup** command to query the name servers for information about various hosts and domains.

nslookup [*DNS_server*]

Syntax Description

DNS_server	FQDN or IP address of the DNS server.

Command History

Release	Modification
3.0	This command was introduced.

To check the name resolution of the server, "iseccv002.lab-autom-ccv.local", run this command:

```
root@center100:~# nslookup iseccv00x.lab-ccv.local

Server:    10.2.3.254
Address 1: 10.2.3.254 _gateway

Name:      iseccv00x.lab-ccv.local
Address 1: 10.2.2.131 iseccv00x.lab-ccv.local
```

To check the name resolution of the server, "8.8.8.8", run this command:

```
root@center100:~# nslookup 8.8.8.8

Server:    208.67.220.220
Address 1: 208.67.220.220 dns.sse.cisco.com

Name:      8.8.8.8
Address 1: 8.8.8.8 dns.google
```

ntpq

Use the **ntpq** command to check the NTP server communication details.

Syntax

ntpq -c peer *IP_address*

Syntax Description

IP_address	IP address of the NTP server

Command History

Release	Modification
3.0	This command was introduced.

This example displays the NTP server communication details:

```
root@center100:~# ntpq -c peer 169.254.0.10
remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        .LOCL.          10 l 159m   64    0    0.000   +0.000   0.000
*aer01-r4d20-dc- .GNSS.           1 u   10  256  377   22.445   -3.473   0.491
```

File and Disk Management Commands

df

Use the **df** command to display statistics about the amount of free disk space in the file system.

df

Command History

Release	Modification
3.0.0	This command was introduced.

This example shows how to check the disk usage of the folder `/data` in your file system:

```
root@center100:~# df -h /data/
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/data_crypt  245G  1.6G  230G   1% /data
```

This example calculates the disk usage of the `/data/` directory, displays it in a human-readable format, and then sorts the output by size:

```
root@center100:~# du -sh /data/* | sort -h
4.0K /data/extensions
4.0K /data/flags
4.0K /data/updates
16K /data/lost+found
32K /data/home
64K /data/license
216K /data/redis
109M /data/log
122M /data/etc
284M /data/var
351M /data/postgresql
774M /data/tmp
```

For more information, see df.

du

Use the **du** command to analyze the disk usage.

du

Command History

Release	Modification
3.0.0	This command was introduced.

This example analyzes the disk usage in the `/data` directory:

```
root@center100:~# du -sh /data/
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/data_crypt  245G  1.6G  230G   1% /data
root@center100:~#
```

This example analyzes the disk usage in the `/data` directory and displays the output in a sorted format:

```
root@center100:~# du-sh /data/* | sort -h
4.0K    /data/extensions
4.0K    /data/flags
4.0K    /data/updates
16K    /data/lost+found
32K    /data/home
64K    /data/license
216K    /data/redis
109M    /data/log
122M    /data/etc
284M    /data/var
351M    /data/postgresql
774M    /data/tmp
```

For more information, see du.

lsblk

Use the **lsblk** command to list information about all available or the specified block devices.

lsblk [**options**] [*device...*]

Syntax Description

device	Name of the block device on the file system.

Command History

Release	Modification
3.0.0	This command was introduced.

To list all block devices in your system, run this command:

```
root@center100:~# lsblk

NAME             MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
loop0              7:0    0 554.2M  1 loop  /
sda                8:0    0   250G  0 disk
|-sda1             8:1    0  1022M  0 part  /system
`-sda2             8:2    0   249G  0 part
  `-data_crypt 251:0    0   249G  0 crypt /data
zram0            252:0    0 125.6G  0 disk  [SWAP]
```

For more information, see lsblk.

rclone

Use the **rclone** command to sync or move Cisco Cyber Vision Center data between local and remote storages.

rclone [flags] [command]

Syntax Description

--failed	Lists failed services on the system.
status <*name of the service*>	Checks the status of the specific service.
restart <*name of the service*>	Restarts the specific service.
stop <*name of the service*>	Stops the specific service.

