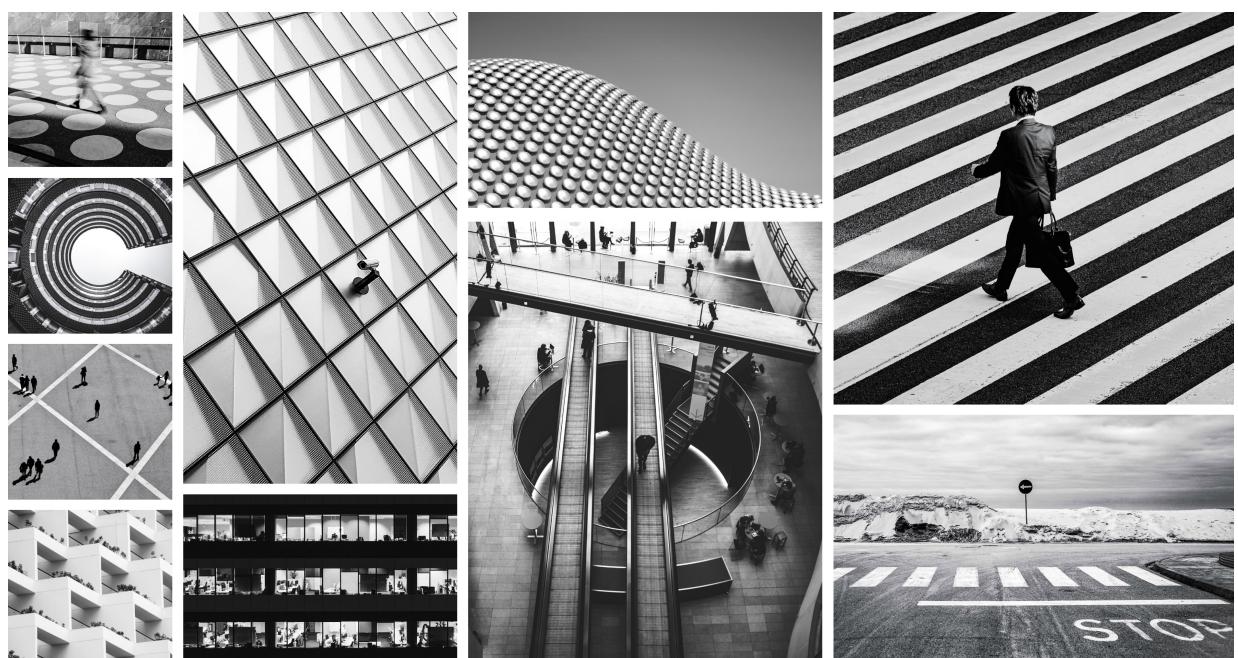




## REST APIs 25.2.10 and 25.2.20

---



The Illumio API is a RESTful API that uses JSON over HTTPS. JSON encodes all data transfer in both directions, so everything sent to and received from the API is encoded in JSON.

## REST API Quick Reference

- [25.3 REST API Reference](#)
- [25.2.20 REST API Reference](#)
- [25.2.10 REST API Reference](#)
- [25.1 REST API Reference](#)
- [24.5 REST API Reference](#)
- [24.2.20 REST API Reference and 24.2.10 REST API Reference](#)
- [23.5 REST API Reference](#)
- [23.2 REST API Reference](#)
- [22.5 REST API Reference](#)
- [22.2 REST API Reference](#)

## Open API Spec in JSON

- [25.4 Open API Spec in JSON](#)
  - [25.3 Open API Spec in JSON](#)
  - [25.2.20 Open API Spec in JSON](#)
  - [25.2.10 Open API Spec in JSON](#)
  - [25.1 Open API Spec in JSON](#)
  - [24.5 Open API Spec in JSON](#)
  - [24.2.20 Open API Spec in JSON and 24.2.10 Open API Spec in JSON](#)
  - [23.5 Open API Spec in JSON](#)
  - [23.2 Open API Spec in JSON](#)
  - [22.5 Open API Spec in JSON](#)
  - [22.2 Open API Spec in JSON](#)

## API Public Schemas (Zipped Files)

- [25.4 API Public Schemas](#)
  - [25.3 API Public Schemas](#)
  - [25.2.20 API Public Schemas](#)
  - [25.2.10 API Public Schemas](#)
  - [25.1 API Public Schemas](#)
  - [24.5 API Public Schemas](#)
  - [24.2.20 API Public Schemas and 24.2.10 API Public Schemas](#)
  - [23.5 API Public Schemas](#)
  - [23.2 API Public Schemas](#)
  - [22.5 API Public Schemas](#)
  - [22.2 API Public Schemas](#)

## Table of Contents

REST APIs 25.2.10 .....	6
New and Changed APIs in 25.2.10 .....	6
New APIs in 25.2.10 .....	6
<b><code>sec_policy_policy_check_get</code></b> .....	6
<b><code>vens_remote_action_put</code></b> .....	7
New Common Schemas .....	8
Changed APIs in 25.2.10 .....	15
Overview of the Illumio REST API .....	18
API Classification and Version .....	20
REST API URIs .....	20
REST API Limits .....	22
HTTP Requests and Responses .....	25
Request Calls Using Curl .....	28
Authentication and API User Permissions .....	29
Authentication .....	29
Required Permissions for API Users .....	34
Session Credentials .....	35
Showing Rule ID in Syslog .....	41
No Log Rules .....	43
API Keys .....	44
REST API Users .....	59
LDAP Authentication .....	64
Overview of Async GET Requests .....	72
Collection vs. Instance .....	72
Async GET Supported APIs .....	72
Async Job Operations .....	74
Overview of Async GET Requests .....	79
Security Policy Objects .....	82
Active vs. Draft .....	82
Security Principals .....	83
Labels .....	86
Label Groups .....	91
Services .....	96
Core Services Detection .....	102
Virtual Services .....	107
Virtual Servers .....	118
IP Lists .....	123
Non-Corporate Public IP Addresses .....	126
PCE Management .....	128
Product Version .....	128
Authentication Settings .....	129
Password Policy .....	130
Supercluster Leader .....	134
PCE Health .....	134
Node Availability .....	140
No Op Public Stable API .....	142
Events .....	143
Organization Settings .....	146
Organization Settings .....	150
Container Clusters .....	150
Access Restrictions and Trusted Proxy IPs .....	166
Rulesets and Rules .....	173
Rulesets .....	173

Rules .....	185
Rule Search .....	193
Rule Hit Count .....	200
Custom Iptables Rules .....	208
Rule-Based Label Mapping .....	215
Machine Authentication .....	220
Enforcement Boundaries .....	223
Provisioning .....	228
Policy Update Mode .....	228
Provisioning (Public Stable) .....	234
Provisioning (Public Experimental) .....	238
Virtual Server Filtering .....	248
RBAC for PCE Users .....	260
RBAC Overview .....	260
User Permissions .....	262
App Owner RBAC Role .....	264
RBAC User Operations .....	265
About RBAC Permissions .....	271
Authorization Security Principals .....	277
Visualization .....	281
Explorer .....	281
Reporting APIs .....	291
Ransomware Protection Dashboard APIs .....	298
VEN Statistics APIs .....	318
Vulnerabilities .....	321
Workloads APIs .....	333
Workload Operations .....	333
Workload Settings .....	344
Workload Interfaces .....	349
Workload Bulk Operations .....	353
Pairing Profiles and Pairing Keys .....	359
Filtering and Aggregating Traffic .....	366
VEN Operations .....	370
Events Administration and REST APIs .....	382
About this guide .....	382
Before reading this guide .....	382
Notational conventions in this guide .....	382
Events Framework .....	383
Events Lifecycle for Resources .....	384
Events Described .....	385
List of Event Types .....	385
Event Types, Syntax, and Record Format .....	395
View and Export Events .....	399
Examples of Events .....	401
Differences from Previous Releases .....	410
Events Monitoring Best Practices .....	410
Events Setup .....	413
Requirements for Events Framework .....	414
Events Settings .....	415
SIEM Integration for Events .....	418
Syslog Forwarding .....	419
Showing Rule ID in Syslog .....	421
Traffic Flow Summaries .....	421
Traffic Flow Types and Properties .....	422
Traffic Flow Summary Examples .....	424

Manage Traffic Flows Using REST API .....	428
Export Traffic Flow Summaries .....	433
Legal Notice .....	435

## REST APIs 25.2.10

The Illumio API is a RESTful API that uses JSON over HTTPS. JSON encodes all data transfer in both directions, so everything sent to and received from the API is encoded in JSON.

## New and Changed APIs in 25.2.10

This topic lists the new and updated REST APIs in 25.2.10.

### New APIs in 25.2.10

- [sec\\_policy\\_policy\\_check\\_get \[6\]](#)
- [vens\\_remote\\_action\\_put \[7\]](#)

#### **sec\_policy\_policy\_check\_get**

This API is used to get all `sec_rules`, `deny_rules`, and `override_deny_rules` based on parameters. It was created to extend the response of the original **allow** endpoint and preserve compatibility with existing tools.

The request format is as follows:

```
GET api/v2/orgs/:xorg_id/sec_policy/:pversion/policy_check?<params>;
```

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "sec_rules",
    "deny_rules",
    "override_deny_rules"
  ],
  "properties": {
    "sec_rules": {
      "description": "Array of allow rules",
      "type": "array",
      "items": {
        "allOf": [
          {
            "$ref": "../common/sec_rules_get.schema.json"
          },
          {
            "rule_set": {
              "$ref": "../common/rule_set.schema.json"
            }
          }
        ]
      }
    }
  }
}
```

```
        },
        "deny_rules": {
            "description": "Array of deny rules",
            "type": "array",
            "items": {
                "allOf": [
                    {
                        "$ref": ".../common/deny_rules_get.schema.json"
                    },
                    {
                        "rule_set": {
                            "$ref": ".../common/rule_set.schema.json"
                        }
                    }
                ]
            }
        },
        "override_deny_rules": {
            "description": "Array of override deny rules",
            "type": "array",
            "items": {
                "allOf": [
                    {
                        "$ref": ".../common/deny_rules_get.schema.json"
                    },
                    {
                        "rule_set": {
                            "$ref": ".../common/rule_set.schema.json"
                        }
                    }
                ]
            }
        }
    }
}
```

**vens remote action put**

The new schema **vens\_remote\_action\_put** is sent by a user to execute a remote action on a VEN. Users authorized to use this method are global administrators, global organization owners, and workload managers.

The required properties include:

- **action** which describes the remote action type
  - **vens** which describes an array of VENs to restart

```

    "vens"
],
"properties": {
  "action": {
    "description": "Remote action type",
    "type": "string",
    "enum": [
      "restart"
    ]
  },
  "vens": {
    "description": "An array of VENs to restart",
    "type": "array",
    "minItems": 1,
    "maxItems": 1,
    "items": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "href"
      ],
      "properties": {
        "href": {
          "description": "VEN URI",
          "type": "string"
        }
      }
    }
  }
}
}

```

## New Common Schemas

- **common deny\_rule\_actor:** The Enforcement Boundary Actor schema describes the actors as workloads and defines the exclusions.

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Enforcement boundary actor",
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "actors": {
        "description": "Rule actors are all workloads ('ams')",
        "type": "string",
        "enum": [ "ams" ]
      },
      "exclusion": {
        "description": "Boolean to specify whether or not the actor is an exclusion - only for labels and label groups",
        "type": "boolean",
      }
    }
  }
}

```

```
        "expose_to": [ "end_user_experimental" ],
        "default": false
    },
    "label": {
        "$ref": "href_object.schema.json"
    },
    "label_group": {
        "$ref": "href_object.schema.json"
    },
    "ip_list": {
        "$ref": "href_object.schema.json"
    },
    "workload": {
        "expose_to": [ "end_user_private_perm" ],
        "$ref": "href_object.schema.json"
    }
}
}
```

- **common deny\_rules\_get**: For deny\_rules, this gets the timestamps when the Enforcement Boundary was created, updated, and deleted. It also defines the users who originally created, updated, and deleted the boundary.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Enforcement boundary",
  "type": "object",
  "required": ["href", "providers", "destinations", "ingress_services"],
  "expose_to": ["end_user_private_perm"],
  "_comment": "Don't set additionalProperties:false here as it collides with usage in allOf, set that in the schema that references this one instead.",
  "properties": {
    "created_at": {
      "description": "Timestamp when this Enforcement Boundary was first created",
      "type": "string",
      "format": "date-time"
    },
    "updated_at": {
      "description": "Timestamp when this Enforcement Boundary was last updated",
      "type": "string",
      "format": "date-time"
    },
    "deleted_at": {
      "description": "Timestamp when this Enforcement Boundary was deleted",
      "type": ["string", "null"],
      "format": "date-time"
    },
    "created_by": {
      "type": ["object", "null"],
      "required": ["href"],
      "properties": {
        "href": {
          "type": "string"
        }
      }
    }
  }
}
```

```

        "description": "User who originally created this Enforcement
Boundary",
        "type": "string"
    }
},
"updated_by": {
    "type": ["object", "null"],
    "required": [ "href" ],
    "properties": {
        "href": {
            "description": "User who last updated this Enforcement
Boundary",
            "type": "string"
        }
    }
},
"deleted_by": {
    "type": ["object", "null"],
    "required": [ "href" ],
    "properties": {
        "href": {
            "description": "User who deleted this Enforcement Boundary",
            "type": "string"
        }
    }
},
"update_type": {
    "$ref": "../common/sec_policy_update_type.schema.json"
},
"href": {
    "description": "The job URI.",
    "type": "string"
},
"providers": { "$ref": "deny_rule_actor.schema.json" },
"destinations": { "$ref": "deny_rule_actor.schema.json" },
"ingress_services": {
    "$ref": "sec_rule_ingress_services.schema.json"
},
"egress_services": {
    "$ref": "sec_rule_egress_services.schema.json"
},
"caps": {
    "$ref": "../common/entity_caps.schema.json"
},
"enabled": {
    "description": "Enabled flag",
    "type": "boolean"
},
"description": {
    "description": "Description",
    "type": [ "string", "null" ]
},
"network_type": {
    "$ref": "../common/rule_network_type.schema.json"
}

```

```

        },
        "override": {
            "description": "When true, the deny rule will override and take
precedence over other user defined allow rules.",
            "default": false,
            "type": "boolean"
        },
        "unscoped_destinations": {
            "description": "Set the scope for rule destinations to All",
            "type": "boolean"
        }
    }
}

```

- **common\_rule\_set**: Parent Rule Set of a Rule.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Enforcement boundary",
    "type": "object",
    "required": ["href", "providers", "destinations", "ingress_services"],
    "expose_to": ["end_user_private_perm"],
    "_comment": "Don't set additionalProperties:false here as it collides
with usage in allOf, set that in the schema that references this one
instead.",
    "properties": {
        "created_at": {
            "description": "Timestamp when this Enforcement Boundary was first
created",
            "type": "string",
            "format": "date-time"
        },
        "updated_at": {
            "description": "Timestamp when this Enforcement Boundary was last
updated",
            "type": "string",
            "format": "date-time"
        },
        "deleted_at": {
            "description": "Timestamp when this Enforcement Boundary was
deleted",
            "type": ["string", "null"],
            "format": "date-time"
        },
        "created_by": {
            "type": ["object", "null"],
            "required": ["href"],
            "properties": {
                "href": {
                    "description": "User who originally created this Enforcement
Boundary",
                    "type": "string"
                }
            }
        },
        "updated_by": {
            "type": ["object", "null"]
        }
    }
}
```

```

"required": [ "href" ],
"properties": {
    "href": {
        "description": "User who last updated this Enforcement Boundary",
        "type": "string"
    }
},
"deleted_by": {
    "type": [ "object", "null" ],
    "required": [ "href" ],
    "properties": {
        "href": {
            "description": "User who deleted this Enforcement Boundary",
            "type": "string"
        }
    }
},
"update_type": {
    "$ref": "../common/sec_policy_update_type.schema.json"
},
"href": {
    "description": "The job URI.",
    "type": "string"
},
"providers": { "$ref": "deny_rule_actor.schema.json" },
"destinations": { "$ref": "deny_rule_actor.schema.json" },
"ingress_services": {
    "$ref": "sec_rule_ingress_services.schema.json"
},
"egress_services": {
    "$ref": "sec_rule_egress_services.schema.json"
},
"caps": {
    "$ref": "../common/entity_caps.schema.json"
},
"enabled": {
    "description": "Enabled flag",
    "type": "boolean"
},
"description": {
    "description": "Description",
    "type": [ "string", "null" ]
},
"network_type": {
    "$ref": "../common/rule_network_type.schema.json"
},
"override": {
    "description": "When true, the deny rule will override and take precedence over other user defined allow rules.",
    "default": false,
    "type": "boolean"
},
"unscoped_destinations": {

```

```
"description": "Set the scope for rule destinations to All",  
"type": "boolean"
```

}

- **common\_sec\_rule\_egress\_services**: Array of objects.

```
{  
  "$schema": "http://json-schema.org/draft-04/schema#",  
  "description": "Array of objects",  
  "type": "array",  
  "items": {  
    "type": "object",  
    "$ref": "../common/href_object.schema.json"  
  }  
}
```

- **common sec\_rules\_get**: For sec\_rules, this gets the timestamps when the Enforcement Boundary was created, updated, and deleted. It also defines the users who originally created, updated, and deleted the boundary.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Enforcement boundary",
  "type": "object",
  "required": ["href", "providers", "destinations", "ingress_services"],
  "expose_to": ["end_user_private_perm"],
  "_comment": "Don't set additionalProperties:false here as it collides with usage in allOf, set that in the schema that references this one instead.",
  "properties": {
    "created_at": {
      "description": "Timestamp when record was first created",
      "type": "string",
      "format": "date-time"
    },
    "updated_at": {
      "description": "Timestamp when record was last updated",
      "type": "string",
      "format": "date-time"
    },
    "deleted_at": {
      "description": "Timestamp when record was deleted",
      "type": ["string", "null"],
      "format": "date-time"
    },
    "created_by": {
      "type": "object",
      "properties": {
        "username": {
          "description": "The username which created this record",
          "type": "string"
        }
      }
    },
    "updated_by": {
      "type": "object",
      "properties": {
        "username": {
          "description": "The username which updated this record",
          "type": "string"
        }
      }
    }
  }
}
```

```

"properties": {
    "username": {
        "description": "The username which last updated this record",
        "type": "string"
    }
},
"deleted_by": {
    "type": ["object", "null"],
    "properties": {
        "username": {
            "description": "The username which deleted this record",
            "type": "string"
        }
    }
},
"update_type": {
    "description": "Type of update",
    "oneOf": [
        {
            "type": "null"
        },
        {
            "type": "string",
            "enum": [ "create", "update", "delete" ]
        }
    ]
},
"update_label": {
    "description": "Type of update",
    "oneOf": [
        {
            "type": "null"
        },
        {
            "type": "string",
            "enum": [ "create", "update", "delete" ]
        }
    ]
},
"href": {
    "description": "URI of object",
    "type": "string"
},
"enabled": {
    "description": "Enabled flag",
    "type": "boolean"
},
"description": {
    "description": "Description",
    "type": [ "string", "null" ]
},
"ingress_services": { "$ref": "sec_rule_ingress_services.schema.json" },
"egress_services": { "$ref": "sec_rule_egress_services.schema.json" },

```

```

    "resolve_labels_as": { "$ref":
"sec_rule_resolve_labels_as.schema.json" },
    "sec_connect": {
        "description": "Whether a secure connection is established",
        "type": "boolean"
    },
    "stateless": {
        "expose_to": [ "end_user_experimental" ],
        "description": "Whether packet filtering is stateless for the rule",
        "type": "boolean"
    },
    "machine_auth": {
        "expose_to": [ "end_user_experimental" ],
        "description": "Whether machine authentication is enabled",
        "type": "boolean"
    },
    "providers": { "$ref":
"sec_policy_rule_sets_sec_rules_providers_get.schema.json" },
    "destinations": { "$ref":
"sec_policy_rule_sets_sec_rules_destinations_get.schema.json" },
    "consuming_security_principals": { "$ref":
"consuming_security_principals_get.schema.json" },
"consuming_security_principals": { "$ref":
"consuming_security_principals_get.schema.json" },
    "unscoped_destinations": {
        "description": "Set the scope for rule destinations to All",
        "type": "boolean"
    },
    "use_workload_subnets": {
        "$ref": "sec_rule_use_workload_subnets.schema.json"
    },
    "rule_set": { "$ref": ".../common/rule_set.schema.json" },
    "log_flow": {
        "description": "If false, the VEN will not log any traffic that
matches this flow.",
        "type": "boolean",
        "expose_to": [ "end_user_private_transitional" ]
    },
    "network_type": { "$ref": ".../common/rule_network_type.schema.json" }
}
}
}

```

## Changed APIs in 25.2.10

The following public APIs have changed in 25.2.10.

### Simplified schema by using a reference for

- **sec\_policy\_allow\_get**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "array",
    "items": {
        "allOf": [
            {

```

```

        "$ref": "sec_policy_rule_sets_sec_rules_get.schema.json"
    }
]
}
}
}

```

The initial schema, which contained the required objects "enabled", "providers", "consumers", and "ub\_service", has been simplified to use a reference to the `sec_policy_rule_sets_sec_rules_get` schema.

### **Additional properties `container_workload` and `kubernetes_workload` were added to**

- `sec_policy_rule_search_consumers`
- `sec_policy_rule_search_providers`

```

{
  "items": {
    "properties": {
      "container_workload_added": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "href"
        ],
        "properties": {
          "href": {
            "description": "Container workload URI",
            "type": "string"
          }
        }
      },
      "kubernetes_workload_added": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "href"
        ],
        "properties": {
          "href": {
            "description": "Kubernetes workload URI",
            "type": "string"
          }
        }
      }
    }
  }
}

```

In the initial API `sec_policy_rule_search_consumers`, in addition to properties "actors", "label", "label\_group", "workload", "virtual\_service", and "ip\_list", two new ones have been added:

### **Additional properties:**

- `kubernetes_workload`: href of `kubernetes_workload` to which the searched rule should apply to

- container\_workload: href of container\_workload to which the searched rule should apply to

### **Property rule\_types added to**

- **sec\_policy\_rule\_search\_post**

```
{
  "properties": {
    "rule_types": {
      "description": "List of rule types",
      "type": "array",
      "items": {
        "type": "string",
        "enum": [
          "sec_rules",
          "deny_rules",
          "override_deny_rules",
          "ip_tables_rules"
        ]
      }
    }
  }
}
```

A new property `rule_types` was added, which lists the rule types: `sec_rules`, `deny_rules`, `override_deny_rules`, and `ip_tables_rules`. Request `rule_types` that should be searched for.

### **Simplified schema**

- **sec\_policy\_rule\_search\_post\_response**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "sec_rules": {
      "description": "Array of allow rules",
      "type": "array",
      "items": {
        "allOf": [
          {
            "$ref": "../common/sec_rules_get.schema.json"
          },
          {
            "rule_set": {
              "$ref": "../common/rule_set.schema.json"
            }
          }
        ]
      }
    },
    "deny_rules": {
      "description": "Array of deny rules",
      "type": "array",
      "items": {
        "allOf": [
          {
            "$ref": "../common/deny_rules_get.schema.json"
          }
        ]
      }
    }
  }
}
```

```

"type": "array",
"items": {
  "allOf": [
    {
      "$ref": "../common/deny_rules_get.schema.json"
    },
    {
      "rule_set": {
        "$ref": "../common/rule_set.schema.json"
      }
    }
  ]
},
"override_deny_rules": {
  "description": "Array of override deny rules",
  "type": "array",
  "items": {
    "allOf": [
      {
        "$ref": "../common/deny_rules_get.schema.json"
      },
      {
        "rule_set": {
          "$ref": "../common/rule_set.schema.json"
        }
      }
    ]
  }
},
"ip_tables_rules": {
  "type": "array",
  "items": {
    "allOf": [
      {
        "$ref": "../common/ip_tables_rules_get.schema.json"
      },
      {
        "rule_set": {
          "$ref": "../common/rule_set.schema.json"
        }
      }
    ]
  }
}
}

```

This schema was simplified using `allOf` and a reference to the two existing schemas.

## Overview of the Illumio REST API

The Illumio API is a RESTful API that uses JSON over HTTPS. JSON encodes all data transfer in both directions, so everything sent to and received from the API is encoded in JSON.

To work with the Illumio API, you need to be authorized by an Illumio administrator and have the appropriate authentication credentials.

## REST API Quick Reference

- [25.3 REST API Reference](#)
- [25.2.20 REST API Reference](#)
- [25.2.10 REST API Reference](#)
- [25.1 REST API Reference](#)
- [24.5 REST API Reference](#)
- [24.2.20 REST API Reference and 24.2.10 REST API Reference](#)
- [23.5 REST API Reference](#)
- [23.2 REST API Reference](#)
- [22.5 REST API Reference](#)
- [22.2 REST API Reference](#)

## Open API Spec in JSON

- [25.4 Open API Spec in JSON](#)
  - [25.3 Open API Spec in JSON](#)
  - [25.2.20 Open API Spec in JSON](#)
  - [25.2.10 Open API Spec in JSON](#)
  - [25.1 Open API Spec in JSON](#)
  - [24.5 Open API Spec in JSON](#)
  - [24.2.20 Open API Spec in JSON and 24.2.10 Open API Spec in JSON](#)
  - [23.5 Open API Spec in JSON](#)
  - [23.2 Open API Spec in JSON](#)
  - [22.5 Open API Spec in JSON](#)
  - [22.2 Open API Spec in JSON](#)

## API Public Schemas (Zipped Files)

- [25.4 API Public Schemas](#)
  - [25.3 API Public Schemas](#)
  - [25.2.20 API Public Schemas](#)
  - [25.2.10 API Public Schemas](#)
  - [25.1 API Public Schemas](#)
  - [24.5 API Public Schemas](#)
  - [24.2.20 API Public Schemas and 24.2.10 API Public Schemas](#)
  - [23.5 API Public Schemas](#)
  - [23.2 API Public Schemas](#)
  - [22.5 API Public Schemas](#)
  - [22.2 API Public Schemas](#)

## API Classification and Version

This section explains the distinction among the Illumio Public Stable, Public Experimental, and private APIs.

### Public Stable APIs

The Public Stable APIs are generally available to all Illumio customers, are documented, and are stable. “Stable” means that Illumio will not introduce any further breaking changes to the API. If a breaking change is required, another version of the API will be introduced, and the previous version will continue to be supported for a minimum of six (6) months.

### Public Experimental APIs

The Public Experimental APIs are generally available to all Illumio customers and are documented, but they are subject to change from release to release. If you use experimental APIs, such as in scripts, be aware that some of them might change. Some of these APIs might be promoted to Public Stable or made no longer available at a future date.

This API guide uses orange headings inside these files to help distinguish which APIs are “Public Experimental”.

### Private APIs

In addition to the Public Stable or Public Experimental APIs, the Illumio Core includes additional Private APIs used by the PCE web console. The private Illumio APIs are not exposed to end-users, are not documented, and are not supported for use.

## Illumio REST API Versions

Illumio REST APIs follow the release versions of other Illumio components, such as the PCE and VEN.

## Illumio REST API Schema Files

Illumio REST API schema files follow the standard JSON schema form described at <http://json-schema.org/>. The file name convention is the Illumio REST API URL name with underscore rather than slashes. For example, for the login API, the payload schema file is named `user_login_get.schema.json`.

## REST API URIs

This section describes the URI syntax used with this API, which can be different depending on the REST call you are making and the types of Illumio resources on which you are operating.

## API Version and Org HREF

The API version and organization HREF are two variables used in every call made to this API.

The current version of the Illumio Core REST API is version 2 (v2), which is represented in method URIs by the `[api_version]` variable.

You can determine the organization HREF for the PCE when you use the login API to authenticate with the PCE and obtain a session token. In method URIs, this value is represented by the [org\_href] variable.

In response to using the login API, the organization HREF is listed as shown, but depends on the version of the API you are using:

```
"orgs": [
  {
    "org_id": 2,
    "org_href": "/orgs/2",
```

Note that both [api\_version] and [org\_href] begin with a forward slash:

- [api\_version] - /api/v2
- [org\_href] - /orgs/2

For example, to get a collection of labels that exist inside an organization, construct the URI as follows, with the API version and the organization HREF shown in blue font:

```
GET [api_version][org_href]/labels
```

To get all of the API keys created by a specific user, construct the URI as follows, with the HREF path to the user shown in a blue font:

```
GET api/v2/orgs/1/api_keys
```

## Port Number

The port number used in the code examples is 8443, which is the default. However, since the port number might be different depending on the implementation, ask your Illumio system administrator which port number to use when making calls to the Illumio Core REST API.

## GET Collections URI Syntax

The base URI for Illumio REST API endpoint for GET collections:

```
GET http://[pce_hostname]:[port][api_version][org_href]/[api_endpoint]
```



### IMPORTANT

When making API calls, the [pce\_hostname] or [pce\_hostname]:[port] should not end with a forward slash ('/'). This is because [api\_version] begins with a forward slash.

For example, the URI for getting a collection of workloads uses this syntax:

```
GET https://pce.my-company.com:8443/api/v2/orgs/1/workloads
```

In the rulesets API, you also have the ability to get all of the rules ("sec\_rules") contained in a ruleset. The URI syntax for this operation is as follows:

```
GET http://[pce_hostname]:[port][api_version][object_href][api_endpoint]
```

For example:

```
GET [api_version][ruleset_href]/sec_rules
```

## **Non-GET Collections URI Syntax**

For the non-GET methods of PUT, POST, and DELETE, the object HREF is listed as the endpoint, as shown here:

```
PUT [api_version][object_href]
```

The relative path of the [api\_version] ("api/v2/") indicates that version 2 of the API is in use.

In the URI above, [org\_href] is not added because it is included in the [object\_href] string. For example, this is the [object\_href] for a workload:

```
/orgs/2/workloads/3e3e17ce-XXXX-42b4-XXXX-1d4d3328b342
```

Another case is performing PUT, POST, or DELETE operations on the rules contained in a ruleset. The URI syntax is the same as a GET operation.

## **Security Policy Items and “:pversion”**

The URI for security policy items is as follows:

```
[pce_host][port][api_version][org_href]/sec_policy/draft/[api_endpoint]
```

This API operates on provisionable objects, which exist in either a **draft** (not provisioned) state or an **active** (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called **:pversion**, which can be set to either **draft** or **active**.

Depending on the method, the API follows these rules:

- For GET operations — **:pversion** can be draft, active, or the ID of the security policy.
- For POST, PUT, DELETE — **:pversion** can be draft (you cannot operate on active items) or the ID if the security policy.

## **REST API Limits**

When making API calls, make sure that you take into account the allowed maximum number of calls per minute, returned objects, or total item count.

**IMPORTANT**

Any tooling that parses the HTTP headers should be changed to allow case-insensitive header name matching in order to retain compatibility with future PCE releases. Refer to RFC 7230, section 3.2, "Header Fields," which states that field names should be case insensitive.

**API Rate Limits and DOS Protection**

The Illumio REST API is rate-limited and allows only a maximum of 500 requests per minute per user session or API key. The rate is set to maintain the PCE performance and service availability and to prevent malicious attackers from attempting to disrupt a service (for example, DoS attacks). If the set rate limit is reached, the call returns an HTTP error 429 Too many requests.

**Limits for Bulk Operations**

In addition to the rate limits described above that are counted for all requests, the unpair workloads and delete traffic flows APIs have a rate limit of 10 calls per minute. There are also two limits on the number of resources that can be operated per call.

API Call and Endpoint	Request Rate Limit	Item Limit	Exposure
<b>Unpair Workloads</b>  PUT [api_version][org_href]/workloads/unpair	10 per minute	1000 workloads per request	Public Stable

**NOTE**

Illumio reserves the right to adjust the rate limit on the Illumio Secure Cloud for given endpoints at any time to ensure all clients receive a high-quality service.

**Ruleset Rules Display Limit**

The PCE web console supports up to 500 rules per ruleset. Rulesets with more than 500 rules cannot be fully displayed in the PCE web console.

**GET Collection Request Limits**

By default, when you perform a synchronous GET request with this API, the maximum number of objects returned is 500.

Some GET APIs provide query parameters to help restrict the number of results, depending on the API. For example, the workloads API provides multiple query parameters for GET collections, such as `label`, `ip_address`, `policy_health`, and more.

If you want to get more than 500 objects from a GET collection, use which runs the request as an offline job. Job results can be downloaded after the job finishes.

## Checking Total Item Count

To find out how many items exist for a given resource, such as whether there are more than 500 workloads in the PCE, first check the number of items using the `max_results` query parameter on a GET collection and then view the header of the response for the total item count for the resource.

If the total item count is less than 500, you can perform a regular GET collection for the results. If the total item count is more than 500, use Asynchronous GET Collections.

For example, make the following GET call on a collection of workloads with the `max_results` query parameter set equal to 1, then check the header to see how many workloads exist in your organization.



### NOTE

When using multiple query parameters, enclose the URI, endpoint, and `query_params` in single or double quotes.

```
GET 'https://pce.mycompany.com:8443/api/v2/orgs/7/workloads?
max_results=1&managed=true'
```

You can check the HTTP response header for the 'X-Total-Count' field, which indicates the total number of workloads. In this example, the total count shows 71 (highlighted in blue font), so a regular GET collection is appropriate. An asynchronous GET collection would be used if the value were more than 500.

```
Cache-Control no-store
Content-Encoding gzip
Content-Type application/json
Date Wed, 07 Sep 2016 14:01:00 GMT
ETag W/"025cc8bfcXXXXXXXXXXXX7900081e7c6cb"
Status 200 OK
Transfer-Encoding chunked
Vary Accept-Encoding
X-Matched-Count 71
X-Request-Id d43a8ce9-XXXX-4453-XXXX-dde79XXX0fa8
X-Total-Count 71
```

## Character Limits on Resource Names

The PCE has a 255-character limit for each name string when naming resources. This JSON property is listed as `name` in the API.

For example, this 255-character limit applies to naming workloads, labels, IP lists, and services.

However, the PCE does not have a character limit for the description field, which typically follows a resource's name.

## HTTP Requests and Responses

This section explains how to formulate HTTP requests and read HTTP responses.

### HTTP Request Headers

Set an `Accept: application/json` header on all `GET` operations (optional for `DELETE` operations):

```
-H 'Accept: application/json'
```

Set a `Content-Type: application/json` header on `PUT` and `POST` operations:

```
-H 'Content-Type: application/json'
```

### HTTP Request Body

Most of the parameters and data accompanying requests are contained in the body of the HTTP request. The Illumio REST API accepts JSON in the HTTP request body. No other data format is currently supported.

### PUT Operations

Illumio REST API `PUT` operations modify a subset of attribute-value pairs for a specified resource. The attributes that are not specified in the `PUT` operation are left unmodified.

For example, to update a user's phone number (using the `Users` API) without modifying the user's address, call `PUT` with a request that only modifies the phone number, and only the phone number is changed.

### Response Header Request-ID

The Illumio REST API provides a useful troubleshooting feature that returns a unique `Request-ID` in the HTTP response header on calls made with this API.

You can provide the `Request-ID` when opening Illumio support tickets, which are designed specifically for operations that produce errors. The `Request-ID` helps Illumio support to troubleshoot specific operations and errors.

If you are using `curl` to make REST API calls to the PCE, you can specify the `curl -D` flag plus a file name to write the response header to a file.

The following example shows a `curl` command to get a collection of workloads that uses the `-D` flag to write the response header to a file named `temp_header`.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/workloads -H "Accept: application/json" -u $KEY:$TOKEN -D temp_header
```

The file contains the response header of the call (highlighted in blue bold font):

```

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 09 Dec 2015 16:58:00 GMT
Content-Type: application/json
Content-Length: 2193032
Connection: keep-alive
Vary: Accept-Encoding
Vary: Accept-Encoding
Status: 200 OK
X-Total-Count: 1406
X-Matched-Count: 1406
ETag: "523d67cbd57b18d0e97bc8e7555142eb"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id:
9722c8b5-94dc-4a50-853a-8e8f22266528

Cache-Control: no-store
Pragma: no-cache

```

## Response Types

The HTTP response includes:

- An HTTP status code
- A response body that contains data in JSON format:
  - Your requested data, if successful
  - An error code and message if there is an error

## HTTP Status Codes — Success

The following table lists all expected success codes returned when you use the Illumio REST API:

HTTP Code	Description
200 OK	Successful operation where JSON body is returned.
201 Created	Successful POST operation where an object was created.
204 No Content	Operation succeeded and nothing was returned.

## HTTP Status Codes — Failure

All Illumio REST API methods (GET, POST, PUT, and DELETE) might fail with an error in the 400 range. Error code 400 usually means that the resource is unavailable (such as trying to update a previously deleted label) or that the URL contains a mistake (such as specifying /labels instead of /labels).

Other errors that might occur:

HTTP Code	Description
<b>400 Bad Request</b>	Something in the curl request was not correct, for example "curl -X -H GET" instead of "curl -H -X GET"
<b>401 Authentication failure or HTTP/1.1 401 Unauthorized</b>	For example, the user attempted to make an API call but forgot to log in, username or password were incorrect or missing, or a missing space before "-u"
<b>403 Authorization failure</b>	For example, the user is not authorized to make the call.
<b>HTTP/1.1 403 Forbidden</b>	For example, using the incorrect HTTP method (like using GET instead of POST), the incorrect org_id parameter was used
<b>404 Invalid URL</b>	
<b>HTTP/1.1 404 Not Found</b>	For example, an incorrect API version number /api/v191/, missing or incorrect org_id. /orgs/{org_id}/, wrong URL, or a misspelled endpoint.
<b>404 Page not found</b>	For example, the wrong org_id in the URI or missing blank space before an option dash, like before -H 'Accept: application/json'
<b>405 Method not allowed</b>	For example, if you perform a POST on a resource that only allows PUT.
<b>406 Invalid payload</b>	The JSON request payload was constructed improperly.

## Other Failure Codes

-bash: -H: command not found HTTP/1.1 401 Unauthorized

- This can occur if more than one query parameter is used and the URI (including the query parameters) is not enclosed in single or double quotes.

Example:

```
'https://pce.my-company.com:8443/api/v2/orgs/2//workloads?managed=true&max_results=1'
```

### curl: (3) Illegal port number

- For example, a blank space is missing between -u uname:'pswd' and the next option, -H 'Accept: application/json'.

parse error: Invalid numeric literal at line 1, column 9

- It can be caused by an incorrect curl command, such as including a path parameter that isn't allowed, like using orgs/org\_id for an endpoint that doesn't use it. This is also a known JSON query bug caused by using -i in a curl command that uses json-query. To see the headers returned from the curl command, remove json-query from the curl command and use -i, for example "curl -i -X GET ..."

curl: (23) Failed writing body

- Can be caused by calling an endpoint that doesn't exist.

The property '#' of type null did not match the following type: object in xxxxxxxx.schema.json

- Can be caused by a missing or incomplete request body.  
[{"token":"input\_validation\_error","message":"Input validation failed. Details: {The property '#/' of type NilClass did not match the following type: object in schema xxxxx.schema.json}"}]
- Is the wrong -H value being used? For example, is -H 'Accept: application/json' being used for a PUT or a POST instead of -H 'Content-Type: application/json'?

## Request Calls Using Curl

This section explains how to use curl commands to work with Illumio APIs by defining some standard options and constants.

### Curl Overview

Curl is a common command-line data transfer tool for making API calls and is especially useful in scripts written for automated tasks.

The syntax for using curl with the API for logging a user into the PCE is as follows:

```
curl -i -X <HTTP method> <uri_of_api> <header> -u $KEY:$TOKEN -Options
```

The syntax for using curl with the API for PUT operations using an API key for authentication is as follows:

```
curl -i -X PUT <URI of API> -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{ "json_property": "property_value", "json_property": "property_value" }'
```

For example:

```
curl -i -X PUT https://scp.illum.io:8443/api/v2/users/11/local_profile/password -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{ "current_password": "NotMyReal_Old*96Password", "new_password": "NotMy*76New!pswd" }'
```

### Curl-specific Options

For the curl examples provided in this API documentation, a few standard curl options are defined as follows.

The user and password to use for server authentication:

```
-u/--user <user:password>
```

For brevity, code examples typically use constants for -u username:'password' arguments. \$TOKEN represents an authentication token (a string enclosed by single quotes to prevent it from unintentionally expanding):

```
-u $KEY:$TOKEN
```

(HTTP) Header to use when getting a web page:

`-H/--header <header>`

(HTTP) Specify an HTTP method to use when communicating with the HTTP server:

`-X/--request <command>`

Example:

`-X POST`

(HTTP) Send the specified data in a POST request to the HTTP server in a way that emulates a user filling in an HTML form and clicking **Submit**:

`-d/--data <data>`

### Example API Call Using CURL

To get all of the API keys of a specific user using the user's session credentials:

```
curl -i -X GET https://scp.illumio.io:8443/api/v2/users/11/api_keys -H  
"Accept: application/json" -u $KEY:$TOKEN
```

### Using Curl with json-query

When using json-query to format the output of curl commands, be aware that due to a json-query bug, this does not work with the curl `-i` option, which displays response headers. When you use the curl `-i` option, such as to see the total number of workloads when using `GET workloads`, you might get various error messages like `curl: (3) illegal port number`. To work around this issue, remove the `-i` option and retry the curl command.

## Authentication and API User Permissions

To use the REST APIs, you must be an authorized Illumio user with credentials to log into the PCE.

You get authorized to perform a specific job according to the privileges granted to you based on the role-based access control (RBAC) and implemented by the Illumio administrator.

The PCE has two types of credentials that you can use to authenticate with it and make REST API calls:

- API keys, which provide a persistent means of authenticating
- Session credentials, which provide a temporary means of authenticating

## Authentication

Before you can use the Illumio REST API to access the PCE, you need to use the Login Users API to authenticate with the Illumio Login Service and obtain an authentication token.

## Authenticate to the Login Service

Before using the Illumio REST API to access the PCE, you need to use the Login Users API to authenticate with the Illumio Login Service and obtain an authentication token. This authentication token expires in 30 seconds.

The URL for the Illumio Login Service for Illumio Core Cloud users is:

- Login Server: `https://login.illum.io:443`
- PCE: `scp1.illum.io`

For SaaS customers, the PCE URL can be different based on their SaaS PCE:

- SCP1 & SCP2 (US)
- SCP3 UK only
- SCP4 APAC
- SCP5 (EMEA)

If you have deployed the PCE as software, then the hostname for the PCE is the value you defined for the '`pce_fqdn`' parameter in the `runtime_env.yml` file.

Once obtained, you can pass the authentication token to the PCE you want to access using the Login API. Once you have authenticated with the PCE and received a session token, you can make additional API calls or create an API Key for persistent access to the PCE's API.

URI to Authenticate with the Login Service

```
POST [api_version]/login_users/authenticate
```

### Create an Authentication Token for the Login Service

To create an authentication token and authenticate with the Login Service, you need to specify the Fully Qualified Domain Name (FQDN) of the PCE you want to access in the call.

Parameter	Description	Type	Re-required
<code>pce_fqdn</code>	<p>Fully Qualified Domain Name (FQDN) of the PCE</p> <p>If you are using Illumio Core Cloud, then the FQDN for the PCE is <code>scp1.illum.io</code>.</p> <p>If you have deployed the PCE virtual appliance in your network, use the FQDN specified during installation.</p>	String	Yes

### Curl Commands for Authentication

When you received your invitation, you created your PCE account using an email and password. Use these credentials to make a call and authenticate now.

If you haven't received an invitation, contact your Illumio administrator.

Example (local users only, use SAML ID for remote users):

- `joe_user@example.com` (username)
- `password` (password)

You also need the FQDN of the Login Server plus the FQDN of the PCE host you want to access:

- The Login Server FQDN for Illumio Core Cloud users is `https://login.illum.io:443`
- The PCE FQDN is `scpl.illum.io`



### **NOTE**

The authorization token returned (`auth_token`) expires after 30 seconds of inactivity. Be ready to call `GET users/login` to create session credentials immediately after making the call to `login_users/authenticate`.

## **Optional Features**

This API was introduced to help avoid issues with misconfigured DNS, which can cause problems with VEN connectivity. Likewise, misconfiguring DHCP can cause problems with IP addresses.

### **Invoke the Optional Features API**

You need a key to invoke `/optional_features` API to enable `editable_dns_client_rule` or `editable_dhcp_client_rule`. Such a key involves a portion that is tightly controlled so that it cannot be randomly generated.

Once the key is generated, it cannot be used in more than one place, meaning an API call provided to customer #1 cannot be replayed at customer #2, who must request their key.

An example of the generated key:

```
secret =
    '...'      # value embedded in code

data = Base64.strict_encode64({
    'pce_fqdn'    => Illumio::RuntimeEnvironment.pce_fqdn,
    'org_id'      => xorg_id,
    'optional_feature'  =>
    'editable_dns_client_rule' ,
    'not_valid_after'   => Time.now.utc.iso8601
})

key = data + OpenSSL::HMAC.hexdigest( 'SHA256' , secret, data)
```

### **Setting Optional Features**

Analytics opt-out

The property `configurable_label_dimension` was added so that the UI users can determine if an organization has enabled user analytics.

Analytics is opt-in by default. If it has been disabled, the UI does not track analytics for that organization.

To set or clear the optional analytics feature, use:

```
{
  name: "ui_analytics", enabled: false|true
}
```

Illumination Classic opt-out

The property illumination\_classic is added to enable or disable the use of that feature.

To set or clear the optional Illumination Classic feature, use:

```
{
  name: "illumination_classic", enabled: false|true
}
```

## **Label-Based Network Detection**

The APIs

- POST /api/v2/orgs/{org\_id}/networks
- PUT /api/v2/orgs/{org\_id}/networks/:network\_id

require that one of the following optional features is enabled :

- label\_based\_network\_detection
- cidr\_network\_detection\_enabled

In addition, both APIs are implementing input validation on payload content:

- If the CIDRs field is provided, the optional feature cidr\_network\_detection\_enabled must be set.
- If the scopes field is provided, the optional feature label\_based\_network\_detection must be enabled.

The example response for the API optional\_features\_put with the label\_based\_network\_detection enabled:

```
"illumination_classic",
  "ransomware_readiness_dashboard",
  "per_rule_flow_log_setting",
  "lightning_default",
  "label_based_network_detection"
]
},
"enabled": {
```

## **labels\_editing\_warning\_for\_enforcement\_mode**

In releases 23.2.10 and 23.4, for the required property name a new optional feature flag for label editing was added: `labels_editing_warning_for_enforcement_mode`.

To enable or disable this flag, use the following CURL command:

```
curl -u ${your_api_key}: ${your_api_secret} -H
"Content-Type: application/json" -X PUT -d
'[{ "name": "labels_editing_warning_for_enforcement_mode", "enabled": true}]'
https:// ${your_pce_server}:8443/api/v2/orgs/${your_ord_id}/optional_features
```

## **windows\_outbound\_process\_enforcement**

In release 23.5, an optional feature flag for the Windows outbound process was added: `windows_outbound_process_enforcement`.

This feature flag can be enabled or disabled using the following CURL command:

```
curl -u ${your_api_key}: ${your_api_secret} -H
"Content-Type: application/json" -X PUT -d
'[{ "name": "windows_outbound_process_enforcement", "enabled": true}]'
https:// ${your_pce_server}:8443/api/v2/orgs/${your_ord_id}/optional_features
```

where you can define the part of the command: "enabled":true or "enabled":false.

## **container\_cluster\_label\_set\_based\_kubernetes\_workload\_instructions**

This flag is enabled by default per organization.

When it's enabled, PCE uses policy deduplication for Kubernetes Workloads under CLAS Container Clusters. This means that we calculate only one workload instruction for every Kubernetes Workload with the same set of labels.

This reduces the number of calculated instructions in production by 70-95%, depending on the customer.

The full list of properties for `optional_features_put` in release 25.2.13 is as follows:

```
"properties": {
    "name": {
        "description": "Name of the feature",
        "type": "string",
        "enum": [
            "ip_forwarding_firewall_setting",
            "ui_analytics",
            "illumination_classic",
            "ransomware_readiness_dashboard",
            "per_rule_flow_log_setting",
            "lightning_default",
            "collector_scanner_filters",
```

```

    "corporate_ips_groups",
    "labels_editing_warning_for_enforcement_mode",
    "label_based_network_detection",
    "cloudsecure_enabled",
    "windows_outbound_process_enforcement",
    "rule_based_label_mapping",
    "core_insights",
    "rule_info_exposure_to_syslog",

"container_cluster_label_set_based_kubernetes_workload_instructions"
]

```

## **hybrid\_policy**

## **Required Permissions for API Users**

To use the REST APIs, you must be an authorized Illumio user with credentials to log into the PCE.

## **User Permissions and the API**

Authentication to the PCE is based on three user roles that allow users to perform specific API operations:

- Organization owner: All GET, POST, PUT, and DELETE APIs
- Administrator: Most GET, POST, PUT, and DELETE APIs
- Read-only: GET only

The PCE also has two other kinds of roles:

- Unscoped: Not bound by label scopes
- Scoped: Bound by label scopes

## **Unscoped Roles**

<b>API Role Name</b>	<b>UI Role Name</b>	<b>Granted Access</b>
owner	Global Organization Owner	Perform all actions: Add, edit, or delete any resource, organization setting, or user account.
admin	Global Administrator	He performs all actions except he cannot change organization settings and cannot perform user management tasks.
read_only	Global Read Only	View any resource or organization setting. Cannot perform any operations.
global_object_provisioner	Global Policy Object Provisioner	Provision rules containing IP lists, services, label groups, and manage security settings. Cannot provision rulesets, virtual services, or virtual servers, or add, modify, or delete existing policy items.

## Scoped Roles

API Role Name	UI Role Name	Granted Access
ruleset_manager	Full Ruleset Manager	Add, edit, and delete all rulesets within the specified scope.
		Add, edit, and delete rules when the Source matches the specified scope. The rule Destination can match any scope.
limited_ruleset_manager	Limited Ruleset Manager	Add, edit, and delete all rulesets within the specified scope.
		Add, edit, and delete rules when the Source and Destination match the specified scope.
ruleset_provisioner	Ruleset Provisioner	Ruleset Managers with limited privileges cannot manage rules that use IP lists, user groups, label groups, or iptables rules as destinations, or rules that allow internet connectivity.
		Provision rulesets within a specified scope. This role cannot provision virtual servers, virtual services, SecureConnect gateways, security settings, IP lists, services, or label groups.

## Session Credentials

While API keys provide a persistent means of authenticating with the PCE, session credentials provide a temporary means of authenticating so you can make Illumio REST API calls.



### IMPORTANT

Any tooling that parses the HTTP headers should be changed to allow case-insensitive header name matching to remain compatible with future PCE releases. For more information, refer to RFC 7230, section 3.2, "Header Fields," which states that field names should be case-insensitive.

Choose a session token or an API key, depending on your programming needs.

## Session Credentials and Tokens

When you create session credentials, an auth\_username and session token are returned, which function as temporary usernames and passwords for making API calls.

Session credentials are used to make all Illumio REST API calls that require authentication and are composed of an auth\_username and a token. They expire after not being used for 30 minutes and reset for another 30 minutes if used within the 30-minute window.

The session token expires after 10 minutes of inactivity.

## When to Use a Session Token

An auth\_username and a session token are useful for one-time use of the API or testing the API. To write a script that performs a one-time use of the API with a session token, use the

Login API to create the `auth_username` and session token. Use those credentials to make other API calls in the script, and then once the script has run, the session token immediately expires when the user logs out.

## What Does a Session Token Look Like?

When you authenticate with the PCE using the Login API, the response returns the credentials needed to make other API calls:

- Your username: `"auth_username": "user_3"`
- Your session token: `"session_token": "xxxxxxxx563199f92af7b705ddca26854205b5233"`

To use the Illumio REST API:

1. Call `login_users/authenticate` using the e-mail address and password you used to create your PCE account to obtain an **authentication token**.



### NOTE

The authorization token expires after 30 seconds, so have the next call formed and ready to paste onto the terminal window before calling `login_users/authenticate`.

2. Call `users/login` with the authentication token to obtain temporary session credentials.

## Use Login API to Create Session Credentials

Unless you're using persistent API credentials, whenever you want to access the Illumio REST API, you must authenticate with the PCE using an `auth_username` and a `session token`. To create these session credentials, call `GET /users/login` with the authentication token previously returned by a call to `POST /login_users/authenticate`.

URI

`GET [api_version]/users/login`

## API Call Using Session Credentials

Once you obtain an `auth_username` and session token from the PCE, you use them to make API calls.

For example, suppose you want to use this session token to get a collection of labels in an organization using the Labels API. In that case, the curl command can be written as shown below using the following authentication:

- `auth_username: user_3`
- Session Token: `xxxxxxxx563199f92af7b705ddca26854205b5233`

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/3/labels -H "Accept: application/json" -u user4:'xxxxxxxx628f5773c47b72dbcd437b4a10d85'
```

## Invoking the PCE APIs Using Multi-Factor Authentication (MFA)

Use the following topics to invoke the PCE APIs using Multi-Factor Authentication (MFA).

## Reference Implementation: Using APIs with Multi-Factor Authentication

You can use multi-factor authentication with APIs. Illumio has a reference implementation that uses MFA with Okta and YubiKey as an authenticator for using APIs.

The following resources are available as part of the reference implementation:

- [config.yml](#)
- [api\\_req\\_poc.rb](#)

### Reference Implementation Considerations

The following are the prerequisites for this reference implementation:

- You must have configured Okta as the IdP for SAML.
- You must have configured Okta to use YubiKey as the authenticator.



#### NOTE

While this configuration has been tested with a specific IdP, you may be able to use this deployment method as a reference implementation for using multi-factor authentication for APIs with other IdPs.

### Instructions for Running the `api_req_poc.rb` File

The following script uses two multi-factor authentication challenges for authenticating with Okta, the IdP:

- YubiKey
- User password

Use the following instructions to run the script:

1. Install the following gems, excluding the default environment gems:
  - 'net/http'
  - json
  - uri
  - logger
  - yaml
  - #gem install nokogiri
2. Configure the config.yml file with the login server FQDN and the PCE FQDN.
3. Export the environment variables using the following:

```
export username = "x@domain.com"
export password = "IAMPass123!"
export yubikey code = "<hold finger on YubiKey and paste the code here>"
```

4. Run the api\_req\_poc.rb file.

### Session Credentials Reference

This topic provides examples of session credentials use.

## Examples

Retrieve a Token

This curl example shows how SaaS local users can use the Illumio Login Service (SAML ID for Remote Users)

```
curl -i -X POST https://login.illum.io:443/api/v2/login_users/authenticate?pce_fqdn=scp1.illum.io -u joe_user@example.com:'password' -H "Content-Type: application/json"
```

Illumio on-premises solutions do not use a login server, so the curl command will look like this:

```
curl -i -X POST -u joe_user@my-company.com:password https://pce.my-company.com:8443/api/v2/login_users/authenticate?pce_fqdn=pce.my-company.com -H "Content-Type: application/json"
```

Response Body to Authenticate with Login Service

The response for the Login Users API is an authentication token (in blue font):

```
{ "auth_token": "xxxxxxxxxxxxxxxxxxxxw89QutJ5WLntqz5jUrI2guA1rZJXKfcbwuF" }
```

### Parameters to create session credentials

Login Service authentication token you obtained using the Login Users API.

Login Users API JSON Schema

This API uses the Illumio Core schema `users_login_get.schema.json`.

### Create Session Token

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/users/login -H "Authorization: Token token=ntqz5jUrI2guA1XzUiLCJlbmMiOiJBMTI4Q0JDLUhZJ"
```

### Response Body

`GET /users/login` returns a temporary `auth_username` and `session_token`.

