*The Johns Hopkins University Applied Physics Laboratory*

*Sarah Heiner*

[*Sarah.Heiner@jhuapl.edu*](mailto:Sarah.Heiner@jhuapl.edu)

*December 2021*

*BPSec Policy Configuration*

Bundle Protocol Security Policy Configuration for ION 4.2

Table of Contents

[1. Introduction 2](#_Toc64472487)

[2. Event Set Commands 2](#_Toc64472488)

[2.1. Event Set Command Elements 2](#_Toc64472489)

[2.2. Adding an Event Set 2](#_Toc64472490)

[2.2.1. Command 2](#_Toc64472491)

[2.2.2. Example Usage 3](#_Toc64472492)

[2.3. Displaying Event Set Information 3](#_Toc64472493)

[2.3.1. Command 3](#_Toc64472494)

[2.3.2. Example Usage 3](#_Toc64472495)

[2.4. Deleting an Event Set 3](#_Toc64472496)

[2.4.1. Command 3](#_Toc64472497)

[2.4.2. Example Usage 4](#_Toc64472498)

[2.5. Listing Event Sets 4](#_Toc64472499)

[3. Event Commands 4](#_Toc64472500)

[3.1. Event Command Elements 4](#_Toc64472501)

[3.1.1. Security Operation Events 5](#_Toc64472502)

[3.1.2. Processing Actions 5](#_Toc64472503)

[3.1.3. Supported Configurations 5](#_Toc64472504)

[3.2. Adding an Event to an Event Set 6](#_Toc64472505)

[3.2.1. Command 6](#_Toc64472506)

[3.2.2. Example Usage 6](#_Toc64472507)

[3.3. Deleting an Event from an Event Set 7](#_Toc64472508)

[3.3.1. Command 7](#_Toc64472509)

[3.3.2. Example Usage 8](#_Toc64472510)

[4. Security Policy Rule Commands 8](#_Toc64472511)

[4.1. Security Policy Rule Command Format 8](#_Toc64472512)

[4.1.1. Filter Criteria 9](#_Toc64472513)

[4.1.2. Specification Criteria 10](#_Toc64472514)

[4.1.3. Event Criteria 11](#_Toc64472515)

[4.2. Adding Security Policy Rules 11](#_Toc64472516)

[4.2.1. Commands 11](#_Toc64472517)

[4.3. Deleting Security Policy Rules 12](#_Toc64472518)

[4.3.1. Command 12](#_Toc64472519)

[4.3.2. Example Usage 13](#_Toc64472520)

[4.4. Displaying Security Policy Rule Information 13](#_Toc64472521)

[4.4.1. Command 13](#_Toc64472522)

[4.5. Find Security Policy Rules 13](#_Toc64472523)

[4.5.1. Command 13](#_Toc64472524)

[4.5.2. Example Usage 14](#_Toc64472525)

[4.6. Listing Policy Rules 15](#_Toc64472526)

# Introduction

The Security Policy Software (SPS) will add commands to the bpsecadmin utility to allow user definition and configuration of security policy rules and security operation event sets. The addition of these commands will not impact the support for any of the commands currently defined in bpsecadmin.

### 

The SPS will support the use of JSON syntax to provide a flexible and lightweight approach to defining security policy.

This document provides an overview of all security policy commands supported by the SPS as well as example commands and an explanation of their impact on the system.

# Event Set Commands

The SPS adds commands to the bpsecadmin utility to allow for the creation and management of event sets. An event set can be associated with one or more security operation events, for which details can be found in Section 3 Event Commands.

All event sets created using the commands below are *named* event sets. By providing a unique name for the event set, the user can then associate that event set with one or more policy rules in the system.

An event set must be created before it can be associated with a security policy rule. In other words, create an event set before adding the security policy rule(s) that must use it.

## Event Set Command Elements

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Element | Description |
| name | String | Event Set Name | The unique name used to identify the security operation event set. |
| desc | String | Description | Optional.  A description of the named event set. Limit: 12 characters. |

## Adding an Event Set

### Command

The following command is used to add a named security operation event set to the system.

