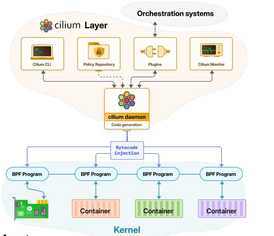
Cilium Cheat Sheet

ISOVALENT | Creators of eBPF cilium tetragon

Cilium is an open source, cloud native solution for providing, securing, and observing network connectivity between workloads, fueled by the revolutionary kernel technology eBPF. At a high level, Cilium offers:

- Transparent API protection and security
- Secure service-to-service communication based on identities
- Secure access to and from external services
- Simple Networking
- Load Balancing
- Bandwidth Management
- Monitoring and Troubleshooting

Components of Cilium



Agent

The Cilium agent (`cilium-agent`) runs as a pod on each K8s node in the cluster. The agent accepts configuration via Kubernetes APIs that describe networking, service load-balancing, network policies, and visibility/monitoring requirements.

The Cilium agent listens for events to learn when containers or workloads are started and stopped. It manages the eBPF programs which the Linux kernel uses to control all network access in/out of those containers.

Client (CLI)

The Cilium CLI client (`cilium`) is a command-line tool that is installed along with the Cilium agent. It interacts with the REST API of the Cilium agent running on the same node. The CLI allows inspecting the state and status of the local agent. It also provides tooling to directly access the eBPF maps to validate their state.

Note

The in-agent Cilium CLI client described here should not be confused with the command line tool for quick installing, managing and troubleshooting Cilium on Kubernetes clusters, which also has the name `cilium`. That tool is typically installed remotely from the cluster and uses `kubeconfig` information to access Cilium running on the cluster via the Kubernetes API. In this document we define usage as either; agent CLI or client CLI.

With Cilium 1.15 and later versions, the in-agent Cilium CLI has been renamed to cilium-dbg.

Operator

The Cilium Operator is responsible for managing duties in the cluster which should logically be handled once for the entire cluster, rather than once for each node in the cluster. The Cilium operator is not in the critical path for any forwarding or network policy decision. A cluster will generally continue to function if the operator is temporarily unavailable. However, depending on the configuration, failure in availability of the operator can lead to:

- Delays in IP Address Management (IPAM) and thus delay in scheduling of new workloads if the operator is required to allocate new IP addresses
- Failure to update the kvstore heartbeat key which will lead agents to declare kvstore unhealthiness and restart.

CNI Plugin

The CNI plugin (`cilium-cni`) is invoked by Kubernetes when a pod is scheduled or terminated on a node. It interacts with the Cilium API of the node to trigger the necessary datapath configuration to provide networking, load-balancing and network policies for the pod.

Installing Cilium

Using the Client CLI:

```
$ cilium install
```

You will see the output:

```
🔮 Auto-detected Kubernetes kind: kind
✨ Running "kind" validation checks
✅ Detected kind version "0.20.0"
ℹ️ Using Cilium version 1.15.1
🔮 Auto-detected cluster name: kind-kind
🔮 Auto-detected kube-proxy has been installed
```

Options

--chart-directory string	Helm chart directory
--datapath-mode string	Datapath mode to use { tunnel | native | aws-eni | gke | azure | aks-byocni } (default: autodetected).
--dry-run	Write resources to be installed to stdout without actually installing them
--dry-run-helm-values	Write non-default Helm values to stdout without performing the actual installation
-h, --help	help for install
--list-versions	List all the available versions without actually installing
--nodes-without-cilium strings	List of node names on which Cilium will not be installed in Helm installation mode, it's assumed that the no-schedule node labels are present and that the infrastructure has set up routing on these nodes to provide connectivity within the Cilium cluster.
--repository string	Helm chart repository to download Cilium charts from (default "https://helm.cilium.io")
--set stringArray	Set helm values on the command line (can specify multiple or separate values with commas: key1=val1,key2=val2)
--set-file stringArray	Set helm values from respective files specified via the command line (can specify multiple or separate values with commas: key1=path1,key2=path2)
--set-string stringArray	Set helm STRING values on the command line (can specify multiple or separate values with commas: key1=val1,key2=val2)
-f, --values strings	Specify helm values in a YAML file or a URL (can specify multiple)
--version string	Cilium version to install (default "v1.13.4")
--wait	Wait for helm install to finish
--wait-duration duration	Maximum time to wait for status (default 5m0s).