These are used in the curl examples as `$KEY:$TOKEN` respectively (if you're not using persistent API credentials).

Example: `-u user_4:'xxxxxxxx628f5773c47b72dbcd437b4a10d85a06a'`

```
{
  "full_name": "Buford T. Justice",
  "local": true,
  "type": "local",
  "href": "/users/4",
  "auth_username": "user_4",
```

```

"inactivity_expiration_minutes": 10,
"start": "2017-10-12 16:49:49 UTC",
"time_zone": "America/Los_Angeles",
"last_login_ip_address": "209.37.96.18",
"last_login_on": "2020-10-12T16:49:49.000Z",
"certificate": {
    "expiration": "2020-11-27T03:09:00.000Z",
    "generated": false
},
"login_url": "https://devtest166.ilabs.io:8443/login",
"orgs": [
    {
        "org_id": 1,
        "org_href": "/orgs/1",
        "display_name": "illum.io",
        "role_scopes": [
            {
                "role": {
                    "href": "/orgs/1/roles/owner"
                },
                "scope": [],
                "href": "/orgs/1/users/4/role_scopes/4"
            }
        ]
    }
],
"session_token": "xxxxxxxxx628f5773c47b72dbcd437b4a10d85a0",
"version_tag": "60.1.0-9701f78bef46f521e3d6dd98f70cd8c220940885",
"version_date": "Tue Sep 12 11:12:46 2020 -0700",
"product_version": {
    "version": "17.1.1",
    "build": "6168",
    "long_display": "17.1.1-6168",
    "short_display": "17.1.1"
}
}
}

```

Optional Feature Schema: optional\_feature.schema.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "additionalProperties": false,
    "description": "PCE Feature",
    "required": [
        "name",
        "enabled"
    ],
    "properties": {
        "name": {
            "type": "string",
            "description": "The name of the feature"
        },
        "preview": {
            "type": "boolean",

```

```

        "description": "Is this a preview feature"
    },
    "enabled": {
        "type": "boolean",
        "description": "Is this feature enabled"
    }
}
}

```

Get the optional features collection: `optional_features_get`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "array",
    "items": {
        "$ref": "optional_feature.schema.json"
    }
}
```

Set the optional features for an organization: `optional_features_put`

The example shows the properties available in the release 24.1.1, which includes the property `rule_based_label_mapping`.

This property was added to support the new APIs presented in Rule-Based Label Mapping.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "array",
    "items": {
        "oneOf": [
            {
                "type": "object",
                "additionalProperties": false,
                "required": [
                    "name",
                    "enabled"
                ],
                "properties": {
                    "name": {
                        "description": "Name of the feature",
                        "type": "string",
                        "enum": [
                            "ip_forwarding_firewall_setting",
                            "ui_analytics",
                            "illumination_classic",
                            "ransomware_readiness_dashboard",
                            "per_rule_flow_log_setting",
                            "lightning_default",
                            "collector_scanner_filters",
                            "corporate_ips_groups",
                            "labels_editing_warning_for_enforcement_mode",
                            "label_based_network_detection",
                            "cloudsecure_enabled",
                            "rule_based_label_mapping"
                        ]
                    }
                }
            }
        ]
    }
}
```

```

        "windows_outbound_process_enforcement",
        "rule_based_label_mapping"
    ]
},
"enabled": {
    "description": "Enable or disable this feature",
    "type": "boolean"
}
}
},
{
    "type": "object",
    "additionalProperties": false,
    "required": [
        "name",
        "enabled"
    ],
    "properties": {
        "name": {
            "description": "Name of the feature",
            "type": "string",
            "enum": [
                "editable_dns_client_rule",
                "editable_dhcp_client_rule"
            ]
        },
        "enabled": {
            "description": "Enable or disable this feature",
            "type": "boolean"
        },
        "key": {
            "description": "Key required to enable the feature. Contact Illumio Support for more details.",
            "type": "string"
        }
    }
}
]
}
}

```

## Showing Rule ID in Syslog

For large customers handling 10K+ messages per second, including rule IDs in the Syslog events will substantially increase the volume of recorded data.

**New Feature:** Added the organization-level feature flag `rule_info_exposure_to_syslog` (disabled by default) in release 25.1.0. This flag controls whether rule ID information is included in syslog messages.

## Adding the Rule ID to Syslog Events

To add the rule IDs to the syslog events, the API `optional_features` was changed by adding the new property `rule_info_exposure_to_syslog`:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": {
    "oneOf": [
      {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "name",
          "enabled"
        ],
        "properties": {
          "name": {
            "description": "Name of the feature",
            "type": "string",
            "enum": [
              "ip_forwarding_firewall_setting",
              "ui_analytics",
              "illumination_classic",
              "ransomware_readiness_dashboard",
              "per_rule_flow_log_setting",
              "lightning_default",
              "collector_scanner_filters",
              "corporate_ips_groups",
              "labels_editing_warning_for_enforcement_mode",
              "label_based_network_detection",
              "cloudsecure_enabled",
              "windows_outbound_process_enforcement",
              "rule_based_label_mapping",
              "core_insights",
              "rule_info_exposure_to_syslog"
            ]
          }
        }
      ]
    ]
  }
}
=====
```

To enable this feature flag, set the `firewall_settings.enable_all_rule_hit_count_enabled` option. This will instruct the VEN to send the rule IDs and the traffic flow payload to the PCE.

```
PUT /api/v2/orgs/:org_id/optional_features
payload
[
  {
    name: "rule_info_exposure_to_syslog",
    enabled: true
  }
]
```

## Enabling the Rule Data

Before implementing the property `rule_info_exposure_to_syslog`, you must update the firewall settings and set the flag `enable_all_rule_hit_count_enabled=true`.

This can be done using API or PCE console.

Setting the flag via the PCE console is explained in Events Administration Guide.

To set the flag `enable_all_rule_hit_count_enabled` via API, use the following command:

```
curl -u api_${ILO_API_KEY_ID}:${ILO_API_KEY_SECRET} -H "Content-Type: application/json" -X PUT -d '{"rule_hit_count_enabled_scopes": []}' https://${ILO_SERVER}/api/v2/orgs/${ILO_ORG_ID}/sec_policy/draft/firewall_settings
```

## No Log Rules

The No Log Rules feature allows you to reduce data volume (and cost) by ignoring traffic that is not worth inspecting from a security perspective, like 443/80 from the internet to public-facing web servers.

The No Log Rules feature enables you to mark rules as "no log," which means that any flows matching those rules will not be logged in the vTap logs on the VEN and will not be reported to the PCE.

To use this feature, you need to enable the flag:

```
[ { "name": "per_rule_flow_log_setting", "enabled": true } ]
```

## Implementing No Log Rules Feature

To implement the No Log Rules feature, follow these steps:

1. Enable the flag `per_rule_flow_log_setting` by setting it to `true`.
2. Write segmentation rules for the flows you want to allow without logging.
3. Use the API to flip the property `log_flow` on these rules.

### When using the No Log Feature:

- In cases where "regular" and "no log" rules overlap, the "no log" rule will always be evaluated first, and the flows will not be logged.
- These rules will only be sent to VENs 22.5.0 or newer.
- If the feature flag is disabled, the "log\_flow" property will not be returned in the API responses, and it will not be possible to set the property using the API.
- There is currently no UI support for this feature - there will be no visual indication that the rules are set to "log\_flow: false".

### Enabling the Feature Flag

To enable the No Log feature flag, execute the cURL command:

```
curl -X PUT -u $API_USER:$API_PASS -H 'Content-type: application/json' --data-binary '[ { "name": "per_rule_flow_log_setting", "enabled": true } ]' $PCE_URL/api/v2/orgs/1/optional_features
```

Verify that the flag was enabled:

```
curl -u $API_USER:$API_PASS $PCE_URL/api/v2/orgs/1/optional_features |
jq '.' | grep "per_rule_flow_log_setting" -C
```

If the commands were successful, you will see the following:

```
{
  "name": "per_rule_flow_log_setting",
  "enabled": true
},
```

## Writing Segmentation Rules

When writing rules to implement the No Log Rules feature, do not provision them immediately.

You might create a separate ruleset for these rules and make it very clear that these are No Log Rules in the name, because there is currently no UI indication whether or not the flag is enabled on the rules.

Once you've created the rules, you'll need to get the rule HREFs. You can do that either by making an API call to the rule sets API for this rule set or by using the developer console.

## Disabling Flow Logging

Disable flow logging for the given rule by making a PUT request to the sec\_rules API with the property **log\_flow** set to **false**.

Execute a CURL and an HREF:

```
curl -X PUT -u $API_USER:$API_PASS -H 'Content-type: application/json' --
data-raw '{ "log_flow": false }' $PCE_URL/api/v2/orgs/1/sec_policy/draft/
rule_sets/3/sec_rules/
```

Use a GET command to verify that the property is now **false**.

```
curl -u $API_USER:$API_PASS -H 'Content-type: application/json' $PCE_URL/api/v2/orgs/1/sec_policy/draft/rule_sets/3/sec_rules/3 |
jq '.'
```

Provision the ruleset, and the No Log rules will be sent down to the relevant VENs.

## API Keys

API keys provide a persistent means of authentication with the PCE and are recommended for scripting purposes.

This Public Stable API allows local users to create user API keys and use them as credentials to access the PCE.

There are two categories of API keys:

- **User-based API keys**

These keys are based on specific owners, allowing them to make API calls to the PCE.

- **Service-based API keys**

These API keys are based on a service instead of a specific user.

## Working with API Keys

When you create an API key, you receive an `api_username` and secret, which function as the username and password for making API calls.

An API key is permanent and does not expire, unless it is deleted.



### IMPORTANT

Any tooling that parses HTTP headers should be modified to support case-insensitive header name matching, thereby retaining compatibility with future PCE releases.

Refer to RFC 7230, section 3.2, "Header Fields," which states that field names should be case-insensitive.

Use API keys to write scripts that run automatically without requiring a human user to authenticate the API call. Unless you are a read-only user, you can create multiple API keys and make API calls in your scripts.

You can also create different API keys for various functions. For example, you might use one API key to script automatic workload pairing and another API key to collect system events from Illumio.

When you create an API key, the response returns both the `auth_username` and the secret needed for authenticating other API calls:

- API username:  
`"auth_username":"api_xxxxxxxxxxxxxx29"` (represented in the code examples in this document as \$KEY)
- API key secret: "secret":  
`"xxxxxxxx5048a6a85ce846a706e134ef1d4bf2ac1f253b84c1bf8df6b83c70d95"` (represented in the code examples in this document as \$TOKEN)

## Get a Collection of all API Keys

You can now get a list of all API keys, both user-based and service account-based.

To query API keys regardless of their type, use this API:

```
GET /api/v2/orgs/:xorg_id/api_keys
```

## Special Characters in API Calls

If a **username** or **name** in an API call contains special characters, these must be encoded for the call to succeed.

For example, for a service account name **sa&1**, instead of

```
api/v2/orgs/1/api_keys?type=service_account&name=sa&1
```

Enter the call as

```
api/v2/orgs/1/api_keys?type=service_account&name=sa%261
```

## Query Keys by Expiration

To retrieve the API keys based on the expiration (active or expired) use these APIs:

```
GET /api/v2/orgs/:xorg_id/api_keys?type=service_account&state=expired
```

```
GET /api/v2/orgs/:xorg_id/api_keys?type=service_account&state=active
```

## Create API Keys in the Web Console

You can create API keys using the Web console with the **User** Menu:

1. In the drop-down **User** menu, select **My API Keys**.  
A list of configured API keys is displayed.  
The message "No API Keys" is displayed if no API keys are configured.
2. To add a new API key, click **Add**.
3. In the **Create API Key** pop-up window, enter the name of the API key in the Name field.  
Optionally, enter a description in the Description field.
4. Click **Save** to save your API key or **Cancel** to close the pop-up window without saving your changes.
5. When the **API Key Created** window appears, click the > button next to "Show credentials" to display the credentials for your API key.  
The following information is displayed:
  - **Key ID:** The unique ID of the API key
  - **Authentication Username:** The username that authenticates the API calls
  - **Secret:** The password for the API key
6. Click **Download Credentials** to download the credentials as a text file. Ensure that you have saved the credential information before clicking Done.

After you click **Done**, the API Keys page displays a summary of your new API key, including the following information:

- Name
- Description
- Key ID
- Authentication Username
- Created On

**NOTE**

The credential information is displayed only once. Ensure that you save it in a secure location, as it is used to access the API for your organization. You must create a new API key if you lose your credential information.

## Service Account-based API Keys

Service account-based APIs enable the creation and management of API keys associated with a service account.

You can manage the expiration of service account-based API keys.

**NOTE**

When Service Accounts were introduced, the following restriction was explicitly added: a Service Account cannot be used to operate on service accounts and user-related resources.

In release 23.4.0, this restriction has been removed: a Service Account api\_key can perform most operations, similar to a user api\_key, except for APIs that require user context. There were no changes to the APIs to support this restriction removal.

Service accounts are always organization-based and specific to a PCE.

Users create their permissions while creating a service account, and an API key is implicitly created. Deleting a service account removes its permissions and all associated API keys.

## Service-based API Key Methods

**Table 1. Service-based API Key Methods**

Functionality	HTTP	URI
List all service account API keys.	GET	[api_version][org_href]/api_keys?type=service_account
To retrieve a specific service account	GET	[api_version][org_href]/service_accounts/:service_account_id
To retrieve all API keys, regardless of the account	GET	api_keys
Create a new service account API key.	POST	[api_version][org_href]/service_accounts
To create a new service account API key after performing the required validation	POST	[api_version][org_href]/service_accounts/:service_account_id/api_keys
Update a service account.	PUT	[api_version][org_href]/service_accounts/:service_account_id
Delete a service account API key.	DELETE	[api_version][org_href]/service_accounts/:service_account_id/api_keys/:key_id
Delete a service account that includes any associated API keys.	DELETE	[api_version][org_href]/service_accounts/:service_account_id

The expiration time of service account-based API keys is defined by owners who can specify the default expiration time.

The key expiration time is specified with a default value in the settings, where the expiration date of an existing API key cannot be modified.

When an expired API key authenticates an API request, the request is rejected, and the audit event triggered by this failure includes the API key's Key ID and its expired status. The details also include the expiration date and the `last_used_at` date.

## Settings

Settings for service account-based API keys specify the default expiration period for service account keys and the retention period for expired keys.

The Public Experimental APIs that manage API key settings are based on the role of the organization administrator (**this\_org\_admin**) and are as follows:

- Support for viewing **api\_key** settings for an organization.

`GET /api/v2/orgs/:xorg_id/settings`

This API now includes the new property `num_assets_requiring_ransomware_protection`, which defines several assets that need ransomware protection for this organization.

- Support for updating **api\_key** settings for an organization.

`PUT /api/v2/orgs/:xorg_id/setting`

API key expiration is now set between -1 and 2147483647 seconds, and expired key retention is a minimum of 0 seconds.

The `settings_put.schema.json` was changed to include the new property `num_assets_requiring_ransomware_protection`. The number of assets that require ransomware protection in a specific organization ranges from one to 9999999.

The new property `ven_maintenance_token` identifies if the tampering protection for the VEN and endpoints is enabled. The default is **not enabled**.

## User-based API Keys

This Public Stable API allows you to manage API keys and make API calls to the PCE.

### Working with User-based API Keys

**Table 2. User API Key Methods**

Functionality	HTTP	URI
Get a collection of API keys	GET	[api_version][user_href]/api_keys
Get an individual API key	GET	[api_version][api_key_href]
Create an API key	POST	[api_version][user_href]/api_keys
Update an API key	PUT	[api_version][api_key_href]
Delete an API key	DELETE	[api_version][api_key_href]

### List User-based API Keys

When you GET an individual API key or a collection of API keys, the response only returns those created by the user who authenticated with the PCE for the session.

This API gets one API key or a collection of API keys created by a specific user. To get a single API key, you need to know the API key's URI, which is returned as an HREF path when you create an API key, and the HREF of the user who created the key.

You can query the user API keys as follows:

```
GET /api/v2/orgs/:xorg_id/api_keys?type=user
```

You can also query the user-based API keys based on their expiration:

```
GET /api/v2/orgs/:xorg_id/api_keys?type=user&state=active
```

```
GET /api/v2/orgs/:xorg_id/api_keys?type=user&state=expired
```

```
GET [api_version][user_href]/api_keys
```

## Create User-based API Keys

This API creates a unique API key and returns an API key ID and secret. You can use these to obtain, update, or delete the key and make other API calls.

To create an API key, you must authenticate using a session token or another API key. To obtain a session token, use the [Users API \[59\]](#) and authenticate with the PCE. You will receive your user ID, user HREF, and a session token that you can use when you call this API to create an API key.



### IMPORTANT

If you use an API key, safely store the key and the secret. Anyone with access to both will have access to the API for your organization.



### IMPORTANT

Due to security concerns, external users are unable to create an API Key, even if their roles permit it.

## Update User-based API Keys

This API enables you to update the name or description of an API key.

To make this call, you need the API key URI, which is returned as an HREF path when you create a User-Based API Key.

```
PUT [api_version][api_key_href]
```

## Delete User-based API Keys

To delete an API key, you need the unique API key ID, which is returned as an HREF path property when you create a new API key or get a single or a collection of API keys.

```
DELETE [api_version][api_key_href]
```

## API Keys Reference

This topic covers parameters, properties, and examples for API keys.

## Parameters

Parameter	Description	Type
user_id	The user ID in the form of an HREF (e.g., 'users/6') of the user who created the API key.	String
api_key_id	This is the actual API key ID. It is used only for DELETE.	String
org_id	Organization ID	Integer
max_results	The maximum number of api keys to return.	Integer
username	The username of the user to filter by	String
name	(POST, PUT, GET) The key name - just a label to be used	String
role	Role URI (JSON-encoded string) to filter on	String
state	State of api keys - active or expired	String
type	Type of principal - User or Service Account	String

Some parameters have been renamed or deprecated to allow differentiation between the type `user` and `service_account`:

- Query parameter `name` is retained for the type `service_account`
- Query parameter `name` is changed to `username` for the type `user`
- Query parameter `service_account_name` was deprecated and consolidated to `name`
- Query parameter `api_key_name` was deprecated and removed as not needed

## Properties

Property	Description	Type
key_id	This is the actual API key ID. Use this query parameter only for a GET instance call.	String
auth_username	Username required for authentication	String
created_at	Timestamp when this key was first created (RFC 3339)	String
name	The key name is just a label to be used.	String
href	URI of the key	date/time

## Examples

Curl Command to Get a Key

The API key is identified in the form of an HREF path property:

```
" /users/11/api_keys/a034248fbcd60b4 "
```

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/users/11/api_keys/
a034248fbcd60b4 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Get a Collection of Keys

To use an API key, store the key and secret safely. Anyone with access to both has access to your organization's API.

Due to security concerns, external users are not allowed to create an API Key even if their roles allow it.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/users/11/api_keys -H
"Accept: application/json" -u $KEY:$TOKEN
```

### Response Body

An API key is represented by its HREF path, as shown here:

```
/users/29/api_keys/1e9bb1787883639d5
```

For example:

```
[
  {
    "href": "/users/29/api_keys/1e9bb1787883639d5",
    "key_id": "1e9bb1787883639d5",
    "auth_username": "api_1e9bb1787883639d5",
    "created_at": "2020-01-27T01:30:22.274Z",
    "name": "my_api_key",
    "description": "my_scripting_key"
  },
  {
    "href": "/users/29/api_keys/1793df73a99255f7e",
    "key_id": "1793df73a99255f7e",
    "auth_username": "api_1793df73a99255f7e",
    "created_at": "2016-03-14T16:20:43.603Z",
    "name": "MyKey",
    "description": "My Special Key"
  }
]
```

### Get All Labels with a Key

If you use an API key to get a collection of labels in an organization, and your API key uses these credentials:

- api\_xxxxxxx64fceee809 is the API key
- xxxxxxxx09137412532289d6ecd10bc89c6b1f608c9a85482e7a573 is the secret (API key password)

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/3/labels -H
"Accept: application/json" -u
```

```
api_xxxxxxxxxx64fce809: 'xxxxxxxx09137412532289d6ecd10bc89c6b1f608c9a85482e7a573
'
```

This document represents session and persistent (API key) credentials as the constants \$KEY:\$TOKEN (without spaces).

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/3/labels -H "Accept: application/json" -u $KEY:$TOKEN
```

## Examples to Create API Keys

URI

```
POST [api_version][user_href]/api_keys
```

An example user HREF looks like this:

```
/users/99
```

```
{
    "name": "my_api_key",
    "description": "my_scripting_key"
}
```

To create a user-based API key

In this curl command, the user authentication (-u) uses the session credentials returned from calling the Login API to log in a user. The API key is passed as a JSON object formatted inside double quotes in the command:

```
curl -i -X POST https://pce.my-
company.com:8443/api/v2/users/14/api_keys -H "Content-Type:application/
json" -u user_14:'xxxxxxxx563199f92af7b705ddca2685' -d
"{"name": "my_api_key", "description": "my_scripting_key"}"
```

Response Body

This example shows the response from creating an API key, which you can use for making other API calls. These values do not expire. The auth\_username functions as the username, and the secret functions as the password when making other API calls:

```
{
    key_id: "xxxxxxxx6654188229"
    secret: "xxxxxxxxxxxxxa6a85ce846a706e134ef1d4bf
              2ac1f253b84c1bf8df6b83c70d95"
    auth_username: api_xxxxxxxxxx6654188229
}
```

These values can now be used to authenticate with the API as follows:

- Username: api\_xxxxxxxxxx29api\_xxxxxxxxxx6654188229
- Password:xxxxxxxxxxxxxa6a85ce846a706e134ef1d4bf2ac1f253b84c1bf8df6b83c70d95

## Parameters for Service-based API Keys

Parameter	Description	Type	Required
org_id	Organization ID.(GET, POST, PUT, DELETE)	Integer	Yes
max_results	Maximum number of service accounts to return (GET)	Integer	No
name	Name of service account to filter by	String	No(GET) Yes(POST)
role	Role URI (JSON-encoded string) to filter on(GET)	String	No
service_account_id	Service account UUID (GET account info, DELETE, PUT, POST)	String	Yes
api_key_id	API Key ID (DELETE)	String	Yes

## Properties for Service-based API Keys

Property	Description	Type	Re-required
name	Service account name	String	Yes
href			Yes
created_at	Timestamp when this service account was first created (RFC 3339)	date/time	Yes
updated_at	Timestamp when this service account was last updated	date/time	Yes
created_by	The user who created this service account  Required: href	Object	Yes
updated_by	The user who last updated this service account	String	Yes
permissions	List of permissions:  required: role, scope	Array	Yes
role	Reference to common/orgs_roles.schema.json		
scope	Reference to org_scope.schema.json		
api_keys	List of associated api_keys  Reference to api_keys_get.schema.json		Yes
api_key	required: "expires_in_seconds"  "type": "integer",  "minimum": -1,  "maximum": 2147483647	Object	
access_restriction	Access restriction assigned to the keys created under this service_account	Object, Null	No
href	URI of service_account	String	
api_keys	List of associated api_keys  Reference to api_keys_get.schema.json	Array	No
expires_in_seconds	Validity of the api_key, in seconds  "type": "integer",  "minimum": -1,  "maximum": 2147483647	String	No

Property	Description	Type	Re- quired
last_login_on	Timestamp when this key was last used	date/time	
account	required:  "href": Associated identity  "type": Type of the account  "name": Name of the account		Yes
service_account_id	API Key ID (DELETE)	String	No
api_key_id	API Key ID (DELETE)	String	Yes

### Parameters for api\_keys\_get (all API keys)

Parameter	Description	Type	Required
key_id	Key ID	String	Yes
auth_username	Username required for authentication	String	Yes
name	The key name is just a label to be used	String	Yes
role	Role URI (JSON-encoded string) to filter on(GET)	String	No
service_account_id	Service account UUID (GET account info, DELETE, PUT, POST)	String	No
api_key_id	API Key ID (DELETE)	String	No

## Response Properties for api\_keys\_get (all API keys)

Property	Description	Type	Required
key_id	Key ID	String	Yes
auth_username	Username required for authentication	String	Yes
created_at	Timestamp when this service account was first created (RFC 3339)	date/time	Yes
name	Service account name	String	Yes
			Yes
href	URI of the key	String	Yes
state	State of the api_key	String	No
expires_in_seconds	Validity of the api_key, in seconds	String	No
created_by	The user who created this api key	String	No
last_login_on	Timestamp when this key was last used  Example: "last_login_on": "2023-04-22T03:54:25Z"	date/time	
account	required:  "href": Associated identity  "type": Type of the account  "name": Name of the account		Yes
access_restriction	Access restriction assigned to the keys created under this service_account	Object, Null	
permissions	List of permissions:  required: role, scope	Array	Yes
role	Reference to common/orgs_roles.schema.json		
scope	Reference to org_scope.schema.json		

## Create a new Service Account API Key

Request

```
{
  "name": "key3",
  "description": "testing key 3",
  "access_restriction": {
    "href": "/orgs/1/access_restrictions/2"
  },
  "permissions": [
    "read"
  ]
}
```

```
{
  "role": { "href": "/orgs/1/roles/ruleset_manager" },
  "scope": [
    {
      "label": {
        "href": "/orgs/1/labels/9",
        "key": "env",
        "value": "Development"
      }
    }
  ]
},
{
  "role": { "href": "/orgs/1/roles/owner" },
  "scope": []
],
{
  "api_key": {
    "expires_in_seconds": 86400
  }
}
}
```

Response

```
{
  "name": "service_account1",
  "description": "testing service_account",
  "href": "/orgs/1/service_accounts/33ed7e04-9b25-
           4c9a-a031-a6b1bd437807",
  "access_restriction": {
    "href": "/orgs/1/access_restrictions/2"
  },
  "permissions": [
    {
      "href": "/orgs/1/permissions/84e5541f-3349-
              41c9-8fdb-9756faf96baa",
      "role": { "href": "/orgs/1/roles/ruleset_manager" },
      "scope": [
        {
          "label": {
            "href": "/orgs/1/labels/9"
          }
        }
      ]
    },
    {
      "role": {
        "href": "/orgs/1/roles/owner"
      },
      "scope": []
    }
  ],
  "api_key": {
    "auth_username": "api_135c247aa6e3b654e",
    "auth_password": "redacted"
  }
}
```

```

    "secret": "ab80cc497f7556e0cd72703c5229d814322c
              301d14d2d8d8c7060d516990097b"
}
}

```

## REST API Users

This Public Stable API allows you to log your User into the PCE, enabling you to obtain a session token to access other REST API calls. This API is your starting point for interacting with the PCE using the REST API.

### Users API Methods

Functionality	HTTP	URI
Authenticate to the Illumio Login Service and obtain a single-use authentication token.	POST	[api_version]/login_users/authenticate
Create a new user.	POST	[api_version][users]
Log in as a user and obtain a session token.	GET	[api_version]/users/login
Log out a user and destroy the session token.	PUT	[api_version][user_href]/log-out
Get a user's information.	GET	[api_version][user_href]
Update the user's information.	PUT	[api_version][user_href]
Change a user's password (a local, non-SSO user).	PUT	[api_version]/login_users/[user_href]/password

### Log In to the PCE

URI to Log In User

GET [api\_version]/users/login

For step-by-step instructions about authenticating to the PCE and using GET /users/login in conjunction with other methods, see [Authentication and API User Permissions \[29\]](#).

### Log Out and Destroy Credentials

This API logs users out of the PCE and destroys the temporary session credentials used to log them in.



#### NOTE

This PUT /logout call is not used with persistent API credentials.

URI to Log Out a User

`PUT [user_href]/logout`

Request Body

The request body is an empty JSON object.

`{ }`

Log Out a User

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/
authentication_services/password_policy -H "Content-Type: application/json"
-u $KEY:$TOKEN -d '{"require_type_symbol": true, "expire_time_days": 90}'
```

## Get User Information

This API retrieves specific information about a user, including the time the user logs into the Illumio PCE, the IP address from which the user logs in, the user's name, and the password.

URI to Get User Information

`GET [user_href]`

## Create a New User

This API creates a new local user.

URI to Create a New User

`POST [api_version][users]`

## Change the User Password

This API method allows currently authenticated users to change their login password.

- The call must be made **by the user currently authenticated** in the session; even an administrator cannot change another user's password.
- An API key is not used with this API.
- The user's login name (typically the user's e-mail address) and login password are used for authentication.
- The user's five most recent passwords are not available for use.

Possible Responses

When you execute the command to change a password, you can receive one of these three messages:

- `204 success`: The password was changed successfully.
- `406`: Validation error such as `invalid`.
- `501`: The password was changed, but the e-mail notification failed.

## REST API Users Reference

This topic covers properties, parameters, and examples for REST API users

## Properties

Property	Description	Type
href	URI of the user.	String
username	Username used for authentication.	String
last_login_on	When the user logged on.	String
last_login_ip_address	The IP address of the system where the user has logged into the PCE.	String
login_count	The number of times the user has logged in.	Integer
full_name	Full name of a user as listed in the PCE web console.	String
time_zone	User's timezone IANA Region name.	String
locked	Indicates if a user account is locked or not. True = locked.	Boolean
effective_groups	A list of group names to which the user belongs.	String
local_profile	Local user profile	Object
updated_at	Date when user account information was last updated in the system.	String
created_at	Date when the user account was created in the system.	String
type	Indicates if the PCE authenticates the user account.	String
	(local) or by a third party SAML-based identity management system (external)	
one_time_password	The time-based one-time password for two-factor authentication. This password is required in addition to username and password for authentication.	String

### Request Example

```
GET https://pce.my-company.com:8443/api/v2/users/5
```

### Get a User's Information

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/users/14 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response Body

In this response, the user is represented in the system by an HREF path property ("href": "/users/14"), which can be used when you want to update the user information.

```
{
  "href": "/users/14",
  "type": "local",
  "effective_groups": [ ],
```

```

"id": 14,
"username": "joe.user@pce.my-company.com",
"full_name": "Ralph W. Emerson",
"time_zone": "America/Los_Angeles",
"locked": false,
"login_count": 75,
"last_login_ip_address": "xxx.37.96.18",
"last_login_on": "2020-08-17T15:42:25.732Z",
"local_profile": {
    "pending_invitation": false
},
"created_at": "2019-10-26T05:24:08.735Z",
"updated_at": "2019-08-17T15:55:40.130Z"
}

```

Request Body for REST API users

Property	Description	Type	Required
full_name	User's full name.	String	No
username	username is an e-mail address such as user@example.com	String	Yes
type	User's type, such as user authenticated as local.	String	Yes
time_zone	The user's timezone IANA region name.	String	No

Create a User

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/users/users
```

Possible Responses

When you execute the command to update a user, you can receive one of these three messages:

- 204 success: A new local user was created successfully.
- 406: Validation error such as invalid.
- 501: The user has been created, but the invitation email failed. The new user cannot register or sign up. If you receive this message, you need to create another local user.

Resend Invitation for a Local User

To resend the invitation to a new local user after an e-mail notification failure, use the following URI:

```
PUT /users/:user_id/local_profile/reinvite
```

## Update User Information

This API updates an Illumio API user's account information.

URI to Update User's Information

`PUT [api_version][user_href]`

#### Request Body

The request body is an empty JSON object.

`{ }`

If you attempt to use a PUT with that URL without a payload, the 406 error shows `No payload provided for PUT request`.

Property	Description	Type	Optional
<code>full_name</code>	User's full name	String	Yes
<code>time_zone</code>	The user's time zone IANA region name	String	Yes

## Log Out a User

Use PUT to log out a user:

```
"logout": {
    "http_method": "PUT",
    "path": "/users/:id/logout",
    "summary": "Logout a specific user and destroy the access
token",
```

Curl Command to log out a User

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/users/12345678/logout
-H "Content-Type: application/json" -u $KEY:$TOKEN
```

where "12345678" is the user ID.

URI to Change the User's Password

`PUT [api_version]/login_users/[user_href]/password`

#### Request Body

Property	Description	Type	Required
<code>password</code>	User's new password must meet these requirements: <ul style="list-style-type: none"> <li>• Have a minimum of 8 characters</li> <li>• Have at least 1 capital letter</li> <li>• Have at least 1 lowercase letter</li> <li>• Have at least 1 number</li> <li>• Not match previously used passwords</li> </ul>	String	Yes

Example Request Body

```
{
    "password": "'new_password'"
}
```

Change the User's Password

```
curl -u 'username'@'company'.com:'existing_password' -X PUT
https://'company'.com:8443/api/v2/login_users/me/password -H "Content-type: application/json" -d '{"password": "'new_password'" }' -i
```

## LDAP Authentication

This Public Experimental API provides user authentication with the PCE using LDAP with OpenLDAP and Active Directory.

LDAP authentication comes in addition to the two previously available methods:

- API keys, which provide persistent authentication, and
- Session credentials, which provide temporary authentication.

### Prerequisites and Limitations

Before configuring LDAP for authentication with the PCE, it is important to provide the required prerequisites and review any limitations.

### Determine Your User Base DN

Before you map your LDAP settings to PCE settings, determine your user base Distinguished Name (DN). The DN is the location in the directory where authentication information is stored.

If you don't have this information, contact your LDAP administrator for assistance.

When configuring the PCE to work with LDAP, be aware of the following:

- PCE uses LDAP protocol version 3 ("v3").
- Supported LDAP distributions include OpenLDAP 2.4 and Active Directory.
- Supported LDAP protocols include LDAP, LDAPS, or LDAP with STARTTLS.

### Limitations

These are the current limitations for LDAP authentication:

- Any locally created user has precedence over an LDAP user of the same name. For example, if the LDAP server has a user with a username attribute (such as `cn` or `uid`) of **johndoe** and the default PCE user of the same name is present, the PCE user takes precedence. Only the local password is accepted. The roles mapped to the local user will take effect upon login. To work around this limitation, you must delete the specific local user.
- LDAP and SAML single sign-on authentication methods cannot be used together. In this release of the PCE, an organization can either use LDAP or SAML single sign-on to authenticate external users.

- This release enables LDAP configuration via REST APIs only.

## LDAP Authentication for the PCE

The PCE supports user and role configuration for LDAP users and groups. You can configure up to three LDAP servers and map users and user groups from your LDAP servers to PCE roles.

Before configuring LDAP, review the 'LDAP Prerequisites and Considerations' topic in this document.

### Authentication Precedence

PCE local authentication takes precedence over any external systems. The PCE authenticates a user in the following order:

1. The PCE first attempts local authentication. If the account expires or fails, the PCE will not try to log in using LDAP authentication.
2. If the local user does not exist, the PCE attempts to log in to LDAP (if enabled).

### Configuration Steps

To configure the PCE to work with LDAP, perform these steps:

1. Enable the PCE to use LDAP authentication. See [Enable LDAP Authentication. \[66\]](#)
2. Set up an LDAP configuration.

When searching for LDAP users, the PCE follows the order in which the servers were configured. By default, the configurable request timeout is 5 seconds. Once the request time expires, the PCE attempts to connect to the next server in the configuration.

For example, assume you configure three LDAP servers in the following order: A, B, and C. The PCE will search the servers in that same order. If it finds a user on server A, it stops even if the same user also exists on servers B and C. The PCE will try to use A's credentials for that user, but if it fails to connect to A, it searches the remaining servers: first B, the search proceeds following the expiration of the connection timeout.

3. Map your LDAP groups to one or more PCE roles.

### Set up the PCE for LDAP Authentication

The PCE supports LDAPS and LDAP with STARTTLS. To use the PCE with secure LDAP with SSL/TLS certificates, add the certificate chain to the local certificate store on the PCE.

### Using REST APIs for LDAP Configuration in the PCE

The following table provides an overview of the REST APIs available to configure the PCE for LDAP Authentication. For information about the parameters for these REST APIs, see LDAP Authentication Reference.

APIs for LDAP Configuration

PCE APIs	HTTP   URI
Retrieve the PCE authentication settings.	GET [api_version]/authentication_settings
Update the PCE authentication settings.	PUT [api_version]/authentication_settings
Retrieve the LDAP configuration.	GET [api_version]/authentication_settings/ldap_configs
Get instance	GET [api_version]/authentication_settings/ldap_configs/:uuid
Create an LDAP configuration.	POST [api_version]/authentication_settings/ldap_configs
Update an LDAP configuration.	PUT [api_version]/authentication_settings/ldap_configs/:uuid
Delete an LDAP configuration.	DELETE [api_version]/authentication_settings/ldap_configs/:uuid
Verify the connection to the LDAP server.	POST [api_version]/authentication_settings/ldap_configs/:uuid/verify_connection

## Enable LDAP Authentication

This section explains how to use the API to enable the PCE for LDAP authentication. Before invoking this API, you must enable the LDAP preview feature in the PCE. For instructions on enabling this preview feature, see [LDAP Authentication](#).

### URI

```
PUT /api/v2/authentication_settings
```

Request Body

Property	Data Type	Required	Description
authentication_type	enum	Yes	The type of authentication

Enum Item	Purpose
Local	Local DB authentication
SAML	SAML authentication enabled
RADIUS	RADIUS authentication enabled
LDAP	LDAP authentication enabled

Example Payload to Configure LDAP Authentication

```
{
  "authentication_type": "LDAP",
}
```

## Response Code

The following response codes can be returned:

- 200 indicates success
- 403 indicates the user is not an org owner
- 406 indicates invalid parameters

## **LDAP Authentication Reference**

This topic covers properties and examples for LDAP authentication.

## LDAP Properties

API Property Name	Type	Re-quired	Description
pce_fqdn	String	No	Regional PCE member FQDN for Supercluster.  For non-supercluster deployment, it is the FQDN of the PCE cluster.
name	String	No	Friendly name of the LDAP server
address	String	Yes	IP address or hostname of the LDAP server  Format:  hostname  or ipv4
port	Integer	Yes	Port number of the LDAP server  636 for LDAPS or  389 for STARTTLS
authentication _method	Enum	Yes	<ul style="list-style-type: none"> <li>• LDAP: Clear text connection</li> <li>• LDAPS: LDAP over SSL/TLS Protocol</li> <li>• STARTTLS: LDAP over SSL/TLS Protocol with handshake connection</li> </ul>
request_timeout_sec-onds	Integer	No	Number of seconds to wait for a response.  default 5 seconds.  Possible values: 1-60
bind_distinguish-ed_name	String	No	Distinguished name (DN) used to bind to the LDAP serv-er.
bind_password	String	No	Bind DN password.  Only applicable for POST or PUT operations; the attribute will not be returned for GET instance or collection APIs.
is_bind_password_set	boolean	No	Flag to indicate whether the Bind DN password is config-ured.  Adding this flag because the API does not return the bind password, and there is a need to indicate if the password has been set for the bind_distinguished_name.  Only applicable for GET operation
user_base_ distin-guished_name	String	Yes	Base DN to search for users

API Property Name	Type	Re-quired	Description
user_distinguished_name_pattern	String	No	Pattern used to create a DN string for a user during login; For example, uid=*,ou=people, where * will be replaced with the username
user_base_filter	String	No	Search filter used to query the LDAP tree for users.
username_attribute	String	Yes	An attribute on a user object that contains the username. For example, uid, sAMAccountName, userPrincipalName
full_name_attribute	String	No	An attribute on a user object that contains the full name. For example, cn, commonName, displayName
user_memberof_attribute	String	No	An attribute on a user object containing group membership information. For example, memberOf, isMemberOf
insecure_disable_tls_certificate_verification	boolean	No	Specifies whether to verify the server certificate when establishing an SSL connection to the LDAP server; default is false

Request body for a multi-node cluster for LDAP authentication

```
{
  "name" : "ldap 1" ,
  "address" : "ldap-1.mycompany.com" ,
  "port" : "10636" ,
  "authentication_method" : "LDAPS" ,
  "request_timeout_seconds" : 4,
  "bind_distinguished_name" : 'CN=admin,CN=Users,DC=mycompany,DC=com' ,
  "bind_password" : 'test1234' ,
  "user_base_distinguished_name" : 'DC=mycompany,DC=com' ,
  "username_attribute" : 'sAMAccountName' ,
  "full_name_attribute" : 'cn' ,
  "user_memberof_attribute" : 'memberof',
}
```

Request body for a supercluster

```
{
  "pce_fqdn" : "devmr01" ,
  "name" : "ldap 1" ,
  "address" : "ldap-1.mycompany.com" ,
  "port" : "10636" ,
  "authentication_method" : "LDAPS" ,
  "request_timeout_seconds" : 4,
  "bind_distinguished_name" : 'CN=admin,CN=Users,DC=mycompany,DC=com' ,
  "bind_password" : 'test1234' ,
  "user_base_distinguished_name" : 'DC=mycompany,DC=com' ,
```

```

"username_attribute" : 'sAMAccountName' ,
"full_name_attribute" : 'cn' ,
"user_memberof_attribute" : 'memberof' ,
}

```

### Response Code

The following response codes can be returned:

- 204 indicates success
- 403 indicates the user is not an org owner
- 406 indicates invalid parameters

## REST API Schema Files

The following schema files for LDAP configuration are available starting with release 19.3.5:

`ldap_config.schema.json`

`authentication_settings_ldap_configs_get.schema.json`

`authentication_settings_ldap_configs_post.schema.json`

`authentication_settings_ldap_configs_put.schema.json`

`authentication_settings_ldap_configs_verify_connection_post.schema.json`

`authentication_settings_get.schema.json`

`authentication_settings_put.schema.json`

## Sample Responses

GET /authentication\_settings

```
{
    "authentication_type" : "LDAP"
}
```

Single-PCE: GET /authentication\_settings/ldap\_configs

```
[
    {
        "href": "/authentication_settings/ldap_configs/acf577c8-839a-4828-90f6-797bfclb54d1",
        "pce_fqdn": "test.io",
        "name": "mycompany",
        "address": "ldap-1.mycompany.com",
        "port": 389,
        "authentication_method": "LDAP",
```

```

    "request_timeout_seconds": 5,
    "bind_distinguished_name": "john.doe@mycompany.com",
    "is_bind_password_set": true,
    "user_base_distinguished_name": "OU=Users,OU=mycompany
Employees,DC=mycompany,DC=com",
    "user_distinguished_name_pattern": null,
    "user_base_filter": "(&(objectcategory=person)(objectclass=user))",
    "username_attribute": "userPrincipalName",
    "full_name_attribute": "cn",
    "user_memberof_attribute": "memberOf",
    "insecure_disable_tls_certificate_verification": false,
    "created_at": "2019-03-07T23:30:13.046Z",
    "updated_at": "2019-03-07T23:30:13.046Z",
    "created_by": {
        "username": "john.doe@mycompany.com"
    },
    "updated_by": {
        "username": "john.doe@mycompany.com"
    }
},
]

```

Supercluster: GET /authentication\_settings/ldap\_configs

```

[
{
    "pce_fqdn": "devmr01",
    "href": "/authentication_settings/ldap_configs/8501dff7-cd3f-4c01-9057-
f2b9b1486348",
    "name": "ldap 1",
    "address": "ldap-1.mycompany.com",
    "port": 389,
    "authentication_method": "STARTTLS",
    "is_bind_password_set": false,
    "user_base_distinguished_name": "DC=ilabs,DC=io",
    "user_distinguished_name_pattern": null,
    "username_attribute": "sAMAccountName",
    "full_name_attribute": "cn",
    "user_memberof_attribute": "memberOf",
    "insecure_disable_tls_certificate_verification": false,
    "created_at": "2018-11-30T18:38:36.634Z",
    "updated_at": "2018-11-30T18:38:36.634Z",
    "created_by": {
        "username": "john.doe@mycompany.com"
    },
    "updated_by": {
        "username": "john.doe@mycompany.com"
    }
}
]
```

## Overview of Async GET Requests

An asynchronous job collects and downloads all matching records as a single job. You can configure a script to continuously poll the job until it is done and then download the job results using the job location HREF listed in the response.

### Collection vs. Instance

GET collection methods return HREF path properties for each resource. Perform other REST operations on individual instances of these resources (such as POST, PUT, and DELETE) using the HREF to identify the resources to operate.

For example, the response body for the API to get a collection of labels returns a list of labels, where each one is identified as an HREF path. In this instance, the general syntax for the API call looks like this:

```
GET https://scp.illum.io:8443[api_version][org_href]labels
```

[org\_href] identifies the organization from which you want to get a collection of labels.

A single label instance in the response is identified by its HREF path:

```
{
  href: "/orgs/2/labels/8"
  key: "env"
  value: "Prod"
  created_at: "2020-01-22T18:24:33Z"
  updated_at: "2020-01-22T18:24:40Z"
  created_by: {
    href: "/users/9"
  }
  updated_by: {
    href: "/users/9"
  }
}
```

To perform other operations on this label (href: "/orgs/2/labels/8"), you can provide this HREF in the API call to operate on this label instance.

For example:

```
PUT https://scp.illum.io:8443/api/v2/orgs/2/labels/8
```

### Async GET Supported APIs

These APIs support async GET collections:

Description	Resource Type	Exposure
agents/update	GET [api_version][org_href]/agents GET [api_version][org_href]/agents/update	Experimental Experimental
audit_log_events	GET [api_version][org_href]/audit_log_events	Experimental
auth_security_principals	GET [api_version][org_href]/auth_securiuty_principals	Experimental
authentication_settings/ password_policy	GET [api_version][org_href]/authentication_settings/ password_policy	Experimental
datafiles	GET [api_version][org_href]/datafiles	Experimental
events	GET [api_version][org_href]/events	Experimental
jobs	GET [api_version][org_href]/jobs	Experimental
labels	GET [api_version][org_href]/labels	Both
network_devices/ network_endpoints	GET [api_version][org_href]/network_devices/network_endpoints	Experimental
network_enforcement_nodes	GET [api_version][org_href]/network_enforcement_nodes	Experimental
node_available	GET [api_version][org_href]/node_available	Both
Pairing Profiles	GET [api_version][org_href]/pairing_profiles	Experimental
permissions	GET [api_version][org_href]/permissions	Experimental
security_principals	GET [api_version][org_href]/sec_policy/draft/security_principals	Experimental
system_events	GET [api_version][org_href]/system_events	
vulnerability_reports	GET [api_version][org_href]/vulnerability_reports	Experimental
<b>sec_policy/draft/</b>		
allow	GET [api_version][org_href]/sec_policy/draft/allow	Experimental
dependencies	GET [api_version][org_href]/sec_policy/draft/depend- encies	Experimental
ip_lists	GET [api_version][org_href]/sec_policy/draft/ip_lists	Both
label_groups	GET [api_version][org_href]/sec_policy/draft/la- bel_groups	Experimental
label_groups/member-of	GET [api_version][org_href]/sec_policy/draft/la- bel_groups/member-of	Experimental
modified_objects	GET [api_version][org_href]/sec_policy/draft/modi- fied_objects	Experimental

Description	Resource Type	Exposure
pending	GET [api_version][org_href]/sec_policy/draft/pending	Experimental
rule_sets	GET [api_version][org_href]/sec_policy/draft/rule_sets	Both
rule_sets/sec_rule	GET [api_version][org_href]/sec_policy/draft/rule_sets/sec_rules	Both
services	GET [api_version][org_href]/sec_policy/draft/services	Both
virtual_service	GET [api_version][org_href]/sec_policy/draft/virtual_services	Both
<b>settings/</b>		
settings	GET [api_version][org_href]/settings	
syslog/destinations	GET [api_version][org_href]/settings/syslog/destinations	Experimental
workloads	GET [api_version][org_href]/settings/workloads	Experimental
<b>users/</b>		
users	GET [api_version][org_href]/users	Stable
api_keys	GET [api_version][org_href]/users/api_keys	Both
orgs	GET [api_version][org_href]/users/orgs	Experimental
login	GET [api_version][org_href]/users/login	Stable
<b>workloads/</b>		
workloads/	GET [api_version][org_href]/workloads	Both
interfaces	GET [api_version][org_href]/workloads/interfaces	Both

## Async Job Operations

To create the asynchronous GET job request, set the following preference:

```
-H 'Prefer: respond-async'
```

Setting this preference executes the request during low-traffic times as an asynchronous job in the background, which lightens network traffic loads.

## Workflow

The workflow for requesting an asynchronous bulk job consists of the following tasks:

1. Create the asynchronous GET job request.
2. Poll the job until the status is "Done" or "Failed."

- 3.** Obtain the HREF of the completed request job.
- 4.** Use the HREF to get the results of the request job.

## Create an Async Job Request

This example demonstrates a request for an asynchronous collection of labels.



### NOTE

Use query parameters for a filtered job request, such as to return only the environment labels: .../labels?key=env

### URI to Create a Job Request

```
GET [api_version]/labels
```

The 'Prefer': respond-async header is highlighted in blue bold font:

```
curl -i -X GET 'https://pce.my-company.com:8443/api/v2/orgs/1/labels' -H  
'Accept: application/json' -H 'Prefer: respond-async' -u $KEY:$TOKEN
```

Response with a Job Status

The response is 202 – Accepted, which includes Location, the header Retry-After and an empty body:

```
Server: nginx
Date: Thu, 14 Jan 2020 23:16:52 GMT
"location": "https://pce.my-company.com:8443/api/v2/orgs/1/jobs/d1775367-1951-4707-aa2e-37a0b9076d31",
Retry-After: 5
Transfer-Encoding: chunked
Connection: keep-alive
Status: 202 Accepted
Cache-Control: no-cache
X-Request-Id: 36aae8ce-82ed-4a6a-8a76-77d2df78daff
```

## Poll the Job

After submitting the job request, poll the job using the suggested Retry-After time to determine when the job is complete.

URI to Get the Status of the Job

The following example demonstrates how to poll the job to determine its status.

```
GET [api_version][org_href]/jobs/[href]
```

Poll the HREF provided in the Location field of the response using the duration specified in Retry-After until the status is either done or failed.

```
curl -i -X GET 'https://pce.my-company.com:8443/api/v2/orgs/1/jobs/[href]'  
-H 'Accept: application/json' -u $KEY:$TOKEN
```

### Async Job Response Properties

The following table defines the properties returned in the response:

Property	Description	Type	Required
href	HREF for resource	String	Yes
status	The current state of the job, to the effect of its success, failure, etc	String	Yes
job_type	An arbitrary designator for the job type or kind, typically supplied by the job requestor	String	No
result	The result produced by the job, typically a URI (with 'href' sub-property), or an error in case of failure.	Object, Null	No
requested_at	Time PCE received request	Date-time	No
requested_by	The URI of the user who requested this job	Object (HREF, required)	No
terminated_at	The time (rfc3339 timestamp) at which this job terminated, either successfully or failingly.",	Date-time	No
created_by	Creator of request	Object (HREF, required)	No

### Async Job Status

If the job status is `running`, the response body includes the following results:

```
{
  "href": "/orgs/1/jobs/43f6e9e3-6a68-4481-87c6-18fd096dafbe",
  "job_type": ":illumio/async_requests",
  "description": "/orgs/1/labels",
  "result": {
  },
  "status": "running",
  "requested_at": "2020-01-14 23:16:52.303166",
  "requested_by": {
    "href": "/users/1"
  }
}
```

## Get Async Job Results

The following example demonstrates how to get job results.