**a {“event\_set” :**

**{**

**“name” : “<event set name>“,**

**“desc” : “<(opt) description>“**

**}**

**}**

### Example Usage

**﻿a {“event\_set” :**

**{**

**“name” : “d\_integrity”,**

**“desc” : “default event set for bib-integrity operations”**

**}**

**}**

This sample command creates an event set by the name ‘d\_integrity’ to be used as the default event set for all bib-integrity security operations. After execution of this command, the d\_integrity event set is available to be associated with security policy rules.

## Displaying Event Set Information

### Command

The following command is used to display the information the system maintains for a named event set. The security operation events and configured, optional processing actions associated with the event set are presented.

**i {“event\_set” :**

**{“name” : “<event set name>“}**

**}**

### Example Usage

**﻿i {“event\_set” :**

**{“name” : “d\_integrity”}**

**}**

This sample command displays the security operation events and their associated optional processing actions for the d\_integrity event set.

## Deleting an Event Set

### Command

The following command is used to delete a named event set from the system. A named event set *cannot* be deleted if it is referenced by a security policy rule. All security policy rules associated with the named event set must be deleted before the event set itself may be deleted.

**d {“event\_set” :**

**{“name” : “<event set name>“}**

**}**

### Example Usage

**﻿d {“event\_set” :**

**{“name” : “d\_integrity”}**

**}**

This sample command deletes the d\_integrity event set from the system.

## Listing Event Sets

The following command is used to list every named event set defined in the system. For each event set, the number of associated security policy rules is displayed as well.

#### **﻿l {“event\_set”}**

# Event Commands

The SPS adds commands to the bpsecadmin utility which allow for the configuration of security operation events and optional processing actions associated with event sets.

A named event set can be modified by adding and deleting events until it is associated with a security policy rule. Once a security policy rule in the system references the event set, it can no longer be modified.

## Event Command Elements

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Element | Description |
| es\_ref | String | Event Set Reference | The name of the event set to be modified by the event command. |
| event\_id | String | Security Operation Event ID | The security operation event for which optional processing actions are to be configured. Event IDs are enumerated in Section 3.1.1. |
| actions | String | Processing Actions | The optional processing action(s) to enable for the specified security operation event. Actions are enumerated in Section 3.1.2.  Actions must be provided as an **array**.  Processing action parameters may be included in this field. The parameter should immediately follow the action it is associated with. Provide each parameter as an ID, value pair.  Supported parameter IDs:   1. **“reason\_code”** 2. **“new\_value”** 3. **“mask”** |

### Security Operation Events

The following security operation events are valid values for the **event\_id** field.

1. **﻿”source\_for\_sop”**
2. **“sop\_added\_at\_source”**
3. **“sop\_misconfigured\_at\_source”**
4. **“verifier\_for\_sop”**
5. **“sop\_misconfigured\_at\_verifier”**
6. **“sop\_missing\_at\_verifier”**
7. **“sop\_corrupted\_at\_verifier”**
8. **“sop\_verified”**
9. **“acceptor\_for\_sop”**
10. **“sop\_misconfigured\_at\_acceptor”**
11. **“sop\_missing\_at\_acceptor”**
12. **“sop\_corrupted\_at\_acceptor”**
13. **“sop\_processed”**

### Processing Actions

The following processing actions are currently implemented by the SPS and are valid values for the **actions** field.

1. **﻿”remove\_sop”**
2. **“remove\_sop\_target”**
3. **“remove\_all\_target\_sops”**
4. **“do\_not\_forward”**
5. **“report\_reason\_code”**

### Supported Configurations

The table below indicates all SPS supported security operation events and processing actions. Cells marked with an **X** indicate that the processing action is permitted to be enabled for that security operation event.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Remove SOP | Remove SOP Target | Remove all Target SOPs | Do Not Forward | Report Reason Code |
| **Source for SOP** |  |  |  |  |  |
| **SOP Added at Source** |  |  |  |  |  |
| **SOP Misconfigured at Source** | X | X | X | X | X |
| **Verifier for SOP** |  |  |  |  |  |
| **SOP Misconfigured at Verifier** | X | X |  | X | X |
| **SOP Missing at Verifier** |  | X |  | X | X |
| **SOP Corrupted at Verifier** | X | X | X | X | X |
| **SOP Verified** |  |  |  |  |  |
| **Acceptor for SOP** |  |  |  |  |  |
| **SOP Misconfigured at Acceptor**  **Acceptor** |  | X |  | X | X |
| **SOP Missing at Acceptor** |  | X |  | X | X |
| **SOP Corrupted at Acceptor** |  | X | X | X | X |
| **SOP Processed** |  |  |  |  |  |

## Adding an Event to an Event Set

### Command

The following command is used to add a security operation event and associated optional processing action(s) to an event set. Multiple processing actions can be specified for a single security operation event.

