Intel

CPA-DPA Spec

Contents

[Service Function Appliance Integration 3](#_Toc452747370)

[Control Path 3](#_Toc452747371)

[Data Path 5](#_Toc452747372)

[DPA IPC 5](#_Toc452747373)

[Command Template 5](#_Toc452747374)

[Response Template 6](#_Toc452747375)

[DPA Broker to DPA Agent Command Reference 6](#_Toc452747376)

[Command Summary 6](#_Toc452747377)

[Command “dpa-info” 6](#_Toc452747378)

[Command “get-statistics” 7](#_Toc452747379)

[Command “set-interface-endpoint-map” 8](#_Toc452747380)

[Command “update-interface-endpoint-map” 9](#_Toc452747381)

[Optional/Not Used 10](#_Toc452747382)

[Command “connection-status” 10](#_Toc452747383)

[Command “set-log-level” 11](#_Toc452747384)

[Command “inspection-start” 12](#_Toc452747385)

[Command “inspection-stop” 12](#_Toc452747386)

[Command “set-system-type” 12](#_Toc452747387)

[Command “clear-statistics” 13](#_Toc452747388)

# Service Function Appliance Integration

## Control Path

The ISC Control Path Agent (CPA) provides ISC the orchestration capability of an appliance. CPA should be deployed and be part of the boot strapping sequence of the appliance.

* Prerequisites
  + Optional: JRE 1.7.0\_65 or higher
  + Mountable CDROM device
* Extraction vmidc.tar on appliance disk
  + CPA include JRE, but if appliance already includes it, you can modify vmidc.sh to replace usage of incorporated JRE with java binary path already installed on the appliance.
* Appliance bootstrap to call “vmidc.sh --start” and shutdown “vmidc.sh --stop”
  + Highly recommended: Open port 8090 as part of bootstrap sequence thus not requiring implementation of enable-firewall-isc-port.py
* Watchdog CPA process
  + Highly recommended: Monitor CPA process. PID is written to /usr/run/cpa.pid.
* Implement python scripts
  + Unless stated otherwise, script accepts not arguments, expected to return zero error code for success or non-zero error code for failure. In case of error, standard out represent error message.
  + get-name.py.
    - Mandatory
    - Called when appliance status is being reported. Get name of the appliance. This name descript the appliance (i.e. vnsp, ngfw, snort)
    - No arguments.
    - Standard output return string representing appliance name.
  + mount-cdrom.py
    - Mandatory
    - Called when CPA needs to mount cd-rom. Script will mount the cd-rom device.
    - No arguments.
  + get-cdrom-mount-path.py
    - Mandatory
    - Called when CPA needs to read files from cd-rom.
    - No arguments.
    - Standard output returns string representing the path location cd-rom had been mounted.
  + enable-firewall-isc-port.py
    - Optional (if not configured on appliance bootstrap).
    - Called every time appliance status is report (currently hard coded to every 3 minutes) Opens incoming port 8090
    - No arguments
  + set-network-info.py
    - Mandatory
    - Called when appliance network configuration is not yet set.
    - Checked each status interval (every 3 minutes).
    - Arguments
      * ip
      * netmask
      * gateway
      * mtu
  + set-mgmt-info.py
    - Mandatory
    - Called when appliance manager configuration is not yet set.
    - Checked each status interval (every 3 minutes).
    - Configure manager discovery and authentication
    - Arguments
      * appliance-name
      * mgr-ip
      * appliance configuration 1
      * appliance configuration 2
      * 0/1. boolean. 1 to re-authenticate.
  + check-authenticated.py
    - Mandatory
    - Called to find out if manager configuration is present
    - Exit code 0 if already authenticated by manager. Non-Zero otherwise.
  + check-discovered.py
    - Mandatory
    - Called each status interval
    - Exit code 0 if already discovered by manager. Non-Zero otherwise.
  + check-inspection-ready.py
    - Mandatory
    - Called each status interval
    - Exit code 0 if appliance is ready to inspect traffic. Non-Zero otherwise.
  + set-cli-password.py
    - Mandatory
    - Called when ISC propagate ssh/console CLI password of appliances
    - Arguments
      * Old password
      * New password
  + persist-file.py
    - Optional (if appliance requires special processing for persisting a file cross reboot).
    - Called every time essential file is being persisted. Perform special handling to ensure file will persist cross reboots
    - Arguments
      * Full path to file name
  + process-mgr-file.py
    - Optional
    - ISC supports propagation of a ‘file’ to all VSS contained appliances
    - Called when ISC push the file (like policies) to appliance. Script handles processing of incoming propagated file.
    - Argument
      * File name

## Data Path

* For NSX
  + Prerequisite: VMware Tools
  + Implement NetX callbacks
* For Openstack
  + Assumption: 2 network interfaces – first is management, second is inspection.
  + Bridge process to read/write packets on inspection interface.
  + Bridge to support tunneling de-encapsulation to extract policy tag and packet direction
* Incorporate and Implement DPAIPC callbacks
  + DPAIPC is a library which provide ISC visibility into some static information and packet processing statistics.
  + Report DPA static info and runtime packet info
  + If applicable, process Security Group to L2/L3 (MAC/IP) mappings. This can be used to translate security groups which are part of policies.