URI to Get Async Job Results

```
GET [api_version][org_href]/datafiles/[href]
```

## Curl Command to Get Async Job Results

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/1/datafiles/[href] -H 'Accept: application/json' -u $KEY:$TOKEN
```

## Response Body with Request Results

When the job is complete, use the HREF in the Result field to obtain the results:

```
{
  "href": "/orgs/1/jobs/43f6e9e3-6a68-4481-87c6-18fd096dafbe",
  "job_type": ":illumio/async_requests",
  "description": "/orgs/1/labels",
  "result": {
    "href": "/orgs/1/datafiles/[href]"
  },
  "status": "done",
  "requested_at": "2020-01-14 23:16:52.303166",
  "terminated_at": "2020-01-14 23:17:05.223047",
  "requested_by": {
    "href": "/users/1"
  }
}
```

## Poll the Query Job Status

After submitting the job request, poll the job using the suggested "Retry-After" duration to determine when the job is complete.

The PCE has four possible status options for the job:

- Pending: Waiting to start
- Running: In progress
- Done: Complete (successful/unsuccessful)
- Failed: Unable to complete (exceeded time limit)

## Get Jobs

Specify the maximum number of jobs to return with the `max_results` query parameter.

Specify the type of job to return with the `job_type` query parameter.

### URI to Get the Status of All Jobs

```
GET [api_version]/jobs
```

## Curl Command to Get All Job Status

```
curl -i -X GET 'https://pce.my-company.com:8443/api/v2/orgs/1/jobs' -H 'Accept: application/json' -u $KEY:$TOKEN
```

## Get a Job

### URI to Get the Status of a Job

GET [api\_version]/jobs/[href]

Curl Command to Get a Job Status

```
curl -i -X GET 'https://pce.my-company.com:8443/api/v2/orgs/1/jobs/[href]'  
-H 'Accept: application/json' -u $KEY:$TOKEN
```

Response - Updated Job

If the job is still running, the response includes a status of "running", as highlighted in blue below:

```
{
  "href": "/orgs/1/jobs/43f6e9e3-6a68-4481-87c6-18fd096dafbe",
  "job_type": ":illumio/async_requests",
  "description": "/orgs/1/labels",
  "result": {
  },
  "status": "running",
  "requested_at": "2016-01-14 23:16:52.303166",
  "requested_by": {
    "href": "/users/1"
  }
}
```

## Delete a Job

URI to Delete a Job

DELETE [api\_version]/jobs/[href]

Curl Command to Delete a Job

```
curl -i -X DELETE 'https://pce.my-company.com:8443/api/v2/orgs/1/jobs/  
[href]' -u $KEY:$TOKEN
```

## Get the Job Results

This example demonstrates how to get job results after polling job returns a status of "done".

The `uuid` path parameter is required. The `filename` path parameter is optional, it specifies the filename to save the job as.

URI to Get Job Results

GET [api\_version][org\_href]/datafiles/[uuid]

Curl Command to Get Job Results

```
curl -i -X GET 'https://yourcompany.com:1234/api/v2/orgs/1/datafiles/  
[uuid]' -H 'Accept: application/json' -u $KEY:$TOKEN
```

Response with Results of Request

```
{
  "href": "/orgs/1/jobs/43f6e9e3-6a68-4481-87c6-18fd096dafbe",
  "job_type": ":illumio/async_requests",
  "description": "/orgs/1/labels",
  "result": {
    "href": "/orgs/1/datafiles/[uuid]"
  },
  "status": "done",
  "requested_at": "2016-01-14 23:16:52.303166",
  "terminated_at": "2016-01-14 23:17:05.223047",
  "requested_by": {
    "href": "/users/1"
  }
}
```

## Overview of Async GET Requests

An asynchronous job collects and downloads all matching records as a single job. You can configure a script to continuously poll the job until it is done and then download the job results using the job location **HREF** listed in the response.

## Collection vs. Instance

GET collection methods return HREF path properties for each resource. Perform other REST operations on individual instances of these resources (such as POST, PUT, and DELETE) using the HREF to identify the resources to operate.

For example, the response body for the API to get a collection of labels returns a list of labels, where each one is identified as an HREF path. In this instance, the general syntax for the API call looks like this:

```
GET https://scp.illum.io:8443[api_version][org_href]labels
```

[**org\_href**] identifies the organization from which you want to get a collection of labels.

A single label instance in the response is identified by its HREF path:

```
{
  href: "/orgs/2/labels/8"
  key: "env"
  value: "Prod"
  created_at: "2020-01-22T18:24:33Z"
  updated_at: "2020-01-22T18:24:40Z"
  created_by: {
    href: "/users/9"
  }
  updated_by: {
    href: "/users/9"
  }
}
```

To perform other operations on this label (href: "/orgs/2/labels/8"), you can provide this HREF in the API call to operate on this label instance.

For example:

```
PUT https://scp.illum.io:8443/api/v2/orgs/2/labels/8
```

## **Async GET Supported APIs**

These APIs support async GET collections:

Description	Resource Type	Exposure
agents/update	GET [api_version][org_href]/agents GET [api_version][org_href]/agents/update	Experimental Experimental
audit_log_events	GET [api_version][org_href]/audit_log_events	Experimental
auth_security_principals	GET [api_version][org_href]/auth_securiuty_principals	Experimental
authentication_settings/ password_policy	GET [api_version][org_href]/authentication_settings/ password_policy	Experimental
datafiles	GET [api_version][org_href]/datafiles	Experimental
events	GET [api_version][org_href]/events	Experimental
jobs	GET [api_version][org_href]/jobs	Experimental
labels	GET [api_version][org_href]/labels	Both
network_devices/ network_endpoints	GET [api_version][org_href]/network_devices/network_endpoints	Experimental
network_enforcement_nodes	GET [api_version][org_href]/network_enforcement_nodes	Experimental
node_available	GET [api_version][org_href]/node_available	Both
Pairing Profiles	GET [api_version][org_href]/pairing_profiles	Experimental
permissions	GET [api_version][org_href]/permissions	Experimental
security_principals	GET [api_version][org_href]/sec_policy/draft/security_principals	Experimental
system_events	GET [api_version][org_href]/system_events	
vulnerability_reports	GET [api_version][org_href]/vulnerability_reports	Experimental
<b>sec_policy/draft/</b>		
allow	GET [api_version][org_href]/sec_policy/draft/allow	Experimental
dependencies	GET [api_version][org_href]/sec_policy/draft/dependencies	Experimental
ip_lists	GET [api_version][org_href]/sec_policy/draft/ip_lists	Both
label_groups	GET [api_version][org_href]/sec_policy/draft/label_groups	Experimental
label_groups/member-of	GET [api_version][org_href]/sec_policy/draft/label_groups/member-of	Experimental
modified_objects	GET [api_version][org_href]/sec_policy/draft/modi-fied_objects	Experimental

Description	Resource Type	Exposure
pending	GET [api_version][org_href]/sec_policy/draft/pending	Experimental
rule_sets	GET [api_version][org_href]/sec_policy/draft/rule_sets	Both
rule_sets/sec_rule	GET [api_version][org_href]/sec_policy/draft/rule_sets/sec_rules	Both
services	GET [api_version][org_href]/sec_policy/draft/services	Both
virtual_service	GET [api_version][org_href]/sec_policy/draft/virtual_services	Both
<b>settings/</b>		
settings	GET [api_version][org_href]/settings	
syslog/destinations	GET [api_version][org_href]/settings/syslog/destinations	Experimental
workloads	GET [api_version][org_href]/settings/workloads	Experimental
<b>users/</b>		
users	GET [api_version][org_href]/users	Stable
api_keys	GET [api_version][org_href]/users/api_keys	Both
orgs	GET [api_version][org_href]/users/orgs	Experimental
login	GET [api_version][org_href]/users/login	Stable
<b>workloads/</b>		
workloads/	GET [api_version][org_href]/workloads	Both
interfaces	GET [api_version][org_href]/workloads/interfaces	Both

## Security Policy Objects

Security policy objects contain information about policy versions, modifications, whether they are still pending and can be reverted, policy dependencies, and policy changes.

### Active vs. Draft

This Public Stable API operates on provisionable objects in either a `draft` (not provisioned) state or an `active` (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, SecureConnect gateways, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — :pversion can be `draft`, `active`, or the ID of the security policy.
- For POST, PUT, DELETE — :pversion can be `draft` (you cannot operate on active items) or the ID if the security policy.

## Security Principals

Security principals are typically unique identifiers for Windows Advanced Directory groups, but they can also be unique identifiers for individuals. This Public Stable API allows you to get (one or many), create (one or bulk), update, and delete security principals.

An array of security principals HREFs can be passed into rules and rulesets in the `consuming_security_principals` array.



### NOTE

The common schema `consuming_security_principals` has been replaced by two other APIs: `consuming_security_principals_get` and `consuming_security_principals_put`

## Security Principals API Methods

Security Principals Methods	HTTP	URI
Get Security Principals	GET	<code>[api_version][org_href]/security_principals/</code>
Get a Security Principal	GET	<code>[api_version][org_href]/security_principals/sid</code>
Create a Security Principal	POST	<code>[api_version][org_href]/security_principals/</code>
Bulk create Security Principals.	PUT	<code>[api_version][org_href]/security_principals/bulk_create</code>
Update a Security Principal	PUT	<code>[api_version][org_href]/security_principals/sid</code>
Delete a Security Principal	DELETE	<code>[api_version][org_href]/security_principals/sid</code>

### Get Security Principals

This GET command, by default, returns information for 100 security principals if `max_results` is not specified.

A maximum value of up to 500 can be specified for `max_results`.

### Update a Security Principal

Use a PUT command to update a security principal.

## Bulk Create Security Principals

This PUT command creates multiple security principals.

A maximum of 2,000 security principals can be added to this API in a call. On success, this API returns an array containing the HREFs of the created security principals.

## Security Principals Reference

This topic provides properties, parameters, and examples of security principals.

### Query parameters for security principals

The only required parameter for all API methods is `org_id`.

Parameter	Description	Type	Required
<code>org_id</code>	Organization ID	Integer	Yes
<code>max_results</code>	Maximum number of entries to return	Integer	No
<code>name</code>	Name of security principal to filter by	String	GET, PUT: No POST: Yes
<code>sid</code>	SID of security principal to filter by	String	GET: No POST, PUT, DELETE: Yes

### Properties for security principals

Parameter	Description	Type
<code>href</code>	URI of security principal	String
<code>sid</code>	Active Directory SID	String
<code>name</code>	Name of security principal	String
<code>used_by_ruleset</code>	Flag to indicate if a ruleset is using this security principal.	Boolean
<code>deleted</code>	Flag to indicate if the security principal has been deleted.	Boolean

### Curl Command to Get Security Principals

```
curl -X GET https://pce.my-company.com:8443/api/v2/security_principals -u $KEY:$TOKEN -H 'Accept: application/json'
```

### Example JSON Response Body

```
{
  "sid": "string",
  "name": "string",
```

```

  "description": "string"
}
```

## Get a specified Security Principal

This GET command returns information about one specific security principal indicated by its sid.

### Curl Command to Get a Security Principal

```
curl -X GET https://pce.my-company.com:8443/api/v2/security_principals/
{sid} -u $KEY:$TOKEN -H 'Accept: application/json'
```

### Example JSON Response Body

```
{
  "sid": "string",
  "name": "string",
  "description": "string"
}
```

## Create a Security Principal

This POST command, on success, returns the HREF of the created security principal.

### Curl Command to Create a Security Principal

```
curl -X POST https://pce.my-company.com:8443/api/v2/security_principals -u
$KEY:$TOKEN -H 'Content-Type: application/json'
```

### Example JSON Request Body

```
{
  "sid": "string",
  "name": "string",
  "description": "string"
}
```

### Curl Command to Bulk Create Security Principals

```
curl -X PUT https://pce.my-company.com:8443/api/v2/security_principals/
bulk_create -u $KEY:$TOKEN -H 'Content-Type: application/json'
```

### Example JSON Request Body

```
[
  {
    "sid": "string",
    "name": "string",
    "description": "string"
  },
  {
    "
```

```

    "sid": "string_2",
    "name": "string_2",
    "description": "string_2"
}
]

```

### Curl Command to Update a Security Principal

```
curl -X PUT https://pce.my-company.com:8443/api/v2/security_principals/{sid} -u $KEY:$TOKEN -H 'Content-Type: application/json'
```

### Example JSON Request Body

```
{
  "name": "string",
  "description": "string"
}
```

### Delete a Security Principal

This command deletes a security principal.

### Curl Command to Delete a Security Principal

```
curl -X DELETE https://pce.my-company.com:8443/api/v2/security_principals/{sid} -u $KEY:$TOKEN
```

This command returns 204 No Content for success.

## Labels

This Public Stable API gets, creates, updates, and deletes labels.

### Labels API Methods

Functionality	HTTP	URI
Get a collection of labels.	GET	[api_version][org_href]/labels
Get an individual label.	GET	[api_version][label_href]
Create a label	POST	[api_version][org_href]/labels
Update a label	PUT	[api_version][label_href]
Delete a label	DELETE	[api_version][label_href]

### Get Labels

This API returns all labels in an organization or a single label. When you get labels, they are returned in the form of an HREF path property, for example: "/orgs/2/labels/1662"

By default, the maximum number of labels returned in a GET collection is 500.



### NOTE

GET returns any label that contains a match, rather than an exact match. For example, a GET request for labels with value=APP could return APP, WEB-APP, and WEBAPP.

## URI to Get Collection of Labels

```
GET [api_version][org_href]/labels
```

## URI to Get an Individual Label

```
GET [api_version][label_href]
```

## Create a Label

This API creates a new label inside an organization for one of the following label types, for which you can provide your string value:

- **Application** ("app"): The type of application the workload supports. Examples are HRM, SAP, Finance, and Storefront.
- **Role** ("role"): The function of a workload. A simple two-tier application consisting of a web server and a database server has two roles: Web and Database.
- **Environment** ("env"): The stage in the development of the application. For example, production, QA, development, and staging.
- **Location** ("loc"): The location of the workload. For example, Germany, the US, Europe, and Asia; or Rack #3, Rack #4, Rack #5; or data center, AWS-east1, AWS-east2, and so on.

## System Default "All" for Labels

The PCE provides a built-in environment, application, and location labels defined as "All" that create broad policies to cover all applications, all environments, and all locations.

For this reason, you cannot create labels of these types defined as "All Applications," "All Environments," or "All Locations" (exactly as written in quotes) to prevent confusion for policy writers.

If you attempt to create labels of these types with the exact name as the system defaults (for example, "All Applications"), you receive an HTTP "406 Not Acceptable" error.

Illumio recommends avoiding the creation of labels with names similar to these default system labels to prevent confusion.

## URI to Create a Label

```
POST [api_version][org_href]/labels
```

## Update a Label

This API allows you to update a label applied to a workload, given that you have the label HREF, which is returned when you get all labels in an organization. For example: "/orgs/2/labels/1662"

### URI to Update a Label

```
PUT [api_version][label_href]
```

## Delete a Label

This API deletes a label from an organization using the label HREF, which is returned when you get a collection of labels in an organization. For example: "/orgs/2/labels/1662"

### URI to Delete a Label

```
DELETE [api_version][label_href]
```

## Labels Reference

This topic provides properties, parameters, and examples for labels.

### Query parameters for labels

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
external_data_reference	A unique identifier within the external data source	String	No
external_data_set	The data source from which a resource originates	String	No
include_deleted	Include deleted labels	Boolean	No
key	Key by which to filter	String	GET: No POST: Yes
max_results	Maximum number of labels to return.	Integer	No
usage	Indicate label usage, including if the label is currently used in an RBAC scope for user permissions if the label is applied to a workload, virtual service, Pairing Profile, selective enforcement, virtual server, or ruleset, and if the label belongs to a label group.	Boolean	No
value	Value on which to filter. Supports partial matches.	String	GET, PUT: No POST: Yes
label_id	Label ID, for [api_version][label_href]	Integer	Yes

## Properties for labels

Property	Description	Type
key	Key in key-value pair	String
value	Value in key-value pair",	String
href	Label URI	
updated_at	Timestamp when this label was last updated	String
created_at	Timestamp when this label was first created	String
external_data_reference	A unique identifier within the external data source	String, Null
external_data_set	The data source from which a resource originates	String, Null

## Curl Command to Get Collection of Labels

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/labels -H
"Accept: application/json" -u $KEY:$TOKEN
```

### Response Body

In the response body, each label returned is identified as an HREF, for example: "/orgs/2/labels/1662"

For example:

```
{
    href: "/orgs/2/labels/1662"
    key: "env"
    value: "Prod"
    created_at: "2020-01-22T18:24:33Z"
    updated_at: "2020-01-22T18:24:40Z"
    created_by: {
        href: "/users/9"
    }
    updated_by: {
        href: "/users/9"
    }
}
{
    href: "/orgs/2/labels/1128"
    key: "role"
    value: "DB"
    created_at: "2020-01-22T18:24:53Z"
    updated_at: "2020-01-22T18:24:59Z"
    created_by: {
        href: "/users/9"
    }
    updated_by: {
```

```

        href: "/users/9"
    }
}

```

### Curl Command to Get a Label

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/labels/8 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response Body

```
{
  href: "/orgs/2/labels/8"
  key: "env"
  value: "Prod"
  created_at: "2020-01-22T18:24:33Z"
  updated_at: "2020-01-22T18:24:40Z"
  created_by: {
    href: "/users/9"
  }
  updated_by: {
    href: "/users/9"
  }
}
```

### An example request body to create a label

```
{
  "key": "role",
  "value": "web"
}
```

### Curl Command to Create a Label

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/labels -H "Content-Type: application/json" -u $KEY:$TOKEN -d '{"key": "role", "value": "web"}'
```

### Response Body

The created label is in the form of an HREF path property. For example, in the response below, the label is identified as "/orgs/2/labels/1677".

```
{
  href: "/orgs/2/labels/1677"
  key: "role"
  value: "my_web_app"
  created_at: "2014-04-18T19:39:27Z"
  updated_at: "2014-04-18T19:39:27Z"
  created_by: {
    href: "/users/76"
  }
  updated_by: {
```

```

        href: "/users/76"
    }
}

```

### An example request body to update a label

To update a label definition, the JSON request body can be constructed as follows:

```
{ "value": "db" }
```

### Curl Command to Update a Label

```
curl -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/labels/1662 -H "Accept: application/json" -u $KEY:$TOKEN -d '{"value": "db"}'
```

### Curl Command to Delete a Label

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/labels/1662 -H "Accept: application/json" -u $KEY:$TOKEN
```

## Label Groups

This Public Stable API helps you write rules more efficiently if the same labels are used repeatedly in rulesets. When you add labels to a label group, the label group can be used in a rule or ruleset scope to represent multiple labels. A label group can also be a member (child) of other label groups.

### Label Groups API Methods

Functionality	HTTP	URI
Get a collection of label groups.	GET	[api_version][org_href]/sec_policy/draft/label_groups
Get an individual label group.	GET	[api_version][label_group_href]
Get an individual label group to see if it is a member of other label groups.	GET	[api_version][label_group_href]/member_of
Create a new label group.	POST	[api_version][org_href]/sec_policy/draft/label_groups
Update an individual label group.	PUT	[api_version][label_group_href]
Delete an individual label group.	DELETE	[api_version][label_group_href]

### Active vs. Draft

This API operates on provisionable objects, which exist in either a **draft** (not provisioned) state or an **active** (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be draft, active, or the ID of the security policy.
- For POST, PUT, DELETE — `:pversion` can be draft (you cannot operate on active items) or the ID if the security policy.

## Get a Collection of Label Groups

This method gets all label groups in your organization. Use this to discover the `label_group_id` to GET a specific label group or for POST, PUT, and DELETE operations.

By default, the maximum number returned on a GET collection of label groups is 500. If you want to get more than 500 label groups, use Asynchronous GET Collections.

URI to Get a Collection of Label Groups

```
GET [org_href]/sec_policy/draft/label_groups
```

URI to Get an Individual Label

```
GET [label_group_href]
```

## Label Group Belonging to Other Groups

This method determines if an individual label group is a member of other label groups. For example, if one label group is also a “child” of three other label groups, the response to this call returns the three “parent” label groups to which the specified label group belongs.

### URI to Check if a Label Group Belongs to Other Label Groups

```
GET [api_version][label_group_href]/member_of
```

#### Response

If the specified label group does not belong to any other label groups, the call returns an HTTP 200 message. If the specified label group belongs to other label groups, the response lists the parent label groups. For example:

```
[  
  {  
    "href": "/orgs/7/sec_policy/draft/label_groups/b51c986b-db35-47d4-  
    ab77-aae570d1f164",  
    "name": "MyLabelsUS"  
  }  
]
```

## Update a Label Group

To update an individual label group, use the HREF of the label group, which is obtained from an API call to get a collection of label groups.

### URI to Update a Label Group

PUT [label\_group\_href]

## Delete a Label Group

To delete an individual label group, specify the HREF of the label group you want to delete, which is obtained from an API call to get a collection of label groups.

### URI to Delete a Label Group

DELETE [api\_version][label\_group\_href]

## Label Groups Reference

This topic provides properties, parameters, and examples for label groups.

### Query parameters for label groups

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
pversion	Security Policy Version	String	Yes
label_group_id	Label Group UUID, for [api_version][label_group_href] and [api_version][label_group_href]/member_of	String	Yes
usage	Include label usage flags.	Boolean	No

## Properties for label groups

Property	Description	Type
href	URI of this label group	String
name	The specific name of a label group to return. Supports partial matches	String
key	Key by which to filter	String
created_at	Timestamp when this label group was first created	String date/time
updated_at	Timestamp when this label group was last updated	String date/time
deleted_at	Timestamp when this label group was deleted	String, null
created_by	The user who originally created this label group  "\$ref": "../common/href_object.schema.json"	
updated_by	The user who last updated this label group  "\$ref": "../common/href_object.schema.json"	
deleted_by	Users who deleted this label group  "\$ref": "../common/href_object.schema.json"	Null
blocked_connection_reject_scopes	Label Group is referenced by Blocked Connection Reject Scopes.  Replaces the property blocked_connection_reject_scope	Boolean
loopback_interfaces_in_policy_scopes	Label Group is referenced by Loopback Interfaces in Policy Scopes.	Boolean
ip_forwarding_enabled_scopes	IP Forwarding Enabled Scopes reference Label Group.	Boolean

Curl Command to Get Label Groups

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/3/sec_policy/draft/label_groups -H "Accept: application/json" -u $KEY:$TOKEN
```

Response

When you get a collection of label groups, each label group is identified by an HREF. You need the HREF to update or delete an individual label group using the API.

```
{
  "href": "/orgs/2/sec_policy/draft/label_groups/4c8e3325-c6dd-4dc2-aadc-971e9de270e4",
```

```

"created_at": "2020-07-25T00:46:52.552Z",
"updated_at": "2020-07-25T00:59:00.177Z",
"deleted_at": null,
"created_by": {
    "href": "/users/3"
},
"updated_by": {
    "href": "/users/3"
},
"deleted_by": null,
"name": "AppGroup1",
"description": null,
"key": "app",
"labels": [],
"sub_groups": [
    {
        "href": "/orgs/2/sec_policy/draft/label_groups/9b30081e-e105-44d8-9945-4c8a30dbe849",
        "name": "AppGroup3"
    }
]
},

```

### Request body to update a label group

This example request body updates the labels contained within a label group.

```
{
    "labels": [
        { "href": "/orgs/28/labels/1100" },
        { "href": "/orgs/28/labels/1098" },
        { "href": "/orgs/28/labels/1099" },
        { "href": "/orgs/28/labels/1101" }
    ],
    "sub_groups": []
}
```

### Curl Command to Update Label Groups

In this example, the label group being updated with the request body from the code example above is identified by its label group HREF.

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/draft/label_groups/3307b3d8-2ca2-48f5-877a-03ada95cd6de -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{"labels": [{"href": "/orgs/28/labels/1100"}, {"href": "/orgs/28/labels/1098"}, {"href": "/orgs/28/labels/1099"}, {"href": "/orgs/28/labels/1101"}], "sub_groups": []}'
```

### Curl Command to Delete a Label Group

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/draft/label_groups/3307b3d8-2ca2-48f5-877a-03ada95cd6de -u $KEY:$TOKEN
```

## Services

This Public Stable API gets, creates, updates, or deletes services. To write services, they must be in the “draft” state, which means they have not been provisioned. To provision changes made to services, use the Security Policy API.

### Services API Methods

Functionality	HTTP	URI
Get a collection of services.	GET	[api_version][org_href]/sec_policy/{pversion}/services
Get an individual service.	GET	[api_version][org_href]/sec_policy/{pversion}/services/service_id
Create a new service.	POST	[api_version][org_href]/sec_policy/draft/services/service_id
Update an individual service.	PUT	[api_version][org_href]/sec_policy/draft/services/service_id
Delete an individual service.	DELETE	[api_version][org_href]/sec_policy/draft/services/service_id

### Active vs. Draft

This API operates on provisioning objects, which exist in either a **draft** (not provisioned) state or an **active** (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be draft, active, or the ID of the security policy.
- For POST, PUT, DELETE — `:pversion` can be draft (you cannot operate on active items) or the ID if the security policy.

### Get Services

This API gets all the services in your organization that are in the “draft” policy state (not yet provisioned).

By default, the maximum number returned on a GET collection of services is 500.

#### URI to Get a Collection of Services

```
GET [api_version][org_href]/sec_policy/draft/services
```

#### URI to Get an Individual Service

```
GET [api_version][service_href]
```

## Create a Service

This method creates an individual service. Once a service is created, it can be used to write rules for a security policy.

### URI to Create a Service

```
POST [api_version][org_href]/sec_policy/draft/services
```

## Update a Service

To update (PUT) an individual service, you need to know its HREF. The HREF of a service is returned when you get a collection of services from the PCE.

### URI to Update an Individual Service

```
PUT [api_version][service_href]
```

## Delete a Service

To delete an individual service, use the HREF of the service you want to delete, which is returned when you get a collection of services.

### URI to Delete an Individual Service

```
DELETE [api_version][service_href]
```

## Services Reference

This topic provides properties, parameters, and examples for service APIs.

## Request Parameters for Services

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
name	Name of service on which to filter. This parameter supports partial matches.	String	GET: No
			POST: Yes
description	Description of the service on which to filter.  This parameter supports partial matches.	String	No
pversion	Security Policy Version	String	Yes
external_data_set	The data source from which the resource originates.	String	No
	For example, if service information is stored in an external database.		
external_data_referenc	A unique identifier within the external data source.	String	No
	For example, if service information is stored in an external database.		
is_ransomware	Services associated with ransomware.	Boolean	No
max_results	The maximum number of results to return using GET.	Integer	No
	The maximum limit for returned services is 500.		
<b>Note:</b> If this parameter is not specified, or a value greater than 500 is specified, a maximum of 500 results are returned.			
name	Name of service on which to filter. This parameter supports partial matches.	String	GET: No
			POST: Yes
port	Specify port or port range to filter results. The range is from -1 to 65535 (0 is not supported).	String	No
proto	Protocol to filter on	Integer	GET: No
			PUT, POST: Yes

## Properties for Services

Properties	Description	Type
href	URI of the service	String
name	Name of service (does not need to be unique)	String
description	Description of the service	String
risk_details	This property contains the object <code>ransomware</code> , which is required to define the Ransomware Dashboard.  It contains the following properties: <ul style="list-style-type: none"><li>• <code>category</code>: Categorization based on Admin or Legacy port used in the service (required)</li><li>• <code>severity</code>: Severity of this service (required)</li><li>• <code>os_platforms</code>: Operating system for this ransomware service, an array with "minItems": 1. (required)</li></ul>	Object, NULL
description_url	Description URL Read-only to prevent XSS attacks	String
process_name	Name of the process.	String
service_ports	Reference to <code>service_ports.schema.json</code>	
windows_services	Reference to <code>windows_services.schema.json</code>	
external_data_set	External data set identifier.	String
		NULL
external_data_reference	External data reference identifier.	String
		NULL

## sec\_policy\_post

This schema section shows how the property `risk_details` was added to define the categorization of services based on the ransomware threat:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "additionalProperties": false,
    "required": [
        "name"
    ],
    "properties": {
        "name": {
            "description": "Name (does not need to be unique)",
            "type": "string"
        },
        "description": {
            "description": "Description",
            "type": "string"
        }
    }
}
```

```
        } ,
        "risk_details": {
            "type": "object",
            "properties": {
                "ransomware": {
                    "type": "object",
                    "properties": {
                        "category": {
                            "description": "Categorization based on Admin or Legacy port used in the service",
                            "type": "string",
                            "enum": [
                                "admin",
                                "legacy"
                            ]
                        }
                    },
                    "severity": {
                        "description": "Severity of this service",
                        "type": "string",
                        "enum": [
                            "low",
                            "medium",
                            "high",
                            "critical"
                        ]
                    }
                },
                "os_platforms": {
                    "description": "Operating system for this ransomware service",
                    "type": "array",
                    "minItems": 1,
                    "items": {
                        "type": "string",
                        "enum": [
                            "windows",
                            "linux"
                        ]
                    }
                }
            }
        }
    }
}
```

## Curl Command to Get All Services

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/draft/services -H "Accept: application/json" -u $KEY:$TOKEN
```

## Curl Example to Get a Service

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/draft/services/91 -H "Accept: application/json" -u $KEY:$TOKEN
```

## Response

Each individual service returned is identified by a service HREF. To GET, PUT, or DELETE an individual service, identify the service using its HREF in the API call.

```
{
  "href": "/orgs/7/sec_policy/active/services/878",
  "created_at": "2017-02-10T18:10:50.324Z",
  "updated_at": "2017-02-10T18:10:50.324Z",
  "deleted_at": null,
  "updated_by": null,
  "deleted_by": null,
  "name": "ICMP ECHO",
  "description": null,
  "description_url": null,
  "process_name": null,
  "service_ports": [
    {
      "icmp_type": 8,
      "icmp_code": null,
      "proto": 1
    },
    {
      "icmp_type": 128,
      "icmp_code": null,
      "proto": 58
    }
  ]
}
```

## Example payload to create a service

```
{
  "name": "RDP",
  "description": "Windows Remote Desktop",
  "service_ports": [
    {
      "port": 3389,
      "proto": 6
    }
  ]
}
```

## Curl Command to Create Windows Service

This example shows how to create a Windows Remote Desktop (RDP) service.

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/
active/services -H "Content-Type:application/json" -u $KEY:$TOKEN -d
'{"name": "RDP", "description": "Windows Remote Desktop", "service_ports": [
  {"port": 3389, "proto": 6}
]}'
```

## Request body to create a service

This example illustrates the request body you can pass to update a service, for example, to change the port used by the Nginx service from its current port number to 8080:

```
{
  "name": "nginx",
  "service_ports": [
    {
      "port": 8080,
      "proto": 6
    }
  ]
}
```

### Curl Command to Update a Service

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/
active/services/79 -H "Content-Type:application/json" -u $KEY:$TOKEN -d
'{"name": "nginx", "service_ports": [{"port": 8080, "proto": 6}]}'
```

### Curl Command to Delete a Service

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/
active/services/79 -u $KEY:$TOKEN
```

## Core Services Detection

This Public Experimental API helps you identify core services and suggests an appropriate label. 51 services can be detected.

Core services (such as DNS, Domain Controller, NTP, and LDP) are essential to your computing environment and run on one or multiple workloads. Identifying and labeling these workloads is important because they are centrally connected, and other applications depend on them.

When you use core service detection to label and write policies for core services, you can save time on application policies and introduce enforcement faster.

Users can change the port numbers on which a specific core service is running so that they can adjust them to their environment. Users cannot change ports using the UI, only the APIs.

The user authorized to manage core services is the Organization Administrator.

The Common schemas for managing core services:

- `core_services_labels.schema.json`
- `core_services_type_ports_def.schema.json`
- `core_services_type_ports.schema.json`

## Core Services API Methods

Functionality	HTTP	URI
Get all detected core services for this organization.	GET	[api_version][org_href]/detected_core_services
Get a detected core service by UUID.	GET	[api_version][org_href]/detected_core_services/<uuid>
Get the core service summary details.	GET	[api_version][org_href]/detected_core_services_summary
Get all core service types for this organization.	GET	[api_version][org_href]/core_service_types
Get the core service type by UUID.	GET	[api_version][org_href]/core_service_types/<uuid>
Accept, reject, or skip the core service recommendation.	PUT	[api_version][org_href]/detected_core_services/:uuid
Edit the suggested labels of a core service type for the organization.	PUT	[api_version][org_href]/core_service_types/:uuid

## Core Services Reference

This topic covers parameters, properties, and examples for core services.

### Query Parameters for core services

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
action	The action taken on the detected core services.	String	No
core_service_type	Get all detected core services for a particular core service type.	String	No
max_results	The maximum results to be returned	Integer	No
detected_core_service_id	UUID of the detected core service	String	Yes

## Properties for core services

Property	Description	Type
href	The href of this detected core service	String
ip_address	The ip address that is detected as core service	String
core_service_type	Get all detected core services of a particular type, such as Splunk/NFS. The href will be given in the query parameter.	String
method_name	The method by which this core service was detected	String
created_at	The date at which core service was detected	date/time
updated_at	The date core service was updated with action information.	date/time
confidence	Confidence of the detected core service.  "minimum": 50, "maximum": 100"	Integer
feedback	Feedback is provided for this core service recommendation, if any.  "maxLength": 500	String
action	Users can accept, skip, or reject the core service determination.	String
labels_applied	Indicates if the end-user applied labels for this workload	Boolean
last_detected_at	The core service detection algorithm last recommended Date core service.	date/time
workload	Reference to <code>traffic_flows_workload.schema.json</code>	Object

## Parameters for detected\_core\_services\_summary

Parameter	Description	Type	Required
core_service_type	The unique identifier for the core service type. A core service type is defined by a name, port information and PCE-recommended labels	String	Yes
recommended	Total number of detected core services which are skipped or no decision has been made yet	Integer	No
accepted	Number of accepted recommendations	Integer	No
rejected	Number of recommendations rejected by the user	Integer	No

## Parameters for core\_services\_types

Parameter	Description	Type	Re- quired
href	The href of this core service type	URI	Yes
name	The name of the core service type	String	Yes
labels	Reference to core_services_labels.schema.json		
created_at	The timestamp at which this core service type was created	String	Yes
updated_at	The timestamp at which this core service type was updated	String	Yes
required_ports	Required ports for this core service type, if any  Reference to core_services_type_ports.schema.json		
optional_ports	Optional ports for this core service type, if any  "\$ref": "core_services_type_ports.schema.json"		
priority	Each IP/workload is identified for one core service type, and they are ordered by priority.  For PUT: "minimum": 1	Integer	No
num_optional_ports_required	Number of optional ports required  For PUT: "maximum": 65535	Integer	No
source	Indicates whether the source is a core service. The default value is true, which means the source is the core service.	Boolean	No

## Sample URLs and Payloads

GET /api/v2/orgs/1/detected\_core\_services/ ddfe5204-ad29-4bcd-9821-fcb62353a985.

```
{
  "href" :
    "/orgs/1/detected_core_services/ddfe5204-ad29-4bcd-9821-fcb62353a985" ,
  "ip_address" :
    "103.10.11.44" ,
  "workload" :
    {
      "hostname" :
        "SE555Q5" ,
      "href" :
        "/orgs/2/workloads/e62d71b3-36c4-4c27-926b-411b93ba6d6f" ,
      "labels" :
        []
    } ,
  "core_service_type" :
    {
      "href" :
        "/orgs/1/core_service_type/3555d1e4-fcb2-49c2-9a4a-215c4d5e86dc"
```

```

} ,
"confidence" :
    100 ,
"method_name" :
    "process_based" ,
"created_at" :
    "2020-08-04T05:02:46.648Z" ,
"updated_at" :
    "2020-08-04T05:02:46.648Z" ,
"last_detected_at" :
    "2020-09-05T05:02:46.648Z"
}

```

**PUT /api/v2/orgs/1/detected\_core\_services/3ddd5204-ad29-4bcd-9821-fcb62353a98f**

Take the appropriate action for the identified core services, such as accepting the recommendation to apply the suggested labels to the workload.

Example

```

1 :
{ "action" : "accept" }

```

Example

```

2 :
{ "action" : "accept" ,
  "workload" :{ "href" :
    "/orgs/2/workloads/e62d71b3-36c4-4c27-926b-411b93ba6d6f" } } # for the
case when an IP is converted to UMWL and accepted as core service

```

Example

```

3 :
{ "action" : "reject" }

```

Example

```

4 :
{ "action" : "reject" ,
  "feedback" : "Not a core service." }

```

Example

```

5 :
{ "action" : "skip" ,
  "feedback" : "Check with Ops if this is a core service." }

```

Example

```

6 :
{ "labels_applied" : true }

```

**GET /api/v2/orgs/ :xorg\_id /core\_service\_types/44dd5204-ad29-4bcd-9821-fcb62353a98f**

```

{
  "href" : "/orgs/2/core_service_type/44dd5204-ad29-4bcd-9821-
fcb62353a98f" ,
  "core_service" : "splunk" ,
  "required_ports" :[{ "port" : 9997 ,

```

```

        "to_port" : 10000 } ],
    "optional_ports" :[{ "port" : 112 }, { "port" : 455 }],
    "labels" : [
    {
        "value" : "app-splunk" ,
        "key" :
            "app"
        "href" : "/orgs/1/labels/2"
    },
    {
        "value" : "role-splunk" ,
        "key" :
            "role" ,
        "href" : "/orgs/1/labels/12"
    } ],
    "created_at" :
        "2020-08-04T05:02:46.648Z" ,
    "updated_at" :
        "2020-08-05T05:02:46.648Z"
}

PUT /api/v2/orgs/ :xorg_id /core_service_types/44dd5204-
ad29-4bcd-9821-fcb62353a98f

{
    "labels" : [
        {
            "href" : "/orgs/1/labels/3"
        },
        {
            "href" : "/orgs/1/labels/10"
        }
    ]
}

```

## Virtual Services

This Public Stable API allows you to write rules per service instead of having to write rules that apply to all the services running on a workload. By binding a workload to individual services, you can isolate one or more services running on a workload and create policies specific to those services. By binding services, you have the flexibility to create a finely-grained, highly-segmented security policy.

You can use it in rules once you have created, provisioned, and bound a virtual service to a specific workload. See [Create an Individual Virtual Service \[109\]](#).

## About Virtual Services

Virtual services can consist of a single service or a collection of explicitly enumerated port/port range and protocol tuples. They can be used directly in a rule as a single entity, or labels representing multiple virtual services can be used to write rules.

Virtual services are dynamically bound to workloads using service bindings. Create a virtual service, and then use a service binding to bind the specific virtual service to a workload. Rules written using a virtual service only apply to the workload to which the service is bound.

Use virtual services in the following scenarios:

- **Apply Rules to a Single Service**

This scenario represents a service or process on a workload, identified by a name or label. You can write a policy that allows other entities to communicate only with that single service. The policy does not need to change if the service is moved to a different workload or a new set of workloads. Only the workload bindings on the virtual service need to be changed. The PCE dynamically calculates the required rules on the updated workloads to allow this virtual service.

- **Applying Rules to one of the many Virtual Services Running on a Workload**

In this case, multiple virtual services run on the workload, each with a different label, and the rule targets a subset of these services. You can write a rule to allow other entities to communicate only with that specific service. The policy does not need to change if this service is moved to a different workload or a new set of workloads.

Only the workload bindings on the virtual service need to be changed. The PCE dynamically calculates the required rules for the updated workloads to allow virtual services.

## Virtual Services API Methods

Functionality	HTTP	URI
Get a collection of virtual services.	GET	[api_version][org_href]/sec_policy/: pversion/virtual_services
Get an individual virtual service.	GET	[api_version][org_href]/sec_policy/: pversion/virtual_services/virtual_service_id
Create a new virtual service.	POST	[api_version][org_href]/sec_policy/ draft/virtual_services
Create a collection of virtual services.	PUT	[api_version][org_href]/sec_policy/ draft/virtual_services/bulk_create
Update a virtual service.	PUT	[api_version][org_href]/sec_policy/ draft/virtual_services/virtual_service_id
Update a collection of virtual services.	PUT	[api_version][org_href]/sec_policy/ draft/virtual_services/bulk_update
Delete a virtual service.	DELETE	[api_version][org_href]/sec_policy/ draft/virtual_services/virtual_service_id

## Active vs. Draft Policy Items

Because virtual services are policy items, changes must be provisioned before they can take effect on your policy. Policy items always exist in either a **draft** (not provisioned) or **active** (provisioned) state.

Security policy items that must be provisioned to take effect include IP lists, rulesets, rules, services, virtual services, label groups, user groups, virtual servers, and PCE security settings.

For these items, the URL of the API call must include the URI element called :pversion, which can be set to either **draft** or **active** when you make the API call.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be `draft` or `active`.
- For POST, PUT, DELETE — `:pversion` can only be `draft` (you cannot operate on provisioned items)

## Get a Collection of Virtual Services

Use this method to get a collection of Virtual Services.

URI to Get a Collection of Virtual Services

```
GET [api_version][org_href]/sec_policy/:pversion/virtual_services
```

## Get an Individual Virtual Service

In the call, you identify the virtual service by its HREF, which can be obtained when you get a collection of virtual services.

## Create an Individual Virtual Service

Use this method to create an individual virtual service. Because a virtual service is a policy item, you must create it in the draft state and then provision the change using the Security Policy API.

Once the virtual service is provisioned, you can use the service binding method to bind the virtual service to a workload.

## URI to Create an Individual Virtual Service

```
POST [api_version][org_href]/sec_policy/draft/virtual_services
```

To create a virtual service, you need the HREF of the service you want to “bind” to a workload. You can obtain a service HREF by calling a GET collection with the service binding API.

Additionally, if you want to add labels to the virtual service, you need the HREF of each label you want to add. Label HREFs can be obtained by calling a GET collection with the labels API. Labels are represented in the JSON request body as an array, opened and closed by square brackets ([ ]).

## Create or Update Virtual Services Collection

This method enables you to create a collection of virtual services in your organization using a single API call instead of creating individual services one at a time.

This capability is useful if you want to keep a set of PCE resources in sync with your internal representation of those resources, such as a configuration management database (CMDB) that serves as the “source of truth” for your PCE resources.

After creating virtual services and adding identifiers to the service properties, you can collect virtual services using query parameters that include the external data reference. You can also run an asynchronous query to get all virtual services through an offline job, which includes the external data references in the response.

The two properties you can use when creating virtual services, `external_data_set` and `external_data_reference` are UTF-8 strings with a maximum length of 255 characters each. The contents must form a unique composite key, meaning that both values of these properties are treated as a unique key. Together, these two properties are recognized as unique keys, even if one is left blank or set to zero.

### **URI to Create a Collection of Virtual Services**

```
PUT [api_version][org_href]/sec_policy/draft/virtual_services/bulk_create
```

### **URI to Update a Collection of Virtual Services**

```
PUT [api_version][org_href]/sec_policy/draft/virtual_services/bulk_update
```

#### **Request Body**

To create a collection of virtual services, pass a JSON object that describes the details of the virtual service. This method's request body and curl command follow the same structure used to create an individual virtual service; you only add multiple virtual service JSON objects instead of just one.

Additionally, the `href` field must be present in the body for each virtual service you update in the `bulk_update`.



#### **NOTE**

Bulk operations are rate-limited to 1,000 items per operation.

### **Update an Individual Virtual Service**

To update (PUT) an individual virtual service, you need to know the HREF of the virtual service you want to update. Virtual service HREFs are returned when you get a collection of virtual services.

### **URI to Update an Individual Virtual Service**

```
PUT [api_version][org_href]/sec_policy/draft/virtual_services/  
virtual_service_id
```

### **Virtual Services and Bindings Reference**

This topic covers properties and examples of virtual services and bindings.

## Virtual Services Query Parameters

Property	Description	Type	Re-required
org_id	Organization ID	Integer	Yes
pversion	Security Policy Version	String	Yes
external_data_reference	A unique identifier within the external data source.  For example, if this virtual service information is stored in an external database.	String, NULL for PUT only	
external_data_set	The data source from which the resource originates.  For example, if this virtual service information is stored in an external database.	String	
name	Name on which to filter. Supports partial matches	String	No
labels	List of lists of label URIs, encoded as a JSON string	String	No
virtual_service_id	Virtual Service ID	String	Yes
service	Service URI	String	No
service_ports.port	Specify port or port range to filter results. The range is from -1 to 65535.	String	No
service_ports.proto	Protocol to filter on.	Integer	No
service_address.fqdn	FQDN configured under service_address property supports partial matches	String	No
service_address.ip	The IP address configured under the service_address property supports partial matches	String	No
usage	Include Virtual Service usage flags.	Boolean	No

## Virtual Services Properties

Property	Description	Type
href	URI of the virtual service	String
created_at	Timestamp when this virtual service was first created	String date/time
updated_at	Timestamp when this virtual service was last updated	String date/time
deleted_at	Timestamp when this virtual service was deleted	String/NULL date/time
name	Name (does not need to be unique)	String
labels	Virtual service labels  References common/label_optional_key_value.schema.json	
update_type	Update type for the virtual service  Reference to common/sec_policy_update_type.schema.json	
external_data_set	The data source from which the resource originates.  For example, if this virtual service information is stored in an external database.	String, NULL for PUT only
external_data_reference	A unique identifier within the external data source.  For example, if this virtual service information is stored in an external database.	String, NULL for PUT only
service_addresses	Reference to virtual_service_service_addresses.schema.json	
ip_overrides	Array of IPs or CIDRs as IP overrides	

### Curl command to get a collection of virtual services

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/sec_policy/active/virtual_services -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response

Each individual virtual service returned is identified by the virtual service HREF. To GET, PUT, or DELETE an individual virtual service, identify the service by its HREF in the API call.

```
[  
{
```

```

    "href": "/orgs/7/sec_policy/draft/virtual_services/1828d8ff-
aeb7-4735-9975-db692813d193",
    "created_at": "2017-10-29T19:41:15.648Z",
    "updated_at": "2017-10-29T19:41:15.648Z",
    "deleted_at": null,
    "created_by": { "href": "/users/14" },
    "updated_by": { "href": "/users/14" },
    "deleted_by": null,
    "update_type": null,
    "name": "Jawoo",
    "description": null,
    "service": { "href": "/orgs/7/sec_policy/draft/services/99" },
    "labels": [
        { "href": "/orgs/7/labels/88" },
        { "href": "/orgs/7/labels/82" },
        { "href": "/orgs/7/labels/92" },
        { "href": "/orgs/7/labels/101" }
    ],
    "ip_overrides": [
        "192.0.1.0",
        "192.168.100.0/24"
    ],
    "apply_to": "host_only"
}
]

```

### URI to Get an Individual Virtual Service

GET [api\_version][virtual\_service\_href]



#### NOTE

For this method, you can specify either draft or active for :pversion.

### Curl Command

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/
draft/virtual_services/89 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response

```
{
    "href": "/orgs/2/sec_policy/draft/virtual_services/6005a35a-1598-4c7b-
a827-be4390f46773",
    "created_at": "2017-12-11T20:56:28.629Z",
    "updated_at": "2017-12-11T21:07:10.407Z",
    "deleted_at": null,
    "created_by": { "href": "/users/9" },
    "updated_by": { "href": "/users/9" },
    "deleted_by": null,
    "update_type": "create",
```

```

"name": "Docker1",
"description": null,
"service": { "href": "/orgs/2/sec_policy/draft/services/5" },
"labels": [
    { "href": "/orgs/2/labels/18" },
    { "href": "/orgs/2/labels/26" },
    { "href": "/orgs/2/labels/126" }
],
"ip_overrides": [
    "192.0.1.0",
    "192.168.100.0/24"
],
"apply_to": "internal_bridge_network"
}

```

### Request Body for an Individual Virtual Service

```
{
  "name": "MyVirtualService",
  "description": "Test",
  "service": { "href": "/orgs/7/sec_policy/draft/services/218" },
  "labels": [
    { "href": "/orgs/7/labels/88" },
    { "href": "/orgs/7/labels/82" },
    { "href": "/orgs/7/labels/92" },
    { "href": "/orgs/7/labels/95" }
  ]
}
```

### Curl Command to create a new virtual service

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/
virtual_services -H "Content-Type: application/json" -u $KEY:$TOKEN.-d
'{
  "name": "MyVirtualService",
  "description": "Test",
  "service": { "href": "/orgs/7/sec_policy/draft/services/218" },
  "labels": [
    { "href": "/orgs/7/labels/88" },
    { "href": "/orgs/7/labels/82" },
    { "href": "/orgs/7/labels/92" },
    { "href": "/orgs/7/labels/95" }
  ]
}'
```

### Request Properties for updating a virtual service

The request properties for updating a virtual service are the same as those for creating a virtual service.

### Request Body

This example request body can be passed to update a virtual service to include a workload binding:

```
{
  "service": { "href": "/orgs/2/sec_policy/draft/services/91" },
  "labels": [
    { "href": "/orgs/2/labels/316" },
    { "href": "/orgs/2/labels/101" },
    { "href": "/orgs/2/labels/102" },
    { "href": "/orgs/2/labels/103" }
  ]
}
```

```

        { "href": "/orgs/2/labels/103" }
    ]
}

```

## Curl Command

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/
sec_policy/draft/virtual_services/256525b6-e7c5-4ad7-b7af-e70586aa1078
-H "Content-Type: application/json" -u $KEY:$TOKEN
-d '{"name":"test","description":null,"service":{"href":"/orgs/2/labels/
316"}, "labels": [{"href":"/orgs/2/labels/101"}, {"href":"/orgs/2/labels/
102"}, {"href":"/orgs/2/labels/103"}]}'
```

## Service Bindings Query Parameters

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
virtual_service	Virtual service href	String	No
service_binding_id	Service Binding ID	String	Yes
workload	The complete HREF of the workload referenced in the service binding.	String	No
external_data_reference	A unique identifier within the external data source. For example, if this virtual service information is stored in an external database.	String	No
external_data_set	The data source from which the resource originates. For example, if this virtual service information is stored in an external database.	String	No

## Service Bindings Properties

Property	Description	Type
virtual_service	Virtual service href, required for POST	Object
bound_service	Bound service href, required for GET	Object
workload	The complete HREF of the workload referenced in the service binding.	String
port_overrides	Reference to port_overrides.schema.json	
external_data_reference	A unique identifier within the external data source. For example, if this virtual service information is stored in an external database.	String, NULL
external_data_set	The data source from which the resource originates. For example, if this virtual service information is stored in an external database.	String, NULL
workload	HREF of the workload	String

## Request Parameters to create a service binding

The request body for creating a service binding is an array of service binding objects. Because this JSON request body is an array, you can create multiple service bindings in a single POST.



### **NOTE**

Ensure that the virtual service you bind to a workload has been provisioned.

This is an example JSON representation of a single service binding:

```
[{"workload": {"href": "/orgs/1/workloads/45c69cf3-4cbb-4c96-81ee-70e94baea1b8"}, "virtual_service": {"href": "/orgs/1/sec_policy/draft/virtual_services/a735332e-5d31-4899-a3a5-fac7055e05c0"}, "port_overrides": [{"port": 14000, "protocol": 6, "new_port": 26000}]}
```

### **Curl Command**

To create a single service binding:

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/service_bindings -H "Content-Type:application/json" -u $KEY:$TOKEN -d '[{"workload": {"href": "/orgs/1/workloads/45c69cf3-4cbb-4c96-81ee-70e94baea1b8"}, "virtual_service": {"href": "/orgs/1/sec_policy/draft/virtual_services/a735332e-5d31-4899-a3a5-fac7055e05c0"}, "port_overrides": [{"port": 14000, "protocol": 6, "new_port": 26000}]}]
```

### **Request Body to Create Multiple Service Bindings**

An example JSON request body for creating multiple service bindings with a different port number:

```
[{"workload": {"href": "/orgs/1/workloads/820efcdc-c906-46b9-9729-26bab7a53223"}, "virtual_service": {"href": "/orgs/1/sec_policy/draft/virtual_services/e38ce044-d2ac-4d7f-aeec-16ef8fb0b15"}, "port_overrides": [{"port": 10000, "protocol": 6, "new_port": 26000}], {"workload": {"href": "/orgs/1/workloads/820efcdc-c906-46b9-9729-26bab7a53223"}, "virtual_service": {"href": "/orgs/1/sec_policy/draft/virtual_services/e38ce044-d2ac-4d7f-aeec-16ef8fb0b15"}, "port_overrides": [{"port": 11000, "protocol": 6, "new_port": 25000}]}]
```

### **Service Binding Request Body**

If you create more than one service binding with a single POST, all of the service bindings must be constructed properly or the POST will fail and no service bindings will be created.

**NOTE**

The response of “failure” indicates the error, but it does not confirm that no service bindings have been created.