Cilium can be uninstalled with:

```
$ cilium uninstall
```

Flags:

-h, --help	help for uninstall
--test-namespace string	Namespace to uninstall Cilium tests from (default "cilium-test")
--timeout duration	Maximum time to wait for resources to be deleted (default 5m0s)
--wait	Wait for uninstallation to have completed

Upgrading Cilium

Upgrade Cilium using the Client CLI.

```
$ cilium upgrade [flags]
```

Flags:

--chart-directory string	Helm chart directory
--datapath-mode string	Datapath mode to use { tunnel | native | aws-eni | gke | azure | aks-byocni } (default: autodetected).
--dry-run	Write resources to be installed to stdout without actually installing them
--dry-run-helm-values	Write non-default Helm values to stdout; without performing the actual upgrade
-h, --help	help for upgrade
--list-versions	List all the available versions without actually installing
--repository string	Helm chart repository to download Cilium charts from (default "https://helm.cilium.io")
--reset-values	When upgrading, reset the helm values to the ones built into the chart
--reuse-values	When upgrading, reuse the helm values from the latest release unless any overrides from are set from other flags. This option takes precedence over HelmResetValues
--set stringArray	Set helm values on the command line (can specify multiple or separate values with commas: key1=val1,key2=val2)
--set-file stringArray	Set helm values from respective files specified via the command line (can specify multiple or separate values with commas: key1=path1,key2=path2)
--set-string stringArray	Set helm STRING values on the command line (can specify multiple or separate values with commas: key1=val1,key2=val2)
-f, --values strings	Specify helm values in a YAML file or a URL (can specify multiple)
--version string	Cilium version to install (default "v1.13.4")
--wait	Wait for helm install to finish
--wait-duration duration	Maximum time to wait for status (default 5m0s)

```
$ cilium upgrade
```

Output:

```
🔮 Auto-detected Kubernetes kind: kind
✨ Running "kind" validation checks
✅ Detected kind version "0.22.0"
ℹ️ Using Cilium version 1.15.1
🔮 Auto-detected cluster name: kind-kind
ℹ️ kube-proxy-replacement disabled
🔮 Auto-detected datapath mode: tunnel
ℹ️ Detecting real Kubernetes API server addr and port on kind
🔮 Auto-detected kube-proxy has not been installed
ℹ️ Cilium will fully replace all functionalities of kube-proxy
```

Checking Cilium Status

Client CLI

```
$ cilium status
```

Flags:

-h, --help	help for status
--ignore-warnings	Ignore warnings when waiting for status to report success
-o, --output string	Output format. One of: json, summary (default "summary")
--wait	Wait for status to report success (no errors and warnings)
--wait-duration duration	Maximum time to wait for status (default 5m0s)
--worker-count int	The number of workers to use (default 5)

Output:

```
    /¯¯\
 /¯¯\__/¯¯\ Cilium:          OK
 \__/¯¯\__/ Operator:        OK
 /¯¯\__/¯¯\ Envoy DaemonSet: disabled (using embedded mode)
 \__/¯¯\__/ Hubble Relay:    disabled
    \__/    ClusterMesh:     disabled

Deployment    cilium-operator  Desired: 1,Ready: 1/1,Available: 1/1
DaemonSet     cilium           Desired: 1,Ready: 1/1,Available: 1/1
Containers:   cilium           Running: 1
              cilium-operator  Running: 1
Cluster Pods:  9/9 managed by Cilium
Helm chart version: 1.15.1
Image versions  cilium   quay.io/cilium/cilium:v1.15.1
                cilium-operator quay.io/cilium/operator-generic:v1.15.1
```