Command History

Release	Modification
4.0	This command was introduced.

To find all available options and flags in the command, run this command:

```
root@center100:~# rclone


Usage:
  rclone [flags]
  rclone [command]

Available Commands:
  about           Get quota information from the remote.
  authorize       Remote authorization.
  backend         Run a backend specific command.
  cat             Concatenate any files and send them to stdout.
  check           Checks the files in the source and destination match.
  cleanup         Clean up the remote if possible.
  config          Enter an interactive configuration session.
  copy            Copy files from source to dest, skipping already copied.
  copyto          Copy files from source to dest, skipping already copied.
  copyurl         Copy url content to dest.
  cryptcheck      Cryptcheck checks the integrity of a crypted remote.
  cryptdecode     Cryptdecode returns unencrypted file names.
  dbhashsum       Produces a Dropbox hash file for all the objects in the path.
  dedupe          Interactively find duplicate files and delete/rename them.
  delete          Remove the contents of path.
  deletefile      Remove a single file from remote.
  genautocomplete Output completion script for a given shell.
  gendocs         Output markdown docs for rclone to the directory supplied.
  hashsum         Produces a hashsum file for all the objects in the path.
  help            Show help for rclone commands, flags and backends.
  link            Generate public link to file/folder.
  listremotes     List all the remotes in the config file.
  ls              List the objects in the path with size and path.
  lsd             List all directories/containers/buckets in the path.
  lsf             List directories and objects in remote:path formatted for parsing.
  lsjson          List directories and objects in the path in JSON format.
  lsl             List the objects in path with modification time, size and path.
  md5sum         Produces an md5sum file for all the objects in the path.
  mkdir          Make the path if it doesn't already exist.
  mount          Mount the remote as a mountpoint.
  move           Move files from source to dest.
  moveto         Move file or directory from source to dest.
  ncdu           Explore a remote with a text based user interface.
  obscure        Obscure password for use in the rclone config file.
  purge          Remove the path and all of its contents.
  rc             Run a command against a running rclone.
  rcat           Copies standard input to file on remote.
  rcd            Run rclone listening to remote control commands only.
  rmdir          Remove the path if empty.
  rmdirs         Remove empty directories under the path.
  selfupdate     Update the rclone binary.
  serve          Serve a remote over a protocol.
  settier        Changes storage class/tier of objects in remote.
  sha1sum        Produces an sha1sum file for all the objects in the path.
  size           Prints the total size and number of objects in remote:path.
  sync           Make source and dest identical, modifying destination only.
  test           Run a test command.
  touch          Create new file or change file modification time.
  tree           List the contents of the remote in a tree like fashion.
  version        Show the version number.

Flags:
  -h, --help   help for rclone

Use "rclone [command] --help" for more information about a command.
```

To enter in an interactive configuration session to set up new remotes and manage existing ones, run this command:

```
root@center100:~# rclone config
n/s/q> n (new remote storage)
name> cvbackupsftp
Storage> 34 (34 / SSH/SFTP Connection)
host> 10.2.3.172
user> user
port> [Enter]  (default 22)
y/g/n> y   (y: enter a password)
password: (enter password)
password: (re enter password)
key_pem> [Enter] (not used)
key_file> [Enter] (not used)
y/g/n> [Enter] (not used)
pubkey_file> [Enter] (not used)
key_use_agent>  [Enter]   (not used)
use_insecure_cipher> [Enter] (not used)
disable_hashcheck> [Enter] (not used)
y/n> n (no advanced setting)
y/e/d> y (validate settings)
e/n/d/r/c/s/q> q (quit config)
```

To move files from a source directory to a destination directory, run this command:

```
root@center100:~# rclone move center100:/tmp/example.txt center101:/tmp/example.txt
```

For more information, see rclone.

General Utilities

scp

Use the **scp** command on the sensor application CLI to securely send the collected PCAP files to other hosts in the network.

scp

This example displays how to share PCAP files with a remote host.