For example, if you use POST to create 10 service bindings and one of the workloads referenced in the JSON payload uses an incorrect URI (HREF), the POST fails with an error message similar to the following message:

```
[ { "token": "invalid_uri", "message": "Invalid URI: { /orgs/1/workloadzzz/ 820efcdc-c906-46b9-9729-26bab7a53223 }" } ]
```

Response Body to get individual or collection of Service Bindings

```
[
  {
    "href": "/orgs/7/service_bindings/287568ad-4a1f-4000-a9fb-e67d1dabce15",
    "virtual_service": { "href": "/orgs/7/sec_policy/active/virtual_services/256525b6-e7c5-4ad7-b7af-e70586aa1078" },
    "workload": { "href": "/orgs/7/workloads/baef2547-2036-4e00-b6f7-3f4be1f7669a" },
    "name": null,
    "hostname": "AssetMgt-proc2",
    "deleted": false },
    "port_overrides": [ { "new_port": 8080, "protocol": 6, "port": 3306 } ]
  },
  {
    "href": "/orgs/7/service_bindings/faebe7bf-0bb7-49a5-868e-8297e038fa9e",
    "virtual_service": { "href": "/orgs/7/sec_policy/active/virtual_services/7b46fce0-4933-4e29-b86c-7a2a71e686ed" },
    "workload": { "href": "/orgs/7/workloads/aee4381b-9836-45b6-b7ab-ae246bf482f" },
    "name": null,
    "hostname": "onlinestore-web2",
    "deleted": false },
    "port_overrides": []
  },
  {
    "href": "/orgs/7/service_bindings/924ad8c2-94bf-40f5-bc4c-13474982bd00",
    "virtual_service": { "href": "/orgs/7/sec_policy/active/virtual_services/256525b6-e7c5-4ad7-b7af-e70586aa1078" },
    "workload": { "href": "/orgs/7/workloads/69fd736b-cd21-4a4c-bdb9-132207c760ce" },
    "name": null,
    "hostname": "test-us",
    "deleted": false },
    "port_overrides": []
  }
]
```

## Curl Command to Get an Individual Service Binding

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/
service_bindings/xxxxxxxxx-4a86-4dd4-b303-23f699d0ebbf -H "Accept:
application/json" -u $KEY:$TOKEN
```

## Curl Command to Get Service Binding Collection

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/
service_bindings -H "Accept: application/json" -u $KEY:$TOKEN
```

## URI to Delete an Individual Service Binding

DELETE [api\_version][service\_binding\_href]

## Curl Command to Delete a Service Binding

Use this curl command to delete the service binding:

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/
service_bindings/xxxxxxxxx-4a86-4dd4-b303-23f699d0ebbf -u $KEY:$TOKEN
```

## Virtual Servers

A virtual server is similar to a workload. It can be assigned labels and has IP addresses, but does not report traffic to the Illumio Core. Each virtual server has only one VIP. The local IP addresses are used as source IP addresses for connection to the pool members (backend servers) when the virtual server operates in SNAT or Auto mode. These IP addresses are likely to be shared by multiple virtual servers on the server load balancer.

A discovered virtual server is a server load balancer (SLB) virtual server (IP address and port(s)) that the NEN has discovered when interrogating SLBs managed by the PCE.

## Virtual Server Methods

There are two groups of methods used to manage virtual servers:

- Methods for virtual servers
- Methods for discovered virtual servers

Functionality	HTTP	URI
Get a list of Virtual Servers.	GET	[api_version][org_href]/sec_policy/:version/virtual_servers
Get a specified Virtual Server.	GET	[api_version][org_href]/sec_policy/:version/virtual_servers/:uuid
Create a Virtual Server object.	POST	[api_version][org_href]/sec_policy/:version/virtual_servers
Modify the enforcement mode, labels, and back-end/source labels of a specified Virtual Server.	PUT	[api_version][org_href]/sec_policy/:version/virtual_servers/:uuid

## Discovered Virtual Servers Methods

You can use only three GET methods for discovered virtual servers.

Functionality	HTTP	URI
Get a list of Discovered Virtual Servers.	GET	[api_version][org_href]/discovered_virtual_servers
Get a specified Discovered Virtual Server.	GET	[api_version][org_href]/discovered_virtual_servers/:uuid
Discovery	GET	[api_version][org_href]/network_enforcement_nodes/virtual_server_discovery_jobs/:uuid

## Virtual Servers Reference

This topic covers virtual server parameters and properties.

## Query Parameters for Virtual Servers

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
pversion	Security Policy Version	String	Yes
discovered_virtual-server	URI of discovered virtual server to filter by	String	No
active_pce_fqdn	FQDN of the PCE	String	No
external_data_reference	A unique identifier within the external data source	String	No
external_data_set	The data source from which a resource originates	String	No
labels	2D array of label URIs, encoded as a JSON string. Filter by virtual server labels.	String	No
max_results	Maximum number of discovered virtual servers to return	Integer	No
mode	Mode of the virtual server(s) to return	String	No
name	Name of virtual server(s) to return. Supports partial matches	String	No
network_enforcement_node.hostname	Hostname of NEN object to filter virtual server(s)	String	No
slb	URI of SLB object to filter virtual server(s)	String	No
vip	Front-end (VIP) address of the virtual server(s). Supports suffix-wildcard matches	String	No
vip_port	Port of front-end VIP of the virtual server(s)	Integer	No
vip_proto	Protocol of front-end VIP of the virtual server(s)	Integer	No

## Properties for Virtual Servers

Property	Description	Type	Re- quired
href	href of virtual server	String	Yes
name	The short friendly name of the virtual server	String	Yes
labels	2D array of label URIs, encoded as a JSON string. Filter by virtual server labels.  "\$ref": "../common/label_optional_key_value.schema.json"	Array	Yes
service	URI of associated service  "\$ref": "../common/href_object.schema.json"		Yes
providers	minItems: 0,  label: "\$ref": "../common/label_optional_key_value.schema.json"  workload: "\$ref": "../common/href_object.schema.json"	Array	Yes
mode	Management mode of the virtual server	String	Yes
discovered_virtual_server	Corresponding discovered virtual server, server URI	String/ Null	Yes
update_type	Reference to common/sec_policy_update_type.schema.json		Yes
created_at	The time (rfc3339 timestamp) at which this virtual server was created	String	Yes
updated_at	The time (rfc3339 timestamp) at which this virtual server was last updated	String	Yes
deleted_at	The time (rfc3339 timestamp) at which this virtual server was deleted	String/ Null	Yes
created_by	The URI of the user who created this virtual server  Reference to common/href_object.schema.json		Yes
updated_by	The URI of the user who last updated this virtual server  Reference to common/href_object.schema.json		Yes
deleted_by	The URI of the user who deleted this virtual server  Reference to common/nullable_href_object.schema.json		Yes

## Discovered Virtual Servers Parameters

Parameter	Description	Type	Re- quired
org_id	Organization ID	Integer	Yes
active_pce_fqdn	FQDN of the PCE	String	No
has_virtual_server	Filter discovered virtual server(s) by whether a virtual server object manages them	Boolean	No
max_results	Maximum number of discovered virtual servers to return	Integer	No
name	Name of discovered virtual server(s) to return. Supports partial matches	String	No
network_enforcement_node	Hostname of NEN object to filter discovered virtual server(s)	String	No
slb	URI of SLB object to filter discovered virtual server(s)	String	No
vip	Frontend (VIP) address of the discovered virtual server(s). Supports suffix-wildcard matches	String	No
vip_port	Port of frontend VIP of the discovered virtual server(s)	Integer	No
vip_proto	Protocol of frontend VIP of the discovered virtual server(s)	Integer	No
virtual_server	URI of virtual server to filter discovered virtual server(s)	String	No
virtual_server_labels	2D array of label URIs, encoded as a JSON string. Filter by virtual server labels. Requires usage of has_virtual_server: true	String	No
virtual_server_mode	Filter discovered virtual server(s) by virtual server mode. Requires usage of has_virtual_server: true	String	No

## Discovered Virtual Server Properties

Property	Description	Type	Re-required
href	href of discovered virtual server	String	Yes
dvs_identifier	NFC-generated unique identifier for discovered virtual server	String	Yes
name	Configured name of virtual server	String	Yes
vip_port	VIP including protocol and port for discovered virtual server.  Reference to <code>common/dvs_vip_port.schema.json</code>		Yes
local_ips	Local IPs of virtual server  Format: ipv4	Array	Yes
mode	Virtual server mode of operation	String	Yes
slb	URI of Service Load Balancer (SLB) object to filter discovered virtual server(s)  Reference to <code>common/href_object.schema.json</code>		Yes
service_checks	Service checks, which has these properties:  • protocol • port	Object	Yes
nfc	DEPRECATED AND REPLACED (USE 'network_enforcement_node' INSTEAD) URI of the NFC for this discovered virtual server  Reference to <code>common/href_object.schema.json</code>		Yes
created_at	The time (rfc3339 timestamp) at which this server load balancer was created	date/time	Yes
updated_at	The time (rfc3339 timestamp) at which this server load balancer was last updated	date/time	Yes
created_by	Reference to <code>common/href_object.schema.json</code>		Yes
updated_by	Reference to <code>common/href_object.schema.json</code>		Yes
network_enforcement_node	URI of the Network Enforcement Node for this discovered virtual server  Reference to <code>common/href_object.schema.json</code>		Yes

## IP Lists

This Public Stable API can get, create, update, and delete IP lists.

IP lists can be used in rules to define sets of trusted IP addresses, IP address ranges, or CIDR blocks allowed into your data center that are allowed to access workloads in your network.

## IP Lists API

Functionality	HTTP	URI
Get a collection of IP lists	GET	[api_version][org_href]/sec_policy/draft/ip_lists
Get an individual IP list	GET	[api_version][ip_list_href]
Create an IP list	POST	[api_version][org_href]/sec_policy/draft/ip_lists
Update an IP list	PUT	[api_version][ip_list_href]
Delete an IP list	DELETE	[api_version][ip_list_href]

## Active vs Draft

This API operates on provisionable objects, which exist in either a **draft** (not provisioned) state or an **active** (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be draft, active, or the ID of the security policy.
- For POST, PUT, DELETE — `:pversion` can be draft (you cannot operate on active items) or the ID if the security policy.

## Get IP Lists

This API allows you to get a collection of IP lists or a single IP list from an organization.

By default, the maximum number returned on a GET collection of IP lists is 500. If you want to get more than 500 IP lists, use Asynchronous GET Collections.

### URI to Get Collection of IP Lists

```
GET [api_version][org_href]/sec_policy/draft/ip_lists
```

### URI to Get an Individual IP List

```
GET [api_version][ip_list_href]
```

### Create an IP List

This API allows you to create IP lists (allowlists) that can be used to create rules in rulesets. An IP list can contain a single IP address or an IP address range.



## WARNING

Please be aware of the following:

0.0.0.0/0 means 0-255 . 0-255 . 0-255 . 0-255 or all possible IP addresses.

0.0.0.0 without the trailing "/0", means a single IP (not ANY IP). This is a rare but sometimes needed object, specifically for DHCP Discovery.

0.0.0.0, when used improperly, might trigger an error, prevent the list from being accepted, and consequently block traffic.

Use the correct syntax for the intended purpose.

## URI to Create an IP List

POST [api\_version][org\_href]/sec\_policy/draft/ip\_lists

## IP Lists Reference

This topic provides properties, parameters, and examples for IP lists.

## Query parameters and properties for IP lists

Parameter	Description	Type	Re-required
org_id	Organization	Integer	Yes
pversion	Security Policy Version	String	Yes
description	Description of IP list(s) to return. Supports partial matches	String	No
external_data_set	The data source from which the resource originates.  For example, if this workload information is stored in an external database.	String	No
external_data_reference	A unique identifier within the external data source.  For example, if this workload information is stored in an external database.	String	No
ip_address	IP address matching the IP lists to return. Supports partial matches	String	No
fqdn	IP lists matching FQDN. Supports partial matches	String	No
max_results	The maximum number of results you want to return when using the  GET method. The maximum limit for returned IP lists is 500.	Integer	No
name	Name of IP list(s) to return. Supports partial matches	String	No
ip_list_id	IP list ID (for [api_version][ip_list_href])	String	Yes
ip_list_attribute	GET: Allow filtering IP Lists that have an attribute assigned.  POST: Specify which attribute should be linked to an IP List.  PUT: Specify which attribute should be linked to an IP List.	String	No

## Non-Corporate Public IP Addresses

The API sec\_policy/rule\_coverage supports non-domain interfaces.

## Security Policy Rule Coverage

Security Principals Methods	HTTP	URI
Get Security Principals	POST	[api_version][org_href]/sec_policy/rule_coverage

## Query Parameters for non-corporate public IP addresses

The property network accepts network\_href to determine if the rule applies to a flow correctly.

Parameter	Description
source	Source entity. Specify labels,such as  "href": "/orgs/14/labels/42"  "href": "/orgs/14/labels/43"
destination	Destination entity  Specify an IP list, such as "href": "/orgs/14/sec_policy/active/ip_lists/14"
network	The network that the source and destination are on
services	Port and protocol, and optional process or Windows service names, of matching rules.

## Response Properties

In release 23.5, in sec\_policy\_rule\_coverage\_post\_response, a new array of rule\_edges was added, which provides a list with a placeholder for each requested source and destination pair.

The previous object rules is replaced with a reference to "\$ref": "#/definitions/rule\_href\_mapping", and the previous array edges is replaced with a reference to "\$ref": "#/definitions/rule\_edges".

```
"rule_edges": {
    "type": "array",
    "description": "A list with a placeholder for each requested source and destination pair",
    "items": {
        "type": "array",
        "description": "A list with with a placeholder for each requested service
                        (per source and destination pair)",
        "items": {
            "type": "array",
            "description": "A list of indexes of matching rules
                            (for each service per source and destination
pair)",
            "items": {
                "type": "string",
                "pattern": "^[0-9]+$"
            }
        }
    }
}
```

Before the release 23.5, the response was as follows:

Parameter	Description
rules	The rules returned in the API response are rules with network_type (such as non-brn) that apply to the given input, such as:  "0": { "href": "/orgs/14/sec_policy/draft/rule_sets/21/sec_rules/220" },  "1": { "href": "/orgs/14/sec_policy/draft/rule_sets/21/sec_rules/223" },  "2": { "href": "/orgs/14/sec_policy/draft/rule_sets/21/sec_rules/237" }
edges	[["0", "1", "2"]]

## PCE Management

As an Illumio administrator, use the APIs listed in this chapter to manage the Policy Compute Engine (PCE).

You can manage many aspects of the PCE through APIs, from authentication and passwords to PCE health.

### Product Version

This API returns the current version of the PCE software.

URI to Get Product Version

```
GET [api_version]/product_version
```

Curl Command to Get Product Version

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/product_version -H "Accept: application/json" -u $KEY:$TOKEN
```

Example Response

The response body has a format similar to this example:

```
{
  "version": "19.3.0",
  "build": 12864,
  "long_display": "19.3.0-12864",
  "short_display": "19.3.0"
}
```

## Authentication Settings

This Public Experimental API gets or updates the authentication settings for the login domain (organization).

These new APIs with the included `saml_configs` setting provide customers an option to sign authN requests.

## API Methods for Authentication Settings

HTTP	URI	Functionality
GET	<code>[api_version]/authentication_settings</code>	Get authentication settings
PUT	<code>[api_version]/authentication_settings</code>	Update authentication settings
GET	<code>[api_version]/authentication_settings/ saml_configs</code>	Gets all SAML configurations where <code>any_org_owner</code> is authorized to use it. The response now includes the PCE signing certificates that IdP will use for the SAML authN request signature validation.
GET	<code>[api_version]/authentication_settings/ saml_configs/:uuid</code>	Get the specified SAML configuration. The response now includes the PCE signing certificates that IdP will use for the SAML authN request signature validation.
PUT	<code>[api_version]/authentication_settings/ saml_config/:uuid</code>	Update the specified SAML configuration. API has been enhanced to enable/disable the signing of a SAML authN request.
POST	<code>[api_version]/authentication_settings/ saml_configs/:uuid/pce_signing_cert</code>	Generate a new certificate for signing SAML AuthN requests.

## Authentication Settings Reference

This topic covers examples of authentication settings.

### Examples

#### Get Authentication Settings

Curl Command to Get Authentication Settings

The `org/:org_id/` path parameter is not specified in this command.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/
authentication_settings -H "Accept: application/json" -u $KEY:$TOKEN
```

Example Default Response

200 OK

```
{ "authentication_type": "Local" }
```

## Update Authentication Settings

Curl Command to Update Authentication Settings

The `org/:org_id/` path parameter is not specified in this command.

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/
authentication_settings/password_policy -H "Content-Type: application/json"
-d '{"authentication_settings": "SAML"}' -u $KEY:$TOKEN
```

Request Properties

Parameter	Description	Type	Required
Local	Local authentication.	String	No
SAML	Authentication with SAML.	String	No

Example Request Body

```
{"authentication_settings": "SAML"}
```

## Password Policy

This Public Experimental API gets or updates the domain password policy.

A default password policy is created automatically when a new login domain (organization) is created. There is only one password policy per login domain, so the same password policy applies to all users.

## Password Policy Methods

Functionality	HTTP	URI
Get the password policy.	GET	[api_version]/authentication_settings/password_policy
Update the password policy.	PUT	[api_version]/authentication_settings/password_policy

## Password Policy Reference

This topic covers properties, parameters, and examples of password policy.

Curl Command Get the Password Policy

The `org/:org_id/` path parameter is not specified in this command.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/
authentication_services/password_policy -H "Accept: application/json" -u
$KEY:$TOKEN
```

Example Default Response: 200 OK

```
{  
  "require_type_number": true,  
  "require_type_lowercase": true,  
  "require_type_uppercase": true,  
  "require_type_symbol": false,  
  "min_characters_per_type": 1,  
  "min_length": 8,  
  "min_changed_characters": 1,  
  "history_count": 1,  
  "expire_time_days": 0,  
  "updated_at": "2019-09-20T03:40:00Z",  
  "updated_by": null  
}
```

## Parameters

Parameter	Description	Type	Req
require_type_number	If true, the password must contain a numerical digit.	Boolean	Yes
require_type_lowercase	If true, the password must contain a lowercase letter.	Boolean	Yes
require_type_uppercase	If true, the password must contain an uppercase letter.	Boolean	Yes
require_type_symbol	If true, the password must contain a symbol, for example:  ! @ # \$ % ^ * ? \u0026 \u003c \u003e	Boolean	Yes
min_characters_per_type	Minimum number of characters for each character type.	Integer	Yes
min_length	Minimum password length.	Integer	Yes
min_changed_characters	Minimum number of changed characters for a new password.  Minimum: 1  Maximum: 4	Integer	Yes
history_count	The number of old passwords to remember.  Minimum: 1  Maximum: 24	Integer	Yes
expire_time_days	Number of days until the password expires.  A value of 0 (zero) means the password never expires.  Minimum: 0  Maximum: 99	Integer	Yes
updated_at	RFC-3339 date-time timestamp of when the password was last updated and automatically recorded by the system.	date-time String	Yes
updated_by	The username of the person who last updated this is automatically recorded by the system.	String	Yes

## Update Password Policy

Curl Command Update the Password Policy

The org/:org\_id/ path parameter is not specified in this command.

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/authentication_services/password_policy -H "Content-Type: application/json" -u $KEY:$TOKEN -d '{"require_type_symbol": true, "expire_time_days": 90}'
```

At least three of the four available character types must be true; otherwise, a 406 Not Acceptable error message will be returned.\*

## Parameters

Parameter	Description	Type	Re- quired
require_type_number	If true, the password must contain a numerical digit.	Boolean	*
require_type_lowercase	If true, the password must contain a lowercase letter.	Boolean	*
require_type_uppercase	If true, the password must contain an uppercase letter.	Boolean	*
require_type_symbol	If true, the password must contain a symbol, for example:  ! @ # \$ % ^ * ? \u0026 \u003c \u003e	Boolean	*
min_characters_per_type	Minimum number of characters for each character type.	Integer	No
min_length	Minimum password length.	Integer	No
min_changed_characters	Minimum number of changed characters for new passwords.  Minimum: 1  Maximum: 4	Integer	No
history_count	The number of old passwords to remember.  Minimum: 1  Maximum: 24	Integer	No
expire_time_days	Number of days the password expires.  A value of 0 (zero) means the password never expires.  Minimum: 0  Maximum: 99	Integer	No

## Example Request Body

Only the parameters to change must be included in the request body.

```
{
  "require_type_number": true,
  "require_type_lowercase": true,
  "require_type_uppercase": true,
  "require_type_symbol": true,
  "min_characters_per_type": 1,
```

```

    "min_length": 8,
    "min_changed_characters": 1,
    "history_count": 1,
    "expire_time_days": 90
}

```

## Supercluster Leader

The Supercluster Leader Public Stable API method checks each PCE in a Supercluster and indicates which PCE is the leader.

### Supercluster Leader API

This call is typically made by a customer's Global Server Load Balancer (GSLB) to monitor the health of the leader.

Possible results:

- If the API returns an HTTP 202 response, the cluster where you made this call is the leader.
- If the API returns an HTTP 404 response, then the cluster where you made this call is a member.

### Get Supercluster Leader

GET [api\_version]/supercluster/leader

Curl Command Get Supercluster Leader

```
curl -X GET 'https://pce.my-company.com:8443/api/v2/supercluster/leader' -i
```

The command is followed by a 200 response:

```

HTTP/1.1 200 OK
Date:Thu, 08 Apr 2021 19:21:59 GMT
Content-Type:application/octet-stream
Content-Length:0
Last-Modified: Thu, 08 Apr 2019 21:55 GMT
ETag: "606f57d3-0"
Accept-Ranges: bytes
Cache-Control:private, must-revalidate
X-Frame-Options: DENY
X-XSS-Protection:1; mode=block
X-Content-Type-Options:nosniff

```

## PCE Health

The Public Stable Health Check API displays health information about a 4X2 Supercluster or a PCE virtual appliance.

**NOTE**

This API is only available for Illumio Core PCE installed on-premises and is not available for Illumio Cloud customers.

**About PCE Health API**

With this API, you can see the following health information:

- How long has the PCE been running, what is its runlevel, and overall health (normal, warning, or error).
- Each node hostname, IP address, uptime, runlevel, and whether the PCE software runs properly.
- Each node type (core or data) indicates whether the data node is the database replica or the primary database. The replication delay for the database replica is also displayed.
- Information about PCE service alerts, such as the number of degraded or failed services in the cluster, so you can see where service failures have occurred.

**PCE Health API Method**

Functionality	HTTP	URI
Check the health of the PCE.	GET	[api_version]/health

**Check PCE Health**

URI to Check PCE Health

GET [api\_version]/health

**PCE Health Reference**

This topic covers properties, parameters, and examples for PCE health.

**Check PCE Health**

Curl Command

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/health -H 'Accept: application/json' -u $KEY:'TOKEN'
```

## Properties

Property	Description	Type
status	<p>Current health status of the PCE. Possible values:</p> <ul style="list-style-type: none"> <li>• <b>normal</b>: When a PCE's health is in a normal state, it means:           <ul style="list-style-type: none"> <li>• All required services are running.</li> <li>• All nodes are running.</li> <li>• CPU usage of all nodes is less than 95%.</li> <li>• Memory usage of all nodes is less than 95%.</li> <li>• Disk usage of all nodes is less than 95%.</li> <li>• Database replication lag is less than or equal to 30 seconds.</li> </ul> </li> <li>• <b>warning</b>: When PCE health is in a warning state, it means:           <ul style="list-style-type: none"> <li>• One or more nodes are unreachable.</li> <li>• One or more optional services are missing, or one or more required services have been degraded.</li> <li>• The CPU usage of any node is greater than or equal to 95%.</li> <li>• Memory usage of any node is greater than or equal to 95%.</li> <li>• Disk usage of any node is greater than or equal to 95%.</li> <li>• Database replication lag is greater than 30 seconds.</li> </ul> </li> <li>• <b>critical</b>: A PCE is considered to be in a critical state when one or more required services are missing. If a PCE enters a critical state, it might not be possible to authenticate to the PCE or get an API response, depending on which services are missing from the PCE.</li> </ul>	String
type	<p>The type of PCE:</p> <ul style="list-style-type: none"> <li>• <b>standalone</b>: Indicates that this PCE is an on-premises 2x2 or 4x2 PCE cluster. Or one of the following types:</li> <li>• <b>leader</b>: Indicates that this PCE is the leader of a Supercluster.</li> <li>• <b>member</b>: Indicates that this PCE is a member of a Supercluster.</li> </ul>	String
fqdn	The fully qualified domain name (FQDN) of the PCE.	String
available_seconds	The length of time that this PCE has been available is measured in seconds.	Number
notifications	<p>Health warnings related to the PCE, which contain the following properties:</p> <ul style="list-style-type: none"> <li>• <b>status</b>: Severity status of this notification. Possible values include: <code>normal</code>, <code>warning</code>, or <code>critical</code>.</li> <li>• <b>token</b>: Description of the notification.</li> <li>• <b>message</b>: Notification message.</li> </ul>	
listen_only_mode_enabled_at	<p>Indicates when listen-only mode was enabled for this PCE.</p> <p>For information about enabling or disabling listen-only mode for a PCE, see PCE Administration Guide.</p>	String
nodes	<p>The nodes that comprise your PCE cluster.</p> <p>For each node of your PCE, this API call returns the following properties:</p> <ul style="list-style-type: none"> <li>• <b>hostname</b>: The node hostname.</li> <li>• <b>ip_address</b>: The node IP address.</li> <li>• <b>runlevel</b>: (Number) The current runlevel of the PCE software on the node. For more information about runlevels and their usage, see PCE Administration Guide.</li> <li>• <b>uptime_seconds</b>: Seconds since this node has been restarted.</li> <li>• <b>cpu</b>: Percentage of the node CPU being used. Includes the following two sub-properties:</li> </ul>	String

Property	Description	Type
	<ul style="list-style-type: none"> <li>• <b>status:</b> Either <code>normal</code>, <code>warning</code>, or <code>critical</code>.</li> <li>• <b>percent:</b> (Number) Percentage of the node CPU being used.</li> <li>• <b>disk:</b> Percentage of the node's disk that is being used. Includes the following two sub-properties:           <ul style="list-style-type: none"> <li>• <b>status:</b> Either <code>normal</code>, <code>warning</code>, or <code>critical</code>.</li> <li>• <b>percent:</b> (Number) Percentage of the node disk being used.</li> </ul> </li> <li>• <b>memory:</b> Percentage of the node's memory that is being used. Includes the following two sub-properties:           <ul style="list-style-type: none"> <li>• <b>status:</b> Either <code>normal</code>, <code>warning</code>, or <code>critical</code>.</li> <li>• <b>percent:</b> (Number) Percentage of the node memory being used.</li> </ul> </li> <li>• <b>services:</b> The status of all PCE services running on the node. Possible statuses for PCE services include:           <ul style="list-style-type: none"> <li>• <code>running</code>: The service is fully running and operational.</li> <li>• <code>not running</code>: The service has stopped running.</li> <li>• <code>partial</code>: The service is running but in a partial state.</li> <li>• <code>optional</code></li> <li>• <code>unknown</code></li> </ul> </li> <li>• <b>generated_at:</b> Timestamp when this information was generated.</li> </ul>	
<code>network</code>	<b>PCE 2x2 or 4x2 Deployment</b>	Array

For a PCE 2x2 or 4x2 deployment, the `network` property provides latency information between the database primary and database replica data nodes in your PCE for policy and traffic data.

This property also indicates which data node in your PCE is the primary database and which is the database replica.

This type of database replication is called `intracluster` in the REST API.

Sub-properties include:

`replication`: The category of properties that provides database replication latency information for a PCE cluster. (For a PCE Supercluster, this information is provided for each PCE in the Supercluster.)

- **type:** Type of replication. `intracluster` for a PCE 2x2 or 4x2 deployment.
- **details:** Includes the following properties:
  - `database_name`: Either `agent` for policy data or `traffic` for traffic data.
  - `primary_fqdn`: The FQDN of the database primarynode.
  - `replica_fqdn`: FQDN of the replica database node.
- **value:** The amount of replication lag between the primary and database replica for both policy and traffic data.
  - **status:** Either `normal`, `warning`, or `critical`.
  - **lag\_seconds:** The amount of lag measured in seconds between the primary and replica databases for both policy and traffic data.

#### Supercluster Deployment

If you have deployed a PCE Supercluster, the PCE health call also returns information about the database replication between the PCE you are currently logged into and all other PCEs in the Supercluster.

In a Supercluster deployment, the security policy provisioned on the leader is replicated to all other PCEs in the Supercluster. Additionally, all PCEs in the Supercluster (leader and members) replicate copies of each workload's context, such as IP addresses, to all other PCEs in the Supercluster.

This other type of database replication for a Supercluster is called `intercluster` in the REST API, and information is provided for all PCEs in the Supercluster.

Property	Description	Type
generated_at	<p>Properties include:</p> <p>replication: The category of properties that provide database replication latency information for a PCE cluster:</p> <ul style="list-style-type: none"> <li>• type: Type of replication. <code>intercluster</code> for a PCE Supercluster deployment.</li> <li>• details: Includes the following properties: <ul style="list-style-type: none"> <li>• fqdn: The FQDN of the primary database of the other PCEs listed in this section.</li> </ul> </li> <li>• value: The amount of replication lag between the PCE you are logged into and one of the other PCEs in the Supercluster.</li> <li>• status: Either <code>normal</code>, <code>warning</code>, or <code>critical</code>.</li> <li>• lag_seconds: The amount of lag measured in seconds between the PCE you are logged into and the other PCE listed in this section.</li> </ul> <p>generated_at</p> <p>The timestamp of when the information was generated.</p>	String

## PCE Health Response

Example response returned from the PCE Health API.

```
[
  {
    "status": "normal",
    "type": "standalone",
    "fqdn": "pce.mycompany.com",
    "available_seconds": 84133,
    "notifications": [],
    "listen_only_mode_enabled_at": null,
    "nodes": [
      {
        "hostname": "pce_core1.mycompany.com",
        "ip_address": "192.0.1.0",
        "type": "core",
        "runlevel": 5,
        "uptime_seconds": 2051301,
        "cpu": {
          "status": "normal",
          "percent": 7
        },
        "disk": [
          {
            "location": "disk",
            "value": {
              "status": "normal",
              "percent": 17
            }
          }
        ],
        "memory": {
          "status": "warning",
          "percent": 85
        },
        "services": {
          ...
        }
      }
    ]
  }
]
```

```

    "status": "normal",
    "services": {
        "running": [
            "agent_background_worker_service",
            "agent_service",
            "agent_traffic_service",
            "auditabile_events_service",
            "collector_service",
            "ev_service",
            "executor_service",
            "fluentd_source_service",
            "login_service",
            "memcached",
            "node_monitor",
            "search_index_service",
            "server_load_balancer",
            "service_discovery_server",
            "traffic_worker_service",
            "web_server",
        ]
    }
},
"generated_at": "2020-03-03T19:38:52+00:00"
},
}
],
"network": {
    "replication": [
        {
            "type": "intracluster",
            "details": {
                "database_name": "agent",
                "primary_fqdn": "bkhoram-qa-6node-v0-pce-1-dbase0"
            },
            "value": {
                "status": "normal",
                "lag_seconds": 0
            }
        },
        {
            "type": "intracluster",
            "details": {
                "database_name": "traffic",
                "primary_fqdn": "bkhoram-qa-6node-v0-pce-1-dbase0"
            },
            "value": {
                "status": "normal",
                "lag_seconds": 0
            }
        }
    ]
},
"generated_at": "2020-03-03T19:38:52+00:00"
}
]

```

## Node Availability

This Public Stable API method allows the Load Balancer to monitor the health of the PCE core nodes in a 2x2 or 4x2 cluster. This feature is only available if the PCE is deployed as software in your data center.



### NOTE

This API call does not require authentication.

URI to Check Node Availability

```
GET [api_version]/node_available
```

## Support Bundle Requests

Several APIs have been introduced to provide a mechanism to generate a support bundle on each node, including a time range and possibly additional options.

## API Methods

Functionality	HTTP	URI
Return the collection of PCE support bundle requests:	GET	[api_version][org_href]/support_bundle_requests
Return a specific PCE support bundle request:	GET	[api_version][org_href]/support_bundle_requests/:uuid
Create a PCE support bundle request.	POST	[api_version][org_href]/support_bundle_requests
Delete the PCE support bundle request.	DELETE	[api_version][org_href]/support_bundle_requests/:uuid

## Health Check from a Load Balancer

In a production deployment, customers run health checks from a Load Balancer. The actual request syntax varies, but here is a sample command for Infoblox:

```
GET /api/v2/node_available HTTP/1.1
```

## Node Availability Reference

This topic covers parameters, properties, and examples of node availability.

## Examples

Check Node Availability

**-X GET** and authentication are not required for this method. The curl **-v** flag provides verbose output.

```
curl -v https://pce.my-company.com:8443/api/v2/node_available
```

Or, you can use `-i -X GET` to return a `200 OK` status if the node is available:

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/node_available
```

Returns `200 OK` if the core node is healthy and can see at least one service running in the PCE cluster.

Otherwise, it returns a `404` error.

For example, if the PCE is healthy and accessible, the response is `200 OK`.

## Parameters for support bundle requests

Property	Description	Type	Required
<code>org_id</code>	Organization ID	Integer	Yes
<code>ending_at</code>	Time at which to exclude entries	String	No
<code>include_logs</code>	Set to true if logs are to be included	Boolean	No
<code>starting_at</code>	Start date for log filtering	String	No
<code>requested_at</code>	A time support bundle was requested.	string(date-time)	Yes

## Properties for support bundle requests

Property	Description	Type	Required
href	URI of this request  Reference to common/href_object.schema.json		Yes
name	The name of the support bundle	String	Yes
download_url	URI of the associated report file  Reference to common/href_object.schema.json		Yes
requested_at	A time support bundle was requested.	String (date-time)	Yes
completed_at	Time support bundle completed.	String, Null (date-time)	Yes
status	A status annunciator indicating the state of this request	String	Yes
include_logs	Set to true if logs are to be included	Boolean	Yes
starting_at	(GET, POST) Start date for log filtering	String, Null (date-time)	Yes
ending_at	End date for log filtering	String, Null (date-time)	Yes

Example for POST

```
{
  "include_logs": true,
  "starting_at": null,
  "ending_at": null
}
```

## No Op Public Stable API

The No Op Public Stable API calls the PCE without performing any operations. This API is used to check connectivity to and from the PCE.

Use this API to verify that new authentication credentials are working after creating a new set of keys.

URI for No Op

GET [api\_version]/noop

Curl Command for No Op

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/noop -H "Accept: application/json" -u $KEY:'TOKEN'
```

## Events

This Public Experimental API gets a collection of events or an individual event.



### NOTE

Use this Events API instead of Audit Events.

Events include logging a user in or out of the PCE, granting a user a role, pairing or unpairing a workload, and creating a label, ruleset, or IP list.

## Event Types

For a complete list of JSON events, descriptions, CEF/LEEF success events, and CEF/LEEF failure events, see [List of Event Types](#)

## Event API Methods

Functionality	HTTP	URI
Get a collection of events.	GET	[api_version][org_href]/events
Get an individual event.	GET	[api_version][event_href]

### Get Events

This API gets a collection of events or a specific event identified by an event ID (in the form of a UUID).

### Get Events Collection

When getting a collection of events, be aware of the following caveats:

- Use the `max_results` query parameter to increase the maximum number of events returned.
- The largest value accepted for `max_results` is 10000. To return more than 10000 events, use Asynchronous GET Collection.

URI to Get a Collection of Events

```
GET [api_version][org_href]/events
```

URI to Get an Individual Event

```
GET [api_version][event_href]
```

## Events Reference

This topic covers properties, parameters, and examples of events.

## Parameters

Parameter	Description	Type
xorg_id	Organization ID in which the event occurred.	Integer
created_by	<p>Information about the person, agent, or system that created the event.</p> <p>Created by <i>system</i>:</p> <ul style="list-style-type: none"> <li>• <b>system</b>: Appears only if the event was generated by the PCE.</li> </ul> <p>Created by <i>user</i> properties:</p> <ul style="list-style-type: none"> <li>• <b>href</b>: URI of the user who created the event.</li> <li>• <b>username</b>: The user name (usually formatted as an e-mail address).</li> </ul> <p>Created by <i>workload</i> properties:</p> <ul style="list-style-type: none"> <li>• <b>href</b>: URI of the agent on the workload that initiated the event.</li> <li>• <b>hostname</b>: The hostname of the workload.</li> </ul>	String
event_type	<p>The type of the event specified by the <code>event_type</code> query parameter if given.</p> <p>If no query parameters are given, all event types are returned.</p> <p>See the response properties table below for types of events returned from a GET call.</p>	String
max_results	<p>Maximum number of events to return.</p> <p>The default is 100, and the maximum is 10000.</p>	Integer
severity	<p>Severity level of the events retrieved. Values include:</p> <ul style="list-style-type: none"> <li>• Warning (<code>warning</code>): A warning that the event is likely to occur if action is not taken.</li> <li>• Error (<code>err</code>)</li> <li>• Information (<code>info</code>): Normal operational messages can be harvested for reporting and measuring throughput, such as user pairing or unpairing workloads in the PCE web console.</li> </ul>	String
status	Status of the event, either <code>success</code> or <code>failure</code> .	String
time-stamp[gte]	Event start timestamp in RFC 3339 format.	String
time-stamp[lte]	Event end timestamp in RFC 3339 format.	String

## Properties

Parameter	Description	Type
event_type	The type of the event specified by the <code>event_type</code> query parameter if given.  If no query parameters are given, all event types are returned.  See the response properties table below for types of events returned from a GET call.	String
status	Status of the event: usually a mapping of <code>api_status_code</code> to a generic result string; nil if no action.  For presentation purposes only.	String
severity	Severity level of the events retrieved. Values include:  <ul style="list-style-type: none"> <li>• Warning (<code>warning</code>): A warning that the event is likely to occur if action is not taken.</li> <li>• Error (<code>err</code>)</li> <li>• Information (<code>info</code>): Normal operational messages can be harvested for reporting and measuring throughput, such as user pairing or unpairing workloads in the PCE web console.</li> </ul>	String
created_by	Information about the person, agent, or system that created the event.  Created by <i>system</i> :  <ul style="list-style-type: none"> <li>• <code>System</code>: This appears only if the PCE generated the event.</li> </ul> Created by <i>user</i> properties:  <ul style="list-style-type: none"> <li>• <code>href</code>: URI of the user who created the event.</li> <li>• <code>username</code>: The user name (usually formatted as an e-mail address).</li> </ul> Created by <i>workload</i> properties:  <ul style="list-style-type: none"> <li>• <code>href</code>: URI of the agent on the workload that initiated the event.</li> <li>• <code>hostname</code>: The hostname of the workload.</li> </ul>	String

## Examples

### Curl Command to Get an Event

You need the ID of the system event you want to get, which is the number at the end of its HREF path property: "/2/events/68632".

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/events/12345
-H "Accept: application/json" -u $KEY:$TOKEN
```

### Curl Command Get Event Collection

In this example, only two events are returned because of `max_events=2`.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/events?
max_results=2 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Example Response

```
[  
  {  
    "href": "/orgs/1/events/xxxxxxxx-5f59-46ab-8f18-xxxxxxxxxx",  
    "timestamp": "2019-09-03T01:xx:xx.xxxZ",  
    "pce_fqdn": "pce.my-company.com",  
    "created_by": {  
      "agent": {  
        "href": "/orgs/1/agents/xxx",  
        "hostname": "xxx-xxxxx-xxxx"  
      }  
    },  
    "event_type": "agent.clone_detected",  
    "status": null,  
    "severity": "info",  
    "action": null,  
    "resource_changes": [ ],  
    "notifications": [  
      {  
        "uuid": "xxxxxxxx-e04b-43bc-a64a-xxxxxxxxxx",  
        "notification_type": "agent.clone_detected",  
        "info": {  
          "agent": {  
            "href": "/orgs/1/agents/xxx",  
            "name": null,  
            "hostname": "xxx-xxxxx-xxxx"  
          }  
        }  
      }  
    ]  
  },  
  {  
    "href": "/orgs/1/events/xxxxxxxx-60a2-4db4-b0f4-xxxxxxxxxx",  
    "timestamp": "2019-09-03T0x:xx:xx.xxxZ",  
    "pce_fqdn": "pce.my-company.com",  
    "created_by": {  
      "agent": {  
        "href": "/orgs/1/agents/xxx",  
        "hostname": "xxx-xxxxx-xxxx"  
      }  
    },  
    ]  
  }  
]
```

## Organization Settings

For Organization Settings examples, see [Organization Settings Reference \[147\]](#).

### Enabling clone detection

Users sometimes need to disable clone detection as a safety valve. For example, if a customer has workloads in a particular environment that behave unexpectedly, they might end up with runaway clones being activated.

Functionality	HTTP	Properties Added	URI
This is for VEN or URI to fetch the current clone detection settings.	GET	automatic_clone_reactivation	/api/v2/orgs/:xorg_id/ settings
Authorization is for the org user and the interservice.		clone_detection_enabled	
This is for the org admin to set new clone detection settings.	PUT	automatic_clone_reactivation	/api/v2/orgs/:xorg_id/ settings
Authorization is for the org admin.		clone_detection_enabled	

Two properties have been added to the schemas `settings_get` and `settings_put`:

- `clone_detection_enabled`
- `automatic_clone_reactivation`

Depending on whether they are added to the PUT or GET method, they require different types of authorization: org admin for PUT and org user or interservice authorization for GET.

- If `automatic_clone_reactivation` is disabled, there is no automatic clone reactivation.

## Organization Settings Reference

This topic covers examples of organizational settings.

### Examples

Example JSON Response Body for Get Events Settings

```
{
  "audit_event_retention_seconds": 180,
  "audit_event_min_severity": "informational",
  "format": "JSON"
}
```

### Update Events Settings

Example JSON Request Body for Update Events

```
{
  "audit_event_retention_seconds": 90,
  "audit_event_min_severity": "informational"
}
```

Example JSON Response Body with Local and Remote Syslog Location Information

```
[
  {
    "href": "/api/v2/orgs/1/settings/syslog/destinations/xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx",
    "pce_scope": [ "remote-my-company0.com", "remote-my-company1.com" ],
    "type": "remote_syslog",
    "description": "remotesyslog",
```

```

    "audit_event_logger": {
        "configuration_event_included": true,
        "system_event_included": false,
        "min_severity": "warning"
    },
    "traffic_event_logger": {
        "traffic_flow_allowed_event_included": true,
        "traffic_flow_potentially_blocked_event_included": true,
        "traffic_flow_blocked_event_included": true
    },
    "node_status_logger": {
        "node_status_included": true
    },
    "remote_syslog": {
        "address" : "my-company-20.com",
        "port" : 12345,
        "protocol" : 6,
        "tls_enabled" : false,
        "tls_verify_cert" : false
    }
}
]

```

Example JSON Response Body with Remote Syslog Location Information

```
{
    "href": "/api/v2/orgs/1/settings/syslog/destinations/xxxxxxxx-xxxx-xxxx-
xxxxxxxxxxxx",
    "pce_scope": [ "remote-my-company0.com", "remote-my-company1.com" ],
    "type": "remote_syslog",
    "description": "remotesyslog",
    "audit_event_logger": {
        "configuration_event_included": true,
        "system_event_included": false,
        "min_severity": "warning"
    },
    "traffic_event_logger": {
        "traffic_flow_allowed_event_included": true,
        "traffic_flow_potentially_blocked_event_included": true,
        "traffic_flow_blocked_event_included": true
    },
    "node_status_logger": {
        "node_status_included": true
    },
    "remote_syslog": {
        "address" : "my-company-20.com",
        "port" : 12345,
        "protocol" : 6,
        "tls_enabled" : false,
        "tls_verify_cert" : false
    }
}
```

Example JSON Request Body to Create a Remote Syslog Destination

```
{
    "pce_scope": [ "my-company0.com", "my-company1.com", "my-
company2.com" ],
    "type": "remote_syslog",
    "description": "remote syslog",
    "audit_event_logger": {
        "configuration_event_included": true,
        "system_event_included": false,
        "min_severity": "warning"
    },
    "traffic_event_logger": {
        "traffic_flow_allowed_event_included": true,
        "traffic_flow_potentially_blocked_event_included": true,
        "traffic_flow_blocked_event_included": true
    },
    "node_status_logger": {
        "node_status_included": true
    },
    "remote_syslog": {
        "address" : "my-company-20.com",
        "port" : 12345,
        "protocol" : 6,
        "tls_enabled" : false,
        "tls_verify_cert" : false
    }
}
```

Example JSON Request Body to Update a Syslog Destination

```
{
    "href": "/api/v2/orgs/1/settings/syslog/destinations/xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx",
    "pce_scope": [ "my-company0.com", "my-company1.com", "my-
company2.com" ],
    "type": "remote_syslog",
    "description": "localhost syslog",
    "audit_event_logger": {
        "configuration_event_included": true,
        "system_event_included": true,
        "min_severity": "informational"
    },
    "traffic_event_logger": {
        "traffic_flow_allowed_event_included": true,
        "traffic_flow_potentially_blocked_event_included": true,
        "traffic_flow_blocked_event_included": true
    },
    "node_status_logger": {
        "node_status_included": false
    },
    "remote_syslog": {
        "address" : "my-company-20.com",
        "port" : 67890,
        "protocol" : 6,
        "tls_enabled" : false,
        "tls_verify_cert" : false
    }
}
```

```

    }
}
```

### **Examples for Enabling clone detection:**

(Only new fields in the schema `settings_get` are shown)

```
"clone_detection_enabled": {
    "description": "When true, clone detection is done for this org",
    "type": "boolean"
},
"automatic_clone_reactivation": {
    "description": "When true, automatic clone reactivation should be done
on clone detection for this org",
    "type": "string",
    "enum": [ "disabled", "windows_domain_joined_workloads_only" ]
}
```

Example reply (only for "clone\_detection\_enabled")

```
{
    "clone_detection_enabled": true,
    "automatic_clone_reactivation": "windows_domain_joined_workloads_only"
}
```

## **Organization Settings**

For Organization Settings examples, see [Organization Settings Reference \[147\]](#).

### **Obtain Permissions from the Census or the Local Database**

A new property `use_census_permissions` was added to the schemas `settings_get` and `settings_put` to indicate whether PCE will obtain permissions from the Census API or a local database.

The flag is set to `TRUE` to get permissions from the census and to `FALSE` for the local database.

Functionality	HTTP	Properties Added	URI
This flag indicates whether the PCE will obtain permissions from the census or a local database.	GET PUT	<code>use_census_permissions</code>	<code>/api/v2/orgs/:xorg_id/ settings</code>

## **Container Clusters**

The Illumio Core uses three groups of APIs to manage container clusters:

- Container Cluster API (GET, POST, PUT, DELETE)

- Container Cluster Workload Profiles API (GET, POST, PUT, DELETE)
- Container Cluster Service Backend API (GET)

## Container Cluster API

A container cluster object stores all the information about a Kubernetes cluster in the PCE by collecting telemetry from Kubelink. Each Kubernetes cluster maps to one container cluster object in the PCE.

Use these methods to get, create, update, or delete container clusters:

Functionality	HTTP	URI
Get the list of container clusters.	GET	[api_version][org_href]/container_clusters
Get the specified container cluster.	GET	[api_version][org_href]/container_clusters/:uuid
Create a container cluster.	POST	[api_version][org_href]/container_clusters
Update the specified container cluster.	PUT	[api_version][org_href]/container_clusters/:uuid
Delete the specified container cluster.	DELETE	[api_version][org_href]/container_clusters/:uuid

## Container Cluster Workload Profiles

When you install an Illumio VEN on a container cluster, all pods are unmanaged or not visible in the PCE. However, Kubelink reports all namespaces on the container clusters, which are then visible via the Container Workload Profiles API.

Each container workload profile maps to a Kubernetes namespace and can be managed or unmanaged. The default state for a profile is unmanaged.

Use these methods to get, create, update, or delete container cluster workload profiles:

Functionality	HTTP	URI
Get the list of container cluster workload profiles.	GET	GET /orgs/:xorg_id/container_clusters/:container_cluster_id/container_workload_profiles
Create container cluster workload profiles.	POST	POST /orgs/:xorg_id/container_clusters/:container_cluster_id/container_workload_profiles
Update the specified container cluster workload profile.	PUT	PUT /orgs/:xorg_id/container_clusters/:container_cluster_id/container_workload_profiles/:container_workload_profile_id
Supports the UI feature for bulk update of container workload profiles.	PUT	PUT /orgs/:xorg_id/container_clusters/:container_cluster_id/container_workload_profiles_update
Delete the specified container cluster workload profile.	DELETE	DELETE /orgs/:xorg_id/container_clusters/:container_cluster_id/container_workload_profiles/:container_workload_profile_id

## Label Restrictions

Kubernetes pods and services running in a namespace (Kubernetes) or project (OpenShift) must be labeled (RAEL) to be included in the policy within Illumio Core. The container workload profile defines how labels will be assigned to pods and services within a namespace.

Using annotations, Illumio labels can be statically assigned from the PCE or defined in the Kubernetes manifest files. For each label key (RAEL), the PCE administrator can define four options:

1. No label will be assigned.
2. One label will be assigned by PCE.
3. Using annotations, Kubernetes can assign a restricted list of labels. Label restrictions prevent unauthorized use of Illumio labels by Kubernetes platform administrators and ensure that labels inherit the correct policies.
4. Any label can be assigned by Kubernetes.

You can set role labels for the following APIs:

- PUT /api/v2/orgs/:xorg\_id/container\_clusters/<:cluster\_id>/ container\_workload\_profiles
- POST /api/v2/orgs/:xorg\_id/container\_clusters/<:cluster\_id>/ container\_workload\_profiles

## Label Assignment Configuration

To clear the label assignment option and go back to the default option (any labels passed at runtime using Kubernetes annotations will be allowed), two options:

Option 1: explicit statement

```
{
  "labels": [
    { "key": "role", "restriction": [] }
  ]
}
```

Option 2: empty payload

```
{
  "labels": []
}
```

## Backend Services Associated with Container Clusters

The Illumio policy model represents Kubernetes services as virtual services. For these services, Kubelink creates virtual services in the PCE and reports the list of Replication Controllers, Daemon Sets, and Replica Sets responsible for managing the pods that support them.

When a match between the Replication Controller and ReplicaSet managing a pod, the PCE creates a binding between the virtual service and the container workload.

The Service Backend matches a virtual service and an application type, such as Deployment or ReplicaSet.

Use this method to get the service backend:

Functionality	HTTP	URI
Get data about the service backend.	GET	GET /orgs/:xorg_id/container_clusters/:container_cluster_id/service_backends

## Kubernetes APIs

### Kubernetes Workload Endpoints

Customers must see the details of Kubernetes workloads in PCE to write policies and troubleshoot any issues.

Two new endpoints have been created for Kubernetes workloads:

- GET /api/v2/orgs/:xorg\_id/kubernetes\_workloads

This API lists all new Kubernetes Workloads in a separate tab/page with separate sorts and filters.

It contains required properties such as `name`, `kind`, `namespace`, as well as optional properties:

```
href, labels, enforcement_mode, visibility_level,
container_workload_profile, container_cluster,
security_policy_applied_at, security_policy_sync_state,
created_at, k8s_label, and k8s_annotations.
```

- GET /api/v2/orgs/:xorg\_id/kubernetes\_workloads

For this API, these changes have been made in release 23.5.0:

two arrays have been removed, `k8s_labels` and `sk8s_annotation`, and replaced with property `metadata`.

Href description has been changed from `URI` of the container workload, to `URI` of the kubernetes workload.

```
"metadata": {
    "$ref": "
        ./common/kubernetes_workloads_metadata.schema.json"
}
```

- GET /api/v2/orgs/:xorg\_id/kubernetes\_workloads/:kubernetes\_workload\_uuid

This API provides a detailed page for the specified Kubernetes workload with custom K8S attributes.

- common\_non\_empty\_label\_scopes.schema.json

This new common schema provides a collection of assigned list of labels. Minimum number is one.

- common\_kubernetes\_workloads\_metadata

The new common schema `kubernetes_workloads_metadata` was added in release 23.5.0, referenced from `kubernetes_workload_get`.

It provides Kubernetes properties, such as labels, annotations, and the UID of the external service.

### Container Clusters Reference

This topic covers properties, parameters, and examples for container clusters.

## Parameters

GET Method

Use the following required and optional parameters:

Parameter	Description	Type	Required
href	URI of the container cluster.	String	Yes
name	User-assigned name of the container cluster.	String	Yes
description	User-assigned description of the container cluster.	String	Yes
nodes		Array	No
machine_id	This parameter has the following property:  • pod_subnet: The pod subnet	Object	Yes
manager_type	Manager of the container cluster (and version).	String	No
network_type	Type of network.	String	No
last_connected	Date-time format.	String	No
online	Online: true/false.	Boolean	No
errors	The object error_type has the following properties:  • audit_event: • href • duplicate_ids • error_type	Object	No
		String	
		Array	
		String	
		String	
kubelink_version	Kubelink software version.	String	No
pce_fqdn	PCE FQDN is used for this container cluster only in the Super-cluster.	String	No
cluster_mode	The new property cluster_mode was added in 23.5.10 to describe the cluster mode for the container cluster. The default is legacy.	String	

POST and PUT methods

Use the following parameters:

Parameter	Description	Type	Required
name	User-assigned name of the cluster	String	Yes
description	User-assigned description of the cluster	String	No

## Curl Examples

Curl Command for GET

```
curl --request GET --url https://pce.my-company.com:8443/api/v2/orgs/1/container_clusters --header 'authorization: Basic YXBpXzE2YjBkYjI0MjJhZGNkYWU5OjA5ZmRjNjA4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzOTQ1ZGMxYzEwZGzlmZjM='
```

Example Response for GET

```
[
  {
    "href": "/orgs/1/container_clusters/445bfa9b-4de4-4c09-9705-496eb04b190f",
    "pce_fqdn": null,
    "name": "k8s2",
    "description": " ",
    "manager_type": "Kubernetes v1.16.2",
    "last_connected": "2019-10-28T22:48:31.228Z",
    "kubelink_version": "2.0.0-master.96e58b",
    "online": true,
    "nodes": [
      [
        {
          "name": "node1",
          "pod_subnet": "10.233.64.0/24"
        },
        {
          "name": "node2",
          "pod_subnet": "10.233.65.0/24"
        },
        {
          "name": "node3",
          "pod_subnet": "10.233.66.0/24"
        }
      ],
      "errors": []
    },
    {
      "href": "/orgs/1/container_clusters/ad678193-8e2f-402b-a864-4947dcc0c6d7",
      "pce_fqdn": null,
      "name": "Openshift 3.11",
      "description": " ",
      "manager_type": "Openshift v3.11.43",
      "last_connected": "2019-10-28T22:50:30.201Z",
      "kubelink_version": "1.0.0-master.a81280",
    }
  ]
]
```

```

"online":true,
"nodes":
[
{
  "name":"ip-172-31-19-198.us-west-2.compute.internal",
  "pod_subnet":"10.128.0.0/23"
},
{
  "name":"ip-172-31-20-168.us-west-2.compute.internal",
  "pod_subnet":"10.131.0.0/23"
},
{
  "name":"ip-172-31-22-56.us-west-2.compute.internal",
  "pod_subnet":"10.130.0.0/23"
},
{
  "name":"ip-172-31-27-241.us-west-2.compute.internal",
  "pod_subnet":"10.129.0.0/23"
}
],
"errors":[
},
{
  "href": "/orgs/1/container_clusters/bef57e90-97d4-4744-
a129-5d35aa12b21b",
  "pce_fqdn":null,
  "name":"k8s3 Cluster",
  "description":"Flannel Vx Lan",
  "manager_type":"Kubernetes v1.13.2",
  "last_connected":"2019-10-28T22:47:59.122Z",
  "kubelink_version":"EYE-60264",
  "online":true,
  "nodes":
  [
    {
      "name":"k8s3master",
      "pod_subnet":"10.244.0.0/24"
    },
    {
      "name":"k8s3minion1",
      "pod_subnet":"10.244.2.0/24"
    },
    {
      "name":"k8s3minion2",
      "pod_subnet":"10.244.1.0/24"
    }
  ],
  "errors":[
  },
  {
    "href": "/orgs/1/container_clusters/d7d62400-7650-4407-ae9b-71803dbb1324",
    "pce_fqdn":null,
    "name":"k8s1 v4",
    "description":"",
    "manager_type":"Kubernetes v1.12.4",

```

```

"last_connected": "2019-10-24T23:58:55.795Z",
"kubelink_version": "EYE-61567",
"online": false,
"nodes":
[
{
  {
    "name": "k8s1master",
    "pod_subnet": "10.244.0.0/24"
  },
  {
    "name": "k8s1minion1",
    "pod_subnet": "10.244.2.0/24"
  },
  {
    "name": "k8s1minion2",
    "pod_subnet": "10.244.1.0/24"
  }
],
"errors": []
}
]

```

Curl Example for POST

```

curl --request POST --url https://pce.my-company.com:8443/api/v2/orgs/1/
container_clusters --header 'authorization: Basic
jI0MjJhZGNkYWU5Oja5ZmRjNja4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzO
TQ1ZGMxYzEwZGJhZTg5NzlmZjM=' --header 'content-type: application/json' --
data '{"name": "test", "description": "test"}'

```

Curl Example for PUT

```

curl --request PUT --url https://pce.my-company.com:8443/api/v2/orgs/1/
container_clusters/1b851d4b-f22d-47be-b744-f3c2dca490a0 --header
'authorization: Basic
YXBpXzE2YjBkYjI0MjJhZGNkYWU5Oja5ZmRjNja4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdk
MWNlMDAzZmMzOTQ1ZGMxYzEwZGJhZTg5NzlmZjM=' --header 'content-type:
application/json' --data '{"name": "test", "description": "test"}'

```

Example Response for POST

```

{
  "href": "/orgs/1/container_clusters/1b851d4b-f22d-47be-b744-
f3c2dca490a0",
  "pce_fqdn": null,
  "name": "test",
  "description": "test",
  "manager_type": null,
  "last_connected": null,
  "kubelink_version": null,
  "online": false,
  "nodes": [],
  "errors": [],
  "container_cluster_token": ""
}

```

```
"1_0dfec0acb8e4bc53e052874874da0c24e7ac98da3b3954e3c9ea6f9860722e84"
}
```

## Parameters for container workload methods

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
container_cluster_id	Cluster UUID	String	Yes
assign_labels	(GET) List of lists of label URIs, encoded as a JSON string  (POST, PUT) Assigned labels	String	No
enforcement_mode	(GET) Filter by enforcement mode.  (PUT) workload enforcement mode	String	No
linked	Filter by linked container workload profiles.	Boolean	No
managed	Filter by managed state	Boolean	No
max_results	Maximum number of container workloads to return.	Integer	No
name	(GET) Name string to match. Supports partial matches.  (POST) A friendly name given to a profile if the namespace is not user-friendly	String	NO
namespace	Namespace string to match. Supports partial matches.	String	No
visibility_level	Filter by visibility level	String	No

## Properties for container workload methods

Property	Description	Type
href	Container Workload Profile URI	String
enforcement_mode	Reference to common/workload_enforcement_mode.schema.json	
managed	If the namespace is managed or not	Boolean
max_results	Maximum number of container workloads to return.	Integer
name	A friendly name given to a profile if the namespace is not user-friendly.	String, Null
namespace	Namespace	String, Null
container_workload_profile_id	Container workload profile UUID	String
labels	Labels to assign to the workload that matches the namespace.  Reference to common/label_restrictions.schema.json	

## Curl Examples

Curl example for GET

```
curl --request GET --url https://pce.my-company.com:8443/api/v2/orgs/1/containerometers iun one table and verified with the Quick Reference._clusters/445bfa9b-4de4-4c09-9705-496eb04b190f/container_workload_profiles --header 'authorization: Basic NjA4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzOTQ1ZGMxYzEwZGJhZTg5NzlmZjM=' --header 'content-type: application/json'
```

Curl Example for POST

```
curl --request POST --url https://pce.my-company.com:8443/api/v2/orgs/1/container_clusters/445bfa9b-4de4-4c09-9705-496eb04b190f/container_workload_profiles --header 'authorization: Basic A5ZmRjNjA4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzOTQ1ZGMxYzEwZGJhZTg5NzlmZjM=' --header 'content-type: application/json' --data '{"name": "test", "description": "test", "assign_labels": [{"href": "/orgs/1/labels/1"}], "mode": "full", "log_traffic": true}'
```

Curl Example for PUT

```
curl --request PUT --url https://pce.my-company.com:8443/api/v2/orgs/1/container_clusters/445bfa9b-4de4-4c09-9705-496eb04b190f/container_workload_profiles/219b49c3-3bb5-4fc0-9913-b76398105e35 --header 'authorization: Basic mRjNjA4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzOTQ1ZGMxYzEwZGJhZTg5NzlmZjM=' --header 'content-type: application/json' --data '{"name": "test", "description": "test", "assign_labels": [{"href": "/orgs/1/labels/1"}], "mode": "full", "log_traffic": true}'
```

Example Response for GET

```
[  
  {  
  
    "href": "/orgs/10/container_clusters/974aec34-e8e7-478d-9ca2-90ebb3642edc/  
    container_workload_profiles/5454cc84-d6be-4e6c-ac62-465f9504fac0",  
      "namespace": "openshift-host-network",  
      "enforcement_mode": "visibility_only",  
      "visibility_level": "flow_summary",  
      "managed": true,  
      "assign_labels": [  
        {  
          "href": "/orgs/10/labels/128"  
        },  
        {  
          "href": "/orgs/10/labels/225"  
        }  
      ],  
      "labels": [  
        {  
          "key": "loc",  
          "assignment": {  
            "href": "/orgs/10/labels/128",  
            "value": "AWS"  
          }  
        },  
        {  
          "key": "env",  
          "assignment": {  
            "href": "/orgs/10/labels/225",  
            "value": "OCP4.6"  
          }  
        }  
      ],  
      "linked": true,  
      "created_at": "2021-08-25T18:11:52.665Z",  
      "created_by": {  
        "href": "/orgs/10/container_clusters/974aec34-  
e8e7-478d-9ca2-90ebb3642edc"  
      },  
      "updated_at": "2021-08-25T18:11:52.665Z",  
      "updated_by": {  
        "href": "/orgs/10/container_clusters/974aec34-  
e8e7-478d-9ca2-90ebb3642edc"  
      }  
    }  
]
```

### **Examples for container\_workload\_profiles/update**

Request

```
{  
  "container_workload_profiles": [  
    {
```

```

        "href": "url_to_some_container_workload_profile"
    },
    {
        "href": "url_to_other_container_workload_profile"
    }
],
"labels": [
{
    "key": "role",
    "assignment": {
        "href": "url_to_label"
    }
}
],
"enforcement_mode": 2,
"visibility_level": "flow_summary",
"managed": true
}

```

#### Example Response

- For success: Response code 204; Response body: none
- If an error occurred on any of the input records:
  - Response code 406;
  - Response body:

```

[
    {
        "token": "input_validation_error",
        "message": "....., record_index=>1, ...",
        "unmanaged_container_workload_profile_labels": ...
            # message contains index of failed record and specific
            error message
    },
    ...
]

```

#### Examples for label restrictions

Set an empty Role label.

```
{
    "labels": [
        {"key": "role", "assignment": {}}
    ]
}
```

Set a Location label.

```
PUT /api/v2/orgs/1/container_clusters/65d1f197-938a-49ef-9343-6f55ec76fd90/
container_workload_profiles/afe4661a-03ef-462f-ada6-ce7334aa9704
```

```
{
    "labels": [
```

```

        { "key": "loc", "restriction": { "href": "/orgs/1/labels/221" } }
    ]
}

```

Set an allow list for the Environment label.