If the security operation event included in the command has already been specified for the event set, the optional processing actions provided in the command will replace the action(s) originally configured for that event in the event set.

**a {“event” :**

**{**

**“es\_ref” : “<event set name>“,**

**“event\_id” : “<security operation event ID>“,**

**“actions” : [{“id”: “<processing action>“,**

**“<(opt.) action parm key>“ : <(opt.) parm value>}, … ,**

**{“id”: “<processing action>“,**

**“<(opt.) action parm key>“ : <(opt.) parm value>}]**

**}**

**}**

### Example Usage

**﻿a {“event” :**

**{**

**“es\_ref” : “d\_integrity”,**

**“event\_id” : “sop\_misconfigured\_at\_acceptor”,**

**“actions” : [{“id” : “remove\_sop\_target”}]**

**}**

**}**

**a {“event” :**

**{**

**“es\_ref” : “d\_integrity”,**

**“event\_id” : “sop\_missing\_at\_acceptor”,**

**“actions” : [{“id” : “report\_reason\_code”,**

**“reason\_code” : 8}]**

**}**

**}**

**a {“event” :**

**{**

**“es\_ref” : “d\_integrity”,**

**“event\_id” : “sop\_corrupted\_at\_acceptor”,**

**“actions” : [{“id” : “remove\_sop\_target”},**

**{“id” : “report\_reason\_code”,**

**“reason\_code”: 8}]**

**}**

**}**

This sample sequence of commands adds events and optional processing actions to the *d\_integrity* event set.

The first command adds the security operation event *sop\_misconfigured\_at\_acceptor* and sets the optional processing action for this event to *remove\_sop\_target*.

The second command sets the *report\_reason\_code* action for the *sop\_missing\_at\_acceptor* event. The reason code (8, for Block Unintelligible) to report is provided as an action parameter, labeled as *reason\_code*.

The third command adds both the *remove\_sop\_target* and *report\_reason\_code* actions for the *sop\_corrupted\_at\_acceptor* event.

## Deleting an Event from an Event Set

### Command

The following command is used to delete a security operation event from a named event set. This results in the removal of *all* optional processing actions configured for that event.

**d {“event” :**

**{**

**“es\_ref” : “<event set name>“,**

**“event\_id” : “<security operation event ID>“**

**}**

**}**

### Example Usage

**﻿d {“event” :**

**{**

**“es\_ref” : “d\_integrity”,**

**“event\_id” : “sop\_corrupted\_at\_acceptor”**

**}**

**}**

The sample command above deletes all optional processing actions for the *sop\_corrupted\_at\_acceptor* event in the *d\_integrity* event set.

# Security Policy Rule Commands

The SPS adds commands to the bpsecadmin utility to allow for the creation and management of security policy rules which may be associated with named event sets.

A security policy rule defines a security operation which is required for the bundle(s) that match its Filter Criteria.

## Security Policy Rule Command Format

A security policy rule command is composed of three components:

1. Filter criteria
2. Specification criteria
3. Event criteria

A security policy rule command follows the general format:

**<action> { “policyrule”:**

**{**

**“desc” : “<(opt.) description>“,**

**“filter” : {<filter criteria>},**

**“spec” : {<specification criteria>},**

**“es\_ref” : {<event criteria>}**

**}**

**}**

### Filter Criteria

**Configuration Note:**

Three fields are present in the filter criteria in which the user can specify an Endpoint ID.

To construct a valid filter, an EID **must** be provided for **at least one** of the following fields:

1. Bundle Source
2. Bundle Destination
3. Security Source

The filter criteria field in a security policy rule command is used to identify :

1. The bundle(s) the rule applies to
2. The block(s) in those bundles that are security targets of the specified security operation.
3. The security policy role the BPA applying the rule must play.