Agent CLI

Display the status of the Cilium daemon

```
$ cilium-dbg status [flags]
```

Flags:

--all-addresses	Show all allocated addresses, not just count
--all-clusters	Show all clusters
--all-controllers	Show all controllers, not just failing
--all-health	Show all health status, not just failing
--all-nodes	Show all nodes, not just localhost
--all-redirects	Show all redirects
--brief	Only print a one-line status message
-h, --help	help for status
-o, --output string	json | yaml | jsonpath='{}'
--timeout duration	Sets the timeout to use when querying for health (default 30s)
--verbose	Equivalent to --all-addresses --all-controllers --all-nodes --all-redirects --all-clusters --all-health

Output:

```
KVStore:                Ok         Consul: 172.17.0.3:8300
ContainerRuntime:       Ok
Kubernetes:             Disabled
Cilium:                 Ok         OK
NodeMonitor:            Listening for events on 2 CPUs with 64x4096 of shared memory
Cilium health daemon:   Ok
Controller Status:      6/6 healthy
Proxy Status:           OK, ip 10.15.28.238, port-range 10000-20000
Cluster health:         1/1 reachable   (2018-04-11T07:33:09Z)
```

Configuring Cilium

Client CLI

Manage the configuration of Cilium, this command will update the *ciliumconfig* configmap.

```
$ cilium config [command]
```

Available Commands:

delete	Delete a key in the configuration
set	Set a key/value pair in the configuration
view	View current configuration

```
$ cilium config view
```

Output:

```
agent-not-ready-taint-key  node.cilium.io/agent-not-ready
arping-refresh-period      30s
auto-direct-node-routes    false
bpf-lb-external-clusterip  false
```

```
$ cilium config set debug true
```

Output:

```
✨ Patching ConfigMap cilium-config with debug=true...
♻️ Restarted Cilium pods
```

Agent CLI

Usage:

```
$ cilium-dbg config [<option>=(enable|disable) ...] [flags]
```

```
$ cilium-dbg config [command]
```

Available Commands:

get	Retrieve cilium configuration

Flags:

-a, --all	Display all cilium configurations
-h, --help	help for config
--list-options	List available options
-n, --num-pages int	Number of pages for perf ring buffer. New values have to be > 0
-o, --output string	json| yaml| jsonpath='{}'
-r, --read-only	Display read only configurations

```
$ cilium-dbg config
```

Output:

```
##### Read-write configurations #####
ConntrackAccounting            : Enabled
ConntrackLocal                 : Disabled
Debug                          : Disabled
DebugLB                        : Disabled
DropNotification               : Enabled
```

Cilium Connectivity Tests

Run a number of prescribed tests in your environment to validate the network connectivity of a Cilium deployment

```
$ cilium connectivity test
```

This command has a number of flags to control which tests are run. The below command will show you all options.

```
$ cilium connectivity test —help
```

Some common flags are:

--agent-daemonset-name string	Name of cilium agent daemonset (default "cilium")
--agent-pod-selector string	Label on cilium-agent pods to select with (default "k8s-app=cilium")
--all-flows	Print all flows during flow validation
--collect-sysdump-on-failure	Collect sysdump after a test fails
--connect-timeout duration	Maximum time to allow initiation of the connection to take (default 2s)
-d, --debug	Show debug messages
--external-cidr string	CIDR to use as external target in connectivity tests (default "1.0.0.0/8")
--external-ip string	IP to use as external target in connectivity tests (default "1.1.1.1")
--external-other-ip string	Other IP to use as external target in connectivity tests (default "1.0.0.1")
--external-target string	Domain name to use as external target in connectivity tests (default "one.one.one.one")
--flow-validation string	Enable Hubble flow validation { disabled | warning | strict } (default "warning")
-h, --help	help for test
--host-net	Use host networking during network performance tests
--hubble	Automatically use Hubble for flow validation & troubleshooting (default true)
--include-upgrade-test	Include upgrade test
--multi-cluster string	Test across clusters to given context
--node-selector stringToString	Restrict connectivity test pods to nodes matching this label (default [])
-p, --pause-on-fail	Pause execution on test failure
--perf	Run network Performance tests
--print-flows	Print flow logs for each test
--test strings	Run tests that match one of the given regular expressions, skip tests by starting the expression with '!', target Scenarios with e.g. '/pod-to-cidr'
--test-namespace string	Namespace to perform the connectivity test in (default "cilium-test")
-t, --timestamp	Show timestamp in messages
-v, --verbose	Show informational messages and don't buffer any lines