Allow a list of Environment labels to be assigned using Kubernetes:

```
PUT /api/v2/orgs/1/container_clusters/65d1f197-938a-49ef-9343-6f55ec76fd90/
container_workload_profiles/afe4661a-03ef-462f-ada6-ce7334aa9704
```

```
{
  "labels": [
    { "key": "env", "restriction": [ { "href": "/orgs/1/labels/176" },
    { "href": "/orgs/1/labels/302" }, { "href": "/orgs/1/labels/303" } ] }
  ]
}
```

Allow any value for the Application label.

```
PUT /api/v2/orgs/1/container_clusters/65d1f197-938a-49ef-9343-6f55ec76fd90/
container_workload_profiles/afe4661a-03ef-462f-ada6-ce7334aa9704
```

```
{
  "labels": [
    { "key": "app", "restriction": [ ] }
  ]
}
```

Multiple ways to assign or allow labels used together in one Container Workload Profile

```
PUT /api/v2/orgs/1/container_clusters/65d1f197-938a-49ef-9343-6f55ec76fd90/
container_workload_profiles/afe4661a-03ef-462f-ada6-ce7334aa9704
```

```
{
  "labels": [
    { "key": "role", "assign": {} },
    { "key": "app", "restriction": [ ] },
    { "key": "env", "restriction": [ { "href": "/orgs/1/labels/176" },
    { "href": "/orgs/1/labels/302" }, { "href": "/orgs/1/labels/303" } ] },
    { "key": "loc", "assign": { "href": "/orgs/1/labels/221" } }
  ]
}
```

Result for the above example:

- **role**: No label will be set; it is an explicit statement (you don't want a **role** label to be assigned).
- **app**: Any value can be set in the annotations for the **app** label key (provided the value exists in PCE).
- **env**: Only the values specified in the allowlist can be set in the annotations for the **env** label key.
- **loc**: The value of the **loc** label key is assigned to the value defined in the payload.

## Properties

Backend services associated with container clusters

Property	Description	Type	Required
name	The name of the container cluster backend.	String	Yes
kind	The type (or kind) of the container cluster backend.	String	Yes
updated_at	The time (rfc339 timestamp) at which the container cluster backend was updated.	String	Yes
created_at	The time (rfc339 timestamp) at which the container cluster backend was created.	String	Yes
virtual_services	Includes the following properties: <ul style="list-style-type: none"> <li>• href: The URI to the associated virtual service</li> <li>• name: The virtual service name</li> </ul>	Object	Yes
		String	
		String	

Curl Example for GET

```
curl --request GET --url https://pce.my-company.com:8443/api/v2/orgs/1/container_clusters/445bfa9b-4de4-4c09-9705-496eb04b190f/service_backends --header 'authorization: Basic YzE2YjBkYjI0MjJhZGNkYWU50ja5ZmRjnja4MDhiMzExZTc2Y2UyNzNmOWNiN2ZhMTA5OTdkMWNlMDAzZmMzOTQ1ZGMxYzEwZGJhZTg5Nz1mZjM='
```

Example Response for GET

```
[
  {
    "name": "58687784f9",
    "kind": "replicasethash",
    "namespace": "kube-system",
    "updated_at": "2020-10-25T20:07:39.741Z",
    "created_at": "2020-10-25T20:07:39.741Z",
    "virtual_service": {
      "href": "/orgs/1/sec_policy/draft/virtual_services/926c2f63-bcd8-42f1-8811-165b34f84334",
      "name": "coredns-k8s2-kube-system"
    }
  },
  {
    "name": "556b9ff8f8",
    "kind": "replicasethash",
    "namespace": "kube-system",
    "updated_at": "2020-10-25T20:07:39.768Z",
    "created_at": "2020-10-25T20:07:39.768Z",
    "virtual_service": {
      "href": "/orgs/1/sec_policy/draft/virtual_services/58b0df03-1151-464e-8352-069e3ad0d7ed",
      "name": "kubernetes-dashboard-k8s2-kube-system"
    }
  }
]
```

```

        }
    }
]

GET /api/v2/orgs/:xorg_id/kubernetes_workloads

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "name",
    "kind",
    "namespace"
  ],
  "properties": {
    "href": {
      "description": "URI of the container workload",
      "type": "string"
    },
    "name": {
      "description": "Container workload name",
      "type": "string"
    },
    "namespace": {
      "description": "k8s namespace where this k8s Workload belongs to",
      "type": "string"
    },
    "kind": {
      "description": "k8s resource kind, e.g. Deployment",
      "type": "string"
    },
    "labels": {
      "type": "array",
      "items": {
        "$ref": "../common/label_optional_key_value.schema.json"
      }
    },
    "enforcement_mode": {
      "$ref": "../common/workload_enforcement_mode.schema.json"
    },
    "visibility_level": {
      "$ref": "../common/workload_visibility_level.schema.json"
    },
    "container_workload_profile": {
      "$ref":
        "container_clusters_container_workload_profiles_get.schema.json"
    },
    "container_cluster": {
      "$ref": "container_clusters_get.schema.json"
    },
    "security_policy_applied_at": {
      "description": "Last reported time when policy was processed by CLAS to the k8s workload (UTC)",
      "type": [

```

```

        "string",
        "null"
    ],
    "format": "date-time"
},
"security_policy_sync_state": {
    "description": "Current state of security policy",
    "type": "string"
},
"created_at": {
    "description": "RFC 3339 timestamp at which this record was created",
    "format": "date-time",
    "type": "string"
},
"updated_at": {
    "description": "RFC 3339 timestamp at which this record was updated",
    "format": "date-time",
    "type": "string"
},
"k8s_labels": {
    "type": "array",
    "items": {
        "type": "object",
        "required": [
            "key",
            "value"
        ],
        "properties": {
            "key": {
                "type": "string"
            },
            "value": {
                "type": "string"
            }
        }
    }
},
"k8s_annotations": {
    "type": "array",
    "items": {
        "type": "object",
        "required": [
            "key",
            "value"
        ],
        "properties": {
            "key": {
                "type": "string"
            },
            "value": {
                "type": "string"
            }
        }
    }
}
}

```

```

        }
    }
}
}

GET /api/v2/orgs/:xorg_id/kubernetes_workloads/:kubernetes_workload_uuid

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Collection of assigned list of labels",
  "type": "array",
  "items": {
    "$ref": "labels.schema.json",
    "minItems": 1
  },
  "uniqueItems": true,
  "minItems": 1
}

common kubernetes_workloads_metadata

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "k8s object metadata",
  "additionalProperties": false,
  "type": "object",
  "properties": {
    "labels": {
      "description": "k8s key/value pairs attached to object that specify identifying attributes",
      "type": "object"
    },
    "annotations": {
      "description": "k8s key/value pairs representing arbitrary non-identifying metadata of object",
      "type": "object"
    },
    "external_service_uid": {
      "description": "k8s object uid of external traffic service (NodePort or LoadBalancer)",
      "type": "string"
    }
  }
}

```

For more information, see Illumio Core for Kubernetes and OpenShift.

## Access Restrictions and Trusted Proxy IPs

To employ automation for managing the PCE environment, you can use API Keys created by an admin user and automate the PCE management tasks. Illumio provides a way to restrict the use of these API keys by IP address, allowing you to block API requests from non-authorized IP addresses.

## Access Restrictions

Access restrictions are configurable entities that contain a list of up to 8 IPv4 IP addresses or CIDR blocks, specifying the source IP addresses of allowed clients. Only the global Org Owner can manage access restrictions within the organization, while other roles are unable to edit or view them.

The following rules apply to access restrictions:

- Each access restriction can be applied to either one or both:
  - API requests are authenticated by API keys
  - API requests are authenticated by **Username/Password** credentials
- The global Org Owners can edit an access restriction after it has been created by modifying the allowed IP list or the options. They can also assign access restrictions to Local and External Users. The API supports the update of access restrictions for a list of users.
- Access restrictions are leader-owned configuration objects that are replicated to all super-cluster regions.
- Access restrictions are enforced as follows:
  - To enforce an API request, determine the user account for that API request using the API key or the user session token, and then find the access restriction that is configured for that user. If no access restriction is assigned to the user, the API request proceeds.
  - If the client IP address for an API request does not satisfy the corresponding user's access restrictions, the request is rejected with a 401 error message.
  - Access restrictions are not enforced on some URLs (node\_available, static JS/CSS content).
- When a request is rejected due to unsatisfied access restrictions, it generates an Event that specifies a failure caused by an invalid source IP address, including the actual IP address and an appropriate error code (403).

## Assignment to Users

Each Access Restriction is a configuration object that specifies a set of allow-list IP addresses or CIDR blocks, designating the allowed client IP address. It also specifies the types of API accesses that are restricted (those authenticated by API Keys or those authenticated by user session tokens).

The Org Owners create and manage access restrictions within their organizations, ensuring a maximum of 50 access restrictions per organization. The Org Owners can assign a single access restriction to each Local or External User (by default, no access restriction is assigned).

## Access Restriction Methods

Functionality	HTTP	URI
Get a list of access restrictions.	GET	/api/v2/orgs/<org_id>/ access_restrictions
Get a specific access restriction.	Get	/api/v2/orgs/<org_id>/ access_restrictions/<id>
Create an access restriction	POST	/api/v2/orgs/<org_id>/ access_restrictions
Update an access restriction.  Same schema as for POST,  but fields such as <code>name</code> or <code>ips</code> might not be required	PUT	/api/v2/orgs/<org_id>/ access_restrictions/<id>
The <code>DELETE</code> endpoint should return an error.  if the specified <code>access_restriction</code> is referenced  by any User or Group.  The existing <code>access_restrictions</code> from all Users  and Groups must be removed before they can be deleted.	DELETE	/api/v2/orgs/<org_id>/ access_restrictions/<id>

## Return Values for Access Restriction

These are the return values for the Access Restriction methods:

Property	Method	Description	Required
<code>href</code>	GET	URI of access restriction	Yes
<code>name</code>	GET, POST	User assigned name of the access restriction	(No GET) (Yes POST)
<code>description</code>	GET, POST	User assigned description of the access restriction	No
<code>ips</code>	GET, POST	Array of ip addresses or CIDR blocks	Yes
<code>enforcement_exclusions</code>	GET, POST	The types of API access methods that are excluded from access restriction enforcement	No

## Trusted Proxy IPs

When a client is connected to the PCE's haproxy server, this connection can traverse one or more load balancers or proxies. Therefore, the source IP address of a client connection to haproxy might not be the actual public IP address of the client.

Proxies and intermediaries often use the `X-Forwarded-For` header (and some other custom headers, like `X-Client-IP`) to pass along the client IP address. The value of this header is a

comma-separated list of one or more IP addresses, where the source IP address seen by the most recent proxy is listed last.

The client IP address used for API requests and Web UI connections comes from the value of the `X-Forwarded-For` header that haproxy sets on the back-end request to the web service. It is set to one of these values:

- Value of the `X-Forwarded-For` header on the incoming request (when `trust_upstream_x_forwarded_for` is `true`)
- Source IP address of the client connection to haproxy (when `trust_upstream_x_forwarded_for` is `false`)

Configurable trusted proxy IPs allow Org Owners to configure a list of IPv4 addresses or CIDR blocks that are considered trusted for setting a client's `X-Forwarded-For` header.

The Org Owner can designate the trusted proxies/intermediaries using this setting. The PCE will consider all others to be untrusted when setting the `X-Forwarded-For` header.

The haproxy is configured to always put the client's source IP address in the `X-Real-IP` header on the back-end request and to pass along any `X-Forwarded-For` headers in the front-end request.

## Trusted Proxy IP Methods

Functionality	HTTP	URI
Get a list of trusted IP proxies	GET	/api/v2/orgs/<org_id>/settings/trusted_proxy_ips
Inter-service API for fetching an orgs' <code>trusted_proxy_ips</code> settings, so that it may be cached locally.  It uses the same schema as the GET endpoint above;  It receives the <code>org_id</code> as a query input	GET	/api/v2/org_trusted_proxy_ips?org_id=<id>
Update <code>trusted_proxy_ips</code> settings for a given org, with the same schema as the GET endpoint (except without the property)  <code>max_trusted_proxy_ips_per_region</code>	PUT	/api/v2/orgs/<org_id>/settings/trusted_proxy_ips

## Trusted Proxy IPs

These are the return values for the Trusted Proxy methods:

Parameter	Method	Description	Req
max_trusted_proxy_ips_per_region	GET	Maximum number of Trusted Proxy IPs allowed for each PCE	Yes
trusted_proxy_ips	GET, PUT	Required:  <code>pce_fqdn</code> : FQDN of PCE region, or null if not in supercluster  <code>ip</code> : IP address or CIDR trusted for handling	Yes

## Organization Access

Changes to the organization access introduced a new common schema:

common ipv4\_ipv6\_subnet

This common schema is replacing the one that is now deleted: common ipv4\_subnet

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "string",
  "pattern": "^(25[0-5]|2[0-4][0-9]|0[1-9]?[0-9][0-9]?)\\.
  {3}(25[0-5]|2[0-4][0-9]|0[1-9]?[0-9]
  [0-9]?)\\/(3[0-2]|0[2-9]?[0-9]))?$$"
}
```

Three organization access APIs have been changed to substitute common/ipv4\_subnet.schema with common/ipv4\_ipv6\_subnet.schema:

- orgs\_access\_restrictions\_post
- orgs\_access\_restrictions\_put

## Access Restrictions Reference

This topic covers examples of access restriction.

### Examples

Create Access Restrictions

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/1/
access_restrictions/
```

Response

```
{
  "name": "sample Access Restriction payload",
  "description": "example",
  "ips": [ "192.168.33.1/16" ],
  "enforcement_exclusions": [ "user_sessions" ]
}
```

### Read an Access Restriction

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/1/
access_restrictions/
```

### **Update an Access Restriction**

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/1/
access_restrictions/1
```

```
{
  "name": "modified Access Restriction payload",
  "description": "example",
  "ips": [ "192.168.33.1/16" ],
  "enforcement_exclusions": [ "user_sessions" ]
}
```

### **Delete the Acess Restriction**

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/1/
access_restrictions/1
```

Curl Command to associate an Access Restriction with an Org Auth Sec Principal (PUT)

```
curl -i -X -PUT https://pce.my-company.com:8443/api/v2/orgs/1/
auth_security_principals/76a0607b-6961-4c74-a98a-8b10775c8a9b
```

```
{
  "name": "test.user@illumio.com",
  "display_name": "test",
  "type": "user",
  "access_restriction": {
    "href": "/orgs/1/access_restrictions/1"
  }
}
```

### **Read a Trusted Proxy IP**

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/1/
access_restrictions/
```

### **Update a Trusted Proxy IP**

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/1/settings/
trusted_proxy_ips/
```

```
{
  "trusted_proxy_ips": [
    {
      "pce_fqdn": null,
      "ip": "66.151.147.0/24"
    },
    {
      "pce_fqdn": null,
      "ip": "192.168.34.0/24"
    }
  ]
}
```

```
]
}
```

## Organization Access

Changes to the organization access introduced a new common schema:

```
common ipv4_ipv6_subnet
```

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "string",
  "oneOf": [
    { "format": "ipv4" },
    { "format": "ipv6" }
  ]
}
```

This common schema is replacing the one that is now deleted: common ipv4\_subnet

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "string",
  "pattern": "^(25[0-5]|2[0-4][0-9]|01?[0-9][0-9]?)\\.(?{3}(25[0-5]|2[0-4][0-9]|01?[0-9][0-9]?)\\.(?{3}(25[0-5]|2[0-4][0-9]|01?[0-9][0-9]?)\\.)?\\/(3[0-2]|0-2)?[0-9])?)?\\$"
}
```

Three organization access APIs have been changed to substitute

common/ipv4\_subnet.schema with

common/ipv4\_ipv6\_subnet.schema:

- orgs\_access\_restrictions\_post
- orgs\_access\_restrictions\_put

```
{
  "properties": {
    "ips": {
      "items": {
        "$ref": {
          "__old": ".../common/ipv4_subnet.schema.json",
          "__new": ".../common/ipv4_ipv6_subnet.schema.json"
        }
      }
    }
  }
}
```

settings\_trusted\_proxy\_ips\_put

```
{
  "properties": {
```

```

"trusted_proxy_ips": {
  "items": {
    "properties": {
      "ip": {
        "$ref": {
          "__old": ".../common/ipv4_subnet.schema.json",
          "__new": ".../common/ipv4_ipv6_subnet.schema.json"
        }
      }
    }
  }
}

```

## Rulesets and Rules

Illumio's security policy includes three rule types: intra-scope rules, extra-scope rules, and custom iptables rules. The scope of a ruleset determines which workloads receive the ruleset's rules:

- **Intra-scope rules** allow communication between Sources and Destinations within a specific scope.
- **Extra-scope rules** permit communication between applications. You can write rules so that destinations within or outside a specified scope can access the providers within a scope. For extra-scope rules, the labels used in the scope must match the labels used by the Source.
- **Custom iptables rules** are needed for your applications as part of the rules managed by the PCE. These rules help preserve configured iptables from native Linux host configurations by allowing you to include them with the rules for your policy.

## Rulesets

This Public Stable API allows you to get, create, update, and delete rulesets. Rulesets contain rules and scopes, which define where the rules apply.

### Ruleset API Methods

Functionality	HTTP	URI
Get a collection of rulesets.	GET	[api_version][org_href]/sec_policy/rule_sets
Get a specified ruleset instance.	GET	[api_version][org_href]/sec_policy/rule_sets/rule_set_id]
Create a ruleset.	POST	[api_version][org_href]/sec_policy/rule_sets
Update a specified ruleset.	PUT	[api_version][org_href]/sec_policy/rule_sets/rule_set_id]
Delete a specified ruleset.	DELETE	[api_version][org_href]/sec_policy/rule_sets/rule_set_id]

## Active vs. Draft

This API operates on provisionable objects, which exist in either a `draft` (not provisioned) state or an `active` (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be draft, active, or the ID of the security policy.
- For POST, PUT, DELETE — `:pversion` can be draft (you cannot operate on active items) or the ID if the security policy.

## Ruleset Components

Rulesets are the core of the Illumio Core policy model and consist of the following elements:

- **Scopes:** Sets of labels (application, environment, and location) that define the boundaries of the rules in a ruleset. If the workloads specified in the rules share the same labels in a ruleset scope, then those workloads and their communications are governed by the rules of the ruleset.

A scope can contain zero or more application, environment, and location labels. A scope can also contain one or more label groups.

If the scope is an empty array ([ ]), then the scope includes all applications, environments, and locations.

If one of the label types is not specified, all instances of that type are permitted. For example, all applications are within the scope if application labels are omitted, but environment and location labels are present.

A label type cannot be used in a rule unless the scope for the label type is “All.” For example, to use a location label, the scope must be an empty array ([]). Alternatively, if an application label and an environment label are present in the scope, the location label cannot be defined within the scope.

A ruleset is not limited to a single scope. A rule can contain multiple scopes depending on the security policy’s needs.



### IMPORTANT

Role labels are not used in scopes but can be used in rules. Never use a role label in a scope.

- **Rules:** A security rule consisting of one or more providers (provides a service over a port and protocol), one or more destinations (consumes the service offered by the Source), and one or more services. A Source or Destination can be an individual workload, a role label representing multiple workloads, IP lists, etc.

## Example Ruleset Scope

Each label in a scope is identified by its HREF. This is the JSON representation of a single ruleset scope with three labels.

Each label must have a different key (role, app, loc, or env). Duplicate label keys are allowed in a scope only if they are in a label group.

```
{
  "scopes": [
    [
      {"label": {"href": "/orgs/7/labels/105"}},
      {"label": {"href": "/orgs/7/labels/88"}},
      {"label": {"href": "/orgs/7/labels/98"}}
    ]
  ]
}
```

## Ruleset Rules



### NOTE

The common schema `consumer_security_principals` has been replaced by two other APIs: `consumer_security_principals_get` and `consumer_security_principals_put`

Ruleset rules define the allowed communication between workloads or between workloads and IP lists.

For information, see Rules.

## Get Rulesets

This method gets all of the rulesets in your organization. This method gets those rulesets that are in the “draft” policy state, which means the current state of rulesets that have not been provisioned.

By default, the maximum number of rulesets returned in a GET collection is 500.



### NOTE

To return more than 500 rulesets, use Asynchronous GET Collection.

URI to Get a Collection of Rulesets

`pversion`: Contains provisionable objects, which exist in either a `draft` (not provisioned) or `active` (provisioned) state.

GET `[api_version][org_href]/sec_policy/:pversion/rule_sets`

URI to Get an Individual Ruleset

`[api_version[org_href]/sec_policy/rule_sets/rule_set_id]`

## Create a Ruleset

This method creates an individual ruleset. The PCE web console supports up to 500 rules per ruleset.



### NOTE

To write more than 500 rules for a particular ruleset, create additional rule-sets, or use the Illumio Core REST API (rulesets with more than 500 rules are not fully displayed in the PCE web console).

URI to Create a Ruleset

```
POST [api_version][ruleset_href]
```

## Update a Ruleset

To update an individual ruleset, you need the HREF of the ruleset you want to update, which can be obtained when you retrieve a collection or an individual ruleset.

If you want to add a single rule to an existing ruleset, use

```
PUT /api/v2/orgs/1/sec_policy/draft/rule_sets/123/sec_rules.
```

## Delete a Ruleset

To delete an individual ruleset, you need the HREF of the ruleset you want to delete, which can be obtained when you get a collection of rulesets.

URI to Delete an Individual Ruleset

```
DELETE [api_version][ruleset_href]
```

## Rulesets Reference

This topic covers parameters, properties, and examples for rulesets.

### Parameters for rulesets

You can use the following query parameters to restrict the query results and retrieve a collection of rulesets.

Parameter	Description	Type	Re- quired
org_id	Organization	Integer	Yes
pversion	Security Policy Version	String	Yes
rule_set_id		Integer	Yes
name	Name of the rulesets to filter, which must be unique. This parameter supports partial matches.	String	No
description	Description of Rule Set(s) to return. Supports partial matches	String	No
external_da- ta_set	The data source from which the resource originates. For example, if ruleset information is stored in an external data-base.	String, Null	No
external_da- ta_reference	A unique identifier within the external data source. For example, if ruleset information is stored in an external data-base.	String, Null	No
enabled	Enabled flag	Boolean	No
update_type	Type of update	String	No
scopes	Rule set scopes  • label: label URI • label_group: label group URI	Array  String	No
rules	Array of rules in this rule set  Required properties:  enabled: Enabled flag  description  external_data_set  external_data_reference  ingress_service	Object	No

## Properties for rulesets

Property	Description	Type	Re- quired
enabled	Enabled flag	Boolean	Yes
name	Name of the rulesets to filter. This parameter supports partial matches.	String	Yes
scopes	Reference to <code>common/rule_set_scopes_get.schema.json</code>		Yes
rules	Array of rules in this rule set  Reference to <code>sec_policy_rule_sets_sec_rules_get.schema.json</code>		Yes
created_at	Timestamp when this rule set was first created	String	Yes
updated_at	Timestamp when this rule set was last updated	String	Yes
deleted_at	Timestamp when this rule set was deleted	String, Null	Yes
created_by	User who originally created this rule set	Object	No
updated_by	User who last updated this rule set	Object	No
deleted_by	User who deleted this rule set	Object, Null	No
update_type	Type of update  Reference to <code>common/sec_policy_update_type.schema.json</code>		No
external_data_set	The data source from which the resource originates. For example, if ruleset information is stored in an external database.	String	No
external_data_reference	A unique identifier within the external data source. For example, if ruleset information is stored in an external database.	String	No
ip_tables_rules	Array of iptables rules in this rule set.  Reference to <code>common/ip_tables_rules_get.schema.json</code>		No

## Properties to create a ruleset

Property	Description	Type	Required
name	Name of the new ruleset, which must be unique.	String	Yes
scopes	Reference to common/rule_set_scopes_put.schema.json		Yes
rules	Reference to sec_policy_rule_sets_sec_rules_post.schema.json		No
ip_tables_rules	Array of custom iptables rules in this rule set. Reference to common/ip_tables_rules_post.schema.json		No
external_data_set	External data set identifier	String, Null	No
external_data_reference	External data reference identifier.	String, Null	No
enabled	Enabled flag	Boolean	Yes
scopes	Reference to common/rule_set_scopes_put.schema.json		Yes
rules	Array of rules in this rule set Reference to sec_policy_rule_sets_sec_rules_post.schema.json		Yes

## Properties for updating a ruleset

Property	Description	Type	Re- quired
name	Name of the ruleset to update, must be unique	String	No
external_data_set	External data set identifier	String, Null	No
external_da- ta_reference	External data reference identifier.	String, Null	No
enabled	Enabled flag	Boolean	Yes
update_type	Type of update  Reference to common/ sec_policy_update_type. schema.json		No
scopes	Reference to common/rule_set_ scopes_ put. schema.json		No
rules	Array of rules in this rule set  Required properties:  "href"  "enabled",  "providers",  "destinations",  "ingress_services",  "resolve_labels_as"	Object	No
destinations	Reference to  sec_policy_rule_sets_ sec_rules_destinations_put. schema.json		
consuming_securi- ty_principals	Reference to  common/consuming_ security_principals_ put.schema.json		
network_type	Reference to  common/rule_network _type. schema.json		
use_workload_sub- nets	Reference to  sec_rule_use_ workload_ subnets.schema.json		

Get a Ruleset

```
$curl -X GET https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/draft/
rule_sets -H "Accept: application/json" -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5| jq
```

Response

```
[  
  {  
    "href": "/orgs/1/sec_policy/draft/rule_sets/1",  
    "created_at": "2023-04-05T23:08:32.578Z",  
    "updated_at": "2023-04-05T23:08:32.632Z",  
    "deleted_at": null,  
    "created_by": {  
      "href": "/users/0"  
    },  
    "updated_by": {  
      "href": "/users/0"  
    },  
    "deleted_by": null,  
    "update_type": null,  
    "name": "Default",  
    "description": null,  
    "enabled": true,  
    "scopes": [  
      []  
    ],  
    "rules": [  
      {  
        "href": "/orgs/1/sec_policy/draft/rule_sets/1/sec_rules/1",  
        "created_at": "2023-04-05T23:08:32.599Z",  
        "updated_at": "2023-04-05T23:08:32.632Z",  
        "deleted_at": null,  
        "created_by": {  
          "href": "/users/0"  
        },  
        "updated_by": {  
          "href": "/users/0"  
        },  
        "deleted_by": null,  
        "update_type": null,  
        "description": "Allow outbound connections",  
        "enabled": true,  
        "providers": [  
          {  
            "ip_list": {  
              "href": "/orgs/1/sec_policy/draft/ip_lists/1"  
            }  
          }  
        ],  
        "destinations": [  
          {  
            "actors": "ams"  
          }  
        ]  
      }  
    ]  
  }]
```

```

"consuming_security_principals": [],
  "sec_connect": false,
  "stateless": false,
  "machine_auth": false,
  "unscoped_destinations": false,
  "network_type": "brn",
  "use_workload_subnets": [],
  "ingress_services": [
    {
      "href": "/orgs/1/sec_policy/draft/services/1"
    }
  ],
  "egress_services": [],
  "resolve_labels_as": {
    "providers": [
      "workloads"
    ],
    "destinations": [
      "workloads"
    ]
  }
},
],
"ip_tables_rules": [],
  "caps": [
    "write",
    "provision"
  ]
},
{
  "href": "/orgs/1/sec_policy/draft/rule_sets/3",
  "created_at": "2023-04-05T23:50:05.591Z",
  "updated_at": "2023-04-06T19:03:49.947Z",
  "deleted_at": null,
  "created_by": {
    "href": "/users/1"
  },
  "updated_by": {
    "href": "/users/1"
  },
  "deleted_by": null,
  "update_type": null,
  "name": "ruleset1"
},
"description": "",
"enabled": true,
"scopes": [
[]
],
"rules": [
  {
    "href": "/orgs/1/sec_policy/draft/rule_sets/3/sec_rules/9",
    "created_at": "2023-04-06T00:58:55.061Z",
    "updated_at": "2023-04-06T00:58:55.088Z",
    "deleted_at": null,
    "created_by": {
      "href": "/users/1"
    }
  }
]
}

```

```

        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    },
    "deleted_by": null,
    "update_type": null,
    "description": "",
    "enabled": true,
    "providers": [
    {
        "label": {
            "href": "/orgs/1/labels/14"
        },
        "exclusion": false
    }
],
    "destinations": [
    {
        "label": {
            "href": "/orgs/1/labels/15"
        },
        "exclusion": false
    }
],
    "consuming_security_principals": [],
    "sec_connect": true,
    "stateless": false,
    "machine_auth": false,
    "unscoped_destinations": false,
    "network_type": "brn",
    "use_workload_subnets": [],
    "ingress_services": [
    {
        "href": "/orgs/1/sec_policy/draft/services/9"
    },
    {
        "port": 23000,
        "proto": 6
    }
],
    "egress_services": [],
    "resolve_labels_as": {
    "providers": [
        "workloads"
    ],
    "destinations": [
        "workloads"
    ]
}
    ],
    "ip_tables_rules": []
}
]
}
}
],
"ip_tables_rules": []

```

```

        "caps": [
            "write",
            "provision"
        ]
    }
]

```

Create a Ruleset

```
$curl -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5-X POST -H 'Content-Type: application/json' -d '
{"name": "ruleset3", "description": "", "scopes": [{"exclusion": false, "label": {"href": "/orgs/1/labels/14"} }]}' https://2x2testvc168.ilabs.io:8443/api/v2/
orgs/1/sec_policy/draft/rule_sets | jq
```

Response

```
{
    "href": "/orgs/1/sec_policy/draft/rule_sets/16",
    "created_at": "2023-04-06T18:46:34.718Z",
    "updated_at": "2023-04-06T18:46:34.727Z",
    "deleted_at": null, "created_by": {
        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    },
    "deleted_by": null,
    "update_type": "create",
    "name": "ruleset3",
    "description": "",
    "enabled": true, "scopes": [
        [
            {
                "label": {
                    "href": "/orgs/1/labels/14"
                },
                "exclusion": false
            }
        ]
    ],
    "rules": [],
    "ip_tables_rules": [], "caps": [
        "write",
        "provision"
    ]
}
```

Update a Ruleset

```
$curl -w "%{http_code}" -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5 -X PUT -H 'Content-Type: application/json' -d '{"scopes": [
    {"label": {"href": "/orgs/1/labels/14"} },
    {"label": {"href": "/orgs/1/labels/14"} }
]}
```

```
15"}]}]}' https://2x2testvc168.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/
rule_sets/14 | jq
```

Response

The ruleset was successfully updated:

204

## Rules

This Public Stable API creates, updates, and deletes individual rules in rulesets.

It also gets a collection of rules from a ruleset.

The Illumio Core allowlist policy model uses rules to define the allowed communications between two or more workloads or between workloads and other entities, such as IP lists, virtual servers, and the internet.

### Rules API Methods

The fundamental structure of a rule (except custom iptables rules) consists of a Source, a service that the source makes available over a network port and protocol, and a Destination of that service.

**Table 3. API Methods for Rules**

Functionality	HTTP	URI
Get rules	GET	sec_policy_rule_sets_sec_rules
Get rules for providers.	GET	sec_policy_rule_sets_sec_rules_providers
Get rules for destinations.	GET	sec_policy_rule_sets_sec_rules_destination
Update rules	PUT	sec_policy_rule_sets_sec_rules
Update rules for providers	PUT	sec_policy_rule_sets_sec_rules_providers
Update rules for destinations	PUT	sec_policy_rule_sets_sec_rules_destinations
Create rules	POST	sec_policy_rule_sets_sec_rules
Delete an individual rule.	DELETE	sec_rule_href

## Deny Rules

This API gets, creates, updates, and deletes deny rules. Deny rules deny communication between sources and destinations.

## Deny Rules API Methods

Functionality	HTTP	URI
Get a collection of deny rules	GET	[api_version][org_href]/sec_policy/:pversion/rule_sets/:rule_set_id/deny_rules
Get a specified deny rule instance.	GET	[api_version][org_href]/sec_policy/:pversion/rule_sets/:rule_set_id/deny_rules/:deny_rule_id
Create a deny rule	POST	[api_version][org_href]/sec_policy/:pversion/rule_sets/:rule_set_id/deny_rules
Update a specified deny rule	PUT	[api_version][org_href]/sec_policy/:pversion/rule_sets/:rule_set_id/deny_rules/:deny_rule_id
Delete a specified deny rule	DELETE	[api_version][org_href]/sec_policy/:pversion/rule_sets/:rule_set_id/deny_rules/:deny_rule_id

## Rules Reference

This topic covers properties, parameters, and examples of rules.

### Parameters

Get a collection of security rules from a ruleset

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
pversion	Security policy version -- draft(not provisioned)  or active (provisioned)	String	Yes
rule_set_id	Ruleset ID	Integer	Yes
external_data_reference	A unique identifier within the external data source.  For example, if this rule information is stored in an external database.	String	No
external_data_set	The data source from which the resource originates.  For example, if this rule information is stored in an external database.	String	No
labels	List of lists of label URLs, encoded as a JSON string	String	No
max_results	Maximum number of Rule Sets to return	Integer	No
name	Name of Rule Set(s) to return. Supports partial matches	String	No

Get an Individual Security Rule from a Ruleset

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
pversion	Security policy version -- draft(not provisioned) or active (provisioned)	String	Yes
rule_set_id	Ruleset ID	Integer	Yes

## Properties to create rules

Property	Description	Type	Re-required
enabled	Indicates if the rule is enabled or disabled.	Boolean	Yes
providers	Entities that can be used as a Source in a rule.  Reference to <code>sec_policy_rule_sets_sec_rules_providers_put.schema.json</code>		Yes
consum	Entities that can be used as a Destination in a rule.  Reference to <code>sec_policy_rule_sets_sec_rules_destinations_put.schema.json</code>		Yes
ingress_services	Reference to <code>sec_rule_ingress_services.schema.json</code>		Yes
resolve_labels_as	Reference to <code>sec_rule_resolve_labels_as.schema.json</code>		Yes
sec_connect	Indicates whether a secure connection is established. If set to true, then the rule will use SecureConnect IPsec encryption for all traffic allowed by the rule.	Boolean	No
stateless	Whether packet filtering is stateless for the rule.  If set to true, then the rule's packet filtering is stateless.  This means that the VEN will instruct the host firewall to not maintain persistent connections for a session.  This type of rule is typically used for datacenter "core services" such as DNS and NTP. You can only create a total of 100 stateless rules in your PCE.  If you need more than 100 stateless rules in your Illumio policy, contact your Illumio Professional Services Representative for more information.	Boolean	No
machine_auth	Whether machine authentication is enabled.  If set to true, then machine authentication is used for the rule, meaning that any hosts defined in the rule have been configured for the PKI-based machine authentication.  Before using this property, your PCE must already be configured for machine authentication.  See the PCE Administration Guide for information on configuring machine authentication for the PCE.	Boolean	No
consuming_security_principals	Reference to <code>common/consuming_security_principals_put.schema.json</code>		
unscoped_destinations	Set the scope for rule destinations to All	Boolean	

Property	Description	Type	Required
network_type	Reference to common/rule_network_type.schema.json		
use_workload_subnets	Reference to sec_rule_use_workload_subnets.schema.json		
label_dimension	<p><b>Providers:</b></p> <p>This property was added for "rule actors that are all workloads (ams)" with a specific label dimension.</p> <p><b>destinations:</b></p> <p>This property was added for "rule actors that are all workloads (ams)" with a specific label dimension.</p> <p>The difference is that for "destinations" APIs, the property label_dimension has an additional specified data type or enum app.</p>		

## Update Rules

This API updates an individual rule inside a ruleset.

URI to Update Rules

```
PUT [api_version][sec_rule_href]
```

The request body and JSON payload is the same as that for creating rules.

## Delete a Rule

This API deletes an individual rule inside a ruleset.

URI to Delete a Rule

```
DELETE [api_version][sec_rule_href]
```

Curl Command to Delete Rule

The curl command for deleting a rule can be structured as follows:

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/sec_policy/draft/rule_sets/152/sec_rules/124 -H "Accept: application/json" -u $KEY:$TOKEN
```

## Examples for Rule Search

Curl Command Examples for Rule Search

```
curl -u API_ID:API_SECRET -X POST -H 'Content-Type: application/json' -d '{"providers": [{"label": {"href": "/orgs/1/labels/
```

```
2" } ] , "destinations": [ { "label": { "href": "/orgs/1/labels/1" } } ] } 'https://
dev6.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/rule_search

curl -u API_ID:API_SECRET -X POST -H 'Content-Type: application/
json' -d '{"providers": [ {"workload": { "href": "/orgs/1/workloads/
4ce873d3-2e5d-4f06-82f5-4ble0ec9ceb2" } } ] } 'https://dev6.ilabs.io:8443/api/v2/
orgs/1/sec_policy/draft/rule_search

curl -u API_ID:API_SECRET -X POST -H 'Content-Type: application/
json' -d '{"ingress_services": [ { "href": "/orgs/1/sec_policy/draft/services/
1" } ] } 'https://dev6.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/rule_search

curl -u API_ID:API_SECRET -X POST -H 'Content-Type: application/json'
-d '{"ingress_services": [ { "port": 11000, "to_port": 12000, "proto": "6" } ] } 'https://dev6.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/rule_search
```

Get a Rule

```
$curl -X GET https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/
active/rule_sets/ -H "Accept: application/json" -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5 | jq
```

Response

```
[ {
  {
    "href": "/orgs/1/sec_policy/active/rule_sets/1",
    "created_at": "2023-04-05T23:08:32.578Z",
    "updated_at": "2023-04-05T23:08:32.632Z",
    "deleted_at": null, "created_by": {
      "href": "/users/0"
    },
    "updated_by": {
      "href": "/users/0"
    },
    "deleted_by": null,
    "name": "Default",
    "description": null,
    "enabled": true, "scopes": [ [] ],
    "rules": [
      {
        "href": "/orgs/1/sec_policy/active/rule_sets/1/sec_rules/1",
        "created_at": "2023-04-05T23:08:32.599Z",
        "updated_at": "2023-04-05T23:08:32.632Z",
        "deleted_at": null, "created_by": {
          "href": "/users/0"
        },
        "updated_by": {
          "href": "/users/0"
        },
        "deleted_by": null,
        "description": "Allow outbound connections",
        "enabled": true,
        "providers": [ {
```

```

"ip_list": {
    "href": "/orgs/1/sec_policy/active/ip_lists/1"
}
}
],
"destinations": [
    {
        "actors": "ams"
    }
],
"consuming_security_principals": [],
"sec_connect": false,
"stateless": false,
"machine_auth": false,
"unscoped_destinations": false,
"network_type": "brn",
"use_workload_subnets": [], "ingress_services": [
    {
        "href": "/orgs/1/sec_policy/active/services/1"
    }
],
"egress_services": [],
"resolve_labels_as": {
    "providers": [
        "workloads"
    ],
    "destinations": [
        "workloads"
    ]
}
}
],

```

Create a Rule

```

curl -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5 -X POST -H 'Content-Type: application/json' -d '{
    "providers": [
        {"label": {"href": "/orgs/1/labels/14"}}
    ],
    "destinations": [
        {"label": {"href": "/orgs/1/labels/15"}}
    ],
    "enabled": true,
    "ingress_services": [
        {"href": "/orgs/1/sec_policy/draft/services/9"}
    ],
    "proto": 6,
    "port": 23000
},
    "network_type": "brn",
    "consuming_security_principals": [],
    "sec_connect": true,
    "machine_auth": false,
    "stateless": false,
    "unscoped_destinations": false,
    "description": "",
    "use_workload_subnets": [],
    "resolve_labels_as": {
        "destinations": [
            "workloads"
        ],
        "providers": [
            "workloads"
        ]
    }
}' https://2x2testvc168.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/rule_sets/3/sec_rules | jq
{
    "href": "/orgs/1/sec_policy/draft/rule_sets/3/sec_rules/9",
    "created_at": "2023-04-06T00:58:55.061Z",
    "updated_at": "2023-04-06T00:58:55.088Z",
    "deleted_at": null,
    "created_by": {
        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    }
},

```

```

"deleted_by": null,
"update_type": "create",
"description": "",
"enabled": true, "providers": [
{
  "label": {
    "href": "/orgs/1/labels/14"
  },
  "exclusion": false
},
],
"destinations": [
{
  "label": {
    "href": "/orgs/1/labels/15"
  },
  "exclusion": false
}
],
"consuming_security_principals": [],
  "sec_connect": true,
  "stateless": false,
  "machine_auth": false,
  "unscoped_destinations": false,
  "network_type": "brn",
"use_workload_subnets": [], "ingress_services": [
{
  "href": "/orgs/1/sec_policy/draft/services/9"
}, {
  "port": 23000,
  "proto": 6
}
],
"egress_services": [],
"resolve_labels_as": {
"providers": [
  "workloads"
],
"destinations": [
  "workloads"
]
}
}
}

```

### Update a Rule

```

curl -w "%{http_code}" -u
api_1c2618a67847c94b8:98c76f7a4563f29cd78b3392684cd5ec09534bafe5197fe8e901d9
5561bdd8f5 -X PUT -H 'Content-Type: application/json' -d '{
"providers": [
{"exclusion":false,"label": {"href": "/orgs/1/labels/14"}}, {"destinations": [
{"exclusion":false,"label": {"href": "/orgs/1/labels/15"}}], "enabled":true, "ingress_services": [{"href": "/orgs/1/sec_policy/draft/services/9"}, {"proto":6, "port":25000}], "network_type": "brn", "consuming_security_principals": [], "sec_connect":true, "machine_auth":false, "stateless":false, "unscoped_

```

```
destinations":false,"description":""},"use_workload_subnets":[]}, "resolve_labels_as":{ "providers":["workloads"], "destinations": ["workloads"]}}' https://2x2testvc168.ilabs.io:8443/api/v2/orgs/1/sec_policy/draft/rule_sets/3/sec_rules/3 | jq
```

Response

The rule was successfully updated:

204

## Rule Search

This Public Experimental method searches for rules across all rulesets. This method is especially useful when your organization has many rules organized in rulesets.

For example, your organization has 192,000 rules organized across 650 rulesets, and you need to know how many rules are applied for SNMP (UDP 161). You can't easily find this information without using this method.



### NOTE

Rule search concurrent requests are now increased to 12 searches on 2x2s and 4x2s.

## Rule Search Methods

**Table 4. Rule Search Methods**

Functionality	HTTP	URI
Create rule search	POST	/api/v2/orgs/:xorg_id/sec_policy/:pversion/rule_search

Rule Search exposes `deny_rules` and `override_deny_rules` in the UI when you search for them in the Rule Search page.

For the changes in the UI, see Security Policy Guide, Policy Check and Rule Search.

**Table 5. New Property**

Property Name	Type	Description	Required
rule_types	Array of enums:  sec_rules  deny_rules  override_deny_rules	Requested rule_types that should be searched for	No

## Rule Search Reference

This topic covers properties, parameters, and examples for rule search.

### Parameters for Rule Search

Parameter	Description
actors	Rule actors are all workloads ('ams')
label	Label URI  required: href
label_group	Label group URI  required: href
workload	Workload URI  required: href
container_workload	href of container_workload to which the searched rule should apply to. Added in 25.2.0.  required: href
kubernetes_workload	href of kubernetes_workload to which the searched rule should apply to. Added in 25.2.0.  required: href
virtual_service	Virtual service URI  required: href
ip_list	Name of Rule Set(URI of associated IP List  required: href

## Schema Changes in Release 25.2.0

- **Schema changes for the following:**
  - sec\_policy\_rule\_search\_providers

- sec\_policy\_rule\_search\_destinations

```

"container_workload": {+
    "type": "object", +
    "additionalProperties": false, +
    "required": [ "href" ], +
    "properties": {+
        "href": {+
            "description": "Container workload URI", +
            "type": "string" +
        } +
    } +
},
"kubernetes_workload": {+
    "type": "object", +
    "additionalProperties": false, +
    "required": [ "href" ], +
    "properties": {+
        "href": {+
            "description": "Kubernetes workload URI", +
            "type": "string" +
        } +
    } +
},
=====

```

This schema has two new properties: container\_workload and kubernetes\_workload

**This schema is simplified by using allOf and two existing schemas:**

- sec\_policy\_rule\_search\_post\_response

```
{
  "$schema": "http://json-schema.org/draft-04/schema#", ,
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "sec_rules": {
      "description": "Array of allow rules",
      "type": "array",
      "items": {
        "allOf": [
          {
            "$ref": "../common/sec_rules_get.schema.json"
          },
          {
            "rule_set": {
              "$ref": "../common/rule_set.schema.json"
            }
          }
        ]
      }
    },
    "deny_rules": {
      "description": "Array of deny rules",
      "type": "array",
      "items": {

```

```

"allOf": [
  {
    "$ref": "../common/deny_rules_get.schema.json"
  },
  {
    "rule_set": {
      "$ref": "../common/rule_set.schema.json"
    }
  }
]
},
"override_deny_rules": {
  "description": "Array of override deny rules",
  "type": "array",
  "items": {
    "allOf": [
      {
        "$ref": "../common/deny_rules_get.schema.json"
      },
      {
        "rule_set": {
          "$ref": "../common/rule_set.schema.json"
        }
      }
    ]
  }
},
"ip_tables_rules": {
  "type": "array",
  "items": {
    "allOf": [
      {
        "$ref": "../common/ip_tables_rules_get.schema.json"
      },
      {
        "rule_set": {
          "$ref": "../common/rule_set.schema.json"
        }
      }
    ]
  }
}
}

```

## Parameter-based Rule Search

This endpoint is used to get all `sec_rules`, `deny_rules`, and `override_deny_rules` based on parameters. It was created to extend the response of the original `allow` endpoint and preserve compatibility with existing tools.

## Policy Check

Functionality	HTTP	URI
Get all rules, deny_rules, and override_deny rules that apply to providers and destinations.	GET	api/v2/orgs/:xorg_id/sec_policy/:pversion/policy_check?<params>

## Parameters for Policy Check

**Table 6. Properties**

Property	Required
xorg_id	Yes
pversion	Yes
src_external_ip	No
src_workload	No
dst_external_ip	No
dst_workload	No
dst_container_workload	No
port	No
protocol	No
network_type	No

Schema for sec\_policy\_policy\_check\_get

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "additionalProperties": false,
    "required": ["sec_rules", "deny_rules", "override_deny_rules"],
    "properties": {
        "sec_rules": {
            "description": "Array of allow rules",
            "type": "array",
            "items": {
                "allOf": [
                    { "$ref": "../common/sec_rules_get.schema.json" },
                    { "rule_set": { "$ref": "../common/
rule_set.schema.json" } }
                ]
            }
        },
        "deny_rules": {
            "description": "Array of deny rules",
            "type": "array",
            "items": {
                "allOf": [
                    { "$ref": "../common/deny_rules_get.schema.json" },
                    { "rule_set": { "$ref": "../common/
rule_set.schema.json" } }
                ]
            }
        }
    }
}
```

```

    "type": "array",
    "items": {
      "allOf": [
        { "$ref": "../common/deny_rules_get.schema.json" },
        { "rule_set": { "$ref": "../common/rule_set.schema.json" } }
      ]
    }
  },
  "override_deny_rules": {
    "description": "Array of override deny rules",
    "type": "array",
    "items": {
      "allOf": [
        { "$ref": "../common/deny_rules_get.schema.json" },
        { "rule_set": { "$ref": "../common/rule_set.schema.json" } }
      ]
    }
  }
}
}

```

**Response Example for sec\_rules**

```
{
  "sec_rules": [
    {
      "href": "/orgs/12346/sec_policy/draft/rule_sets/39024006/sec_rules/39024014",
      "created_at": "2024-09-27T14:05:58.245Z",
      "updated_at": "2024-09-27T14:05:58.271Z",
      "deleted_at": nil,
      "created_by": {
        "href": "/users/0"
      },
      "updated_by": {
        "href": "/users/0"
      },
      "deleted_by": nil,
      "update_type": "create",
      "description": nil,
      "enabled": true,
      "providers": [
        {
          "ip_list": {
            "href": "/orgs/12346/sec_policy/draft/ip_lists/39024003",
            "created_at": "2024-09-27T14:05:50.537Z",
            "updated_at": "2024-09-27T14:05:50.548Z",
            "deleted_at": nil,
            "created_by": {
              "href": "/users/0"
            },
            "updated_by": {
              "href": "/users/0"
            }
          }
        }
      ]
    }
  ]
}
```

```

        "deleted_by": nil,
        "update_type": nil,
        "name": "Any (0.0.0.0/0 and ::/0)"
    }
}
{
    "label": {
        "href": "/orgs/12346/labels/1058977347",
        "key": "app",
        "value": "destinationApp"
    },
    "exclusion": false
}
],
"destinations": [
    {
        "label": {
            "href": "/orgs/12346/labels/1058977347",
            "key": "app",
            "value": "destinationApp"
        },
        "exclusion": false
    }
],
"consuming_security_principals": [],
"sec_connect": false,
"stateless": false,
"machine_auth": false,
"unscoped_destinations": false,
"network_type": "brn",
"use_workload_subnets": [],
"ingress_services": [
    {
        "port": 3306,
        "proto": 6
    }
],
"egress_services": [],
"resolve_labels_as": {
    "providers": [
        "workloads"
    ],
    "destinations": [
        "workloads"
    ]
}
"rule_set": {
    "href": "/orgs/12346/sec_policy/draft/rule_sets/
39024006",
        "created_at": "2025-01-22T09:43:31.326Z",
        "updated_at": "2025-01-22T09:43:31.393Z",
        "deleted_at": nil,
        "created_by": {
            "href": "/users/0"
        },
        ...
    }
}

```

```

    "updated_by": {
        "href": "/users/0"
    },
    "deleted_by": nil,
    "update_type": "create",
    "name": "Test rule",
    "enabled": true,
    "scopes": [
        []
    ],
    "caps": [
        "write",
        "provision"
    ],
    "update_label": "create"
}
},
],
"deny_rules": [
=====
], "override_deny_rules": [
=====

```

## Rule Hit Count

The Rule Hit Count feature is configured so only certain VENs can compute the rule hit counts and send the rule ID to the PCE.

### Enabling Rule Hit Count

The Rule Hit Count feature is disabled by default on all the VENs and the PCE.

To use the Rule Hit Count feature, you must first enable it on the PCE and the relevant VENs.

#### Enable Rule Hit Count on a VEN

Use the following API to enable the feature on a VEN on all scopes:

#### **PUT api/v2/orgs/:xorg\_id/sec\_policy/draft/firewall\_settings**

```
{
    "rule_hit_count_enabled_scopes": []
}
```

This sample API can be used to enable features in specific scopes. This example enables the features on all VENs with labels 7 and 12.