### DPA IPC

* Stream based protocol. Synchronous calls, broker sends command, waits for response
* All commands are json objects, all response are json objects
* All command parameters are key/value pairs
* All response values are key/value pairs
* Response is required for every command
* Keys of object are not order dependent
* Every command object must contain the key “cmd” and have a string value which is the name of the command. All parameter names are implied by the command
* Every response object must contain the key “status” and have an integer value. The meaning of the status is defined for each command
* If either the client or the server receives detects an invalid JSON object it must close the connection. Server is then responsible for reconnecting
* If the stream is idle for more than 30 seconds either side should close the connection.
* Client can handle multiple connections at once put will be singly threaded.
* All commands and responses shall be terminated by a NVT newline (CR LF).
* Newline characters shall appear only at the end of a complete object (e.g. command or response).
* Server must listen on TCP port 10613 only on the loopback (127.0.0.1) address.

### Command Template

{

“cmd” : “<command-name>”,

“<param-name1>” : <parm-value1>,

“<param-name2>” : <parm-value2>,

…..

}

### Response Template

{

“status” : <status as integer>,

“error”: “error string”

“response”: {

<DATA TYPE DETERMINED BY REQUEST>

}

}

## DPA Broker to DPA Agent Command Reference

### Command Summary

The DPA Agent will support the following commands

|  |  |
| --- | --- |
| Command | Description |
| dpa-info | Information about the Data Path Agent, e.g. name and version |
| get-statistics | Return the current statistics |
| If Security group sync is Supported following API’s are applicable | |
| set-interface-endpoint-map | Sets the service profile to policy id map |
| update-interface-endpoint-map | Update a single entry in the service profile map |

## Command “dpa-info”

Return information about the DPA and the IPC library

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “dpa-info” |

**Example**

{

“cmd” : “dpa-info”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| Status | Integer | One of the following  200 |
| response | Object | Contains the following: |
| DPA-name | String | Name of the DPA |
| DPA-version | String | Version of the DPA |
| IPC-version | String | Version of the DPA library |
| dpa-pid | Long | DPA Process ID |

{

“status” : 200,

“response”: {

“DPA-name” : “Stonesoft DPA”,

“DPA-version” : “1.1.0 Patch 2”,

“IPC-version” : “1.1”,

“dpa-pid” : 1319

}

}

## Command “get-statistics”

Return the current statistics about filter statistics.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| cmd | string | “get-statistics” |
| filter | string | Optional name to return statistics about |

**Example**

{

“cmd” : “get-statistics”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Statistics returned  400 – Statistics not ready  401 – filter not found |
| response | Object | Contains the following: |
| TBD | ???? | To be determined |

{

“status”: 200,

“response”: {

"currentTicks": "1460448811388087",

"workloadInterfaces": "0",

"rx": "5",

"txSva": "5",

"txResource": "0",

"dropSva": "0",

"dropError": "0",

"dropResource": "0",

"rxInQueue": "0",

"rxSize": "4095",

"txInQueue": "0",

"txSize": "40"

}

}

## Command “set-interface-endpoint-map”

Replace the entire service profile map. The DPA agent upon receiving this message will replace all entries in its internal cache to the values in the call. The server MUST issue this call upon first connection and on each subsequent reconnection before sending a update-interface-endpoint-map.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| cmd | string | “set-interface-endpoint-map” |
| map | array | An array of service profile entry objects |
| service profile entry | | |
| interfaceTag | String | VMWare service profile id/ Security group interface tag |
| endpointGroupList | Object | List of endpoint groups |

|  |  |  |
| --- | --- | --- |
| EndpointGroup | | |
| id | String | NSX Security group ID or null in case of openstack |
| name | String | Name of security group |
| type | String | IP or MAC |
| addresses | List<String> | List of addresses. IP or mac(cannot have mixed) |