Configuring Cilium Hubble

Client CLI

```
$ cilium hubble
```

Available Commands:

disable	Disable Hubble observability using Helm
enable	Enable Hubble observability using Helm
port-forward	Forward the relay port to the local machine
ui	Open the Hubble UI

Cilium Troubleshooting

Client CLI

Collects information required to troubleshoot issues with Cilium and Hubble by using the sysdump command. This output is useful to include when opening issues on the Cilium GitHub repository.

```
$ cilium sysdump [flags]
```

Flags:

--cilium-bugtool-flags stringArray	Optional set of flags to pass to cilium-bugtool command.
--cilium-namespace string	The namespace Cilium is running in. If not provided then the --namespace global flag is used (if provided)
--cilium-operator-namespace string	The namespace Cilium operator is running in
--cilium-spire-namespace string	The namespace Cilium SPIRE installation is running in
--cni-configmap-name string	The name of the CNI config map (default "cni-configuration")
--debug	Whether to enable debug logging
--extra-label-selectors stringArray	Optional set of labels selectors used to target additional pods for log collection.
-h, --help	help for sysdump
--hubble-flows-count int	Number of Hubble flows to collect. Setting to zero disables collecting Hubble flows. (default 10000)
--hubble-flows-timeout duration	Timeout for collecting Hubble flows (default 5s)
--logs-limit-bytes int	The limit on the number of bytes to retrieve when collecting logs (default 1073741824)
--logs-since-time duration	How far back in time to go when collecting logs (default 876h0m0s)
--node-list string	Comma-separated list of node IPs or names to filter pods for which to collect gops and logs
--output-filename string	The name of the resulting file (without extension)
--profiling	Whether to enable scraping profiling data (default true)
--quick	Whether to enable quick mode (i.e. skip collection of 'cilium-bugtool' output and logs)

Agent CLI

Request available debugging information from agent

```
$ cilium-dbg debuginfo [flags]
```

Flags:

-f, --file	Redirect output to file(s)
--file-per-command	Generate a single file per command
-h, --help	help for debuginfo
--output strings	markdown| html| json| jsonpath='{}'
--output-directory string	directory for files (if specified will use directory in which this command was ran)

These remaining commands are focused on using the Agent CLI `cilium-dbg` running inside of the Cilium Agent pods.

Manage Cilium BGP

Access the BGP control plane

```
$ cilium-dbg bgp [command]
```

Available Commands:

peers	List current state of all peers
route-policies	List configured route policies
routes	List routes in the BGP Control Plane's RIBs

```
$ cilium-dbg bgp routes advertised ipv4 unicast
```

Output:

```
VRouter  Peer        Prefix          NextHop     Age
65001    172.0.0.1   10.244.0.0/24   172.0.0.2   3m25s

Attrs
[{Origin: i} {AsPath: 65001} {Nexthop: 172.0.0.2}]
```

list connected bgp peers:

```
$ cilium-dbg bgp peers
```

Output:

```
Local AS   Peer AS   Peer Address   Session       Uptime
65001      65000     172.0.0.1:179  established   4m40s

Family         Received   Advertised
ipv4/unicast   1          1
```

Manage Cilium Endpoints

Manage endpoints

```
$ cilium-dbg endpoint [command]
```

Available Commands:

config	View & modify endpoint configuration
disconnect	Disconnect an endpoint from the network
get	Display endpoint information
health	View endpoint health
labels	Manage label configuration of endpoint
list	List all endpoints
log	View endpoint status log

```
$ cilium-dbg endpoint list
```