This example displays how to check the current date on the sensor:

```
sh-5.0# scp /iox_data/appdata/*.pcap.gz user@10.2.3.172:/srv/pub/date
```

For more information, see scp.

top

Use the **top** command to identify the most used processes or services and their CPU and memoray usage.

top

This example displays how to display the sorted list of system processes:

```
root@center100:~# top
top - 16:18:32 up 17:19,  1 user,  load average: 0.52, 0.52, 0.61
Tasks: 299 total,   1 running, 298 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.0 us,  0.3 sy,  0.0 ni, 97.5 id,  0.1 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :  32083.2 total,    886.5 free,   5711.7 used,  25485.0 buff/cache
MiB Swap:  64166.5 total,  64159.5 free,      7.0 used.  22045.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 201043 postgres  20   0 4280628 106044  94696 S   7.3   0.3   0:01.15 postgres
   1871 root      20   0  716204  11056   3688 S   2.7   0.0  38:28.49 sysinfod
   1037 rabbitmq  20   0 3544340 244040   6040 S   0.7   0.7  24:36.89 beam.smp
   3058 root      20   0 2379016 342896  34996 S   0.7   1.0   5:41.36 influxd
   1027 snmp      20   0   12112   7688   4332 S   0.3   0.0   1:17.47 snmpd
   1651 postgres  20   0 4262732   3.9g   3.9g S   0.3  12.4   1:58.89 postgres
   1704 redis     20   0   98652   5764   3396 S   0.3   0.0   3:56.79 redis-server
   2145 sbs-sys+  20   0  725244  40024  10012 S   0.3   0.1   2:13.35 sysinfod-sensor
   3148 root      20   0 9032316 985508  24232 S   0.3   3.0   8:56.12 java
 200422 ntp       20   0  278332 141940 137392 S   0.3   0.4   0:01.06 postgres
      1 root      20   0    9652   7912   5144 S   0.0   0.0   0:05.91 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
```

For more information, see top.

vi

Use the **vi** command to edit files on a Cisco Cyber Vision Center or sensor application.

vi *filename*

Syntax Description

i	Switches to the insert mode.
Esc	Switches to the command mode. You can use the following commands in the command mode: • **:w** — Save and continue editing • **:wq** — Save and quit/exit vi • **:q!** — Quit vi and do not save changes.

Command History

Release	Modification
4.0	This command was introduced.

This example shows how to open a file using the "vi" editor:

```
root@center100:~# vi myfile.txt
```

For more information, see vi.

ps

Use the **ps** command to list the processes that currently run on the system.

ps

This example displays how to report a snapshot of the current processes:

```
root@center100:~# ps
PID TTY          TIME CMD
   1359 pts/0    00:00:00 bash
   4311 pts/0    00:00:00 ps
```

For more information, see ps.

openssl

Use the **openssl** command to check the details of the certificate installed on the Cisco Cyber Vision Center.

openssl

Command History

Release	Modification
3.0	This command was introduced.

To check the certificate installed on the Center, run this command:

```
root@center100:~# openssl x509 -in  /data/etc/ca/center-cert.pem -text -nooutrclone

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            a7:14:d1:0c:a4:e6:cd:26:e6:2b:62:21:05:67:28:66
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = Cisco Cyber Vision Center CA VMware-420fa93e303899ac-b12eefd0c6
        Validity
            Not Before: Jun  5 15:59:09 2024 GMT
            Not After : Aug  4 15:59:09 2026 GMT
        Subject: CN = center100.sentryo.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
[…]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                1D:3C:0D:72:5A:52:E0:2B:05:BF:9D:72:64:4D:9A:76:D8:E9:D1:DE
            X509v3 Authority Key Identifier:
                keyid:A1:E5:28:AC:C6:2E:F4:FD:B8:47:D5:CF:8E:45:BC:EE:48:E9:90:5D
               DirName:/CN=Cisco Cyber Vision Center CA VMware-420fa93e303899ac-b12eefd0c6

                serial:00

            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
                DNS:center100.sentryo.local
[…]
```