**Example**

{

"cmd": "set-interface-endpoint-map",

"map": [{

"interfaceTag": "serviceprofile-01",

"endpointGroupList": [{

"id": "foo-1",

"name": "foo",

"type": "MAC",

"addresses": ["ff: ff: ff: ff: ff: ff",

"f1: ff: ff: ff: ff: ff"]

}],

"interfaceTag": "serviceprofile-02",

"endpointGroupList": [{

"id": "foo-2",

"name": "foo2",

"type": "IP",

"addresses": ["10.0.0.1",

"10.0.0.2"]

}]

}]

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  400 – Missing Parameters  401 – Update failed |

{

“status” : 200

}

## Command “update-interface-endpoint-map”

Add/replace/delete one or more entries in the service profile map. The DPA agent upon receiving this message will replace the specified entries in its internal cache with the values in the call. A null or missing value for a policy-id indicates a deleted entry.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| cmd | string | “update-interface-endpoint-map” |
| map | array | An array of service profile entry objects (see “set-interface-endpoint-map”) |

**Example**

{

“cmd” : “update-serviceprofile-map”,

"map": [{

"interfaceTag": "serviceprofile-01",

"endpointGroupList": [{

"id": "foo-1",

"name": "foo",

"type": "MAC",

"addresses": ["ff: ff: ff: ff: ff: ff",

"f1: ff: ff: ff: ff: ff"]

}],

"interfaceTag": "serviceprofile-02",

"endpointGroupList": [{

"id": "foo-2",

"name": "foo2",

"type": "IP",

"addresses": ["10.0.0.1",

"10.0.0.2"]

}]

}]

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  400 – Missing Parameter  401 – Update failed unspecified  402 – Update failed service profile not found |

{

“status” : 200

}

# Optional/Not Used

## Command “connection-status”

Return the current connection status. Used as a heartbeat to keep the connection in case the connection has gone idle. The server should request status if it has not made a request in the last 15 seconds.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| cmd | string | “connection-status” |

**Example**

{

“cmd” : “connection-status”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| Status | Integer | One of the following  200 - Connection Active  201 – Connection Active and DPA has not updated statistics in the last 10 seconds |
| last-update | String | ISO-8601 string of the time in UTC of the last DPA statistics update |
| current-time | String | ISO-8601 string of the time in UTC of the DPA machine |

**Example**

{

“status” : 200,

“response”: {

“last-update” : “2014-03-19T01:47:32Z”,

“current-time” : “2014-03-19T01:47:34Z”

}

}

## Command “set-log-level”

Sets dpa log level

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “set-log-level” |
| Filter (optional) | string | Regex describing the subsystem to set log level for.  If missing or null default is “.\*” |
| Level | string | One of the following:  NOTSET=0  CRITICAL =10  ERROR=20  WARNING=30  INFO=40  DEBUG=50 |

**Example**

Sets all filters to CRITICAL log level

{

“cmd” : “set-log-level”,

“Level”: “CRITICAL”

}

Sets subsystem “netx” , “rings” or “packets” to DEBUG log level

{

“cmd” : “set-log-level”,

“Filter” : “{netx|packets|rings}”

“Level”: “DEBUG”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  400 – Missing Parameters  401 – Update failed |

{

“status” : 200

}

## Command “inspection-start”

Indicates that the DPA should start inspecting traffic and sending it to the system. DPAIPC will call a registered callback when “inspection-start” is invoked.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “inspection-start” |

**Example**

{

“cmd” : “inspection-start”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  401 – inspection start callback not registered |

{

“status” : 200

}

## Command “inspection-stop”

Indicates that the DPA should stop inspecting traffic and sending it to the system. DPAIPC will call a registered callback when “inspection- stop” is invoked.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “inspection- stop” |

**Example**

{

“cmd” : “inspection- stop”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  401 – inspection stop callback not registered |

{

“status” : 200

}

## Command “set-system-type”

Indicates that the DPA what system type it is. DPAIPC will call a registered callback when “set-system-type” is invoked.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “set-system-type” |
| system-type | string | The type of the system |

**Example**

{

“cmd” : “set-system-type”

“system-type” : “AKA9502”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  400 – missing parameter  401 – set-system-type callback not registered |

{

“status” : 200

}

## Command “clear-statistics”

Indicates that the DPA must clear all statistics associated with the get-statistics command. DPAIPC will call a registered callback when “clear-statistics” is invoked.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Cmd | string | “clear-statistics” |

**Example**

{

“cmd” : “clear-statistics”

}

|  |  |  |
| --- | --- | --- |
| Return | Type | Description |
| status | Integer | One of the following  200 - Update successful  401 – clear statistics callback not registered |

{

“status” : 200

}