Output:

```
ENDPOINT POLICY (ingress) POLICY (egress)  IDENTITY
         ENFORCEMENT      ENFORCEMENT
217      Disabled         Disabled         1862

LABELS (source:key[=value])                IPv6   IPv4   STATUS
k8s:app.kubernetes.io/component=metrics    10.0.0.41      ready
k8s:app.kubernetes.io/instance=prometheus
```

```
$ cilium-dbg endpoint get 217
```

Output:

```
[
  {
    "id": 217,
    "spec": {
      "options": {
        "ConntrackAccounting": "Enabled",
        "ConntrackLocal": "Disabled",
        "Debug": "Disabled",
        "DebugLB": "Disabled",
        "DebugPolicy": "Disabled",
        "DropNotification": "Enabled",
        "MonitorAggregationLevel": "Medium",
        "PolicyAuditMode": "Disabled",
        "PolicyVerdictNotification": "Enabled",
        "SourceIPVerification": "Enabled",
        "TraceNotification": "Enabled"
      }
    },
    "status": {
      "controllers": [
        {
          "configuration": {
            "error-retry": true,
```

```
$ cilium-dbg endpoint log 217
```

Output:

```
Timestamp              Status   State
2024-03-21T09:48:16Z   OK       ready
2024-03-21T09:48:16Z   OK       ready
2024-03-21T09:48:16Z   OK       regenerating

Message
Successfully regenerated endpoint program (Reason: one or more identities created or deleted) Completed endpoint regeneration with no pending regeneration requests
Regenerating endpoint: one or more identities created or deleted
```

Manage Load Balancing

Manage services & loadbalancers

```
$ cilium-dbg service [command]
```

Available Commands:

delete	Delete a service
get	Display service information
list	List services
update	Update a service

```
$ cilium-dbg service list
```

Output:

```
ID  Frontend            Service    Type     Backend
1   10.96.0.1:443       ClusterIP  1 => 172.18.0.2:6443 (active)
2   10.96.144.206:443   ClusterIP  1 => 172.18.0.2:4244 (active)
3   10.96.0.10:53       ClusterIP  1 => 10.0.0.139:53 (active)
                                   2 => 10.0.0.63:53 (active)
```

Updating a service will not return any output response

```
$ cilium-dbg service update --frontend 127.0.0.1:80 \
  --backends 127.0.0.2:90,127.0.0.3:90 \
  --id 20
```

Accessing BPF Maps

Direct access to the BFP maps

```
$ cilium-dbg bpf [command]
```

Available Commands:

auth	Manage authenticated connections between identities
bandwidth	BPF datapath bandwidth settings
config	Manage runtime config
ct	Connection tracking tables
endpoint	Local endpoint map
fs	BPF filesystem mount
ipcache	Manage the IPCache mappings for IP/CIDR ↔ Identity
ipmasq	ip-masq-agent CIDRs
lb	Load-balancing configuration
metrics	BPF datapath traffic metrics
nat	NAT mapping tables
nodeid	Manage the node IDs
policy	Manage policy related BPF maps
recorder	PCAP recorder
sha	Manage compiled BPF template objects
srv6	Manage the SRv6 routing rules
tunnel	Tunnel endpoint map
vtep	Manage the VTEP mappings for IP/CIDR ↔ VTEP MAC/IP

```
$ cilium-dbg bpf lb list
```

Output:

```
SERVICE ADDRESS    BACKEND ADDRESS (REVNAT_ID) (SLOT)
10.96.94.145:8080 0.0.0.0:0 (11) (0) [ClusterIP, non-routable]
                  10.0.0.41:8080 (11) (1)
10.96.202.41:8080 10.0.0.91:8080 (10) (1)
                  0.0.0.0:0 (10) (0) [ClusterIP, non-routable]
10.96.144.206:443 0.0.0.0:0 (2) (0) [ClusterIP, InternalLocal, non-routable]
```

```
$ cilium-dbg bpf ct list global
```

