BPSEC POLICY CONFIGURATION

Bundle Protocol Security Policy Configuration for ION 4.0.2

The Johns Hopkins University Applied Physics Laboratory

February 2021

Table of Contents

1. 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.

2. 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 many 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 policy rules in the system.

An event set must be created before it can be associated with a security policy rule.

2.1. 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.
type	String	List type	The structure type to be listed by the SPS (“**event_set**” to list all event sets).

2.2. Adding an Event Set

2.2.1. 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>”
     }
  }
```

2.2.2. 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.

2.3. Displaying Event Set Information

2.3.1. 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>”}
  }
```

2.3.2. 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.

2.4. Deleting an Event Set

2.4.1. 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>"}
  }
```

2.4.2. Example Usage

```
d {"event_set" :
     {"name" : "d_integrity"}
  }
```

This sample command deletes the d_integrity event set from the system.

2.5. 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 {"type" : "event_set"}
```

3. 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.

3.1. 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: a) **“reason_code”** b) **“new_value”** c) **“mask”**

3.1.1. Security Operation Events

The following security operation events are valid values for the **event_id** field.

a) **"source_for_sop**
b) **"sop_added_at_source”**
c) **"sop_misconfigured_at_source”**
d) **"verifier_for_sop”**
e) **“sop_misconfigured_at_verifier”**
f) **"sop_missing_at_verifier”**
g) **"sop_corrupted_at_verifier”**
h) **"sop_verified”**
i) **"acceptor_for_sop”**
j) **"sop_misconfigured_at_acceptor”**
k) **"sop_missing_at_acceptor”**
l) **"sop_corrupted_at_acceptor”**
m) **"sop_processed"**

3.1.2. Processing Actions

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

a) **“remove_sop”**
b) **“remove_sop_target”**
c) **"remove_all_target_sops”**
d) **"do_not_forward”**
e) **"report_reason_code”**

3.1.3. 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		X		X	X
SOP Missing at Acceptor		X		X	X
SOP Corrupted at Acceptor		X	X	X	X
SOP Processed					

3.2. Adding an Event to an Event Set

3.2.1. 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 label>” : <(opt.) parm value>}, … ,
                      {"id": “<processing action>”,
        “<(opt.) action parm label>” : <(opt.) parm value>}]
    }
```

```
  }
```

3.2.2. 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, request_storage, and report_reason_code actions for the sop_corrupted_at_acceptor event.

3.3. Deleting an Event from an Event Set

3.3.1. 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>”
      }
   }
```

3.3.2. 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.

4. 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.

4.1. 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>}
     }
}
```

4.1.1. Filter Criteria

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.

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:

a) Bundle Source
b) Bundle Destination
c) Security Source

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 `scid` 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 `req_scid` 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:

			a) “**sec_source**” or “**s**” b) “**sec_verifier**” or “**v**” c) “**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.
scid	Integer	Security Context ID	The ID of the security context to use when applying the security operation to the bundle.

4.1.2. 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: a) “**bib-integrity**” b) “**bcb-confidentiality**”
scid	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: a) “**key_name**” b) “**iv**” c) “**salt**” d) “**icv**” e) “**intsig**”

			f) “**bek**” g) “**bekicv**”

4.1.3. 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.

4.2. Adding Security Policy Rules

4.2.1. Commands

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

```
a {"policyrule" :
      {
         "desc"   : “<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>,
            "scid"     : <security context ID>
            },
         "spec" :
            {
             "svc"      : “<security service>”,
             “scid”     : <security context ID>,
             "sc_parms" : [{"id": <ID>, "value": <SC parm>}, … ,
                           {"id": <ID>, "value": <SC parm>}]
            },
        "es_ref"  : “<event set name>”
        }
```

```
    }
```

4.2.1.1.1. 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,
         "scid"     : "BIB-HMAC-SHA-256"
      },
     "spec":
      {
         "svc"      : "bib-integrity"
         "sc_parms" : [{"id":"key_name","value":"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.

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.

4.3. Deleting Security Policy Rules

4.3.1. 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.

4.3.2. 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.

4.4. Displaying Security Policy Rule Information

4.4.1. Command

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

```
i {"policyrule" : <security policy rule id>}
```

4.4.1.1. Example Usage

```
i {"policyrule" : 2}
```

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

4.5. Find Security Policy Rules

4.5.1. Command

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

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.

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`**.

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>”,
     "scid"    :  <security context ID>,
     "role"    :  “<security policy role>”
     }
  }
```

4.5.2. 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:

a) The security operation that will be added to the bundle by the security source node.
b) The target block type of the security operation.
c) The security context that will be used to add the required security service.

4.6. 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 {“type” : “policyrule”}
```