When the security policy role identified in the filter criteria field is Security Source, the bundle for which the rule applies to is identified using the EID(s) and security target type information provided in the filter criteria.

**Configuration Note:**

The **sc\_id** field is present in the **filter criteria** when the associated security policy role is *Security Verifier* or *Security Acceptor*. The security context is identified in the **specification criteria** for a *Security Source.*

When the security policy role identified in the filter criteria field is either Security Verifier or Acceptor, the bundle for which the rule applies to is identified using the EID(s) and security target type information provided. Then, to identify the block in the bundle for which the rule applies, the **tgt** and **sc\_id** fields must be used.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Value Type | Element | Description |
| rule\_id | Integer | Security Policy Rule ID | A user-provided value to uniquely identify the security policy rule.  Rule ID can be assigned to any uint16\_t value except 0, which is reserved. |
| desc | String | Description | Optional.  A description of the security policy rule (limit: 64 characters). |
| role | String | Security Policy Role | Security policy roles are: Security Source, Security Verifier, or Security Acceptor. These options are shortened to:   1. **“sec\_source”** or **“s”** 2. **“sec\_verifier”** or **“v”** 3. **“sec\_acceptor”** or **“a”**. |
| src | String | Bundle Source EID Expression | The endpoint identifier of the bundle source. The EID may contain an end-of-string wildcard character: ‘~’.  For example, the EIDs “ipn:1.1”, “ipn:1.~”, and “ipn~” are all valid entries for this field. |
| dest | String | Bundle Destination EID Expression | The endpoint identifier of the bundle destination. The EID may contain an end-of-string wildcard character: ‘~’. |
| sec\_src | String | Security Source EID Expression | The endpoint identifier of the security source, which may be different from the bundle source. The EID may contain an end-of-string wildcard character: ‘~’. |
| tgt | Integer | Security Target Block Type | The block type number of the security operation’s target. |
| sc\_id | Integer | Security Context ID | The ID of the security context to use when applying the security operation to the bundle.   * **BIB-HMAC-SHA2** is sc\_id **2** * **BCB-AES-GCM** is sc\_id **8** |

### Specification Criteria

The specification criteria field in a security policy rule command provides the security service and security context information required to apply the rule to the bundle.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Value Type | Element | Description |
| svc | String | Security Service | The SPS supports the following security services:   1. “bib-integrity” 2. “bcb-confidentiality” |
| sc\_id | Integer | Security Context ID | The ID of the security context to use when applying the security operation to the bundle. |
| sc\_parms | String | Security Context Parameters | Parameter(s) to be used by the security context when applying the security operation to the bundle. Parameters must be represented as ID-value pairs in an array.  Supported security context parameter IDs are:   1. ﻿”key\_name” 2. “iv” 3. “salt” 4. “icv” 5. “intsig” 6. “bek” 7. “bekicv” |

### Event Criteria

The event criteria field in a security policy rule command is used to associate a security policy rule with either a named event set or an anonymous event set. The event set determines how security operation events are handled.

**The SPS currently supports named event sets only.**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Value Type | Element | Description |
| es\_ref | String | Event Set Reference | The name of the event set to associate with the security policy rule. |

## Adding Security Policy Rules

### Commands

The following command is used to add a security policy rule referencing a **named event set** to the system:

**a {“policyrule” :**

**{**

**“desc” : “<(opt.)description>“,**

**“filter” :**

**{**

**“rule\_id” : <security policy rule ID>,**

**“role” : “<security policy role>“,**

**“src” : “<bundle source>“,**

**“dest” : “<bundle destination>“,**

**“sec\_src” : “<security source>“,**

**“tgt” : <security target block type>,**

**“sc\_id” : <security context ID>**

**},**

**“spec” :**

**{**

**“svc” : “<security service>“,**

**“sc\_id” : <security context ID>,**

**“sc\_parms” : [{“<parm1 ID>” : “<parm1 value>”}, … ,**

**{“<parmn ID>” : “<parmn value>”}]**

**},**

**“es\_ref” : “<event set name>“**

**}**

**}**

##### Example Usage

The following command is used to add a security policy rule to the system and associate it with a named event set.