Output:

```
UDP OUT 10.0.0.82:59993 -> 10.0.0.139:53 expires=4413
RxPackets=1 RxBytes=260 RxFlagsSeen=0x00 LastRxReport=4353
TxPackets=1 TxBytes=96 TxFlagsSeen=0x00 LastTxReport=4353
Flags=0x0000 [ ] RevNAT=3 SourceSecurityID=42426 IfIndex=0
UDP SVC 10.0.0.82:45215 -> 10.96.0.10:53 expires=4413
RxPackets=0 RxBytes=5 RxFlagsSeen=0x00 LastRxReport=0
TxPackets=0 TxBytes=0 TxFlagsSeen=0x00 LastTxReport=4353
Flags=0x0000 [ ] RevNAT=3 SourceSecurityID=0 IfIndex=0
```

```
$ cilium-dbg bpf policy list
```

Output:

```
Endpoint ID: 3156
Path: /sys/fs/bpf/tc/globals/cilium_policy_03156

POLICY DIRE  LABELS  PORT/PROTO PROXY PORT AUTH  TYPE BYTES PACKETS
Allow  Ingr  reser:  9080/TCP   NONE       disa  0    0       24
             reserv
Allow  Ingr  reser:  9200/TCP   NONE       disa  0    0       24
Allow  Ingr  k8s:ap:  ANY       NONE       disa  0    0       0
             k8s:io
             k8s:io
             k8s:io
             k8s:io
Allow  Egre  k8s:ap:  ANY       NONE       disa  0    0       0
             k8s:io
             k8s:io
             k8s:io
             k8s:io
Allow  Egre  cidr:l:  80/TCP    17324      disa  0    0       24
             reserv
```

View the Userspace BPF Map

Manage services & loadbalancers

```
$ cilium-dbg map [command]
```

Available Commands:

events	Display cached list of events for a BPF map
get	Display cached content of given BPF map
list	List all open BPF maps

```
$ cilium-dbg map list
```

Output:

```
Name                Num entries  Num errors  Cache enabled
cilium_ipcache      26           0           true
cilium_policy_03136 0            0           false
cilium_policy_02738 0            0           false
cilium_policy_02631 0            0           false
```

```
$ cilium-dbg map events cilium_ipcache
```

Output:

```
key="172.18.0.2/32" value="identity=1 encryptkey=0 tunnelendpoint=0.0.0.0" time=2024-03-21T09:47:51Z action=update desiredState=sync lastError="<nil>"
key="10.0.0.117/32" value="identity=1 encryptkey=0 tunnelendpoint=0.0.0.0" time=2024-03-21T09:47:51Z action=update desiredState=sync lastError="<nil>"
key="10.0.0.198/32" value="identity=4 encryptkey=0 tunnelendpoint=0.0.0.0" time=2024-03-21T09:47:51Z action=update desiredState=sync lastError="<nil>"
key="0.0.0.0/0" value="identity=2 encryptkey=0 tunnelendpoint=0.0.0.0" time=2024-03-21T09:47:51Z action=update desiredState=sync lastError="<nil>"
```

```
$ cilium-dbg map get cilium_ipcache
```

Output:

```
Key             Value
10.0.0.63/32    identity=45098 encryptkey=0 tunnelendpoint=0.0.0.0
10.0.0.155/32   identity=21366 encryptkey=0 tunnelendpoint=0.0.0.0
10.0.0.9/32     identity=50136 encryptkey=0 tunnelendpoint=0.0.0.0
10.96.31.77/32  identity=16777217 encryptkey=0 tunnelendpoint=0.0.0.0

State   Error
sync
sync
sync
sync
```

Validate Encryption between nodes

```
$ cilium-dbg encrypt [command]
```

Available Commands:

flush	Flushes the current IPsec state
status	Display the current encryption state

```
$ cilium-dbg encrypt [command]
```

Output:

```
Encryption: Wireguard
Interface: cilium_wg0
Public key: r44U/DzhLA1CeMXS69nkQ6npl1F9D0wu3x4A7MjU3Q4=
Number of peers: 2
```