```
{
    "rule_hit_count_enabled_scopes": [
        [
            {
                "label": {

```

```

        "href": "/orgs/1/labels/7"
    }
}
{
    "label": {
        "href": "/orgs/1/labels/12"
    }
}
]
]
}

```

Commit or provision these DRAFT changes.

### **POST /api/v2/orgs/:xorg\_id/sec\_policy**

```

{
    "update_description": "Enable rule hit count",
    "change_subset": {
        "firewall_settings": [
            {
                "href": "/orgs/1/sec_policy/draft/firewall_settings"
            }
        ]
    }
}

```

#### **Disable the feature Rule Hit Count on all VENs:**

### **PUT api/v2/orgs/:xorg\_id/sec\_policy/draft/firewall\_settings**

The property `rule_hit_count_enabled_scopes` was added to this API:

```
{
    "rule_hit_count_enabled_scopes": []
}
```

#### **Enable Rule Hit Count on a PCE**

Use the following API to enable the feature on a PCE:

### **PUT /api/v2/orgs/:xorg\_id/report\_templates/rule\_hit\_count\_report**

```
{
    "enabled": true
}
```

#### **Generating Rule Hit Count Reports**

A Rule Hit Count report can be either a scheduled report generated on a recurrent basis or an ad-hoc report.

To generate the Rule Hit Count report, two new schemas have been introduced:  
`rule_hit_count_report_params` and `rule_set_lists`:

## Rule Hit Count Reports

A Rule Hit Count report can be either a scheduled report generated on a recurrent basis or an ad-hoc report. To generate the Rule Hit Count report, two new schemas have been introduced: `rule_hit_count_report_params` and `rule_set_lists`:

### `rule_hit_count_report_params`

The new schema returns the rule hit count statistics for all the rules in a ruleset during the specified time range.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Returns the rule hit count stats for all the rules in a ruleset during the specified time-range",
    "type": "object",
    "additionalProperties": false,
    "required": [
        "report_time_range",
        "rule_sets"
    ],
    "properties": {
        "report_time_range": {
            "description": "Time range the report is built across",
            "type": "object",
            "oneOf": [
                {
                    "$ref": "report_time_range_definitions.schema.json#/definitions/custom_date_range"
                },
                {
                    "$ref": "report_time_range_definitions.schema.json#/definitions/last_num_days"
                }
            ]
        },
        "rule_sets": {
            "$ref": "rule_set_lists.schema.json"
        },
        "max_results": {
            "description": "maximum number of rules to return in the specified time-range in descending order of rule creation time",
            "minimum": 0,
            "maximum": 200000,
            "type": "integer"
        }
    }
}
```

### `rule_set_lists`

This schema returns the rule hit count statistics for all the rules in a ruleset during the specified time range.

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
```

```

"description": "Returns the rule hit count stats for all the rules in
a
ruleset during the specified time-range",
"type": "array",
"items": {
    "type": "object",
    "additionalProperties": false,
    "required": [
        "href"
    ],
    "properties": {
        "href": {
            "description": "HREF of the ruleset",
            "type": "string"
        }
    }
}
}

```

## Generate an Ad-hoc Report

The following API can create a report for the last x number of days. The example generates a rule hit count report for all rule sets for the last 30 days.

POST /api/v2/orgs/:xorg\_id/reports

```

{
    "report_template": {
        "href": "/orgs/1/report_templates/rule_hit_count_report"
    },
    "description": "My first rule hit count report",
    "report_parameters": {
        "report_time_range": {
            "last_num_days": 30
        },
        "rule_sets": []
    },
    "send_by_email": true
}

```

The example response:

```

{
    "href": "/orgs/1/reports/d1b80240-ffa5-4e99-b2a0-c3d4946efe03",
    "report_template": {
        "href": "/orgs/1/report_templates/rule_hit_count_report",
        "name": "Rule Hit Count Report"
    },
    "description": "My first rule hit count report",
    "created_at": "2023-11-03T07:52:04.018Z",
    "updated_at": "2023-11-03T07:52:04.018Z",
    "progress_percentage": 0,
    "generated_at": null,
    "status": "pending",
    "report_parameters": {

```

```

        "report_time_range": {
            "last_num_days": 30
        },
        "rule_sets": [ ]
    },
    "send_by_email": true,
    "created_by": {
        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    }
}

```

To create a report for a custom date range, use the following:

```

{
    "report_template": {
        "href": "/orgs/1/report_templates/rule_hit_count_report"
    },
    "description": "My first rule hit count report",
    "report_parameters": {
        "report_time_range": {
            "start_date": "2023-10-03T00:00:00Z",
            "end_date": "2023-11-03T23:59:59Z"
        },
        "rule_sets": []
    },
    "send_by_email": true
}

```

## Check the Status of the Report

Use a GET API and the HREF from the POST response to check the status of the report:

GET /api/v2/orgs/:xorg\_id/reports/:report\_uuid

```

{
    "href": "/orgs/1/reports/d1b80240-ffa5-4e99-b2a0-c3d4946efe03",
    "report_template": {
        "href": "/orgs/1/report_templates/rule_hit_count_report",
        "name": "Rule Hit Count Report"
    },
    "description": "My first rule hit count report",
    "created_at": "2023-11-03T07:52:04.018Z",
    "updated_at": "2023-11-03T07:52:05.233Z",
    "progress_percentage": 100,
    "generated_at": "2023-11-03T07:52:05.233Z",
    "status": "done",
    "report_parameters": {
        "rule_sets": [],
        "report_time_range": {
            "last_num_days": 30
        }
    },
}

```

```

    "send_by_email": true,
    "created_by": {
        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    }
}

```

## Map the Ruleset HREF to the Ruleset

When you map the ruleset HREF to the ruleset, the response contains the HREF of the user:

```

{
    "href": "/orgs/2293773/sec_policy/active/rule_sets/9851624184900724",
    "created_at": "2024-07-24T18:24:31.400Z",
    "updated_at": "2024-07-31T21:46:43.166Z",
    "deleted_at": null,
    "created_by": {
        "href": "/users/9851624184872975"
    },
    "updated_by": {
        "href": "/users/9851624184872975"
    },
    "deleted_by": null,
    "name": "App19492 | Env19492",
    "description": null,
    "external_data_set": "illumio_policy_generator",
    "external_data_reference": "9851624185012145 | 9851624185012146",
    "enabled": true,
    "scopes": [
        [
            {
                "label": {
                    "href": "/orgs/2293773/labels/9851624185012145"
                },
                "exclusion": false
            },
            {
                "label": {
                    "href": "/orgs/2293773/labels/9851624185012146"
                },
                "exclusion": false
            }
        ]
    ],
    "rules": [
        {
            "href": "/orgs/2293773/sec_policy/active/rule_sets/9851624184900724/sec_rules/9851624184910709",
            "created_at": "2024-07-24T18:24:31.548Z",
            "updated_at": "2024-07-24T18:24:31.589Z",
            "deleted_at": null,
            "created_by": {
                "href": "/users/9851624184872975"
            },
            "updated_by": {
                "href": "/users/9851624184872975"
            },
            "deleted_by": null,
        }
    ]
}

```

```

"description": "",
"enabled": true,
"providers": [
  {
    "label": {
      "href": "/orgs/2293773/labels/9851624185012147"
    },
    "exclusion": false
  },
  {
    "label": {
      "href": "/orgs/2293773/labels/9851624185012144"
    },
    "exclusion": false
  }
],
"destinations": [
  {
    "label": {
      "href": "/orgs/2293773/labels/9851624185012147"
    },
    "exclusion": false
  },
  {
    "label": {
      "href": "/orgs/2293773/labels/9851624185012144"
    },
    "exclusion": false
  }
],
"consuming_security_principals": [],
"sec_connect": false,
"stateless": false,
"machine_auth": false,
"unscoped_destinations": false,
"network_type": "brn",
"use_workload_subnets": [],
"ingress_services": [
  {
    "port": 8080,
    "proto": 17
  }
],
"egress_services": [],
"resolve_labels_as": {
  "providers": ["workloads"],
  "destinations": ["workloads"]
}
},
"ip_tables_rules": [],
"deny_rules": [],
"caps": ["write", "provision"]
}

```

## Download the Report

When the report's status is completed, it is emailed to the user who created the report if the option `send_by_email` is set. Once the status of the report is set to "done," the report can be downloaded using the download API as follows:

```
GET /api/v2/orgs/:xorg_id/reports/:report_uuid/download
```

The sample response that can be saved as CSV:

```

Rule HREF,Rule Name,Rule Set HREF,Rule Set Name,Rule Hit Count,Days Since
Last Hit,
    Last Updated Timestamp,Last Updated By,Start Date,End Date
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/23,"",/orgs/1/
sec_policy/active/rule_sets/
1,Default,0,-1,2023-08-07T22:55:37-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/21,"",/orgs/1/sec_policy/
active/
rule_sets/1,Default,0,-1,2023-07-25T04:48:09-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/19,"",/orgs/1/sec_policy/
active/
rule_sets/1,Default,0,1,2023-07-25T04:35:31-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/8,"",/orgs/1/sec_policy/
active/
rule_sets/1,Default,0,-1,2023-07-21T16:34:08-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/3,"",/orgs/1/sec_policy/
active/
rule_sets/1,Default,0,1,2023-07-20T04:22:23-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/rule_sets/1/sec_rules/1,Allow outbound
connections,/
    orgs/1/sec_policy/active/rule_sets/
1,Default,0,1,2023-07-25T04:52:39-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/enforcement_boundaries/5,my test deny rule
with iplists,
    "",",0,-1,2023-07-20T03:00:05-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/enforcement_boundaries/3,ransomware_deny_rule2,"",
    ",0,1,2023-06-30T17:16:38-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z
/orgs/1/sec_policy/active/enforcement_boundaries/1,ransomware deny rule,"",
    ",0,-1,2023-06-07T23:32:07-07:00,
    /users/1,2023-10-04T00:00:00Z,2023-11-02T23:59:00Z

```

## Schedule a Recurrent Report

You need to create a report schedule to create a recurring report. In this example, the "Monthly Rule Hit Count Report" has been generated for the last 30 days and will be emailed to the person who requested it.

To create a report schedule:

```
{
  "report_template": {
    "href": "/orgs/1/report_templates/rule_hit_count_report"
  },
  "report_parameters": {
    "report_time_range": {
      "last_num_days": 30
    },
    "rule_sets": []
}
```

```

} ,
"send_by_email": true,
"report_generation_frequency": "monthly",
"name": "Monthly Rule Hit Count Report",
}

```

Other API Changes to Support the Rule Hit Count Feature:

`sec_policy_label_groups_get`

The property `rule_hit_count_enabled_scopes` was added.

```

"properties": {
    "rule_hit_count_enabled_scopes": {
        "description": "Label Group is referenced by Rule Hit
                        Count Enabled Scopes",
        "type": "boolean"
    }
}

```

`sec_policy_firewall_settings_get`

The property `rule_hit_count_enabled_scopes` was added.

```

{
    "properties": {
        "rule_hit_count_enabled_scopes": {
            "description": "Workloads that match the scope
                            will have rule hit count enabled",
            "$ref": "../common/rule_set_scopes_get.schema.json"
        }
    }
}

```

For the following APIs, a reference to the schema `rule_hit_count_report_params` was added.

`report_schedules_get`

`report_schedules_put`

`report_schedules_post`

`reports_post`

`report_templates_get`

## Custom Iptables Rules

This Public Stable API allows you to leverage preexisting iptables rules on Linux workloads and add them as rules to rulesets.

You can use the rules API to create custom iptables rules in situations where your Linux workloads have preexisting iptables rules configured that you would like to keep in addition to the rules you create using Illumio Core.

If you configured iptables on Linux workloads before using Illumio Core, when you pair a workload, the VEN assumes control of the iptables to enact policy and disables any pre-programmed iptables. To solve this, you can use the Rules API to leverage your own iptables rule configurations in a ruleset.

## Custom Iptables Rules

These terms clarify the relationship between your iptables rules and Illumio Core rules:

- **iptables:** Linux host configuration before the VEN is installed
- **Rules:** Configurations in the PCE that define the allowed communication between two or more workloads or other entities (IP lists, labels representing multiple workloads, and label groups)
- **Custom iptables rules:** PCE rules that leverage your iptables rule configurations that get programmed on your workloads by the VEN and managed by the PCE

## How Custom iptables Rules Work

Custom iptables rules in the PCE consist of a list of predefined iptables statements and the entities that receive the rule definitions. Each rule can have a list of iptables configurations, which allows you to group a sequence of rules for a specific function. Custom iptables rules are programmed after the Illumio PCE generates the iptables rules and are provisioned.

Before custom iptables rules are sent to the VEN, they are checked for any unsupported tokens (such as names of firewall chains already in use by Illumio, matching against IP sets, and semicolons). The rule cannot be saved or provisioned if an unsupported token is included.

If the VEN fails to apply a custom iptables rule because of a missing package or an incorrectly formatted rule:

- Error is reported to the PCE and is logged as two audit events:  
“Firewall config failure” (`fw_config_failure`) and  
“Failed to apply policy changes” (`policy_deploy_failed`).
- The error is displayed in the VEN health status.
- The new policy is not used, and the last known successful policy is used instead.

For policy distribution and enforcement, the VEN creates a custom chain that contains the rules for each table or chain in the iptables. Each custom chain is appended to the end of its corresponding chain in the correct table. When the VEN requests the policy, the `iptables` command is sent, including where the chain should be placed.

For security reasons, custom iptables rules only support rules in the `mangle`, `nat`, and `filter` tables.

The following table describes the permitted actions for each iptables type:

Table Name	Chain Names	Custom Rules
raw	prerouting, output	No
mangle	prerouting, input, output, forward, postrouting	Yes
nat	prerouting, output, postrouting	Yes
filter	input, output, forward	Yes
security	input, output, forward	No

## Create a Custom iptables Rule.

This method allows you to create a rule that can contain custom iptables.

Create a Custom iptables Rule.

POST [api\_version/[rule\_set\_href]]/sec\_rules

## Custom Iptables Rules Reference

This topic covers parameters, properties, and examples for iptable rules.

### Query Parameters

Parameter	Description	Type	Required
name	Ruleset name (must be unique)	String	Yes
scopes	Scope for the ruleset, which consists of a list of labels, with each list having at least one application, environment, and/or location label	Array	Yes
external_data_set	External data set identifier.	String	No
external_data_reference	External data reference identifier	String	No
enabled	Whether the ruleset is enabled or not	Boolean	Yes
rules	Standard (non-iptables) rules	String	Yes
iptables_rules	Rules that use iptables (see the following table for properties)	String	Yes

## Custom iptables\_rules Properties

Property	Description	Type	Required
enabled	Whether the rule is currently enabled	Enum	Yes
ip_version	Whether IPv4 or IPv6 is used	String	Yes
description	Description of ruleset	String	No
actors	Entities that receive the ruleset.	String	Yes
statements	Rules for iptables (table, chain name, and parameters), which consist of the following elements:	String	Yes
	<ul style="list-style-type: none"> <li>• <b>table_name</b>: Name of iptables table, which is <code>nat</code>, <code>mangle</code>, or <code>filter</code></li> <li>• <b>chain_name</b>: Name of iptables chain, which is <code>prerouting</code>, <code>input</code>, <code>output</code>, <code>forward</code>, or <code>postrouting</code></li> <li>• <b>parameters</b>: Remaining iptables rules (excluding table name and chain name)</li> </ul>		

## Examples

### Request Body

In this example, a ruleset named `test_ipt_rs` contains two iptables rules.



### NOTE

Each iptables rule can contain multiple statements.

```
{
  "name": "test_ipt_rs",
  "enabled": true,
  "scopes": [
    [
      { "label": { "href": "/orgs/1/labels/24" } },
      { "label": { "href": "/orgs/1/labels/27" } },
      { "label": { "href": "/orgs/1/labels/21" } }
    ],
    [
      {
        "enabled": true,
        "actors": [ { "label": { "href": "/orgs/1/labels/11" } } ],
        "statements": [
          {
            "table_name": "mangle",
            "chain_name": "PREROUTING",
            "parameters": "-i eth0 -p tcp --dport 2222 -j MARK --set-mark 2222"
          },
        ]
      }
    ]
  ]
}
```

```

        {
            "table_name": "nat",
            "chain_name": "PREROUTING",
            "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j
REDIRECT --to-port 3333"
        },
        {
            "table_name": "filter",
            "chain_name": "INPUT",
            "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j
ACCEPT"
        }
    ],
    "ip_version": "4"
},
{
    "enabled": true,
    "actors": [{ "actors": "ams" }],
    "statements": [
        {
            "table_name": "nat",
            "chain_name": "POSTROUTING",
            "parameters": "-o eth1 -s 192.0.2.10! -d
198.51.100.0/24 -j MASQUERADE"
        }
    ],
    "ip_version": "4"
}
]
}

```

Create Custom iptables Rule.

```

curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/
sec_policy/draft/rule_sets -H "Content-Type:application/json"
-u $KEY:$TOKEN-d '{
  "name": "test_ipt_rs",
  "enabled": true,
  "scopes": [
    []
  ],
  "ip_tables_rules": [
    {
      "enabled": true,
      "actors": [
        {
          "label": {
            "href": "/orgs/1/labels/11"
          }
        }
      ],
      "ip_version": "4"
    }
  ]
}, {
  "table_name": "mangle",
  "chain_name": "PREROUTING",
  "parameters": "-i eth0 -p tcp --dport 2222 -j MARK --set-mark 2222"
},
{
  "table_name": "nat",
  "chain_name": "PREROUTING",
  "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j REDIRECT --to-port 3333"
},
{
  "table_name": "filter",
  "chain_name": "INPUT",
  "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j ACCEPT"
}, {"ip_version": "4"}, {"enabled": true, "actors": [{"actors": "ams"}]}, {"statements": [{"table_name": "nat", "chain_name": "POSTROUTING", "parameters": "-o eth1 -s 10.0.0.2 ! -d 172.17.0.0/16 -j MASQUERADE"}], "ip_version": "4"}]'

```

Response Body

Property	Description	Type
href	Identifier for the resource	String

## Response

```
{
  "href": "/orgs/1/sec_policy/draft/rule_sets/17",
  "created_at": "2023-02-24T23:19:01.020Z",
  "updated_at": "2023-02-24T23:19:01.020Z",
  "deleted_at": null,
  "created_by": {
    "href": "/users/1"
  },
  "updated_by": {
    "href": "/users/1"
  },
  "deleted_by": null,
  "name": "test_ipt_rs",
  "description": null,
  "enabled": true,
  "scopes": [
    [
      {
        "label": { "href": "/orgs/1/labels/24" }
      },
      {
        "label": { "href": "/orgs/1/labels/27" }
      },
      {
        "label": { "href": "/orgs/1/labels/21" }
      }
    ],
    [
      {
        "label": { "href": "/orgs/1/labels/15" }
      },
      {
        "label": { "href": "/orgs/1/labels/16" }
      },
      {
        "label": { "href": "/orgs/1/labels/17" }
      }
    ]
  ],
  "rules": [],
  "ip_tables_rules": [
    {
      "href": "/orgs/1/sec_policy/draft/rule_sets/17/ip_tables_rules/20",
      "created_at": "2023-02-24T23:19:01.280Z",
      "updated_at": "2023-02-24T23:19:01.280Z",
      "deleted_at": null,
      "created_by": {
        "href": "/users/1"
      },
      "updated_by": {
        "href": "/users/1"
      },
      "deleted_by": null,
      "description": null,
      "enabled": true,
      "actors": [
        {
          "actors": "ams"
        }
      ],
      "ip_version": "4",
      "statements": [
        {
          "table_name": "nat",

```

```

    "chain_name": "POSTROUTING",
    "parameters": "-o eth1 -s 192.0.2.0 ! -d 198.51.100.0/24 -j
MASQUERADE"
}
]
},
{
  "href": "/orgs/1/sec_policy/draft/rule_sets/17/ip_tables_rules/18",
  "created_at": "2023-02-24T23:19:01.229Z",
  "updated_at": "2023-02-24T23:19:01.229Z",
  "deleted_at": null,
  "created_by": {
    "href": "/users/1"
  },
  "updated_by": {
    "href": "/users/1"
  },
  "deleted_by": null,
  "description": null,
  "enabled": true,
  "actors": [
    {
      "label": {
        "href": "/orgs/1/labels/11",
        "key": "loc",
        "value": "test"
      }
    }
  ],
  "ip_version": "4",
  "statements": [
    {
      "table_name": "filter",
      "chain_name": "INPUT",
      "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j ACCEPT"
    },
    {
      "table_name": "nat",
      "chain_name": "PREROUTING",
      "parameters": "-i eth0 -p tcp -m mark --mark 2222 -j REDIRECT
--to-port 3333"
    },
    {
      "table_name": "mangle",
      "chain_name": "PREROUTING",
      "parameters": "-i eth0 -p tcp --dport 2222 -j MARK --set-mark
2222"
    }
  ]
}
}

```

## Rule-Based Label Mapping

New APIs for managing the new feature, Rule-based label mapping, are the following:

### Label Mapping API Methods

Functionality	HTTP	URI
Returns the collection of label mapping rules.	GET	/orgs/:xorg_id/label_mapping_rules
Creates a new label-mapping rule.	POST	/orgs/:xorg_id/label_mapping_rules
Deletes multiple label mapping rules	PUT	/orgs/{org_id}/label_mapping_rules/delete
Gets the instance of a single label-mapping rule.	GET	/orgs/:xorg_id/label_mapping_rules/:label_mapping_rule_id
Updates the instance of a single rule.	PUT	/orgs/:xorg_id/label_mapping_rules/:label_mapping_rule_id
Deletes the specified label-mapping rule.	DELETE	/orgs/:xorg_id/label_mapping_rules/:label_mapping_rule_id
Reorders label-mapping rules.	PUT	/orgs/{org_id}/label_mapping_rules/{label_mapping_rule_id}/reorder
This asynchronous API runs a set of label-mapping rules on a set of workloads.	POST	/orgs/:xorg_id/label_mapping_rules/run
Gets the status of the async job to run the rules.	GET	/orgs/:xorg_id/label_mapping_rules/run/:job_uuid
Downloads the results of the run rules job.	GET	/orgs/:xorg_id/label_mapping_rules/run/:job_uuid/download
Assign labels from the results of the label-mapping rules run the job.	PUT	/:xorg_id/label_mapping_rules/run/:job_uuid/assign_labels
Bulk label update	PUT	[api_version][label_href]label_mapping_rules_update

### Bulk label update

If an organization has 500 rules defined, the UI must make up to 500 individual calls in an enable/disable rules operation.

This issue is resolved using the "bulk update" endpoint, which was chosen instead of a "bulk enable/disable" API because it provides additional flexibility to support future use cases.

`label_mapping_rule_label_assignments.schema.json`

The property `label_assignment` was deleted and replaced with a reference to `label_mapping_rule_label_assignments.schema.json` for the following APIs:

- label\_mapping\_rules\_post
- label\_mapping\_rules\_get
- label\_mapping\_rules\_put

## Rule-Based Label Mapping Reference

This topic covers examples of rule-based label mapping.

### Examples

#### Example response for GET /api/v2/orgs/:xorg\_id/label\_mapping\_rules

```
[
  {
    "href": "/orgs/1/label_mapping_rules/
48ed8903-878e-4010-859a-63d19be797c3",
    "enabled": true,
    "position": 40,
    "created_at": "2024-04-24T06:54:00.530Z",
    "updated_at": "2024-04-24T06:54:00.535Z",
    "expression": {
      "property": "hostname",
      "values": [
        "this"
      ],
      "operator": "starts_with"
    },
    "created_by": {
      "href": "/users/2"
    },
    "updated_by": {
      "href": "/users/2"
    },
    "label_assignments": [
      {
        "label": {
          "href": "/orgs/1/labels/24"
        }
      },
      {
        "label": {
          "href": "/orgs/1/labels/11"
        }
      },
      {
        "label": {
          "href": "/orgs/1/labels/20"
        }
      }
    ]
  },
  {
    "href": "/orgs/1/label_mapping_rules/d1479032-
f7cb-479f-87bd-1bc7bd816a74",
    "enabled": true,
    "position": 41,
  }
]
```

```

"created_at": "2024-04-26T18:18:10.238Z",
"updated_at": "2024-04-26T18:18:10.256Z",
"expression": {
    "property": "hostname",
    "values": [
        "perf-workload-1"
    ],
    "operator": "equals"
},
"created_by": {
    "href": "/users/1"
},
"updated_by": {
    "href": "/users/1"
},
"label_assignments": [
    {
        "label": {
            "href": "/orgs/1/labels/9"
        }
    }
],
{
    "href": "/orgs/1/label_mapping_rules/d4ddb653-56e9-4150-
a93e-8a734c510c03",
    "enabled": true,
    "position": 43,
    "created_at": "2024-04-28T00:05:38.146Z",
    "updated_at": "2024-04-28T00:05:38.150Z",
    "expression": {
        "property": "hostname",
        "values": [
            "perf-workload-2"
        ],
        "operator": "equals"
    },
    "created_by": {
        "href": "/users/2"
    },
    "updated_by": {
        "href": "/users/2"
    },
    "label_assignments": [
        {
            "label": {
                "href": "/orgs/1/labels/7"
            }
        }
    ]
}
]

```

**Example Request for POST /api/v2/orgs/:xorg\_id/label\_mapping\_rules**

```
{
    "expression": {
        "logical_operator": "and",
        "child_expressions": [
            {
                "property": "os",
                "operator": "equals",
                "values": ["linux"]
            },
            {
                "property": "process",
                "operator": "contains",
                "values": ["mysql"]
            }
        ]
    },
    "label_assignments": [
        {
            "label": {
                "key": 'os',
                "value": 'Linux'
            }
        },
        {
            "label": {
                "key": 'role',
                "value": 'Database'
            }
        }
    ]
}
```

**Example Response for POST /api/v2/orgs/:xorg\_id/label\_mapping\_rules**

```
{
    "href": "/orgs/1/label_mapping_rules/4512e359-bda3-49d1-8f9e-b9a03357e4ee",
    "enabled": true,
    "position": 2,
    "created_at": "2024-04-18T23:45:49.237Z",
    "updated_at": "2024-04-18T23:45:49.290Z",
    "expression": {
        "property": "os",
        "operator": "contains",
        "values": [
            "windows"
        ]
    },
    "created_by": {
        "href": "/users/1"
    },
    "updated_by": {
        "href": "/users/1"
    }
}
```

```

        } ,
        "label_assignments": [
            {
                "label": {
                    "key": "os",
                    "value": "Windows"
                }
            }
        ]
    }
}

```

**Example Request to Enable Multiple Rules**

```
[
    {
        "href": "/orgs/1/label_mapping_rules/84e19944-2ae1-4985-9ace-
ca2e109da80e",
        "enabled": true
    },
    {
        "href": "/orgs/1/label_mapping_rules/15c3711c-3821-4719-8c10-
a3e89966b66e",
        "enabled": true
    }
]
```

**Example Request to Disable Multiple Rules**

```
[
    {
        "href": "/orgs/1/label_mapping_rules/84e19944-2ae1-4985-9ace-
ca2e109da80e",
        "enabled": false
    },
    {
        "href": "/orgs/1/label_mapping_rules/15c3711c-3821-4719-8c10-
a3e89966b66e",
        "enabled": false
    }
]
```

**Example Request to Update Multiple Rules**

```
[
    {
        "href": "/orgs/1/label_mapping_rules/84e19944-2ae1-4985-9ace-
ca2e109da80e",
        "expression": {
            "property": "hostname",
            "operator": "equals",
            "values": [
                "foo"
            ]
        },
        "label_assignments": [
            {
                "label": {

```

```

        "href": "/orgs/1/labels/46"
    }
}
]
},
{
    "href": "/orgs/1/label_mapping_rules/15c3711c-3821-4719-8c10-
a3e89966b66e",
    "expression": {
        "logical_operator": "and",
        "child_expressions": [
            {
                "values": [
                    "/usr/bin/node"
                ],
                "operator": "equals",
                "property": "process"
            },
            {
                "values": [
                    {
                        "port": 3000,
                        "proto": 6
                    }
                ],
                "operator": "equals",
                "property": "port"
            }
        ]
    },
    "label_assignments": [
        {
            "label": {
                "href": "/orgs/1/labels/44"
            }
        }
    ]
}
]
```

## Machine Authentication

This Public Experimental API allows you to configure unmanaged workloads and rules for machine authentication if you have configured the PCE to use machine authentication.

Before you start writing rules, you need to complete the following tasks:

- Configure an unmanaged (no VEN) workload on which you want to use machine authentication with the client certificate X.509 Subject distinguished name (`distinguished_name`) issued from the CA. You do not need to set this property using machine authentication with managed workloads (with VENs installed).
- Configure rules for machine authentication by setting the `machine_auth` flag to true on each rule. You can also optionally set SecureConnect (`sec_connect`) if you want the traffic data to be encrypted using IPsec.

Once you have done these two tasks, you can use these unmanaged workloads in machine authentication-based rules.

## Configure Machine Authentication

The machine authentication workload property for the certificate distinguished name is required for those hosts or systems where you have not installed a VEN, such as a laptop or other server whose IP address is unknown or changes often.

You can set the `distinguished_name` when you first create (POST) the unmanaged workload, which is passed in the JSON request payload.



### NOTE

For information on creating an unmanaged workload, see [Create an Unmanaged Workload](#). [335]

Use this URI to configure machine authentication when you create a new unmanaged workload:

```
POST [api_version][org_href]/workloads
```

If you want to enable machine authentication on an existing unmanaged workload, you need to know the workload HREF, which can be obtained from the command GET on a collection of Workloads.

The workload HREF is highlighted in blue:

```
/orgs/7/workloads/XXXXXXXX-9611-44aa-ae06-fXXX8903db65
```

Use this URI to configure machine authentication for an existing unmanaged workload:

```
PUT [api_version][workload_href]
```

## Configure Machine Authentication on Rule

For a rule to use machine authentication, you need to configure it on the rule when you create or update it.

Use this URI to configure machine authentication for a new rule:

```
POST [api_version][rule_set_href]/sec_rules
```

If you want to enable machine authentication on an existing rule, you need to know the HREF of the rule. For example:

```
/orgs/3/sec_policy/draft/rule_sets/152/sec_rules/124
```

Use this URI to configure machine authentication for an existing rule:

```
PUT [api_version][sec_rule_href]
```

## Configure Machine Authentication on an Existing Rule

This topic covers flags to enable machine authentication and SecureConnect encryption for the rule.

**Table 7. Flags for machine authentication**

Parameter	Description
machine_auth	An optional boolean flag, set to true, enables machine authentication for the rule.
sec_connect	An optional boolean flag enables SecureConnect (host-to-host traffic encryption) for the rule.

This example shows the JSON payload for updating a rule to enable machine authentication, but with SecureConnect disabled.

```
{
  "providers": [{"label": {"href": "/orgs/1/labels/1"} }],
  "sec_connect": false,
  "destinations": [
    {
      "actors": "ams"
    },
    "consuming_security_principals": [],
    "unscoped_destinations": false,
    "description": "",
    "ingress_services": [{"proto": 6}],
    "resolve_labels_as": {
      "providers": ["workloads"],
      "destinations": ["workloads"]
    },
    "enabled": true,
    "machine_auth": true
  }
}
```

Configure Machine Authentication for a Rule:

```
curl -i -X PUT https://pce.my-company.com/api/v2/
orgs/1/sec_policy/draft/rule_sets/152/sec_rules/124 -H "Content-
Type:application/json" -u $KEY:$TOKEN -d '{
  "providers": [{"label": {"href": "/orgs/1/labels/1"} }],
  "sec_connect": false,
  "destinations": [
    {"actors": "ams"}
  ],
  "consuming_security_principals": [],
  "ingress_services": [{"proto": 6}],
  "unscoped_destinations": false,
  "description": "",
  "resolve_labels_as": {
    "providers": ["workloads"],
    "destinations": [
      "workloads"
    ]
  },
  "enabled": true,
  "machine_auth": true
}'
```

## Enforcement Boundaries



### WARNING

Enforcement Boundaries are still available in 24.2 APIs. However, they are being replaced by the Deny rules explained in the topic <https://docs.illumio.com/core/24.2/Content/Guides/security-policy/create-security-policy/rules.htm>

## Enforcement Boundaries in the REST API

The RBAC roles Global Org Owner and Global Admin can manage Enforcement Boundaries without restrictions.

You can only use Enforcement Boundaries with managed workloads. You cannot apply Enforcement Boundaries to NEN-controlled or other unmanaged workloads.

One or more ports on a workload are enforced ("port enforcement"), leaving the remaining ports unenforced. Instead of configuring workloads directly, enforcement is controlled using policies.

Workloads have to be placed in `selective` mode when using Enforcement Boundaries. Therefore, to use an Enforcement Boundary, you need to perform two separate configurations:

- Set the workload policy state to `selective`.
- Create a security policy with a scope that includes the workload.

## Enforcement Boundaries Methods

Functionality	HTTP	URI
View the configured enforcement boundaries.	GET	[api_version][org_href]/sec_policy/:version/enforcement_boundaries/:id
Edit the specified enforcement boundary.	PUT	[api_version][org_href]/sec_policy/:version/enforcement_boundaries/:id
Create a new enforcement boundary.	POST	[api_version][org_href]/sec_policy/:version/enforcement_boundaries
Delete the specified enforcement boundary	DELETE	[api_version][org_href]/sec_policy/:version/enforcement_boundaries/:id

## Enforcement Boundaries Reference

This topic covers examples of enforcement boundaries.

**WARNING**

APIs for Enforcement Boundaries are still available in release 24.2.

## Parameters

Parameter	Description	Type	Required
org_id	Organization ID	Integer	Yes
pversion	Security Policy Version	String	Yes
labels	List of lists of label URIs, encoded as a JSON string	String	No
max_results	Maximum number of Rule Sets to return	Integer	No
name	Filter by name supports partial matching	String	No
service	Service URI	String	No
service_ports.port	Specify port or port range to filter results. The range is from -1 to 65535.	String	No
service_ports.proto	Protocol to filter on	Integer	No
enforcement_boundary_id	Enforcement boundary ID	Integer	Yes

## Properties

Property	Description	Type	Re-required
href	URI of the selective enforcement rule	String	Yes
name	Name of the selective enforcement rule	String	Yes
providers	<p><code>label</code></p> <p>.....Label URI. The required parameter is <code>href</code>.</p> <p><code>label_group</code></p> <p>.....Label group URI. The required parameter is <code>href</code>.</p> <p><code>ip_list</code></p> <p>.....IP List URI. The required parameter is <code>href</code>.</p> <p><code>actors</code></p> <p>.....Label group URI. The required parameter is <code>href</code>.</p>	Array	Yes
destinations	<p><code>label</code></p> <p>.....Label URI. The required parameter is <code>href</code>.</p> <p><code>label_group</code></p> <p>.....Rule actors are all workloads ('ams').</p> <p><code>ip_list</code></p> <p>.....IP List URI. The required parameter is <code>href</code>.</p> <p><code>actors</code></p> <p>.....Rule actors are all workloads ('ams').</p>	Array	Yes
ingress_services	<p>Collection of services that are enforced</p> <p><code>port:</code></p> <p>The port number, or the starting port of a range. If unspecified, this will apply to all ports for the given protocol.</p> <p>minimum: 0, maximum: 65535</p> <p><code>to_port:</code></p> <p>Upper end of port range; this field should not be included if specifying an individual port.</p> <p>minimum: 0, maximum: 65535</p>	Array	Yes

Property	Description	Type	Re- quired
	<p>proto:</p> <p>Transport protocol (numeric) enum: 6,17</p>		
created_at	<p>Timestamp when this Enforcement Boundary was first created.</p> <p>Format date-time</p>	String	No
updated_at	<p>Timestamp when this Enforcement Boundary was last updated.</p> <p>Format date-time</p>	String	No
deleted_at	<p>Timestamp when this Enforcement Boundary was deleted</p>	String	No
created_by	<p>A user who originally created this Enforcement Boundary</p> <p>Required parameter href.</p>	String	No
updated_by	<p>A user who last updated this Enforcement Boundary</p> <p>Required parameter href.</p>	String	No
deleted_by	<p>A user who deleted this Enforcement Boundary</p> <p>Required parameter href.</p>	String	No
update_type	Type of update.	String	No
enabled	<p>For POST: The optional enabled boolean field can be provided in the payload. If it is not provided, the newly created enforcement boundary object is enabled by default.</p> <p>For PUT: The optional boolean value for the enabled field in the payload is: "enabled": true</p>	Boolean	No

## Examples

### Get Enforcement Boundaries

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/draft/enforcement_boundaries -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response

In this response, the former scope property is replaced with providers, and another property destinations was added. The required properties are: name, providers, destinations, and in-  
gress\_services(formerly enforced\_service).

```
{
  "href": "/orgs/1/sec_policy/draft/enforcement_boundaries/1",
```

```

"created_at": "2021-09-21T21:48:40.228Z",
"updated_at": "2021-09-21T21:48:40.241Z",
"deleted_at": null,
"created_by": {
    "href": "/users/1"
},
"updated_by": {
    "href": "/users/1"
},
"deleted_by": null,
"update_type": "create",
"name": "Dev to Prod separation",
"providers": [
    {
        "label": {
            "href": "/orgs/1/labels/7",
            "key": "env",
            "value": "Production"
        }
    }
],
"destinations": [
    {
        "label": {
            "href": "/orgs/1/labels/9",
            "key": "env",
            "value": "Development"
        }
    }
],
"ingress_services": [
    {
        "href": "/orgs/1/sec_policy/draft/services/1",
        "created_at": "2021-09-21T16:31:16.266Z",
        "updated_at": "2021-09-21T16:31:16.292Z",
        "deleted_at": null,
        "created_by": {
            "href": "/users/0"
        },
        "updated_by": {
            "href": "/users/0"
        },
        "deleted_by": null,
        "update_type": null,
        "name": "All Services",
        "service_ports": [
            {
                "proto": -1
            }
        ]
    }
],
"caps": [
    "write",
    "provision"
]

```

```
],
"workload_counts":{

}
}
```

Create Enforcement Boundaries

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/
draft/enforcement_boundaries -H "Content-Type: application/json" -u
$KEY:$TOKEN -d '{"name": "eb1", "providers": [{"actors": "ams"}],
"destinations": [{"actors": "ams"}], "ingress_services": [{"port": 1,
"proto": 6}]}'
```

Edit Enforcement Boundaries

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/
draft/enforcement_boundaries/1 -H "Content-Type: application/json" -u
$KEY:$TOKEN -d '{"name": "a4"}'

{
  "name": "a name here",
  "providers": [
    {"label": "/orgs/1/labels/13"},
    {"label": "/orgs/1/labels/15"},
    {"ip_list": "/orgs/1/sec_policy/draft/ip_lists/22"}
  ],
  "destinations": [
    {"actors": "ams"}
  ],
  "ingress_services": [
    {"href": "/orgs/1/sec_policy/draft/services/20"},
    {"port": 22, "proto": 6},
    {"port": 8080, "to_port": 8088, "proto": 6}
  ]
}
```

## Provisioning

Use the Public Stable Provisioning API to implement all current changes to your security policy, such as additions, changes, and deletions.

The Public Experimental Provisioning API supplies information about unprovisioned changes to security policy items.

Finally, the Policy Update Mode API controls when policy updates are applied to workloads.

## Policy Update Mode

This Public Experimental API controls when policy updates are applied to workloads.

## Overview of Policy Update Mode

The PCE has two policy update options:

- **Adaptive:** Apply policy changes as soon as you provision.
- **Static:** Apply policy changes later, such as during a scheduled maintenance window.

By default, the PCE policy update mode is set to `Adaptive`, but you can configure `Static` policy update mode for specific sets of workloads identified by scopes. Workloads that share the same labels configured for static policy update scope receive policy changes from the PCE. Still, those changes *will not be applied* until a user or an orchestration system instructs the PCE to apply those changes.

Configuring static policy update mode requires defining a scope that contains one or more environment, application, or location and role labels. If a label type is not defined in the scope, that label type is interpreted as `All`. For example, if the policy update scope is

```
Application = Checking, Location = China,
```

The PCE interprets the scope as

```
Application = Checking, Location = China, Environment = All.
```

## Methods for policy update

Functionality	HTTP	URI
Get the current policy update mode for your organization.	GET	[api_version][org_href]/sec_policy/draft/firewall_settings
Change the policy update mode for your organization.	PUT	[api_version][org_href]/sec_policy/draft/firewall_settings

### Get Policy Update Mode

You can use this method to get your organization's current policy update mode settings, which is part of your PCE security settings. This method contains a variable (`:pversion`) that can be used to return the security settings with `active` (currently provisioned) or `draft` state for your organization.

URI To Get Policy Update Mode

```
GET [api_version][org_href]/sec_policy/draft/firewall_settings
```

### Change Policy Update Mode

The Change Policy Update Mode sets your organization's draft policy update mode, which might include adding or removing a policy scope.

The draft state of your policy update mode can be modified, but not the currently active (provisioned) version. First, change to the draft policy update mode, and then provision those changes.

## URI To Change Policy Update Mode

```
PUT [api_version][org_href]/sec_policy/draft/firewall_settings
```

### Remove all Static Policy Scopes

To remove all static policy scopes, pass an empty JSON array:

```
PUT [api_version][org_href]/sec_policy/draft/firewall_settings
{ "static_policy_scopes": [ ] }
```



#### NOTE

The policy update mode is set to Adaptive when all static policy scopes are removed.

## Policy Update Mode Reference

This topic covers properties, parameters, and examples for policy update mode.

### Parameters

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
pversion	Security Policy Version	String	Yes
max_results	Maximum number of policy objects to return (per type)	Integer	No

### Properties

The current `firewall_settings` resource specifies a combination of IPsec / IKE authentication method (PSK or certificate) for SecureConnect and Machine Authentication.

Parameter	Description	Type	Required
update_type	Type of update	String	Yes
static_policy_scopes	Scopes that have static policy application mode  Reference to <code>common/rule_set_scopes_get.schema.json</code>		No
max_results	Maximum number of policy objects to return (per type)	Integer	No
ike_authentication_type	IKE authentication type to use for IPsec (SecureConnect and Machine Authentication)	String	No
allow_captive_portal_outbound	Defines whether or not to open the endpoint firewall to all out-bound traffic when the VEN discovers a captive portal scenario	Boolean	No
containers_inherit_host_policy_scopes	A workload that matches the scope will apply the policy it receives to itself and the containers it hosts.  Reference to <code>common/rule_set_scopes_get.schema.json</code>		
blocked_connection_reject_scopes	Scopes whose blocked connection action will be rejected  Reference to <code>common/rule_set_scope_get.schema.json</code>		
loopback_interfaces_in_policy_scopes	Workloads that match the scope will apply policy on loopback interfaces and the loopback interface's IPs will be distributed to peers.  Reference to <code>common/rule_set_scope_get.schema.json</code>		

## Draft or Active Policy Update Mode

Variable	Description
:pversion	Allows you to get:  <ul style="list-style-type: none"> <li>• <b>active:</b> The currently provisioned security settings, including policy update mode</li> <li>• <b>draft:</b> The draft state of any changed security settings that have not yet been provisioned, including policy update mode</li> </ul>

## Examples

### Curl Command Get Active Policy Update Mode

This curl example gets the active (currently provisioned) security settings for your organization, which includes the policy update mode settings.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/sec_policy/active/firewall_settings -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response Body

The `static_policy_scopes` property in the response (in blue ) indicates that two static scopes have been configured for policy update.

Each scope is defined as a JSON array of labels, which includes an Application, Environment, and a Location label. The labels in the scope are identified by their HREFs.

```
{
  "href": "/orgs/7/sec_policy/active/firewall_settings",
  "created_at": "2015-10-23T22:01:01.151Z",
  "updated_at": "2017-09-02T19:08:55.623Z",
  "deleted_at": null,
  "created_by": { "href": "/users/0" },
  "updated_by": { "href": "/users/14" },
  "deleted_by": null,
  "update_type": null,
  "allow_dhcp_client": true,
  "log_dropped_multicast": true,
  "log_dropped_broadcast": false,
  "allow_traceroute": true,
  "allow_ipv6": true,
  "allow_igmp": false,
  "track_flow": true,
  "system_rule_log_flow": false,
  "allow_path_mtu_discovery": true,
  "network_detection_mode": "single_private_brn",
  "static_policy_scopes": [
    [
      { "label": { "href": "/orgs/7/labels/83" } },
      { "label": { "href": "/orgs/7/labels/86" } },
      { "label": { "href": "/orgs/7/labels/94" } }
    ],
    [
      { "label": { "href": "/orgs/7/labels/82" } },
      { "label": { "href": "/orgs/7/labels/100" } },
      { "label": { "href": "/orgs/7/labels/89" } },
      { "label": { "href": "/orgs/7/labels/94" } }
    ]
  ],
  "secure_connect_certs": {
    "default_issuer_name_match": "test",
    "scoped_certificates": []
  }
}
```

Properties to change the policy update.

Property	Description	Type	Re-required
static_policy_scopes	<p>A set of up to four labels, one or more of the type Application, Environment, Role, and Location.</p> <p>Each label in the policy scope is identified by its HREF, nested in a JSON array.</p> <p>Before updating the organization policy update mode, make sure you have the exact set of labels you want to use and their HREFs.</p>	JSON array of strings	Yes

## Request Body

This example shows the request body for two policy update scopes. The first has a single label scope, and the second scope has a set of three labels.

```
{
  "static_policy_scopes": [
    [
      { "label": { "href": "/orgs/1/labels/8" } }
    ],
    [
      { "label": { "href": "/orgs/1/labels/2" } },
      { "label": { "href": "/orgs/1/labels/8" } },
      { "label": { "href": "/orgs/1/labels/11" } }
    ]
  ]
}
```

## Curl Command to Update Policy Update Mode

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/7/
firewall_settings -H "Content-Type: application/json" -u $KEY:$TOKEN -d
'{"static_policy_scopes": [{"label": {"href": "/orgs/1/labels/8"}}, {"label": {"href": "/orgs/1/labels/2"}}, {"label": {"href": "/orgs/1/labels/8"}}, {"label": {"href": "/orgs/1/labels/11"}}]}'
```

## Response

The response for a successful change to your policy update mode is an HTTP 204 No Content Operation. No data is returned.

## Curl Command to Remove Static Policy Scopes

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/7/
firewall_settings -H "Content-Type: application/json" -u $KEY:$TOKEN -d
'{"static_policy_scopes": []}'
```

## Provisioning (Public Stable)

This Public Stable API provisions all current changes (additions, changes, and deletions) to your security policy.

This API can also return a collection of provisioning or individual provisioning versions.

To get information about un-provisioned changes to security policy items, find provisioning dependencies, delete un-provisioned security policy items, revert the last provisioned items, and check whether a security rule exists that allows communications between two workloads. See [Provisioning - Public Experimental \[238\]](#).

## Provisioning API Methods

Functionality	HTTP	URI
Provision of the current set of modified security policy items.	POST	[api_version][org_href]/sec_policy
Get a list of all provisioned security policy versions.	GET	[api_version][org_href]/sec_policy
Get a specific version of a provisioned security policy.	GET	[api_version][sec_policy_version_href]

### Provision All Items

Policy item additions, modifications, and deletions must be provisioned before they affect workloads.

URI to Provision All Items

```
POST api_version][org_href]/sec_policy
```

### Get an Individual Provision Version

This method gets a specific version of a provisioned policy.

Every time a security policy is provisioned, it gets a unique version ID, which takes the form of an HREF. This HREF can be obtained from a GET of all security policy provisioned versions and then used in this call.

URI to Get an Individual Version of a Provisioned Policy

```
GET [api_version][sec_policy_version_href]
```

## Provisioning Reference (Public Stable)

This topic covers examples of public stable provisioning API.

### Examples

Provision All Items

This example passes a provisioning comment using the curl -d option (lowercase d) followed by the comment '{ "update\_description": "make active" }'. This operation provisions all draft policy items.

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy -H "Content-Type: application/json" -u $KEY:$TOKEN -d '{"update_description": "make active"}'
```

### Response

After provisioning the draft security policy, the response provides information related to the operation, including the version HREF of the provisioning.

You can use a provision history HREF to get all modified items for a particular version.

The response also indicates how many workloads were affected, when the provisioning was done, which user did it, and any provided message.

```
{
  "href": "/orgs/2/sec_policy/80",
  "commit_message": null,      "version": 80,
  "workloads_affected": 3,
  "object_counts": 3,
  "created_at": "2020-26T21:48:46.446Z",
  "created_by": { "href": "/users/18" }
}
```

## Provision Individual Items

### Curl Example

The request body uses update\_description instead of commit\_message, and instead of entities, define an array of pending HREFs for each method as appropriate.

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{"change_subset": {"rule_sets": [{"href": "/orgs/2/sec_policy/draft/rule_sets/843"}], "ip_lists": [{"href": "/orgs/2/sec_policy/draft/ip_lists/151"}]}, "update_description": "Provisioning a ruleset and an ip list"}'
```

### Request Body Prototype

The security policy POST request body has this format. Only define the methods used in the call and don't include any unused methods in the request body.

```
{
  "update_description": "string",
  "change_subset": {
    "label_groups": [
      {
        "href": "string"
      }
    ]
  }
}
```

```

] ,
"services": [
    {
        "href": "string"
    }
],
"rule_sets": [
    {
        "href": "string"
    }
],
"ip_lists": [
    {
        "href": "string"
    }
],
"virtual_services": [
    {
        "href": "string"
    }
],
"firewall_settings": [
    {
        "href": "string"
    }
],
"enforcement_boundaries": [
    {
        "href": "string"
    }
],
"secure_connect_gateways": [
    {
        "href": "string"
    }
],
"virtual_servers": [
    {
        "href": "string"
    }
]
]

```

## Restore the Previous Security Policy

This API creates draft changes of the previous security policy's changes. When this API is called, the draft changes should not be present in the PCE.

Curl Command to Restore the Security Policy

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/127/
restore -H "Content-Type: application/json" -u $KEY:$TOKEN -d {}
```

## Get All Provision Versions

This method gets the full history of all provisioned security policy versions.

### URI to Get All Provisioned Versions

```
GET [api_version][org_href]/sec_policy
```

### Get the Provision Versions

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/127/restore -H "Content-Type: application/json" -u $KEY:$TOKEN -d {}
```

### Response

Note that the field selective\_enforcement\_rules was renamed to enforcement\_boundaries in the object\_counts property.

```
{
    "href": "string",
    "version": "string",
    "workloads_affected": 0,
    "commit_message": "string",
    "object_counts": {
        "rule_sets": 0,
        "ip_lists": 0,
        "services": 0,
        "virtual_services": 0,
        "label_groups": 0,
        "virtual_servers": 0,
        "firewall_settings": 0,
        "secure_connect_gateways": 0,
        "enforcement_boundaries": 0
    },
    "created_at": "string",
    "created_by": {
        "href": "string"
    }
}
```

### Curl Command to Get Version

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/79 -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response

```
{
    "href": "string",
    "version": "string",
    "workloads_affected": 0,
    "commit_message": "string",
    "object_counts": {
        "rule_sets": 0,
        "ip_lists": 0,
        "services": 0,
```

```
"virtual_services": 0,  
"label_groups": 0,  
"virtual_servers": 0,  
"firewall_settings": 0,  
"secure_connect_gateways": 0,  
"enforcement_boundaries": 0  
},  
"created_at": "string",  
"created_by": {  
    "href": "string"  
}  
}
```

## Provisioning (Public Experimental)

This Public Experimental API gets information about un-provisioned changes to security policy items (rulesets, IP lists, security settings, labels and label groups, services, virtual services, and user groups). You can also find provisioning dependencies, delete unprovisioned security policy items, revert the last provisioned items, and check whether a security rule exists that allows communications between two workloads.

To provision security policy items and get information about one or more provisioned items, see "Provisioning - Public Stable".

## Provisioning API Methods

Functionality	HTTP	URI
Get the collection of modified ( <code>draft</code> ) security policy items pending provisioning.	GET	<code>[api_version][org_href]/sec_policy/pending</code>
Check whether a rule exists between two workloads that allow communication.	GET	<code>[api_version][sec_policy_version_href]/allow</code>
Get the collection of all policy items that were modified in a specific version of a security policy.	GET	<code>[api_version][sec_policy_version_href]/modified_objects</code>
Delete all un-provisioned security policy item modifications (all un-provisioned <code>draft</code> changes) pending provisioning.	DELETE	<code>[api_version][org_href]/sec_policy/pending</code>
Revert a specified list of pending uncommitted security policy items.  This method allows you to select specific items to revert.	PUT	<code>[api_version][org_href]/sec_policy/delete</code>
Determine if a specific set of objects can be provisioned or if they depend on other objects that need to be provisioned.	POST	<code>[api_version]/sec_policy/draft/dependencies</code>
Used to see the policy impact before provisioning.  This API is referencing <code>sec_policy_change_subset.schema.json</code> , which contains the property <code>change_subset</code>	POST	<code>[api_version]/sec_policy/impact</code>

## Provisionable Policy Items

The following security policy items require provisioning before they can affect managed workloads (workloads with a VEN installed). The total sum of these policy items constitutes the security policy.

- **IP Lists:** IP addresses, IP ranges, and CIDR blocks allowed to access managed workloads.
- **Label Groups:** Labels can be managed in label groups.
- **Rulesets:** Policy items that include labels and rules to define permitted communication between workloads and between groups.
- **Pairing Profiles** A Pairing Profile applies certain properties to workloads as they pair with the PCE, such as labels and workload policy states.
- **Security Settings:** General network security settings, such as ICMP echo reply, allow or disable IPv6, and connectivity settings.
- **Services:** Definitions or discovery of existing services on your workloads.
- **Virtual Servers:** Allows rules that allow communication with workloads managed by a load balancer.
- **Virtual Services:** A virtual service is a single service (a port/protocol set) that can be used directly in a rule as a single entity. Labels that represent multiple virtual services can also be used to write rules.
- **Enforcement Boundaries:** By narrowing the scope for segmentation, enforcement boundaries facilitate the implementation of allow lists, enabling users to achieve a high level of system maintainability using a simple policy mode.

When the security policy is provisioned, the PCE recalculates any changes made to policy configurations and then transmits those changes to the VENs installed on the workloads.

## Policy Provisioning States

This API operates on provisionable objects, which exist in either a `draft` (not provisioned) state or an `active` (provisioned) state.

Provisionable items include label groups, services, rulesets, IP lists, virtual services, firewall settings, enforcement boundaries, and virtual servers. For these objects, the URL of the API call must include the element called `:pversion`, which can be set to either `draft` or `active`.

Depending on the method, the API follows these rules:

- For GET operations — `:pversion` can be `draft`, `active`, or the ID of the security policy.
- For POST, PUT, DELETE — `:pversion` can be `draft` (you cannot operate on active items) or the ID if the security policy.

### Get All Items Pending Provisioning

This method gets a list of all modified policy items pending provisioning.

URI to Get All Policy Items Pending Provisioning

This API allows the user to view a list of all policy objects pending provisioning bucketed by type. The UI uses this to generate the "draft changes" page.