**﻿a {“policyrule” :**

**{**

**“desc” : “Verify payloads originating from any endpoint**

**destined for ipn:2.1”,**

**“filter” :**

**{**

**“rule\_id” : 1,**

**“role” : “sec\_verifier”,**

**“src” : “ipn:~”,**

**“dest” : “ipn:2.1”,**

**“tgt” : 1,**

**“sc\_id” : 8**

**},**

**“spec”:**

**{**

**“svc” : “bib-integrity”**

**“sc\_parms” : [{”key\_name”:”hmac\_key256”}]**

**},**

**“es\_ref” : “d\_integrity”**

**}**

**}**

This sample security policy rule is used to require the verification of an integrity security operation targeting the payload block of any bundle with a destination of ipn:2.1. That integrity operation must use the security context BIB-HMAC-SHA-256, which has a security context ID of 8.

During verification, the integrity signature for the payload block is generated using the BIB-HMAC-SHA-256 security context and the key for verification as a is provided as a security context parameter with ID “key\_name” and value “hmac\_key256”.

The *d\_integrity* event set provides the optional processing actions for this rule. Note that this event set must be defined in the system before it is referenced in the security policy rule.

## Deleting Security Policy Rules

### Command

The following command is used to delete an existing security policy rule:

**d {“policyrule” :**

**{“rule\_id” : <security policy rule ID>}**

**}**

The **rule\_id** value 0 is reserved. If **rule­\_id** is set to 0 in the delete command, all defined security policy rules are deleted. Otherwise, the security policy rule with the matching ID will be removed.

**Configuration Note:**

Deleting a security policy rule requires use of the **rule\_id** to uniquely identify that security policy rule. Filter criteria are not used to identify the security policy rule to remove as this can result in unintuitive behavior.

**Example:** If a user defined a security policy rule with *general* filter criteria (src: “ipn~” and dest: “ipn:1.~”) and then issued a delete command with more *specific* filter criteria (src: “ipn:1.1”, dest: ipn:1.~”), the rule with more general filter criteria would be matched and removed. For this reason, we use rule IDs.

### Example Usage

**﻿d {“policyrule” :**

**{“rule\_id” : 2}**

**}**

This command deletes security policy rule 2 from the system.

**﻿d {“policyrule” :**

**{“rule\_id” : 0}**

**}**

This command deletes all defined security policy rules from the system.

## Displaying Security Policy Rule Information

### Command

The following command displays the information maintained for the security policy rule matching the provided rule ID.

**i {“policyrule” :**

**{“rule\_id” : <security policy rule ID>}**

**}**

#### Example Usage

**i {“policyrule” :**

**{“rule\_id” : 2}**

**}**

This command displays the details for the security policy rule with ID 2.

## Find Security Policy Rules

### Command

The find command provides the policy rule ID(s) of the rule(s) matching the provided filter criteria.

**Configuration Note:**

To find every policy rule matching the filter, the **type** field is set to **all**.

To find the best match for the filter, the **type** field is set to **best**.

The result of a **best** policy rule find command can be used to determine the security policy rule that will be applied to a bundle with the provided characteristics.

Use the **all** policy rule find command to determine/verify the rule ID of a policy rule matching the filter criteria given. This command is particularly helpful if the user wants to issue a **delete** or **info** command but is unsure of the rule ID to provide.

**f {“policyrule” :**

**{**

**“type” : “all” | “best”,**

**“src” : “<bundle source>“,**

**“dest” : “<bundle destination>“,**

**“sec\_src” : “<security source>“,**

**“sc\_id” : <security context ID>,**

**“role” : “<security policy role>“**

**}**

**}**

### Example Usage

**﻿f {“policyrule” :**

**{**

**“type” : “all”,**

**“src” : “ipn:2.1”**

**“dest” : “ipn:3.1”**

**}**

**}**

This command can be used to find all security policy rules that apply to bundles originating from ipn:2.1 (the bundle source) that are destined for ipn:3.1 (the bundle destination”. The rule IDs for all matching policy rules will be displayed.

**﻿f {“policyrule”:**

**{**

**“type” : “best”,**

**“src” : “ipn:2.1”,**

**“dest” : “ipn:3.1”,**

**“role” : “s”**

**}**

**}**

This command can be used to check what policy rule may be applied to a bundle.