Manage FQDN Proxy

View the FQDN cache of names to address resolution, and clear as necessary

```
$ cilium-dbg fqdn [command]
```

Available Commands:

cache	Manage fqdn proxy cache
names	Show internal state Cilium has for DNS names / regexes

```
$ cilium-dbg fqdn cache list
```

Output:

```
Endpoint Source     FQDN      TTL  ExpirationTime   IPs
1624     connection jobs-ap:  0    0001-01-01T00:   10.0.0.102
1624     connection jobs-ap:  0    0001-01-01T00:   10.0.0.102
342      lookup     elastic:  28   2024-03-21T11:   10.96.31.77
943      connection jobs-ap:  0    0001-01-01T00:   10.0.0.127
```

Displaying Cilium Agent Metrics

Access metric status

```
$ cilium-dbg metrics [command]
```

Available Commands:

list	List all metrics

```
$ cilium-dbg metrics list
```

Output:

```
Metric                          Labels                          Value
cilium_agent_api_proc:          method="DELETE" path=           0.053763
cilium_agent_api_proc:          method="DELETE" path=           0.001639
cilium_agent_api_proc:          method="GET" path="/v           0.000051
cilium_bpf_map_pressu:          map_name="ipcache"              0.000397
cilium_bpf_map_pressu:          map_name="lb4_backend           0.000366
cilium_bpf_map_pressu:          map_name="lb4_reverse           0.000366
cilium_bpf_map_pressu:          map_name="lb4_service           0.000351
cilium_errors_warnin:           level="error" subsyst           0.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
cilium_errors_warnin:           level="warning" subsy           0.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
```

Display BPF program events

The monitor displays notifications and events emitted by the BPF programs attached to endpoints and devices. This includes:

- Dropped packet notifications
- Captured packet traces
- Policy verdict notifications
- Debugging information

```
$ cilium-dbg monitor [flags]
```

Flags:

--from []uint16	Filter by source endpoint id
-h, --help	help for monitor
--hex	Do not dissect, print payload in HEX
-j, --json	Enable json output. Shadows -v flag
--monitor-socket string	Configure monitor socket path
-n, --numeric	Display all security identities as numeric values
--related-to []uint16	Filter by either source or destination endpoint id
--to []uint16	Filter by destination endpoint id
-t, --type []string	Filter by event types [agent capture debug drop l7 policy-verdict recorder trace trace-sock]
-v, --verbose bools[=false]	Enable verbose output (-v, -vv) (default [])

```
$ cilium-dbg monitor --related-to 1624
```

Output:

```
Listening for events on 8 CPUs with 64x4096 of shared memory
Press Ctrl-C to quit
<- Request http from 90 ([k8s:io.cilium.k8s.namespace.labels.kubernetes.io/metadata.name=tenant-jobs k8s:io.cilium.k8s.namespace.labels.name=tenant-jobs k8s:io.cilium.k8s.policy.cluster=default k8s:io.cilium.k8s.policy.serviceaccount=default k8s:io.kubernetes.pod.namespace=tenant-jobs k8s:app=crawler]) to 1624 ([k8s:app=loader k8s:io.cilium.k8s.namespace.labels.kubernetes.io/metadata.name=tenant-jobs k8s:io.cilium.k8s.namespace.labels.name=tenant-jobs k8s:io.cilium.k8s.policy.cluster=default k8s:io.cilium.k8s.policy.serviceaccount=default k8s:io.kubernetes.pod.namespace=tenant-jobs]), identity 752->20780, http Forwarded POST http://loader:50051/loader.Loader/LoadCv => 0
```

Manage Security Policies

```
$ cilium-dbg policy [command]
```

Available Commands:

delete	Delete policy rules
get	Display policy node information
import	Import security policy in JSON format
selectors	Display cached information about selectors
validate	Validate a policy
wait	Wait for all endpoints to have updated to a given policy revision

```
$ cilium-dbg policy get
```