```
GET [api_version][org_href]/sec_policy/pending
```

### Revert All Items Pending Provisioning

This method reverts (undoes) the current set of non-provisioned security policy modifications (all non-provisioned draft changes).

```
DELETE [api_version][org_href]/sec_policy/pending
```

### Get Security Policy Dependencies

This public experimental API allows the user to determine a particular policy object's provisioning (or revert) dependencies. An object also buckets the response JSON and has the same schema change.

URI to Get Specific Security Policy Dependencies

```
POST /sec_policy/draft/dependencies
```

### Get Rules Allowing Communication

This method gets a list of all rules that allow communication between two workloads (and other entities) for a specific version of a provisioned security policy.

By default, the maximum number returned on a GET collection with this API is 500.

## Check for Rules Between Workloads

```
GET /api/v2/orgs/{org_id}/sec_policy/{pversion}/allow
```

## Revert a List of Items Pending Provisioning

This API allows the user to revert a subset of policy objects via the change\_subset field. via the change\_subset field.

The field selective\_enforcement\_rules was replaced with enforcement\_boundaries.

## Revert a Specific List of Items Pending Provisioning

```
PUT [api_version][org_href]/sec_policy/delete
```

## Get Modified Items in a Provisioned Version

This method collects all modified policy items in a specific security policy version.

Every time the security policy is provisioned, it gets a version in the form of an HREF. You can obtain the HREF by getting all provisioned versions of your security policy. You can use that provision version HREF when calling this method.

## URI to Get All Modified Items in a Specific Provisioned Version

```
GET [api_version][sec_policy_version_href]/modified_objects
```

## Provisioning (Public Experimental) Reference

This topic covers examples of public experimental APIs for provisioning.

### Examples

#### Get Items Pending Provisioning

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/pending -H "Accept:application/json" -u $KEY:$TOKEN
```

#### Response

```
-----  
],  
"virtual_services": [  
  {  
    "name": "string",  
    "href": "string",  
    "updated_by": null,  
    "updated_at": "2021-05-03T00:24:56Z",  
    "update_type": "create",  
    "caps": [  
      "write"  
    ]  
  }  
]
```

```

        }
    ],
"
    enforcement_boundaries
": [
{
    "name": "string",
    "href": "string",
    "updated_by": null,
    "updated_at": "2021-05-03T00:24:56Z",
    "update_type": "create",
    "caps": [
        "write"
    ]
}
]
}

```

The field `selective_enforcement_rules` was replaced with `enforcement_boundaries`.

#### Revert a Specific List of Items Pending Provisioning

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/pending -u $KEY:$TOKEN
```

```
{
    "change_subset": {
        "label_groups": [
            {
                "href": "string"
            }
        ],
        "services": [
            {
                "href": "string"
            }
        ],
        "rule_sets": [
            {
                "href": "string"
            }
        ],
        "ip_lists": [
            {
                "href": "string"
            }
        ],
        "virtual_services": [
            {
                "href": "string"
            }
        ],
        "firewall_settings": [
            {
                "href": "string"
            }
        ]
    }
}
```

```

        }
    ],
    "secure_connect_gateways": [
        {
            "href": "string"
        }
    ],
    "virtual_servers": [
        {
            "href": "string"
        }
    ],
    "enforcement_boundaries": [
        {
            "href": "string"
        }
    ]
}
}

```

If an empty request body is given,

```
{}
```

then all objects will be reverted.

#### Curl Command to Revert a Pending Rule

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/1/sec_policy/
delete -H "Accept: application/json" -H "Content-Type: application/json" -u
api_1fc24761346777702:'26c55be6892762b65f27aacc795076767f16ffcd7e9fde323a307
e5fd286eb8d' -d '{"change_subset":{"rule_sets":[{"href":"/orgs/1/sec_policy/
draft/rule_sets/3"}]}}'
```

## Security Policy Properties

Parameter	Description
change_subset	<p>Defines a hash of provisionable or revertible objects identified by their HREFs.</p> <p>Includes label groups, services, rulesets, IP lists, virtual services, and virtual servers.</p> <p>Each individual object of a specific type (for example, <code>rule_sets</code>) is represented in the request body as an array of HREFs for those object types.</p> <p>For <code>POST /api/v2/orgs/:xorg_id/sec_policy/impact</code>:</p> <ul style="list-style-type: none"> <li>If provided, the impact will be calculated only on <code>change_subset</code>.</li> <li>If missing, the impact will be calculated on all of the pending items.</li> </ul>
operation	<p>Determines if there are dependencies for <i>provisioning</i> or <i>reverting</i> the specified objects:</p> <ul style="list-style-type: none"> <li><code>commit</code>: Specify this value to check for dependencies before <i>provisioning</i> an object.</li> <li><code>revert</code>: Specify this value to check for dependencies before <i>reverting</i> an object that is in a <i>draft</i> state.</li> </ul> <p>Subproperties of <code>change_subset</code> that represent provisionable objects</p>
label_groups	List of label groups in the draft state to check for provisioning dependencies identified by label group HREF.
services	List of services in the draft state to check for provisioning dependencies identified by service HREF.
rule_sets	List of rulesets in the draft state to check for provisioning dependencies identified by rule_set HREF.
ip_lists	List of IP lists in the draft state to check for provisioning dependencies, identified by IP list HREF.
virtual_services	List of virtual services in the draft state to check for provisioning dependencies identified by virtual service HREF.
	Reference to <code>common/href_object.schema.json</code>
virtual_servers	List of virtual servers in the draft state that you want to check for provisioning dependencies identified by virtual server HREF.
	Reference to <code>common/href_object.schema.json</code>
firewall_settings	Reference to <code>common/href_object.schema.json</code>
enforcement_boundaries	Reference to <code>common/href_object.schema.json</code>

## Request Body

```
{
  "operation": "commit",
  "change_subset": {
    "enforcement_boundaries": [
      {
        "id": "12345"
      }
    ]
  }
}
```

```

        "href": "/orgs/2/sec_policy/draft/enforcement_boundaries/51"
    }
]
}
}
}

```

Check for Provisioning Dependencies

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/7/sec_policy/draft/dependencies -H "Content-Type: application/json" -u $KEY:$TOKEN -d '{"operation": "commit", "change_subset": {"rule_sets": [{"href": "/orgs/1/sec_policy/draft/rule_sets/9"}, {"href": "/orgs/1/sec_policy/draft/rule_sets/3"}], "virtual_services": [{"href": "/orgs/1/sec_policy/draft/virtual_services/xxxxxxxxx-adeb-4895-8ff2-60c5b9833d9e"}, {"href": "/orgs/1/sec_policy/draft/virtual_services/xxxxxxxxx-12bc-4cfa-99ef-330c399bc78c"}]}'
```

Response

The response indicates that the field `selective_enforcement` was replaced with `enforcement_boundaries` following the change in the request.

```

"$ref": "../common/href_object.schema.json"
},
},
-
"selective_enforcement_rules": {
+
"enforcement_boundaries": {
"type": "array",
"items": {
"$ref": "../common/href_object.schema.json"
}
}
}

```

If there are no dependencies for either commit or revert, the response returns an empty array.

[ ]

Curl command example: Get all modified items in a specific provisioned version

```
curl -X GET /orgs/{org_id}/sec_policy/{pversion}/modified_objects -u $KEY:$TOKEN -H 'Accept: application/json'
```

Response (similar to the following)

```
{
  "update_type": null,
  "object_type": null,
  "href": null,
  "name": "string",
  "updated_at": "2021-05-03T00:24:56Z",
  "updated_by": null,
}
```

Required properties `updated_at` and `updated_by` have been added and `modified_by` and `modified_at` have been deleted.

Provide query parameters in the URI that specify the source workload IP address or HREF, the service HREF, and the destination workload HREF. You can obtain a workload HREF with a [GET call on the Workloads API \[333\]](#).

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
pversion	Security policy version	String	Yes
src_external_ip	The external IP of the source workload	String	No
OR	or		
src_workload	The URI of the source workload		
dst_external_ip	The external IP of the destination workload	String	No
OR	OR		
dst_workload	The URI of the destination workload		
service	The specific service to check	String	No
port	The specific port number to check	Integer	No
protocol	The specific protocol number to check	Integer	No

### Curl Command to Get Rules Between Workloads

The workloads and the service are identified by their HREFs:

```
curl -X GET /orgs/{org_id}/sec_policy/{pversion}/allow -u $KEY:$TOKEN -H
'Accept: application/json'
```

### Response

```
[
  {
    "href": "string",
    "enabled": true,
    "description": "string",
    "service": {
      "href": "string"
    },
    "ub_service": null,
    "sec_connect": true,
    "providers": [
      {
        "actors": "string",
        "label": {
          "href": "string"
        },
        "label": {
          "label": "string"
        }
      }
    ]
  }
]
```

```

"agent": {
    "href": "string"
},
"workload": {
    "href": "string"
},
"bound_service": {
    "href": "string"
},
"virtual_server": {
    "href": "string"
},
"ip_list": {
    "href": "string"
}
}
],
"destinations": [
{
    "actors": "string",
    "label": {
        "href": "string"
    },
    "agent": {
        "href": "string"
    },
    "workload": {
        "href": "string"
    },
    "bound_service": {
        "href": "string"
    },
    "ip_list": {
        "href": "string"
    }
}
]
}

```

Example for POST /api/v2/orgs/1/sec\_policy/impact

Each of the allowed properties such as `ip_lists`, `label_groups`, and `services` can be included in the request body of the POST call and the response schema defines the format and values of this API request for the example in the request body.

`sec_policy_impact_post_response.schema.json`

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "required": [
        "num_sets",
        "num_managed_workloads",
        "num_container_workloads",
        "num_unmanaged_workloads"
    ],
    "properties": {

```

```

"num_sets": {
  "description": "number of affected sets",
  "type": "integer"
},
"num_virtual_servers": {
  "description": "number of affected virtual servers",
  "type": "integer"
},
"num_managed_workloads": {
  "description": "number of affected workloads of type Workload",
  "type": "integer"
},
"num_container_workloads": {
  "description": "number of affected workloads of type
ContainerWorkload",
  "type": "integer"
},
"num_unmanaged_workloads": {
  "description": "number of affected unmanaged workloads",
  "type": "integer"
},
"all_workloads_optimization": {
  "description": "flag to indicate if all-workloads-optimization has
been used",
  "type": "boolean"
}
}
}

```

## Virtual Server Filtering

Filtering of the discovered virtual servers and draft virtual server endpoints makes it easier to manage large numbers of virtual servers.

The existing Public Experimental API endpoints for virtual servers have been changed to support the required filters and associated UI operations. You can now filter a discovered virtual server collection by:

- name
- SLB (API uses href as per conventions)
- VIP: IP, proto, port (any or all)
- virtual server href

## Virtual Server Endpoints

New filters have been added for the following existing endpoints:

- GET /orgs/:xorg\_id/discovered\_virtual\_servers
- GET /orgs/:xorg\_id/sec\_policy/:pversion/virtual\_servers

**NOTE**

These Interface endpoints are available only for API version V2.

## **Virtual Server Discoveries**

Virtual server discovery happens passively once the Server Load Balancer (SLB) is configured and the Network Enforcement Node (NEN) receives the SLB configuration changes. However, users might want to be able to run virtual server discovery on demand.

The new schema `network_enforcement_nodes_virtual_server_discovery_jobs_put.schema.json` is used to create a virtual server discovery job request that contains the `slb_name`, the virtual server `ip_address`, and the port. NEN picks up the request, launches the discovery of the virtual server information, and posts the results back.

### **Discovery Job On-demand**

Use the following API:

```
POST /api/v2/orgs/1/network_enforcement_nodes/virtual_server_discovery_jobs
```

where the required properties are:

`slb_name`

- Description: Name of the SLB to interrogate.
- Format: String

`virtual_server_infos`

- Description: An array of `virtual_server_info` objects consisting of `virtual_server` port and IP address
- Format: Array of Objects

### **Check the Status of Discovery Job**

To find out the results of the discovery request use the following command:

```
GET /api/v2/orgs/1/network_enforcement_nodes/
virtual_server_discovery_jobs/:job_uuid
```

## Discovered Virtual Servers

Filter	URI Example	Notes
name	/discovered_virtual_servers?name	Supports partial and incomplete matches
slb	/discovered_virtual_servers?slb= /orgs/1/slbs/<uuid>	
vip	/discovered_virtual_servers ? vip=10.1	Supports suffix matches, e.g. 10.1 matches any IP address that starts with "10.1", "10.100", ... but not "110.x"
vip-proto	/discovered_virtual_servers? vip_proto=6	
vip_port	/discovered_virtual_servers? vip_port=80	
has_virtual_server	/discovered_virtual_servers?has_virtual_server=true	The virtual_server_mode and virtual_server_labels MUST be used with has_virtual_server=true, otherwise an error will be raised.
virtual_server_mode	/discovered_virtual_servers?virtual_server_mode=enforced	Options for this filter are "unmanaged" or "enforced"
virtual_server_labels	/discovered_virtual_servers? virtual_server_labels=[[/orgs/1/labels/2, /orgs/1/labels/3], [/orgs/1/labels/4]]	(JSON encoded array of arrays)
virtual_server	/discovered_virtual_servers ? virtual_server = /orgs/1/sec_policy/draft/virtual_servers/<uuid>	

## Virtual Servers

Filter	URI Example	Notes
name	/virtual_servers?name=myvip	Supports partial and incomplete matches
slb	/virtual_servers?slb=/orgs/1/slbs/<uuid>	
vip	/virtual_servers?vip=10.1	Supports suffix matches, e.g., 10.1 matches any IP address that starts with "10.1", "10.100", ... but not "110.x"
vip-proto	/virtual_servers?vip_proto=6	
vip_port	/virtual_servers?vip_port=80	
mode	/virtual_servers?mode=enforced	Options for this filter are "unmanaged" or "enforced"
labels	/virtual_servers?[[/orgs/1/labels/2, /orgs/1/labels/3], [/orgs/1/labels/4]] (JSON encoded array of arrays)	
discovered_virtual_server	/virtual_servers?discovered_virtual_server =/orgs/1/discovered_virtual_servers/<uuid>	

## Schema Changes

discovered\_virtual\_servers

The following object has been added to the schema:

```
{
[... existing fields ...]
"virtual_server" : {
    "href": "/orgs/1/sec_policy/draft/virtual_servers/fbae7cd2-04c3-4d7b-a628-2d69a9d64a71",
    "update_type" : "create", # or "update", "delete", null
    "mode": "enforced", # or "unmanaged"
    "labels" [
        { "href": "/orgs/1/labels/2", "key": "role", "value": "database" },
        { "href": "/orgs/1/label/12", "key": "env", "value": "production" }
    ]
}
}
```

virtual\_servers

The "mode" and "vip\_port" fields have been added to the "discovered\_virtual\_server sub-object" to reflect the filtering result.

{

```
[... existing fields ...]
"discovered_virtual_server" : {
    "dvs_identifier" : "5111ecf75c61544720d800cce97a624d" ,
    "href" : "/orgs/1/discovered_virtual_servers/c1cd1f00-7b48-4c43-
a099-f758acla9b40" ,
    "mode" : "snat" ,
    "name" : "Common/vip1" ,
    "vip_port" : {
        "port" : "80" ,
        "protocol" : 6 ,
        "vip" : "10.0.0.109"
    }
}
}

slb_config
```

This schema has been deprecated. It was used for `nfc` (Network Function Controller), which is now deprecated.

## Virtual Server Filtering Reference

This topic covers examples of virtual server filtering.

### Examples

#### Curl Command for Discovered Virtual Servers

```
curl -i -u
api_1bbac8b7295e9b512:343461267jks009651245343461267jks00965124b27074fa181f1
edb3bb4a3 https://2x2testvc27.ilabs.io:8443/api/v2/orgs/1/
discovered_virtual_servers
```

#### Response Body

```
[ {
    "href": "/orgs/1/discovered_virtual_servers/52044aea-14db-4510-
a1c6-00231230034",
    "dvs_identifier": "96803bd07185cd093dd800231230034",
    "name": "Common/QL_VIP_1",
    "nfc": {
        "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
    },
    "slb": {
        "href": "/orgs/1/slbs/84a1cd93-142f-480d-b9f8-00231230034"
    },
    "vip_port": [
        "vip": "172.16.27.88",
        "protocol": 6,
        "port": "8080"
    ],
    "local_ips": ["172.16.26.18", "172.16.27.18"],
    "mode": "snat",
    "snat_type": "snat_pool",
    "snat_pool_ips": ["172.16.26.27", "172.16.26.18", "172.16.27.18"],
```

```

"service_checks": [ {
    "protocol": 1
},
"created_at": "2021-02-26T08:32:02.131Z",
"updated_at": "2021-02-26T08:32:02.131Z",
"created_by": {
    "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
},
"updated_by": {
    "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
}
},
{
    "href": "/orgs/1/discovered_virtual_servers/073c40ec-7357-44f4-
a66d-002312300349",
    "dvs_identifier": "b679034796cdde929a000231230034",
    "name": "Common/QL_VIP_2",
    "nfc": {
        "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
    },
    "slb": {
        "href": "/orgs/1/slbs/84a1cd93-142f-480d-b9f8-00231230034"
    },
    "vip_port": {
        "vip": "172.16.27.71",
        "protocol": 6,
        "port": "8080"
    },
    "local_ips": ["172.16.26.18", "172.16.27.18"],
    "mode": "snat",
    "snat_type": "snat_pool",
    "snat_pool_ips": ["172.16.26.28", "172.16.26.18", "172.16.27.18"],
    "service_checks": [ {
        "protocol": 1
    }],
    "created_at": "2021-02-26T08:32:02.177Z",
    "updated_at": "2021-02-26T08:32:02.177Z",
    "created_by": {
        "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
    },
    "updated_by": {
        "href": "/orgs/1/nfcs/0bcf6c3d-f588-44c7-a269-00231230034"
    }
}
]

```

### Response Body, another example

```
[
{
    "href": "/orgs/1/discovered_virtual_servers/5db1ce10-263a-44fb-8c0c-
a2312dfb2e6a",
    "dvs_identifier": "the_test_dvs-1",
    "name": "Test DVS No. 1",
    "vip_port": {
        "vip": "30.55.148.143",
        "port": "8080"
    }
}
```

```

    "protocol": 6,
    "port": "8001"
},
"local_ips": [
    "10.0.0.1"
],
"mode": "snat",
"slb": {
    "href": "/orgs/1/slbs/8798cea8-1fd4-40e3-a2f1-adae6f094766"
},
"nfc": {
    "href": "/orgs/1/nfcs/7c6ae23f-5532-41b0-9ble-cf9c689de0ae"
},
"network_enforcement_node": {
    "href": "/orgs/1/network_enforcement_nodes/7c6ae23f-5532-41b0-9ble-
cf9c689de0ae"
},
"snat_type": "snat_local_ips",
"service_checks": [],
"created_at": "2022-09-09T22:19:49.915Z",
"updated_at": "2022-09-09T22:19:49.915Z",
"created_by": {
    "href": "/users/0"
},
"updated_by": {
    "href": "/users/0"
},
"virtual_server": {
    "href": "/orgs/1/sec_policy/draft/virtual_servers/
ba700305-29b9-4d6a-8bed-c8476753c327",
    "update_type": null,
    "mode": "enforced",
    "labels": [
        {
            "href": "/orgs/1/labels/14",
            "key": "role",
            "value": "LBL-ROLE-1"
        },
        {
            "href": "/orgs/1/labels/15",
            "key": "loc",
            "value": "LBL-LOC-1"
        }
    ]
}
{
    "href": "/orgs/1/discovered_virtual_servers/de5f454e-e95b-40c8-a128-
fc27a1bed840",
    "dvs_identifier": "the_test_dvs-2",
    "name": "Test DVS No. 2",
    "vip_port": {
        "vip": "34.92.48.237",
        "protocol": 6,
        "port": "8002"
    },

```

```

"local_ips": [
    "10.0.0.2"
],
"mode": "snat",
"slb": {
    "href": "/orgs/1/slbs/8798cea8-1fd4-40e3-a2f1-adae6f094766"
},
"nfc": {
    "href": "/orgs/1/nfcs/7c6ae23f-5532-41b0-9ble-cf9c689de0ae"
},
"network_enforcement_node": {
    "href": "/orgs/1/network_enforcement_nodes/7c6ae23f-5532-41b0-9ble-
cf9c689de0ae"
},
"snat_type": "snat_local_ips",
"service_checks": [],
"created_at": "2022-09-09T22:19:49.919Z",
"updated_at": "2022-09-09T22:19:49.919Z",
"created_by": {
    "href": "/users/0"
},
"updated_by": {
    "href": "/users/0"
},
"virtual_server": {
    "href": "/orgs/1/sec_policy/draft/virtual_servers/
e1502bf3-0992-4167-b84f-eaebd73cc2d7",
    "update_type": null,
    "mode": "enforced",
    "labels": [
        {
            "href": "/orgs/1/labels/28",
            "key": "role",
            "value": "LBL-ROLE-2"
        },
        {
            "href": "/orgs/1/labels/29",
            "key": "loc",
            "value": "LBL-LOC-2"
        }
    ]
}
}

```

### Curl Command for Virtual Servers

```

curl -i -u
api_1bcab8b7295e9b512:343461267jks00965124500jkjdmnwe00231230034dfd256124fa1
81f1edb3bb4a3 https://2x2testvc27.ilabs.io:8443/api/v2/orgs/1/sec_policy/
draft/virtual_servers

```

### Response Body

```

[ {
    "href": "/orgs/1/sec_policy/draft/virtual_servers/

```

```

5c7aeb96-56e2-4af8-8b4e-00231230034",
  "created_at": "2021-02-26T08:38:15.298Z",
  "updated_at": "2021-02-26T08:39:21.676Z",
  "deleted_at": null,
  "created_by": {
    "href": "/users/1"
  },
  "updated_by": {
    "href": "/users/1"
  },
  "deleted_by": null,
  "update_type": null,
  "name": "Common/QL_VIP_1",
  "description": "",
  "discovered_virtual_server": {
    "href": "/orgs/1/discovered_virtual_servers/52044aea-14db-4510-
alc6-00231230034"
  },
  "dvs_name": "Common/QL_VIP_1",
  "dvs_identifier": "96803bd07185cd093dd800231230034",
  "labels": [
    {
      "href": "/orgs/1/labels/1185",
      "key": "role",
      "value": "Database_VIP_1"
    },
    {
      "href": "/orgs/1/labels/1178",
      "key": "app",
      "value": "Application_1"
    },
    {
      "href": "/orgs/1/labels/1176",
      "key": "loc",
      "value": "test_place_1"
    },
    {
      "href": "/orgs/1/labels/1174",
      "key": "env",
      "value": "Production"
    }
  ],
  "service": {
    "href": "/orgs/1/sec_policy/draft/services/1"
  },
  "providers": [
    {
      "label": {
        "href": "/orgs/1/labels/1183",
        "key": "role",
        "value": "Web"
      }
    },
    {
      "label": {
        "href": "/orgs/1/labels/1178",
        "key": "app",
        "value": "Application_1"
      }
    },
    {
      "label": {
        "href": "/orgs/1/labels/1176",
        "key": "loc",
        "value": "test_place_1"
      }
    }
  ]
}

```

```

        "key": "loc",
        "value": "test_place_1"
    },
},
{
    "label": {
        "href": "/orgs/1/labels/1174",
        "key": "env",
        "value": "Production"
    }
},
"mode": "unmanaged"
}
]

```

### **Virtual Server discoveries: example for request**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Details of Virtual Servers to discover",
    "type": "object",
    "additionalProperties": false,
    "required": ["slb_name", "virtual_server_infos"],
    "properties": {
        "slb_name": {
            "description": "Name of SLB to interrogate",
            "type": "string"
        },
        "virtual_server_infos": {
            "description": "IP address and port info of Virtual Servers to discover",
            "type": "array",
            "additonalProperties": false,
            "minItems": 1,
            "items": {
                "type": "object",
                "required": ["ip_address", "port"],
                "properties": {
                    "ip_address": {
                        "description": "Virtual Server IP address",
                        "type": "string"
                    },
                    "port": {
                        "description": "Virtual Server port",
                        "type": "integer"
                    }
                }
            }
        }
    }
}
```

### **Sample Response**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Details of Virtual Servers discovery job",
    "type": "object",
    "additionalProperties": false,
    "properties": {

```

```

    "href": {
      "description": "URI of Virtual Servers discovery job",
      "type": "string"
    }
  }

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Details of Virtual Servers discovery job",
  "type": "object",
  "additionalProperties": false,
  "required": ["status", "created_at", "created_by"],
  "properties": {
    "href": {
      "description": "URI of the requested discovery job",
      "type": "string"
    }
    "status": {
      "description": "The current state of the request",
      "type": "string",
      "enum": ["pending", "running", "done"]
    },
    "created_at": {
      "description": "The time (rfc3339 timestamp) at which this job was created",
      "type": "string",
      "format": "date-time"
    },
    "completed_at": {
      "description": "The time (rfc3339 timestamp) at which this job was completed",
      "type": "string",
      "format": "date-time"
    },
    "created_by": {
      "type": "object",
      "required": ["href"],
      "properties": {
        "href": {
          "description": "User who originally created this Virtual Server discovery job",
          "type": "string"
        }
      }
    },
    "connection_state": {
      "description": "Status of most recent connection to the SLB device",
      "type": "string",
      "enum": ["pending", "successful", "cannot_resolve",
"cannot_connect", "bad_credentials", "bad_certificate", "bad_request",
"dup_device"]
    },
    "virtual_server_infos": {
      "description": "Information of individual virtual server discovered",
      "type": "array",

```

```

    "minItems": 1,
    "items": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
            "ip_address": {
                "description": "Virtual server IP address",
                "type": "string"
            },
            "port": {
                "description": "Virtual server port",
                "type": "integer"
            },
            "discovered_virtual_server": {
                "description": "Discovered Virtual Server. Null indicates not found",
                "type": "object",
                "required": [ "href" ],
                "properties": {
                    "href": {
                        "description": "URI of Discovered Virtual Server",
                        "type": "string"
                    }
                }
            }
        }
    }
}

```

If a virtual server is discovered, the response might look as follows:

```

{
    "status" : "done",
    "created_at" : "2021-7-19T07:20:50.52Z",
    "created_by" : {
        "href" : "api/v2/orgs/1/users/1"
    },
    "connection_state" : "successful",
    "completed_at" : "2021-7-19T07:20:54.97Z",
    "virtual_server_infos" : [
        {
            "ip_address" : "10.2.4.54",
            "port" : 443,
            "discovered_virtual_server" : {
                "href" : "api/v2/orgs/1/discovered_virtual_servers/7a597ef0-6609-4927-9eee-ce403517d850"
            }
        },
        {
            "ip_address" : "10.23.23.2",
            "port" : 8443,
            "discovered_virtual_server" : {
                "href" : "api/v2/orgs/1/discovered_virtual_servers/6a597ef0-6609-4927-9eee-ce403517d850"
            }
        }
    ]
}

```

If the connection was not established, the response might look as follows:

```
{
    "status" : "done",

```

```

"connection_state" : "cannot_connect",
"created_at" : "2021-7-19T07:20:50.52Z",
"completed_at" : "2021-7-19T07:20:54.97Z",
"created_by" : {
    "href" : "api/v2/orgs/1/users/1"
}
}

```

## RBAC for PCE Users

As an Illumio administrator, use the Role-based Access Control (RBAC) API to assign privileges and responsibilities to users as follows:

- Establish the least required privileges to perform a job.
- Limit access to the smallest operation set to perform a job.
- Separate users' duties, such as giving responsibility or delegating authority to a specific team.
- Allow access based on roles and scopes. Scopes in the Illumio Core specify the domain boundaries granted to a user.
- Manage user authentication and authorization.

## RBAC Overview

Role-based Access Control (RBAC) is an API that creates, updates or deletes permissions for users and groups. These users and groups are managed locally by the PCE or externally by a single sign-on (SSO) identity source (IdP).

## RBAC Terms and Concepts

Become familiar with the following RBAC concepts before using this API.

### User

A user is a PCE account that provides login or API access to the PCE. A user can be managed locally by the PCE or externally through an IdP.

### Permission

A permission represents a combination of a user's account, an RBAC role, and an optional scope. You can grant multiple permissions to a user, depending on your requirements. A permission is a three tuple consisting of a role, a scope, and an authorization security principal:

- **Role:** User personas associated with a set of allowed operations, such as creating new labels or provisioning policy changes. Roles can be one of two general types: unscoped and scoped.
- **Unscoped roles** (or roles with “global scopes”) do not restrict the types of resources a user can operate. This means that the role is not affected by any label scopes.
- **Scoped roles** use one or more unique application, environment, and location labels (each with a label HREF, key, and value) to restrict user or group permissions to only those

objects that share the same labels. Specifically, scoped roles allow certain users to create and provision rules and rulesets.

- **Scope:** A set of three labels (one of each type for Application, Environment, and Location) that restricts operations to those workloads sharing the same labels as the scope label set.
  - GET, POST, and PUT permissions methods for the Ruleset Manager (limited or full) or Ruleset Provisioner roles have a required scope parameter. When granting permissions, choose a scope restricting which resources these users can use in a ruleset or which resources they can provision.
  - A scope contains zero or more applications, environment, and location labels. Each label in the scope is identified by its HREF. A scope can also contain zero or more label groups.
  - If the scope is an empty array ([ ]), it includes all applications, environments, and locations.
  - If one of the label types is not specified, all instances of that type are permitted. For example, all applications are within the scope if application labels are omitted but environment and location labels are present.
- **Authorization Security Principal:** The binding that connects a user account with its permissions (a role, and depending on the role, scopes).



### NOTE

If you are using an external identity provider to manage user access to the PCE, make sure that your identity provider is configured and those external users have been added to the PCE before you use this API to assign user permissions.

## Grant Permissions Workflow

Granting user permissions with the REST API follows this general workflow:

### 1. Create a local user (optional)

This step creates a new local PCE user with no permissions and sends an e-mail invitation to the user's e-mail address. (If you use an external identity source to manage user access to the PCE, skip this step.)

### 2. Create an authorization security principal.

An authorization security principal is binding between a user or group, an RBAC role, and optional scopes.

### 3. Grant permissions by assigning a role and scope to the authorization security principal

Once a user account has been associated with an authorization security principal, you can assign an RBAC role to the account and add custom scopes if the user role requires them.

## List User Roles and Role Names

The APIs GET roles and GET role\_name have been promoted from Internal to Public Experimental.

They allow users to list roles and role names.

Functionality	HTTP	URI
Get the roles in the organization.	GET	[api_version] /orgs/:xorg_id/roles
Get information for this role name.	GET	[api_version]/orgs/:xorg_id/roles/:role_name

## User Permissions

### Organization-wide Default User Permissions

This Public Experimental API supplies an organization-wide default user permission and allows users to log into the PCE and view resources. These resources must not be explicitly assigned to any RBAC roles or scopes.

### About Default User Permissions

If you use an external identity source for user management, you might want to block some of those users from the PCE without removing them from your identity source. *Deleting* the organization-wide read-only permission allows you to achieve this.

When the read-only user permission is disabled for your organization, users not explicitly assigned this permission cannot log into the PCE and access Illumio resources. If users, without permission, attempt to log into the PCE, their external identity source authenticates them, but the PCE immediately logs them out.

To disable organization-wide read-only permissions:

1. Get a collection of all authorization security principals in your organization, and search the response for the one named `null`. Once you find this authorization security principal, note its full HREF.
2. Get the HREF of the permissions object associated with the `null` authorization security principal. Keep a record of the JSON object for this permission if you want to re-enable the permission at a later date.
3. Delete the permission associated with the `null` authorization security principal.

### Get a Collection of Authorization Security Principals

The first step in disabling the organization-wide read-only permission is to get a collection of all authorization security principals in your organization.

### Get Permission for Null Auth Security Principal

To get the permission object associated with the `null` authorization security principal, call the GET Permissions API with the query parameter value set to the HREF for the `null` authorization security principal, similar to the curl command:

```
curl -i -X GET -H "Accept: application/json" -u $KEY:$TOKEN https://pce.mycompany.com:8443/api/v2/orgs/7/permissions?auth_security_principal=/orgs/7/auth_security_principals/a23ea011-4191-49e6-a22a-d3dba4fb8058
```

Response

The response returns the HREF of the permission associated with the organization-wide read-only permission.

```
{
    "href": "/orgs/7/permissions/14c92849-e88e-4930-8804-3245565619e5",
    "role": {
        "href": "/orgs/7/roles/read_only"
    },
    "scope": [],
    "auth_security_principal": {
        "href": "/orgs/7/auth_security_principals/a23ea011-4191-49e6-a22a-d3dba4fb8058"
    }
}
```

## Delete Null Authorization Security Principal Permission

Keep a record of the permission object returned in case you want to re-enable the permission in the future.

Delete the read-only permission HREF to disable it.

Curl Command to Delete Null Authorization Security Principal Permission

```
curl -i -X DELETE -H "Accept: application/json" -u $KEY:$TOKEN https://pce.mycompany.com:8443/api/v2/orgs/7/permissions?auth_security_principal=/orgs/7/auth_security_principals//orgs/7/permissions/14c92849-e88e-4930-8804-3245565619e5
```

Response

An HTTP 200 response is returned on successfully deleting the organization-wide read-only permission.

## Re-Enable Organization Read-Only Permission

If the organization-wide read-only permission was disabled, you can re-enable it by recreating the permission object. This object must be constructed exactly as the object returned to you when you got the permission. The request body below illustrates the JSON structure of this permission object.

## Reference and Examples for Default User Permissions

Curl Command Get Auth Security Principals Collection

```
curl -i -X GET https://pce.mycompany.com:8443/api/v2/orgs/7/auth_security_principals -H "Accept: application/json" -u $KEY:$TOKEN
```

Example Response Body

The null authorization security principal in the following example is highlighted in blue:

```
[  
.....
```

```
{
    "href": "/orgs/7/auth_security_principals/a23ea011-4191-49e6-a22a-d3dba4fb8058",
    "name": null,
    "display_name": null,
    "type": "group"
},
.....
]
```

URI to Enable the Organization-Wide Read-Only Permission

POST [api\_version][permission\_href]

Request Body

```
{
    "role": {
        "href": "/orgs/7/roles/read_only"
    },
    "auth_security_principal": {
        "href": "/orgs/7/auth_security_principals/a23ea011-4191-49e6-a22a-d3dba4fb8058"
    },
    "scope": []
}
```

Curl Command to Enable Organization Read-Only Permission

```
curl -i -X POST https://pce.mycompany.com:8443/api/v2/orgs/7/permissions
-H "Content-Type: application/json" -u $KEY:$TOKEN -d '{"role": {
    "href": "/orgs/7/roles/read_only"}, "auth_security_principal": {"href": "/orgs/auth_security_principals/a23ea011-4191-49e6-a22a-d3dba4fb8058"}, "scope": []}'
```

Response

An HTTP 201 response is returned when the organization-wide read-only permission is successfully recreated.

## App Owner RBAC Role

The App Owner RBAC (Role-Based Access Control) role hides information in the PCE that is not relevant to the user with that role. At the same time, the app owners can write effective rules to secure their apps and restrict visibility within the PCE to the permitted scopes for users.

RBAC previously restricted only the write permission for users while the read permission was unrestricted, and every user had visibility into PCE. The App Owner RBAC role also restricts the read permission to correspond to the user roles. It accelerates enterprise-wide expansion so that the customers who acquired Illumio for a single application can expand faster.

The introduction of the App Owner role solves these problems because it does the following:

- Accelerates micro-segmentation deployment by allowing for scaling after an organization implements micro-segmentation with smaller applications.
- Assures compliance with good security practices so that users cannot view the sensitive information they are not allowed to see.
- Eliminates the complexity of building a custom portal. The App Owners can use Illumio REST APIs instead of the custom UIs created by customers.

App Owners are responsible for managing vulnerabilities in their applications for which the PCE owners can assign scoped roles.

## **App Owner Roles**

Roles of Ruleset Managers, Ruleset Provisioners, and Workload Managers are assigned to users and user groups. They can be expanded to provide users with additional read/write permissions. All permissions are additive.

### **Ruleset Manager with Scoped Reads**

This RBAC role requires written permission from the owner to change the policy. Users with this role can see only the content related to their location in the PCE instead of having full read-only access to all the PCE content.

The role now also supports scoped reads.

### **Ruleset Provisioner with Scoped Reads**

This RBAC role can provision policy changes to workloads. Users with this role can see only the content related to their location in the PCE instead of having full read-only access to all the PCE content.

The role now also supports scoped reads.

### **Ruleset Viewer**

This RBAC role has access to the PCE to manage one or multiple applications. Users with this role can get a view of their application and its dependencies, but they cannot see information about other applications.

### **Workload Manager with Scoped Reads**

This RBAC role provides a control for managing workloads. Users with this role can see only the content related to their scope in the PCE instead of having full read-only access to all the PCE content.

The role now also supports scoped reads.

## **RBAC User Operations**

This Public Stable API creates, updates, re-invites local users, and converts user status (a local user to an external user or an external user to a local user). This API is intended only for local users managed by the PCE, not users managed by an external identity source (IdP).

## API Methods

Functionality	HTTP	URI
Get a collection of users.	GET	[api_version]/users
GET an individual user.	GET	[user_href]
Get all the organizations the user has accessed after logging in ( <i>this endpoint is Public Experimental</i> )	GET	[api_version][user_href]/orgs
Create a local user and send an e-mail invitation.	POST	[api_version]/users
Convert an external user to a local user	POST	[user_href]local_profile
Delete a local user and convert to an external user	DELETE	[user_href]local_profile
Re-invite a local user.	PUT	[user_href]local_profile/reinvite
For authenticated users: change your password by sending a request to the agent service.	PUT	[user_href]local_profile/password

### Get RBAC Users

These methods get a collection of users or individual users in the organization.

By default, the maximum number of users returned from a GET collection is 500. To get more than 500 users, use Asynchronous GET Collections.

URI to Get a Collection of Local Users

```
GET [api_version]/users
```

URI to Get an Individual User

```
GET [user_href]
```

### Create a Local User

This method creates local users whom the PCE manages.

URI to Create a Local User

```
POST [api_version]/users
```

### User Profiles

Change a user's status by converting a local user to an external user or vice versa.

### Convert Local to External User

This method converts a local user to an external user by *deleting* the local user account profile.

Use the user HREF, obtained from the response when a user logs into the PCE using the Login API or from the GET collection response.

For example: /users/14

### Convert External User to Local User

This method converts externally managed users to local users managed by the PCE.

### Re-invite a Local User

If you have already created a local user, but that user has not logged in yet for the first time, you can use this method to resend the email invitation. Once they receive the invitation, they can log into the PCE and complete their PCE user account registration.

## RBAC User Operations Reference

This topic covers parameters, properties, and examples for RBAC user operations.

### Parameters

Parameter	Description	Type	Required
type	Indicates that the user created is a local user managed by the PCE.	String	No
id	User ID	Integer	Yes

## Properties

Property	Description	Type	Required
href	User URI	String	Yes
username	Identify a local user by an e-mail address, which must meet these requirements: <ul style="list-style-type: none"><li>• Be unique</li><li>• Use the format xxxx@yyyy.zzz</li><li>• Be 255 characters or less.</li></ul>	String (email)	Yes
last_login_on	This is populated automatically after a login.	String	Yes
last_log-in_ip_address	This is populated automatically after a login.	String	Yes
login_count	Number of times this user logged in	Integer	Yes
full_name	User's full name	String	Yes
time_zone	Time Zone IANA Region Name	String	Yes
type	User's type, i.e., user authenticated locally or remotely via SAML.	String	Yes
updated_at	Timestamp when this user was last updated	String	Yes
created_at	Timestamp when this user was first created	String	Yes
current_password	The current password that you want to change	String	Yes
new_password	New password to set	String	Yes

## Examples

### Convert Local to External User

URI to Convert a Local User to an External User

```
DELETE [user_href]/local_profile
```

Example

```
DELETE https://pce.my-company.com:8443/api/v2/users/14/local_profile
```

Convert Local User to External User

```
curl -i -X >DELETE https://pce.my-company.com:8443/api/v2/users/14/local_profile -H "Accept: application/json" -u $KEY:$TOKEN
```

### Convert External User to Local User

URI to Convert an External User to a Local User

```
POST [user_href]/local_profile
```

Example

```
POST https://pce.my-company.com:8443/api/v2/users/14/local_profile
```

## Re-invite a Local User

URI to Re-invite a Local User

```
PUT [user_href]/local_profile/reinvite
```

Example

```
PUT https://pce.my-company.com:8443/api/v2/users/14/local_profile/reinvite
```

Curl Command Get Collection of Local Users

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/users?type=local -H "Accept: application/json" -u $KEY:$TOKEN
```

Response

```
[  
  {  
    "href": "/users/99",  
    "type": "local",  
    "effective_groups": [],  
    "id": 99,  
    "username": "joe.user@example.com",  
    "full_name": "Joe User",  
    "time_zone": "America/Los_Angeles",  
    "locked": false,  
    "login_count": 1,  
    "last_login_ip_address": "192.x.x.x",  
    "last_login_on": "2016-03-11T08:19:17.587Z",  
    "local_profile": { "pending_invitation": false },  
    "created_at": "2016-03-08T20:58:05.882Z",  
    "updated_at": "2016-03-11T08:19:17.588Z"  
  }  
  ....  
  {  
    "href": "/users/56",  
    "type": "local",  
    "effective_groups": [],  
    "id": 56,  
    "username": "jeff.user@example.com",  
    "full_name": "Jeff User",  
    "time_zone": "America/New_York",  
    "locked": false,  
    "login_count": 21,  
    "last_login_ip_address": "192.x.x.x",  
    "last_login_on": "2017-05-26T14:22:37.643Z",  
    "local_profile": { "pending_invitation": true },  
    "created_at": "2016-05-02T07:16:21.725Z",  
    "updated_at": "2017-05-26T14:23:04.625Z"
```

```

    }
]
```

## Pending Invitation

Users with "pending\_invitation": "true" in the response have not yet accepted the invitation to log in and create an account.

```
{
  "href": "/users/56",
  "type": "local",
  "effective_groups": [],
  "id": 56,
  "username": "jeff.user@example.com",
  "full_name": "Jeff User",
  "time_zone": "America/New_York",
  "locked": false,
  "login_count": 21,
  "last_login_ip_address": "192.x.x.x",
  "last_login_on": "2017-05-26T14:22:37.643Z",
  "local_profile": { "pending_invitation": true },
  "created_at": "2016-05-02T07:16:21.725Z",
  "updated_at": "2017-05-26T14:23:04.625Z"
}
```

Request body to create a local user

```
{
  "username": "joe_user@mycompany.com",
  "display_name": "Joe User",
  "type": "local"
}
```

## Curl Command to Create a Local User

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/users -H
"Content-Type: application/json" -u $KEY:$TOKEN -d '{"username": "joe_user@mycompany.com", "display_name": "Joe User", "type": "user"}'
```

## Curl Command to Convert External User to Local User

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/users/14/
local_profile -H "Content-Type: application/json" -u $KEY:$TOKEN
```

## Curl Command to Reinvite a Local User

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/users/14/
local_profile/reinvite -H "Content-Type: application/json" -u $KEY:$TOKEN
```

## About RBAC Permissions

This Public Experimental API grants permissions to PCE users and groups. It also returns a collection of permissions in the organization, gets individual user permissions, and updates and deletes permissions.



### NOTE

In addition to labels, label groups have been added as part of the response and parameters because they are now supported in user scopes.

## API Methods

Functionality	HTTP	URI
Get a list of all RBAC permissions for the organization (schema and query parameter format change)	GET	[api_version]orgs/{org_id}/permissions
Get an individual permission (schema change)	GET	[api_version]{org_id}/permissions/{permission_id}
Grant permission (schema change)	POST	[api_version]orgs/{org_id}/permissions
Update a permission (schema change)	PUT	[api_version]orgs/{org_id}/permissions/ {permission_id}

## New Schema and Query Parameter

For the above endpoints, the `org_scope.schema.json` is now used instead of `labels_summary.schema.json` and `labels.schema.json`.

For the endpoint `GET /api/v2/orgs/1/permissions`, the query parameter is changed from

```
scope: ["/orgs/1/labels/5", "/orgs/1/labels/3"]
```

to

```
scope: [{"label": {"href": "/orgs/1/labels/5"}}, {"label": {"href": "/orgs/1/labels/3"}}]
```

## Ruleset Manager and Ruleset Provisioner

Suppose you grant a user or group the Ruleset Manager or the Ruleset Provisioner role. In that case, you can also associate a scope to the role so you can control which rulesets they can add and provision.

A default read-only user permission is organization-wide and inherited by all users. This global permission allows users with no permissions explicitly granted to access the PCE.

**NOTE**

For more information, see [Organization-Wide Default User Permission \[262\]](#).

**Role HREF Syntax**

An RBAC role is identified in the REST API by its HREF, the exact syntax of which is based on the PCE organization HREF [`org_href`].

```
[org_href]/roles/[role_name]
```

For example, if you wanted to grant a user permission with the Global Object Provisioner role, and your PCE organization HREF is `/org/6`, the role HREF would look like:

```
/orgs/6/roles/global_object_provisioner
```

**Scoped Permissions**

The permission for this scoped role consists of the following elements:

- A scope for the role (application, environment, and location labels)
- The role
- An authorization security principal associated with a user account

**NOTE**

See the scope parameter change explained in [Use the New Schema and Query Parameter \[271\]](#).

**Unscoped Permissions**

Request - Unscoped Permission

In this request for unscoped permission, the required `scope` property is defined as an empty JSON array (`[ ]`).

**NOTE**

When the `scope` parameter is empty, the change explained in [Use the New Schema and Query Parameter \[271\]](#) does not apply.

**RBAC Permissions Reference**

This topic covers parameters, properties, and examples for RBAC permissions.

## Parameters

### Unscoped Roles

API Role Name	UI Role Name	Granted Access
owner	Global Organization Owner	Perform all actions: Add, edit, or delete any resource, security settings, or user accounts.
admin	Global Administrator	Perform all actions except cannot change security settings and cannot perform user management tasks.
read_only	Global read-only	View any resource or security settings. Cannot perform any operations.
global_object_provisioner	Global Policy Object Provisioner	Provision rules containing IP lists, services, label groups, and manage security settings. Cannot provision rulesets, virtual services, or virtual servers or add, modify, or delete policy items.

### Scoped Roles

API Role Name	UI Role Name	Granted Access
ruleset_manager	Full Ruleset Manager	Add, edit, and delete all rulesets within a specified scope.  Add, edit, and delete rules when the Source matches a specified scope.  The rule Destination can match any scope.
limited_ruleset_manager	Limited Ruleset Manager	Add, edit, and delete all rulesets within a specified scope.  Add, edit, and delete rules when the Source and Destination match the specified scope. Ruleset managers with limited privileges cannot manage rules that use IP lists, user groups, label groups, iptables rules as destinations, or rules that allow internet connectivity.
ruleset_provisioner	Ruleset Provisioner	Provision rulesets within a specified scope.  Cannot provision virtual servers, virtual services, SecureConnect gateways, security settings, IP lists, services, or label groups.
scope		The <code>org_scope.schema.json</code> is now used instead of <code>labels_summary.schema.json</code> and <code>labels.schema.json</code> .

## Parameters for RBAC Permissions

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
auth_security_principal	<p>The authorization security principal is associated with the permission. It is not needed to get individual permission or to delete a permission.</p> <p>The HREF of the authorization security principal</p> <p>(auth_security_principal) associated with the user or group being granted the permission.</p> <p>The HREF of an authorization security principal is returned when you create a new one, or you can GET a collection of authorization security principals in your PCE.</p>	String	POST:Yes PUT:No
role	<p>The RBAC role is associated with the permissions.</p> <p>An RBAC role is identified in the REST API by its HREF, the exact syntax of which is different for every user and is based on the PCE organization.</p> <p>HREF [org_href]. For example:</p> <pre>[org_href]/roles/[role_name]</pre> <p>For example, to grant a user permission with the Global ObjectProvisioner role with a PCE organization HREF of /org/6, the role HREF would be:</p> <pre>/orgs/6/roles/global_object_provisioner</pre> <p>(For additional information about these roles and their associated capabilities, see PCE Administration Guide, <a href="#">Role-Based Access for Application Owners</a>.</p> <p>Unscoped roles:</p> <ul style="list-style-type: none"> <li>• owner</li> <li>• admin</li> <li>• read_only</li> <li>• global_object_provisioner</li> </ul> <p>Scoped roles:</p> <ul style="list-style-type: none"> <li>• ruleset_manager</li> <li>• limited_ruleset_manager</li> <li>• ruleset_provisioner</li> </ul>	String	Yes
scope	Scope to filter on, where the scope is in the format defined in org_scope.schema.json	String	No
permission_id	UUID of the permission	String	Yes

## Properties

Parameter	Description	Type	Re-required
org_id	Organization	Integer	Yes
permission_id	UUID of the permission. Used to get, update, and delete an individual permission	String	Yes
auth_security_principal	The authorization security principal is associated with the permission. It is not needed to get individual permission or to delete a permission.  Reference to auth_security_principal_uri. schema.json		
role	The RBAC role is associated with the permissions.  Reference to common/orgs_roles.schema.json		Yes
scope	Scope to filter on, where the scope is in the format defined in org_scope.schema.json		Yes

## Examples

Curl Command Get Permissions with a Specific Role

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/permissions?role=ruleset_provisioner -H "Accept: application/json" -u $KEY:$TOKEN
```

Example Request Body with orgs\_permission.schema.json

```
{
  "scope": [
    {
      "label_group": {
        "href": "/orgs/1/sec_policy/active/label_groups/7d480df0-f5e1-4d1e-b088-d8105150a883"
      }
    },
    {
      "label": {
        "href": "/orgs/1/labels/12"
      }
    }
  ],
  "role": {
    "href": "/orgs/1/roles/limited_ruleset_manager"
  },
  "auth_security_principal": {
    "href": "/orgs/1/auth_security_principals/177027ca-c3fe-4610-ac14-fe5cba173af5"
  }
}
```

Example Response for a Scoped Permission

The response shows the new permission (at the top) that has been created and identified by its HREF:

```
{
  "href": "/orgs/2",
  "display_name": "Luke",
  "permissions": [
    {
      "href": "/orgs/2/permissions/23dde367-41ea-4752-
bfe5-16c173aad1a5",
      "role": {
        "href": "/orgs/2/roles/limited_ruleset_manager"
      },
      "scope": [
        {
          "label": {
            "href": "/orgs/2/labels/452",
            "key": "app",
            "value": "App1"
          }
        }
      ],
      "auth_security_principal": {
        "href": "/orgs/2/auth_security_principals/04b63b79-9883-4e84-
acc5-f727f1c67fa1"
      }
    },
    .....
  ]
}

{
  "scope": [],
  "role": { "href": "/orgs/7/roles/owner" },
  "auth_security_principal":{ "href": "/orgs/7/auth_security_principals/
xxxxxxxxx-e4bf-4ba5-bd77-ccfc3a8ad999" }
}
```

Response - Unscoped Permission

```
{
  "href": "/orgs/7/permissions/51d9207c-354b-45de-9bf5-d1b613ac3719",
  "role": { "href": "/orgs/7/roles/owner" },
  "scope": [],
  "auth_security_principal":{ "href": "/orgs/7/auth_security_principals/
xxxxxxxxx-e4bf-4ba5-bd77-ccfc3a8ad999" }
}
```

## Authorization Security Principals

This Public Experimental API gets, creates, updates, and deletes authorization security principals.

An authorization security principal connects a user account with its permissions, which consist of a role and optional scopes.

## API Methods

Functionality	HTTP	URI
Get a collection of authorization security principals in an organization.	GET	[api_version][org_href]/auth_security_principals
Get an individual authorization security principal.	GET	[api_version][auth_security_principal_href]
Create an individual authorization security principal.	POST	[api_version][org_href]/auth_security_principals
Update an authorization security principal.	PUT	[api_version][auth_security_principal_href]
Delete an authorization security principal.	DELETE	[api_version][auth_security_principal_href]

## Get Authorization Security Principals

This method gets an individual or a collection of authorization security principals in your organization.

By default, the maximum number of authorization security principals returned from a GET collection is 500.

URI to Get a Collection of Authorization Security Principals

```
GET [api_version][org_href]/auth_security_principals
```

URI to Get an Individual Authorization Security Principal

Use the auth\_security\_principal\_id in a GET collection response (the last set of numbers in an HREF field).