In this example, the user enters the relevant bundle details – its source node is ipn:2.1 and its destination is ipn:3.1. Providing the Security Source role will refine the policy rule search further.

The result provided from this command will be the rule ID of the security policy rule identifying the current node as the Security Source for bundles matching the provided criteria, if such a rule exists. If further information is desired the user should then use the policy rule info command with the provided rule ID to determine information such as:

1. The security operation that will be added to the bundle by the security source node.
2. The target block type of the security operation.
3. The security context that will be used to add the required security service.

## Listing Policy Rules

The following command is used to list every security policy rule currently defined in the system. The rule ID and optional description for each rule will be displayed.

**l {“policyrule”}**

# Deprecation of Transmit and Receive Rules

The transmit and receive rules used to require security services for the Streamlined Bundle Security Protocol (SBSP) implementation in ION for BPv6 were migrated to the BPSec implementation in ION for BPv7. The bpsecadmin utility supported both transmit and receive rules and the JSON security policy rules in ION releases 4.0.2, 4.1, and 4.1.1.

Starting in the ION Open Source 4.2 release, transmit and receive rules are deprecated and **security policy rules are the supported method for policy configuration** using the bpsecadmin utility.

The conversion between transmit/receive rules and security policy rules is straightforward, as all transmit/receive rule fields translate directly into the JSON syntax shown throughout this manual. Examples of this conversion are provided to assist network operators in making this transition.

### Adding a Security Policy Rule

﻿

A rule following the transmit/receive syntax requiring a BIB on the Payload Block may look like the following:

**a bibrule ipn:2.\* ipn:3.\* 1 'BIB-HMAC-SHA256' bibkey**

To convert this rule to the updated security policy syntax, we will examine each field individually.

**a** – Indicates that the user is adding a rule.

**bibrule** – This rule applies to the bib-integrity security service.

**ipn:2.\*** - The bundle source for this rule is any EID matching this pattern.

**ipn:3.\*** - The bundle destination for this rule is any EID matching this pattern.

**1** – The target block type for this rule is 1: the Payload Block.

**BIB-HMAC-SHA2** – The security context, corresponding to ID 2 to be used when applying the bib-integrity service.

**bibkey** – The **name of the key** to use when applying the bib-integrity service.

To create this transmit/receive rule as a security policy rule, there are a few changes to make to the rule contents. The first step in creating a security policy rule is to define an event set to associate it with. Section 2.2, Adding an Event Set, describes this process in detail. For simplicity, we can define an empty Event Set to use for this example:

**a {“event\_set” :**

**{**

**“name” : “default\_es”,**

**“desc” : “default event set for example purposes”**

**}**

**}**

With the event set for this example defined, the following security policy rules can be added. The first security policy rule identifies the Security Source BPA, and should be added at the node which will create the BIB and add it to the bundle.

**a {“policyrule” :**

**{**

**“desc” : “<(opt.)description>“,**

**“filter” :**

**{**

**“role” : “sec\_source”,**

**“src” : “ipn:2.\*“,**

**“dest” : “ipn:3.\*“,**

**“tgt” : 1**

**},**

**“spec” :**

**{**

**“svc” : “bib-integrity”,**

**“sc\_id” :** 2**,**

**“sc\_parms” : [{“key\_name” : “bibkey”}]**

**},**

**“es\_ref” : “default\_es“**

**}**

**}**

The second security policy rule identifies the Security Acceptor BPA and shouldbe added at the node which will process the BIB, checking the integrity of its target, the Payload block, and removing it from the bundle. Security Verifier rules can also be added using a nearly identical format, with the security role changed to “sec\_verifier”.

**a {“policyrule” :**

**{**

**“desc” : “<(opt.)description>“,**

**“filter” :**

**{**

**“role” : “sec\_acceptor”,**

**“src” : “ipn:2.\*“,**

**“dest” : “ipn:3.\*“,**

**“tgt” : 1,**

**“sc\_id” :** 2

**},**

**“spec” :**

**{**

**“svc” : “bib-integrity”,**

**“sc\_parms” : [{“key\_name” : “bibkey”}]**

**},**

**“es\_ref” : “default\_es“**

**}**

**}**