Output:

```
[
    {
        "endpointSelector": {
            "matchLabels": {
                "k8s:io.kubernetes.pod.namespace": "tenant-jobs"
            }
        },
        "ingress": [
            {
                "fromEndpoints": [
                    {
                        "matchLabels": {
                            "k8s:io.kubernetes.pod.namespace": "tenant-jobs"
                        }
                    }
                ]
            }
        ],
        "egress": [
            {
                "toEndpoints": [
                    {
                        "matchLabels": {
                            "k8s:io.kubernetes.pod.namespace": "tenant-jobs"
                        }
                    }
                ]
            }
        ],
```

Displaying Cilium Agent Metrics

Access metric status

```
$ cilium-dbg metrics [command]
```

Available Commands:

list	List all metrics

```
$ cilium-dbg metrics list
```

Output:

```
Metric                          Labels                          Value
cilium_agent_api_proc:          method="DELETE" path=           0.053763
cilium_agent_api_proc:          method="DELETE" path=           0.001639
cilium_agent_api_proc:          method="GET" path="/v           0.000051
cilium_bpf_map_pressu:          map_name="ipcache"              0.000397
cilium_bpf_map_pressu:          map_name="lb4_backend           0.000366
cilium_bpf_map_pressu:          map_name="lb4_reverse           0.000366
cilium_bpf_map_pressu:          map_name="lb4_service           0.000351
cilium_errors_warnin:           level="error" subsyst           0.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
cilium_errors_warnin:           level="warning" subsy           0.000000
cilium_errors_warnin:           level="warning" subsy           1.000000
```

Inspect Statedb for stored devices, addresses and routes

```
$ cilium-dbg statedb [command]
```

Available Commands:

devices	Show contents of table "devices"
dump	Dump StateDB contents as JSON
l2-announce	Show contents of table "l2-announce"
node-addresses	Show contents of table "node-addresses"
routes	Show contents of table "routes"

```
$ cilium-dbg statedb devices
```

Output:

```
Name       Index  Selected  Type    MTU     HWAddr
lo         1      false     device  65536
eth0       5      true      veth    1500    02:42:ac:12:00:02
cilium_net 2      false     veth    1500    32:5b:be:d4:48:06
cilium_host 3     false     veth    1500    7e:cd:bf:a9:ea:90

Flags                 Addresses
up|loopback           127.0.0.1, ::1
up|broadcast|multicast  172.18.0.2, fc00:f853:ccd:e793::2, fe
up|broadcast|multicast
up|broadcast|multicast  10.0.0.117
```

```
$ cilium-dbg statedb routes
```

Output:

```
Destination   Source      Gateway     LinkIndex  Table  Scope
127.0.0.0/8   127.0.0.1                 1          255    254
127.0.0.1/32  127.0.0.1                 1          255    254
127.255.255.255/32  127.0.0.1           1          255    253
0.0.0.0/0     172.18.0.1                5          254    0
172.18.0.0/16 172.18.0.2                5          254    253
172.18.0.2/32 172.18.0.2                5          255    254
10.0.0.117/32 10.0.0.117                3          255    254
10.0.0.117/32                           3          254    253
10.0.0.0/24   10.0.0.117  10.0.0.117  3          254    0
0.0.0.0/0                               1          2004   254
```

Modifying the output of cilium-dbg

All the list commands will return a pretty printed list with the information retrieved from Cilium Daemon. If you need something more detailed you can use JSON output, to get the JSON output you can use the global option -o json

```
$ cilium-dbg endpoint list -o json
```

Moreover, Cilium also provides a JSONPath support, so detailed information can be extracted.

```
$ cilium-dbg endpoint list -o jsonpath='{[*].id}'
```

```
29898 38939 56326
```

```
$ cilium-dbg endpoint list -o jsonpath='{range [*]}{@.id}{"="}{@.status.policy.spec.policy-enabled}{"\n"}{end}'
```

```
29898=none
38939=none
56326=none
```

ISOVALENT
now part of cisco