```
GET [api_version][org_href]/auth_security_principals/{auth_security_principal_id}
```

## Create an Authorization Security Principal

This method creates an individual authorization security principal.

URI to Create an Authorization Security Principal

`POST [api_version][org_href]/auth_security_principals`

## Update an Authorization Security Principal

To update an individual authorization security principal, use its HREF, which is obtained from the response from a GET collection.

URI to Update an Individual Authorization Security Principal

`PUT [api_version][auth_security_principal_href]`

## Delete an Authorization Security Principal

To delete an authorization security principal, use its HREF, which is returned in the response to a GET collection request.

URI to Delete an Individual Authorization Security Principal

`DELETE [api_version][auth_security_pincipal_href]`

## Authorization Security Principals Reference

This topic covers properties, parameters, and examples of authorization security principals.

### Parameters

Parameters used for Auth Security Principals are:

Parameter	Description	Type	Required
<code>org_id</code>	Organization Id	Integer	Yes
<code>name</code>	Name of the authorization security principal.  • If the user is local (managed by the PCE), the <code>namemust be the e-mail address of the local user.</code> • If an external IDP manages the user or group, use the name that identifies the external user or group in the external system.	String	GET, PUT: No  POST: Yes
<code>type</code>	One of two types of users, either <code>user</code> or <code>group</code> .  POST: Yes	String	GET, PUT: No
<code>auth_security_principal_id</code>	UUID of the auth_security_principal. Required for <code>[api_version][auth_security_principal_href]</code>	String	Yes
<code>display_name</code>	An optional display name for the authorization security principal.	String	No
<code>access_restriction</code>	Access restriction assigned to this user	String  NULL	No

## Properties

Property	Description	Type	Required
href	URI of auth_security_principal	String	Yes
name	Name of the authorization security principal.	String	GET, PUT: No
	<ul style="list-style-type: none"> <li>If the user is local (managed by the PCE), the name must be the e-mail address of the local user.</li> <li>If the user or group is managed by an external IdP, use the name that identifies the external user or group in the external system.</li> </ul>		POST: Yes
type	One of two types of users, either user or group.	String	GET, PUT: No
			POST: Yes
auth_security_principal_id	UUID of the auth_security_principal. Required for [api_version][auth_security_principal_href]	String	Yes
display_name	An optional display name for the authorization security principal.	String	No
access_restriction	Access restriction assigned to this user	String	No
		NULL	

## Examples

### Curl Command to Get Authorization Security Principals

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/auth_security_principals -H "Accept: application/json" -u $KEY:$TOKEN
```

### Response

Each individual authorization security principal returned is identified by its HREF. You can use the HREF to GET, PUT, or DELETE an authorization security principal.

```
{
  "href": "/orgs/7/auth_security_principals/97cb9898-027b-474e-9807-19e04460dfb0",
  "name": "jimmyjo@illum.io",
  "display_name": "Jimmy Joe Meeker",
  "type": "user"
},
.....
{
  "href": "/orgs/7/auth_security_principals/db7a2657-dcb8-4237-a6e7-7269cdbaea5d",
  "name": "foxy.brown@illumio.com",
  "display_name": "Foxy Brown",
  "type": "user"
}
]
```

**Curl Command to Get an Authorization Security Principal**

```
curl -i -X GET -H "Accept: application/json" -u $KEY:$TOKEN https://pce.my-company.com:8443/api/v2/orgs/2/auth_security_principals/db7a2657-dcb8-4237-a6e7-7269cdbaea5d
```

**Request Body - Local User Authorization Security Principal**

```
{
  "type": "user",
  "name": "joe_user@illumio.com",
  "display_name": "Joe User"
}
```

**Response Body - Local User Authorization Security Principal**

```
{
  "href": "/orgs/7/auth_security_principals/e8c232d2-e4bf-4ba5-bd77-ccfc3a8ad999",
  "name": "joe_user@illumio.com",
  "display_name": "Joe User",
  "type": "user"
}
```

**Request Body - External Group User Authorization Security Principal**

```
{
  "type": "group",
  "name": "jCQN=Bank-Admin,OU=EU,DC=Acme,DC=com",
  "display_name": "Provisioners for Bank Accounts"
}
```

**Response Body - External Group Authorization Security Principal**

```
{
  "href": "/orgs/7/auth_security_principals/e8c232d2-e4bf-4ba5-bd77-ccfc3a8ad777",
  "name": "jCQN=Bank-Admin,OU=EU,DC=Acme,DC=com",
  "display_name": "Acme Bank Admins",
  "type": "group"
}
```

**Curl Command Create an Authorization Security Principal**

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/auth_security_principals -u $KEY:$TOKEN -H "Content-Type:application/json" -d '{"type": "user", "name": "joe_user@illumio.com", "display_name": "Joe User"}'
```

**Request body to update an individual authorization security principal**

```
{
  "type": "user",
  "name": "joe_user2@illumio.com",
```

```

    "display_name": "Joe User"
}

```

### Curl Command Create an Authorization Security Principal

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/sec_policy/draft/services/79 -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{"type": "user", "name": "joe_user2@illumio.com", "display_name": "Joe User"}'
```

### Curl Command Delete the Authorization Security Principal

```
curl -i -X DELETE -H "Accept: application/json" -u $KEY:$TOKEN https://pce.my-company.com:8443/api/v2/orgs/2/auth_security_principals/e8c232d2-e4bf-4ba5-bd77-ccfc3a8ad777
```

## Visualization

In addition to reviewing workloads and traffic with the PCE web console, you can analyze the traffic flows and get insight into the exposure to vulnerabilities using the Visualization API.

The Explorer API is used to search and analyze PCE traffic flows. It queries the PCE's traffic database and analyzes these flows for auditing, reporting, and troubleshooting. The VEN adds the DNS names to the flow summary logs and sends them to the PCE, while the Explorer API appends the DNS names to allow auditors and analysts to view them without performing reverse look-ups on random IP addresses.

Vulnerability Maps combine Illumio's Application Dependency Map with vulnerability data from Qualys Cloud Platform to provide insights into the exposure of vulnerabilities and attack paths across your applications.

## Explorer

The Public Experimental Explorer APIs search and analyze PCE traffic flows for auditing, reporting, and troubleshooting. You can search for traffic flows between workloads or hosts, labeled workloads, or IP addresses, and you can restrict the search by specific port numbers and protocols.

## Asynchronous Queries API Methods

The maximum returned results when using the deprecated method **POST [api\_version][org\_href]/traffic\_flows/traffic\_analysis\_queries** was 100,000, a reasonable number a user can view in the UI.

When Explorer captures all traffic flows into a CSV file to build rules offline, the queries take longer to return, traffic data contains more than 100,000 rows, and so on. Explorer queries are required to support both the single-node and multi-node Explorer in the SuperCluster environment.

The limitation of 100,000 results was raised to 200,000 to support SuperCluster environments in Explorer better.

Functionality	HTTP	URI
Create a new async traffic query.	POST	[api_version][org_href]traffic_flows/async_queries
Get a collection of async traffic queries.	GET	[api_version][org_href]traffic_flows/async_queries
Download the completed async traffic query results.	GET	[api_version][org_href]traffic_flows_async/queries/:uuid/download
Update an async traffic query (request cancellation of the queued async query).	PUT	[api_version][org_href]traffic_flows/async_queries/:uuid
Delete the completed async traffic query.	DELETE	[api_version][org_href]traffic_flows/async_queries/:uuid

## Explorer Reference

This topic contains properties and examples for working with Explorer.

**POST [api\_version][org\_href] traffic\_flows/async\_queries**

Property	Description	Type
query_name	Name of the query	String
sources	Source labels, workloads, or IP addresses to include or exclude in the search.  The response can contain up to five matching IP addresses.	Object
	<b>Note:</b> The response returns <code>sources</code> as <code>destinations</code> .  Sources are treated as destinations for the purposes of the request; the response returns the source of an individual flow as <code>src</code> .	
	<b>Sub-properties:</b> <ul style="list-style-type: none"><li>• <code>include</code>: Targets that can be included are workloads, labels, or IP addresses identified by their HREF and structured as an array of JSON objects. If this property is left empty, then <code>include</code> means consider “ALL” or “ANY” of the object type.</li><li>• <code>exclude</code>: Targets that can be excluded are workloads, labels, or IP addresses identified by their HREF and structured as a JSON array.<ul style="list-style-type: none"><li>• When IP List is present in the Destination part of a traffic query, traffic from workloads that belong to that IP List will not be returned by default. If users want to see that traffic, they need to set <code>exclude_workloads_from_ip_list_query: false</code></li><li>• When IP List is present in the Source part of traffic query, traffic to workloads that belong to that IP List will not be returned by default. If the user wishes to see that traffic, they need to set <code>exclude_workloads_from_ip_list_query: false</code></li></ul></li></ul>	
destinations	Target labels, workloads, or IP addresses to include or exclude in the search.  The response returns <code>targets</code> as <code>providers</code> .	Object
	<b>Required sub-properties:</b> <ul style="list-style-type: none"><li>• <code>include</code>: Targets that can be <i>included</i> are workloads, labels, or IP addresses identified by their HREF and structured as an array of JSON objects. If this property is left empty, then <code>include</code> means consider “ALL” or “ANY” of the object type.</li><li>• <code>exclude</code>: Targets that can be <i>excluded</i> are workloads, labels, or IP addresses identified by their HREF and structured as JSON. If this property is left empty, then <code>exclude</code> means exclude “NONE” of the object types.</li></ul>	
services	Services (5-tuple of port/to_port/proto/process/service) to include or exclude. Not all properties of the service subobjects are required.	
	<b>Required properties:</b> <ul style="list-style-type: none"><li>• <code>include</code>: List of included services (5-tuple of port/to_port/proto/process/service)</li><li>• <code>exclude</code>: List of excluded services (5-tuple of port/to_port/proto/process/service).</li></ul>	
	Properties of the <code>include</code> and <code>exclude</code> subobjects: <ul style="list-style-type: none"><li>• <code>port</code>: Port Number (integer 0-65535). Also the starting port when specifying a range.</li><li>• <code>to_port</code>: High end of port range inclusive if specifying a range. If not specifying a range then don't send this:</li></ul>	

Property	Description	Type
	<ul style="list-style-type: none"> <li>• <b>proto</b>: Protocol number. For the expected proto values see <a href="#">IANA Protocol Numbers</a>.</li> <li>• <b>process_name</b>: name of the process</li> <li>• <b>windows_service_name</b>: name of the Windows service</li> </ul>	
<b>policy_decisions</b>	<p>List of policy decisions. Allows you to filter the query based on policy decision:</p> <ul style="list-style-type: none"> <li>• <b>allowed</b>: Allowed traffic.</li> <li>• <b>potentially_blocked</b>: Allowed but potentially blocked traffic. This type of traffic occurs when a workload VEN is in the test policy state.</li> <li>• <b>blocked</b>: Blocked traffic.</li> <li>• <b>unknown</b></li> </ul>	Array of strings
<b>boundary_decisions</b>	<p>List of boundary decisions</p> <ul style="list-style-type: none"> <li>• <b>blocked</b>: blocked due to boundary</li> <li>• <b>override_deny_rule</b>: overridden deny rule</li> <li>• <b>blocked_non_illumio_rule</b>: Deny rule not written by Illumio</li> </ul>	Array
<b>max_results</b>	Maximum number of flows to return. Limit is 200,000	Integer
<b>exclude_workloads_from_ip_list_query</b>	Exclude workload traffic when IP List is provided either in the Destination or Source part of the traffic query.	Boolean
<b>data_sources</b>	<p>This property was added in release 24.1 and describes the data sources of the flows to be <b>included</b> or <b>excluded</b>.</p> <ul style="list-style-type: none"> <li>• <b>include:include</b> "server", "endpoint", "flowlink", "scanner"</li> <li>• <b>exclude:</b> "server", "endpoint", "flowlink", "scanner"</li> </ul>	

**GET [api\_version][org\_href] traffic\_flows/async\_queries\_download**

Property	Description	Type	Req
src	Reference to traffic_flows_endpoint.schema.json		Yes
dst	Reference to traffic_flows_endpoint.schema.json		Yes
service	Reference to traffic_flows_service.schema.json		Yes
num_connections	The number of times this flow was seen	Integer	Yes
policy_decision	Policy decision made	String	Yes
draft_policy_decision	The draft policy decision of the flow (added in release 23.2.10)	String	No
timestamp_range	Timestamp ranges for the flow detected. Required properties are: first_detected last_detected	Object	Yes
caps	Reference to rbac_permission_types.schema.json		Yes
client_type	The type of client that reported this flow	String	No

**Example Async Explorer Queries****Curl command for POST traffic\_flows\_async\_queries**

```
curl -i -u
api_1195cf055cf8a834c:148af87ecc980900eaf10d6c54e6c0f607b22e0dbf768dd007e51
e731096282 https://devtest0.ilabs.io:8443/api/v2/orgs/1/traffic_flows/
async_queries -H "Content-Type: application/json" -X POST -d '{"sources": {
  "include": [
    {
      "workload": {
        "href": "/orgs/1/workloads/a3ffb374-f6c6-4cce-ac57-642c66f1498f"
      }
    }
  ],
  "exclude": []
}, "destinations": {"include": [
  {}
], "exclude": []}, "services": {"include": [], "exclude": [
  {}
]}, "sources_destinations_query_op": "and", "start_date": "2016-01-29T17:04:03.149Z", "end_date": "2021-01-29T17:06:03.151Z", "policy_decisions": [
  {}
], "max_results": 1000, "query_name": "worklaod test"}'
```

**Response**

```
HTTP/1.1 202 Accepted
content-location: 7734501b-74a2-47a4-9ded-77bf4ceea938
content-type: application/json
content-length: 615
x-request-id: 00c8fa00-dbd8-4a28-a5c7-354fb5ae3886
cache-control: no-store
x-frame-options: DENY
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
{"status": "queued", "href": "/orgs/1/traffic_flows/async_queries/7734501b-74a2-47a4-9ded-77bf4ceea938", "created_by": {"href": "/users/1"}, "query_parameters": {"sources": {"include": [{"workload": {"href": "/orgs/1/}}]}}
```

```
workloads/a3ffb374-f6c6-4cce-ac57-642c66f1498f"}}], "exclude": []
}, "destinations": {"include": [], "exclude": []}, "services": {"include": []
}, "exclude": []
}, "sources_destinations_query_op": "and", "start_date": "2016-01-29T17:04:03.149Z", "end_date": "2021-01-29T17:06:03.151Z", "policy_decisions": []
}, "max_results": 1000, "query_name": "worklaod test"}, "created_at": "2021-04-09T20:50:30Z", "updated_at": "2021-04-09T20:50:30Z"
}
```

### Curl command for GET traffic\_flows/async\_queries

This query gets the collection of all async jobs for the current user, including anything that was already submitted.

```
curl -i -u
api_1195cf055cf8a834c:148af87ecc980900eaf10d6c54e6c0f607b22e0dbf768dd007e51
e731096282 https://devtest0.ilabs.io:8443/api/v2/orgs/1/traffic_flows/
async_queries
```

### Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 1510
x-request-id: fcf065e5-e465-4161-ba98-542182734c38
cache-control: no-store
x-frame-options: DENY
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
[{"matches_count": 1984, "flows_count": 1000, "status": "completed", "href": "/orgs/1/traffic_flows/async_queries/88675fbd-a88e-44bd-b358-2d6f2fc4f95a", "result": "/orgs/1/traffic_flows/async_queries/88675fbd-a88e-44bd-b358-2d6f2fc4f95a/download", "created_by": {"href": "/users/1"}, "query_parameters": {"sources": {"include": [{"workload": {"href": "/orgs/1/workloads/a3ffb374-f6c6-4cce-ac57-642c66f1498f"}]}], "exclude": []}, "destinations": {"include": [], "exclude": []}, "services": {"include": []
}, "exclude": []
}, "sources_destinations_query_op": "and", "start_date": "2016-01-29T17:04:03.149Z", "end_date": "2021-01-29T17:06:03.151Z", "policy_decisions": []
}, "max_results": 1000, "query_name": "worklaod tesrrrrrt"}, "created_at": "2021-04-09T20:50:19Z", "updated_at": "2021-04-09T20:50:27Z"}, {"matches_count": 1984, "flows_count": 1000, "status": "completed", "href": "/orgs/1/traffic_flows/async_queries/7734501b-74a2-47a4-9ded-77bf4ceea938", "result": "/orgs/1/traffic_flows/async_queries/7734501b-74a2-47a4-9ded-77bf4ceea938/download", "created_by": {"href": "/users/1"}, "query_parameters": {"sources": {"include": [{"workload": {"href": "/orgs/1/workloads/a3ffb374-f6c6-4cce-ac57-642c66f1498f"}]}], "exclude": []}, "destinations": {"include": []
}, "exclude": []
}, "services": {"include": []
}, "sources_destinations_query_op": "and", "start_date": "2016-01-29T17:04:03.149Z", "end_date": "2021-01-29T17:06:03.151Z", "policy_decisions": []
}, "max_results": 1000, "query_name": "worklaod test"}, "created_at": "2021-04-09T20:50:30Z", "updated_at": "2021-04-09T20:50:32Z"
}
```

## Curl command for GET traffic\_flows/async\_queries/:uuid

This query gets a specific job included in the collection.

```
curl -i -u $KEY:$TOKEN https://devtest0.ilabs.io:8443/api/v2/orgs/1/traffic_flows/async_queries/88675fbd-a88e-44bd-b358-2d6f2fc4f95a
```

### Response

```
HTTP/1.1 200 OK
content-type: application/json
content-length: 756
x-request-id: f328b845-8542-4b96-a128-43aefdf7ba5a
cache-control: no-store
x-frame-options: DENY
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
{"matches_count":1984,"flows_count":1000,"status":"completed",
 "href":"/orgs/1/hanges for22.4.0 from the Wj/async_queries/88675fbd-a88e-44bd-b358-2d6f2fc4f95a",
 "result":"/orgs/1/traffic_flows/async_queries/88675fbd-a88e-44bd-b358-2d6f2fc4f95a/download",
 "created_by":{"href":"/users/1"},"query_parameters":{"sources":{"include":
 [[[{"workload":{"href":"/orgs/1/workloads/a3ffb374-f6c6-4cce-ac57-642c66f1498f"}]]],"exclude":[],"destinations":{"include":
 [[[],"exclude":[]]},"services":{"include":[],"exclude":[]}},"sources_destinations_query_op":"and","start_date":"2016-01-29T17:04:03.149Z",
 "end_date":"2021-01-29T17:06:03.151Z","policy_decisions":[],"max_results":1000,"query_name":"worklaod
 tesrrrrrt"},"created_at":"2021-04-09T20:50:19Z","updated_at":"2021-04-09T20:50:27Z"}
```

### Response for GET traffic\_flows/async\_queries/:uuid\_download

```
{
  "dst": {
    "ip": "10.244.0.1",
    "workload": {
      "href": "/orgs/1/workloads/35d8efea-f230-4027-a8ee-5f20626c4d21",
      "name": "wl3",
      "labels": [
        {
          "key": "env"reserpine for
          "href": "/orgs/1/labels/7",
          "value": "Production"
        },
        {
          "key": "loc",
          "href": "/orgs/1/labels/11",
          "value": "Amazon"
        },
        {
          "key": "role",
          "href": "/orgs/1/labels/3",
          "value": "Administrator"
        }
      ]
    }
  }
}
```

```

        "value": "API"
    } ,
    {
        "key": "B-label",
        "href": "/orgs/1/labels/15",
        "value": "b_label_2"
    }
],
"managed": false,
"os_type": "linux",
"endpoint": false,
"hostname": "",
"enforcement_mode": "visibility_only"
}
},
"src": {
    "ip": "10.0.2.15",
    "workload": {
        "href": "/orgs/1/workloads/fc3801b8-05ec-4954-a957-7f5673123389",
        "name": "wl2",
        "labels": [
            {
                "key": "env",
                "href": "/orgs/1/labels/7",
                "value": "Production"
            },
            {
                "key": "loc",
                "href": "/orgs/1/labels/11",
                "value": "Amazon"
            },
            {
                "key": "role",
                "href": "/orgs/1/labels/3",
                "value": "API"
            }
        ],
        "managed": false,
        "os_type": "linux",
        "endpoint": false,
        "hostname": "",
        "enforcement_mode": "visibility_only"
    }
},
"caps": [],
"state": "snapshot",
"dst_bi": 0,
"dst_bo": 0,
"seq_id": 2,
"network": {
    "href": "/orgs/1/networks/fbeeb98d-4ed6-428d-9f71-69f542bfd8fd",
    "name": "Corporate"
},
"service": {
    "port": 3306,

```

```

    "proto": 6
  },
  "flow_direction": "outbound",
  "num_connections": 1,
  "policy_decision": "unknown",
  "timestamp_range": {
    "last_detected": "2022-09-01T20:35:22Z",
    "first_detected": "2022-09-01T20:35:22Z"
  }
}

```

## Parameters for Database Usage Metrics

The organization flow Database Usage Metrics has the following required parameters:

Parameters	Description	Type	Re- quired
flows_days	Organization's total number of days of flow data	Integer	Yes
flows_days_limit	Organization's limit on the total number of days of flow data  Limit was increased from 90 to 97	Integer	Yes
flows_oldest_day	Organization's oldest day of flow data (yyyy-mm-dd)	String	No
flows_size_gb	Organization's limit on the total number of gigabytes of flow data	Number	Yes
flows_size_gb_limit	Organization's limit on the total number of gigabytes of flow data	Number	Yes
server	Define the server's total flow data per organization for the total number of days, limit on the total number of days, oldest days, size in gigabytes, and so on.	Object	No
endpoint	Organization's total number of days of endpoint flow data.	Object	No
backlog	Total gigabytes used to store flow data input files	Object	No
updated_at	Timestamp in UTC when these flow metrics were generated	String, date	No

Parameters for `server`

Parameters	Description	Type
num_flows_days	Organization's total number of days of the server flow data	Integer
num_flows_days_limit	Organization's limit on the total number of days of server flow data	Integer
flows_oldest_day	Organization's oldest day of server flow data (yyyy-mm-dd)	String, date
flows_size_gb	Organization's limit on the total number of gigabytes of server flow data	Number
flows_size_gb_limit	Organization's limit on the total number of gigabytes of server flow data	Number
num_daily_tables	The number of server daily tables, including Flowlink and Cloud, counted once for each unique day	Number
num_weekly_tables	The number of server weekly tables, including Flowlink and Cloud, counted once for each unique week	Number

Parameters for endpoint

Parameters	Description	Type
num_flows_days	Organization's total number of days of the endpoint flow data	Integer
num_flows_days_limit	Organization's limit on the total number of days of endpoint flow data	Integer
flows_oldest_day	Organization's oldest day of endpoint flow data (yyyy-mm-dd)	String, date
flows_size_gb	Organization's limit on the total number of gigabytes of endpoint flow data	Number
flows_size_gb_limit	Organization's limit on the total number of gigabytes of endpoint flow data	Number
num_daily_tables	The number of endpoint daily tables, counted once for each unique day	Number
num_weekly_tables	The number of endpoint weekly tables, counted once for each unique week	Number

Parameters for backlog

Parameters	Description	Type
total_disk_used_gb	Total gigabytes used to store flow data input files	Number
total_file_count	Total number of flow data input files	Integer

An example response looks such as the following:

```
{
  "org_id":1,
  "server":{
```

```

    "flows_size_gb":2.53228759765625,
    "num_flows_days":95,
    "flows_oldest_day":"2023-02-06",
    "num_daily_tables":7,
    "num_weekly_tables":13,
    "flows_size_gb_limit":26,
    "num_flows_days_limit":90
},
"endpoint": {
    "flows_size_gb":0.34337615966796875,
    "num_flows_days":6,
    "flows_oldest_day":"2023-05-11",
    "num_daily_tables":6,
    "num_weekly_tables":0,
    "flows_size_gb_limit":26,
    "num_flows_days_limit":14
},
"flows_days":95,
"flows_size_gb":2.8644485473632812,
"flows_days_limit":90,
"flows_oldest_day":"2023-02-06",
"flows_per_second":0.0,
"flows_size_gb_limit":26,
"updated_at":"2023-05-16T22:36:25Z"
}

```

## Reporting APIs

Reporting APIs allow users to generate application reports. Instead of first exporting generated data, such as traffic flows, and then using other tools to create reports, users can now use built-in reports.

Users can request one-time or recurring reports and specify time ranges and report types.

Reporting APIs belong to several groups based on their use:

### Reporting API Types

#### Reporting Schedules

These APIs allow the Global Organization Administrator (`this_global_org_user`) to manage report schedules.

Each report can be generated once or recurring, where the recurrence is specified during report configuration.

The default time range is 30 days, and other possible values are 1 day, 7 days, 14 days, 30 days, 60 days, and 90 days.

Functionality	HTTP	URI
Returns a collection of report schedules.	GET	[api_version][org_href]/report_schedule
Returns a scheduled report for the provided UUID.	GET	[api_version][org_href]/report_schedule/:report_schedule_id
Updates a report schedule for the provided UUID.	PUT	[api_version][org_href]/report_schedule/:report_schedule_id
Create a new report schedule.	POST	[api_version][org_href]/report_schedule
Deletes a report schedule for the provided UUID.	DELETE	[api_version][org_href]/report_schedule/:report_schedule_id

## Defining Report Schedule Query

To define the query for report schedules, reference the required schemas (explained in [Schemas to Define a Report](#)).

- `executive_summary_report_params.schema.json`
- `traffic_flow_report_params.schema.json`
- `report_app_groups.schema.json`
- `custom_date_range.schema.json`
- `ves_report_params.schema.json`

## Report Templates

These APIs enable the Global Organization Administrator (`this_global_org_user`) to manage report templates. In each report template, users can specify the type, time range, recurrence, and suitable parameters for the report.

Functionality	HTTP	URI
Lists the collection of all available report templates for this user and organization.	GET	[api_version][org_href]/report_templates
This API is used to enable or disable a specific report type, which can only be done by the organization administrator.	PUT	[api_version][org_href]/report_templates

## Defining Report Templates Query

To define the query for report templates, reference the required schemas (explained in [Schemas to Define a Report](#)).

## On-Demand Reports

The user authorized as the Global Organization Administrator (`this_global_org_user`) can download various kinds of reports, create them on demand, or add the property `report_format` to determine the format in which the report will be generated.

Functionality	HTTP	URI
Returns a collection of reports.	GET	[api_version][org_href]/reports
Returns a report for the provided UUID.	GET	[api_version][org_href]/reports/:report_id
Downloads a specific report.	GET	[api_version][org_href]/reports/:report_id/download
Creates a new on-demand report.	POST	[api_version][org_href]/reports
Cancels a report if it's not yet completed/failed	PUT	[api_version][org_href]/reports/:report_id
Added a new property <code>report_format</code> , which determines the format in which the report should be generated	POST	[api_version][org_href]/reports_schedules
Added a new property <code>report_format</code> , which determines the format in which the report should be generated	PUT	[api_version][org_href]/reports_schedules

## Report Settings

Report Settings define how many days a report will be stored or persisted.

The user authorized as the Global Organization Administrator (this\_global\_org\_user) can manage, list, or update report settings.

Functionality	HTTP	URI
Get the report settings for an organization.	GET	[api_version][org_href]/report_settings
Updates the report settings for an organization.	PUT	[api_version][org_href]/report_settings

## Schemas to Define a Report

These schemas are referenced and used to define the content of a report:

- `executive_summary_report_params.schema.json`

Reports parameters for the executive summary report, such as `report_time_range` (Time range the report is built across) and references to `report_time_range_definitions.schema.json#/definitions/custom_date_range` or `report_time_range_definitions.schema.json#/definitions/last_num_days`.

- `traffic_flow_report_params.schema.json`

Report parameters for the traffic flow query report.

- `report_app_groups.schema.json`

This is the App Group Schema for reports.

- `custom_date_range.schema.json`

Provides the time range across which the report is built.

- `common_legacy_workload_modes.schema.json`

Provides the assigned labels summary with the label URI, as well as the key and value in the key-value pair.

- `report_time_range_definitions.schema.json`

Provides the report parameters for the executive summary report, such as Start date for the range, End date for the range, and the Last x number of days the report is built across.

- `labels_summary.schema.json`

Provides the assigned labels summary with properties such as label URI, Key in key-value pair, and Value in key-value pair.

- `ves_report_params.schema.json`

Provides report parameters for the new VES (Vulnerability-Exposure Score) report type.

## Reporting APIs Reference

This topic contains properties and examples for reporting APIs

### Reporting Parameters and Properties

Parameter	Description	Parameter Type	Format
<code>org_id</code>	Organization	path	Integer
<code>report_schedule_id</code>	UUID of the report schedule	String	date/time

Property	Description	Parameter Type
href	Report Schedule URI	URI, required
report_template	Template for the report	Object, required
report_generation_frequency	How often to generate a report: in addition to daily, weekly, monthly, and quarterly reports, you can schedule to receive the report only once.	String
report_parameters	Any specific parameters required for this report template. These parameters are added as references:	Object, required
	<ul style="list-style-type: none"> <li>• executive_summary_report_params. schema.json</li> <li>• traffic_flow_report_params. schema.json</li> <li>• explorer_report_params. schema.json</li> <li>• ves_report_params.schema.json</li> <li>• ransomware_risk_report_params. schema.json</li> </ul>	
created_at	Timestamp (rfc3339 timestamp) in UTC when this report schedule was created	String
created_by	The URI of the user who created this report schedule	URI
updated_at	Timestamp (rfc3339 timestamp) when this report schedule was last updated.",	String
updated_by	The URI of the user who updated this report schedule	URI

## Properties for Report Templates

Property	Description	Parameter Type	Required
href	Report Template URI	String	Yes
name	Display name for this report template  maxLength: 255	String	Yes
report_parameters	Any specific parameters required for this report template to define one of the report types, using one of the listed schemas:  <ul style="list-style-type: none"> <li>• executive_summary_report_params.schema.json</li> <li>• traffic_flow_report_params.schema.json</li> <li>• explorer_report_params.schema.json</li> <li>• ves_report_params.schema.json</li> <li>• ransomware_risk_report_params. schema.json</li> </ul>	Object	Yes

## Parameters for On-Demand Reports

Parameter	Description	Parameter Type	Required
href		Integer	Yes
report_template	Template for the report	Object	Yes
status	Current status of this report	String	Yes
report_parameters	Any specific parameters required for this report template to define one of the report types, using one of the listed schemas:	Object	Yes
	<ul style="list-style-type: none"> <li>• executive_summary_report_params.schema.json</li> <li>• traffic_flow_report_params.schema.json</li> <li>• explorer_report_params.schema.json</li> <li>• ves_report_params.schema.json</li> <li>• ransomware_risk_report_params.schema.json</li> </ul>		
send_by_email	Flag for whether to send user reports by email.	Boolean	True/false
progress_percentage	Progress percentage for this report.	Integer	"minimum": 0, "maximum": 100

## Report Templates Examples

GET /orgs/:xorg\_id/report\_templates

List the report templates for this user and organization.

```
[
  {
    "href": "/orgs/1/report_templates/executive_summary_report",
    "name": "Executive Summary",
    "report_parameters": {}
  },
  {
    "href": "/orgs/1/report_templates/traffic_flow_report",
    "name": "Traffic Flow Query",
    "report_parameters": {
      "app_groups": []
    }
  }
]
```

This request references the following two schemas:

- executive\_summary\_report\_params.schema.json
- traffic\_flow\_report\_params.schema.json

## Report Schedules Examples

POST /api/v2/orgs/1/report\_schedules

Request to create a new report schedule:

```
{
  "report_template": {
    "href": "/orgs/1/report_templates/traffic_flow_report"
  },
  "name": "John's Traffic Flow Report - Quarterly",
  "report_generation_frequency": "quarterly",
  "report_parameters": {
    "report_time_range": {
      "last_num_days": 90
    },
    "app_groups": [
      ]
  }
}
```

Response (201 created)

```
{
  "href": "/orgs/1/report_schedules/8a08b381-c8fe-4837-b9c6-071c70861369",
  "report_template": {
    "href": "/orgs/1/report_templates/traffic_flow_report"
  },
  "name": "John's Traffic Flow Report - Quarterly",
  "report_generation_frequency": "quarterly",
  "report_parameters": {
    "app_groups": [],
    "report_time_range": {
      "last_num_days": 90
    }
  }
}
```

## On-demand Reports

POST /api/v2/orgs/1/reports

Request to create an on-demand report in the PDF format (`report_format`):

```
{
  "report_template": {
    "href": "/orgs/1/report_templates/executive_summary_report"
  },
  "description": "John's Executive Summary Report",
  "report_parameters": {
    "report_time_range": {
      "last_num_days": 30
    }
  },
  "report_format": "pdf"
}
```

Response

```
{
  "href": "/orgs/1/reports/be9b68ec-c35a-49bb-9400-f78c9950e321",
  "report_template": {
    "href": "/orgs/1/report_templates/executive_summary_report",
    "name": "Executive Summary Report"
  },
  "description": "John's Executive Summary Report",
  "created_at": "2021-01-15T05:45:27.130Z",
  "updated_at": "2021-01-15T05:45:27.130Z",
  "progress_percentage": 0,
  "generated_at": null,
  "status": "queued",
  "report_parameters": {
    "report_time_range": {
      "last_num_days": 30
    }
  },
  "created_by": {
    "href": "/users/1"
  },
  updated_by": {
    "href": "/users/1"
  }
}
```

## Report Settings

**GET /orgs/:xorg\_id/settings/reports**

Request to list report settings:

```
{
  "href": "/orgs/1/report_settings",
  "report_retention_days": 1,
  "enabled": true,
  "max_queued_reports": 25
}
```

## Ransomware Protection Dashboard APIs

The Ransomware Dashboard is powered by the two main APIs: `time_series` and `risk_summary`.

Multiple APIs are used to manage Ransomware Dashboard features and to generate reports about protection statistics:

## Risk Summary APIs

Risk summary APIs are the following:

**GET /api/v2/orgs/:xorg\_id/app\_groups/:app\_group\_id/risk\_details**

This API is added under specific `app_groups` to represent the `name` and `os_platforms` of the ransomware service.

This API is referencing the schema `workloads_by_exposure`:

### **workloads\_by\_exposure**

The schema `workloads_by_exposure` describes the number of workloads with one or more critically risky services as its highest risk. It is also used by the API `risk_summary_get`.

### **GET /api/v2/orgs/1/app\_groups/risk\_summary**

This API, added under `app_groups`, returns a ransomware risk summary for each app group.

This API is referencing the common schema `workload_exposure_severity`:

### **common workload\_exposure\_severity**

The common schema `workload_exposure_severity` describes exposed ransomware severity for a workload.

### **workloads\_get**

This Public Stable API was changed to support the Ransomware Dashboard in the following way:

One new object was added: `risk_summary`, which explains the risk summary for the workload. This object includes a required object `ransomware`, which supplies these properties:

- `workload_exposure_severity`
- `ransomware_protection_percent`
- `last_updated_at`

### **workloads\_risk\_details\_get**

This API, which supplies the risk details, you can see in action on the Workloads page, tab Ransomware Protection.

In addition to the organization admin, the users with access to the workload can view the ransomware protection details for that workload or how many risky ports are protected and how many risky ports are not protected.

### **workload\_ransomware\_services**

This schema is referenced from `workloads_risk_details_get` to supply the required service data:

- Service location and name
- Service Port and Protocol

- Severity and Protection state of this service
  - Status of the port on the workload
  - Active and Draft policy that applies to the Port

Information about the operating systems has been added for the ransomware service: Windows and Linux.

## **settings\_get**

This Public Stable API now includes two new properties: `num_assets_requiring_ransomware_protection` and `cloud_secure_tenant_id`.

```
},
  "num_assets_requiring_ransomware_protection": {
    "description": "number of assets that need ransomware protection for
this org",
    "type": [
      "integer",
      "null"
    ]
  },
=====
},
  "cloud_secure_tenant_id": {
    "description": "Cloud Secure tenant id corresponding to this
organization",
    "type": "string"
  }
}
}
```

## **settings\_put**

This Public Stable API was changed to include the new property `num_assets_requiring_ransomware_protection`, which provides a number of assets that need ransomware protection in a specific organization (1 - 9999999). Number of assets is between one and 9999999.

```
"properties": {
    "num_assets_requiring_ransomware_protection": {
        "description": "number of assets that need ransomware protection for
this org",
        "type": "integer",
        "minimum": 1,
        "maximum": 9999999
    },
=====
},
"cloud_secure_tenant_id": {
    "description": "Cloud Secure tenant id corresponding to this
organization",
    "type": "string"
}
```

```

    }
}
```

## Risky Services APIs

The new widget for the Ransomware Dashboard displays risky services and their protection coverage scores.

### **GET /api/v2/orgs/:xorg\_id/sec\_policy/:pversion/services**

The API now contains the property `average_protection_percent`, which is the average amount of ransomware protection percentages for all service ports in the service.

To include the needed data in the response to this request from the UI, the query parameter `include_ransomware_protection_percent` is used and set to `true` to include the average percentage data.

## Summary Reports

The APIs used to generate summary reports are:

- `reports_risk_summary_ransomware_timeseries_statistics_post`
- `reports_risk_summary_ransomware_timeseries_statistics_post_response`.
- `reports_risk_summary_get`
- `reports_time_series_statistics_post`
- `reports_time_series_statistics_post_response`

### **reports\_risk\_summary\_ransomware\_timeseries\_statistics\_post**

This API is used to show the time series data:

- Number of managed workloads
- Percent of the ransomware protection coverage
- Number of workloads by exposure

Data is presented with the granularity of `day`, `week`, `month`, and `quarter`, where the default is `day`.

Payloads for this API can be such as the following:

```
[ { "property": "num_workloads_by_exposure", "resolution": "day" } ]
```

```
[ { "property": "ransomware_protection_coverage_percent", "resolution": "day" } ]
```

### **reports\_risk\_summary\_ransomware\_timeseries\_statistics\_post\_response**

This API gives the start and end date of the time period.

### **reports\_risk\_summary\_get**

Security administrators use this API to view how many workloads are ransomware protection-ready and then assess the degree of protection in their whole system. This schema supplies the required information to run the Ransomware Dashboard:

- Number of total workloads
- Number of protected workloads
- Number of risky ports by the severity of their risk exposure (low, medium, high, and critical)
- Workload protection by the port type (admin and legacy)
- Ransomware protection coverage percent
- Date when the status was last updated

The property `risky_ports_by_category` was added to support the widget "Risky ports by type" in the UI.

Four required properties are added for the ransomware objects:

- `top_risky_applications`
- `top_risky_services`
- `num_risky_services`
- `num_unenforced_workloads`

### **`num_protected_unprotected_ports`**

This schema is referenced from `reports_risk_summary_get.schema.json` to supply the number of protected and unprotected ports for a specified risk level.

Other APIs that support Summary Reports

- `report_schedules_post`
- `report_schedules_put`
- `reports_schedules_get`
- `report_templates_get`
- `reports_get`

**APIs referencing `ransomware_risk_report_params` through the property `report_parameters`**

### **`reports_time_series_statistics_post`**

This schema supplies the granularity of the time series data.

The API `reports_time_series_statistics_post` includes these properties:

- `num_managed_workloads`, which is requested by the payload. The resolution might be `day`, `week`, `month`, and `quarter`, which defines what the UI will show. The default value is "day".
- `ransomware_protection_coverage_percent`: Percent of the ransomware protection coverage
- `num_workloads_by_exposure`: Number of workloads by exposure

Data is presented with the granularity of `day`, `week`, `month`, and `quarter`, where the default is `day`.

## **reports\_time\_series\_statistics\_post\_response**

This API specifies the time series data about the protected workloads.

This API gives the percentage of the end date of the counted period.

It is referencing the schema num\_workloads\_by\_exposure\_time\_series.

## **Ransomware Dashboard Reference**

This topic contains properties and examples for the Ransomware dashboard.

### **Ransomware API Details**

**GET /api/v2/orgs/:xorg\_id/app\_groups/:app\_group\_id/risk\_details**

```
=====
"workload_risk_summary_by_service": {
  "description": "Workload risk summary by ransomware service",
  "type": "array",
  "items": {
    "type": "object",
    "required": [
      "href",
      "name",
      "os_platforms",
      "port",
      "protocol",
      "severity",
      "num_unprotected_workloads",
      "num_protected_workloads",
      "average_ransomware_service_protection_coverage_percent"
    ]
  }
=====
```

This API is referencing the schema workloads\_by\_exposure:

```
=====
},
"workloads_by_exposure": {
  "description": "Workloads of this app group by Ransomware Exposure",
  "type": "object",
  "$ref": "workloads_by_exposure.schema.json"
}
=====
```

### **workloads\_by\_exposure**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "critical": {
```

```

    "description": "Number of workloads that have 1 or more critically
                    risky services as its highest risk",
    "type": "integer"
},
"high": {
    "description": "Number of workloads that have 1 or more high risk
                    services as its highest risk",
    "type": "integer"
},
"medium": {
    "description": "Number of workloads that have 1 or more medium risk
                    services as its highest risk",
    "type": "integer"
},
"low": {
    "description": "Number of workloads that have 1 or more low risk
                    services as its highest risk",
    "type": "integer"
},
"fully_protected": {
    "description": "Number of workloads that have no risky services
                    and are fully protected",
    "type": "integer"
}
}
}

```

**GET /api/v2/orgs/1/app\_groups/risk\_summary**

```

=====
},
"risk_summary": {
"description": "Risk Summary for this app group",
"type": "object",
"required": [
    "ransomware"
],
=====

```

**common workload\_exposure\_severity**

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Exposed ransomware severity for workload",
    "type": "string",
    "enum": [
        "critical",
        "high",
        "medium",
        "low",
        "fully_protected"
    ]
}
```

**workloads\_get**

```
{
  "properties": {
    "risk_summary": {
      "description": "Risk Summary for this workload",
      "type": "object",
      "required": [
        "ransomware"
      ],
      "properties": {
        "ransomware": {
          "type": [
            "object",
            "null"
          ],
          "required": [
            "workload_exposure_severity",
            "ransomware_protection_percent",
            "last_updated_at"
          ],
          "properties": {
            "workload_exposure_severity": {
              "$ref": "../common/workload_exposure_severity.schema.json"
            },
            "ransomware_protection_percent": {
              "description": "Ransomware protection percentage for this
                workload",
              "type": "number"
            },
            "last_updated_at": {
              "description": "The time at which the ransomware stats are
                last computed at",
              "type": "string",
              "format": "date-time"
            }
          }
        }
      }
    }
  }
}
```

**workloads\_risk\_details\_get**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "risk_details": {
      "type": "object",
      "required": [
        "ransomware"
      ],
      "properties": {
        "type": [
          "object",
          "null"
        ],
        "properties": {

```

```

    "details": {
      "type": "array",
      "items": {
        "$ref": "workload_ransomware_services.schema.json"
      }
    },
    "last_updated_at": {
      "description": "The time at which the protection stats were last computed at",
      "type": "string",
      "format": "date-time"
    }
  }
}

```

Sample Response for `workloads_risk_details_get`

```
{
  "risk_details": {
    "ransomware": {
      "services": [
        {
          "href": "/api/v2/orgs/8/workloads/23131cf5-1d70-42de-9242-39055338d0ef",
          "name": "SSH",
          "port": 22,
          "protocol": 17,
          "severity": "low",
          "port_status": "listening",
          "protection_state": "unprotected",
          "active_policy": "allowed",
          "draft_policy": "blocked",
          "recommendation": "add_boundary"
        },
        {
          "href": "/api/v2/orgs/8/workloads/23131cf5-1d70-42de-9242-39055338d0ef",
          "name": "SSH",
          "port": 22,
          "protocol": 6,
          "severity": "high",
          "port_status": "listening",
          "protection_state": "protected",
          "active_policy": "allowed",
          "draft_policy": "blocked",
          "recommendation": "has_draft_policy_needs_provisioning"
        }
      ],
      "last_updated_at": "2023-01-21 23:32:42.679673"
    }
  }
}
```

Sample Responses for `workloads_risk_details_get` when the evaluation concludes there is no risk for the workload.

When the results are not yet computed:

```
{
  "risk_details": {
    "ransomware": null
  }
}
```

The full response looks as follows:

```
[
  {
    "property": "num_managed_workloads",
    "time_series": [
      {
        "start_date": "2022-10-31",
        "end_date": "2022-11-2",
        "count": 120
      },
      {
        "start_date": "2022-10-24",
        "end_date": "2022-10-30",
        "count": 115
      },
      {
        "start_date": "2022-10-17",
        "end_date": "2022-10-23",
        "count": 110
      },
      {
        "start_date": "2022-10-10",
        "end_date": "2022-10-16",
        "count": 100
      }
    ]
  }
]
```

### **workload\_ransomware\_services**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "href",
    "port",
    "protocol",
    "severity",
    "port_status",
    "protection_state",
    "active_policy",
    "draft_policy"
  ],
  "properties": {
```

```

  "href": {
    "description": "Reference of the service",
    "type": "string"
  },
  "name": {
    "description": "Name of the service",
    "type": "string"
  },
  "port": {
    "description": "Port Number",
    "type": "integer",
    "minimum": 0,
    "maximum": 65535
  },
  "proto": {
    "description": "Protocol Number",
    "type": "integer"
  },
  "severity": {
    "description": "Severity of this service",
    "type": "string",
    "enum": [
      "low",
      "medium",
      "high",
      "critical"
    ]
  },
  "category": {
    "description": "Category of this service",
    "type": "string",
    "enum": [
      "admin",
      "legacy"
    ]
  },
  "port_status": {
    "description": "Status of the port on the workload",
    "type": "string",
    "enum": [
      "listening",
      "inactive"
    ]
  },
  "protection_state": {
    "description": "Protection state of this service",
    "type": "string",
    "enum": [
      "unprotected",
      "protected_open",
      "protected_closed"
    ]
  },
  "active_policy": {
    "description": "Active Policy that applies to this port",

```

```

    "type": "string",
    "enum": [
        "allowed",
        "allowed_across_boundary",
        "blocked_by_boundary",
        "blocked_no_rule"
    ]
},
"draft_policy": {
    "description": "Draft Policy that applies to this port",
    "type": "string",
    "enum": [
        "allowed",
        "allowed_across_boundary",
        "blocked_by_boundary",
        "blocked_no_rule"
    ]
},
"recommendation": {
    "description": "Recommendation for this port based on enforcement state, allow and deny rules and active/draft rule",
    "type": "string",
    "enum": [
        "add_boundary",
        "has_draft_policy_needs_provisioning"
    ]
}
}
}

```

Additional information about Windows and Linux.

```

{
  "properties": {
    "os_platforms": {
      "description": "Operating system for this ransomware service",
      "type": "array",
      "minItems": 1,
      "items": {
        "type": "string",
        "enum": [
          "windows",
          "linux"
        ]
      }
    }
  }
}

```

### **settings\_get**

New property num\_assets\_requiring\_ransomware\_protection.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "href": {
      "description": "Org Setting URI",
      "type": "string",
      "format": "uri"
    },
    "num_assets_requiring_ransomware_protection": {
      "description": "number of assets that need ransomware protection
                     for this org",
      "type": [
        "integer",
        "null"
      ]
    }
  },
  =====
}

settings_put
```

New property `num_assets_requiring_ransomware_protection` provides a number of assets that need ransomware protection in a specific organization (1 - 9999999).

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "num_assets_requiring_ransomware_protection": {
      "description": "number of assets that need ransomware protection
                     for this org",
      "type": "integer",
      "minimum": 1,
      "maximum": 9999999
    }
  },
  =====
```

### **GET /api/v2/orgs/:xorg\_id/sec\_policy/:pversion/services**

```
=====
{
  "properties": {
    "risk_details": {
      "properties": {
        "ransomware": {
          "properties": {
            "average_protection_percent": {
              "description": "This is the average of ransomware
                             protection percentage for all
                             the service ports in this service.",
              "type": "number"
            }
          }
        }
      }
    }
  }
}
```

```

        }
    }
}

=====

```

Example response with the parameter `include_ransomware_protection_percent` set to `true`:

```

[
{
    "href": "/orgs/2/sec_policy/draft/services/4852",
    "created_at": "2020-01-13T23:31:21.710Z",
    "updated_at": "2020-01-13T23:31:21.750Z",
    "deleted_at": null,
    "created_by": {
        "href": "/users/142"
    },
    "updated_by": {
        "href": "/users/142"
    },
    "deleted_by": null,
    "update_type": null,
    "name": "IST Common POPv3",
    "description": "Post Office Protocol v3",
    "description_url": null,
    "process_name": null,
    "external_data_set": "illumio_segmentation_templates",
    "external_data_reference": "1000032 -- Universal - Version 1",
    "service_ports": [
        {
            "port": 110,
            "proto": 6
        }
    ],
    "risk_details": {
        "ransomware": {
            "category": "legacy",
            "severity": "low",
            "os_platforms": [
                "linux",
                "windows"
            ],
            "average_protection_percent": 0.2
        }
    }
}

```

#### **reports\_risk\_summary\_ransomware\_timeseries\_statistics\_post**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "array",
    "items": {
        "type": "object",
        "required": [

```

```

    "property"
  ] ,
  "properties": {
    "property": {
      "description": "The property for which time series data is
requested.",
      "type": "string",
      "enum": [
        "num_managed_workloads",
        "ransomware_protection_coverage_percent",
        "num_workloads_by_exposure"
      ]
    },
    "resolution": {
      "type": "string",
      "description": "The granularity for the time series data. E.g.
day, week, month, quarter",
      "enum": [
        "day",
        "week",
        "month",
        "quarter"
      ],
      "default": "day"
    },
    "max_results": {
      "type": "integer",
      "default": 5
    }
  }
}
}

```

### **reports\_risk\_summary\_ransomware\_timeseries\_statistics\_post\_response**

A sample response of risk\_summary:

```
{
  "ransomware": {
    "num_total_workloads": 2,
    "num_protected_workloads": 0,
    "workloads_by_exposure": {
      "critical": 2,
      "high": 0,
      "medium": 0,
      "low": 0,
      "fully_protected": 0
    },
    "risky_ports_by_severity": {
      "critical": {
        "num_protected_ports": 0,
        "num_unprotected_ports": 6
      },
      "high": {
        "num_protected_ports": 0,

```

```

    "num_unprotected_ports": 8
  },
  "medium": {
    "num_protected_ports": 0,
    "num_unprotected_ports": 20
  },
  "low": {
    "num_protected_ports": 0,
    "num_unprotected_ports": 14
  }
},
"risky_ports_by_category": { ---- New section
  "admin": {
    "num_protected_ports": 0,
    "num_unprotected_ports": 26
  },
  "legacy": {
    "num_protected_ports": 0,
    "num_unprotected_ports": 22
  }
},
"ransomware_protection_coverage_percent": 0.0,
"last_updated_at": "2023-11-27T22:08:09Z"
}
}
}

```

A sample response of `ransomware_timeseries_statistics` with `ransomware_protection_coverage_percent`

```
[
  {
    "property": "ransomware_protection_coverage_percent",
    "time_series": [
      {
        "start_date": "2023-11-27",
        "end_date": "2023-11-27",
        "data": {
          "percentage": 59.67
        }
      },
      {
        "start_date": "2023-11-26",
        "end_date": "2023-11-26",
        "data": {
          "percentage": 56.0
        }
      },
      {
        "start_date": "2023-11-25",
        "end_date": "2023-11-25",
        "data": {
          "percentage": 23.8
        }
      },
      {
        "start_date": "2023-11-24",
        "end_date": "2023-11-24",
        "data": {
          "percentage": 23.8
        }
      }
    ]
  }
]
```

```

    "start_date": "2023-11-24",
    "end_date": "2023-11-24",
    "data": [
        "percentage": 23.0
    }
},
{
    "start_date": "2023-11-23",
    "end_date": "2023-11-23",
    "data": [
        "percentage": 5.0
    }
}
]
}
]

```

A sample response of `ransomware_timeseries_statistics` with `num_workloads_by_exposure`:

```

[
{
    "property": "num_workloads_by_exposure",
    "time_series": [
        {
            "start_date": "2023-11-27",
            "end_date": "2023-11-27",
            "data": {
                "critical": 2,
                "high": 0,
                "medium": 0,
                "low": 0,
                "fully_protected": 0
            }
        },
        {
            "start_date": "2023-11-26",
            "end_date": "2023-11-26",
            "data": {
                "critical": 2,
                "high": 0,
                "medium": 0,
                "low": 0,
                "fully_protected": 0
            }
        },
        {
            "start_date": "2023-11-25",
            "end_date": "2023-11-25",
            "data": {
                "critical": 2,
                "high": 0,
                "medium": 0,
                "low": 0,
                "fully_protected": 0
            }
        }
    ]
}
]
```

```

        }
    },
    {
        "start_date": "2023-11-24",
        "end_date": "2023-11-24",
        "data": {
            "critical": 2,
            "high": 0,
            "medium": 0,
            "low": 0,
            "fully_protected": 0
        }
    },
    {
        "start_date": "2023-11-23",
        "end_date": "2023-11-23",
        "data": {
            "critical": 2,
            "high": 0,
            "medium": 0,
            "low": 0,
            "fully_protected": 0
        }
    }
]
}
]

```

### **reports\_risk\_summary\_get**

The property `risky_ports_by_category` was added to support the widget "Risky ports by type" in the UI.

```

"risky_ports_by_category": {
    "description": "Risky ports by Port type",
    "type": "object",
    "properties": {
        "admin": {
            "$ref": "num_protected_unprotected_ports.schema.json"
        },
        "legacy": {
            "$ref": "num_protected_unprotected_ports.schema.json"
        }
    }
}

```

Four required properties are added for the ransomware object:

```

"required": [
    "ransomware"
],
"properties": {
    "ransomware": {
        "type": [

```

```

    "object",
    "null"
],
"required": [
    "num_total_workloads",
    "num_protected_workloads",
    "workloads_by_exposure",
    "risky_ports_by_severity",
    "risky_ports_by_category",
    "top_risky_applications",
    "top_risky_services",
    "num_risky_services",
    "num_unenforced_workloads",
    "last_updated_at"
]
}

```

### **num\_protected\_unprotected\_ports**

This schema is referenced from `reports_risk_summary_get.schema.json` to supply the number of protected and unprotected ports for a specified risk level:

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "required": [
        "num_protected_ports",
        "num_unprotected_ports"
    ],
    "properties": {
        "num_protected_ports": {
            "description": "Number of protected ports for this risk level,  
across all protection ready workloads",
            "type": "integer"
        },
        "num_unprotected_ports": {
            "description": "Number of unprotected ports for this risk level,  
across all protection ready workloads",
            "type": "integer"
        }
    }
}

```

### **APIs that support Summary Reports**

- `report_schedules_post`
- `report_schedules_put`
- `reports_schedules_get`
- `report_templates_get`
- `reports_get`

These five APIs are referencing `ransomware_risk_report_params` through the property `report_parameters`:

```

report_parameters: {
    "description": "Any specific parameters required for this report"
}

```

```

template",
  "type": "object",
  "anyOf": [
    {
      "$ref": "executive_summary_report_params.schema.json"
    },
    {
      "$ref": "traffic_flow_report_params.schema.json"
    },
    {
      "$ref": "explorer_report_params.schema.json"
    },
    {
      "$ref": "ves_report_params.schema.json"
    },
    {
      "$ref": "ransomware_risk_report_params.schema.json"
    }
  ]
}

reports_time_series_statistics_post

```

Data is presented with the granularity of day, week, month, and quarter, where the default is day.

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": {
    "type": "object",
    "required": [
      "property"
    ],
    "properties": {
      "property": {
        "description": "The property for which time series data is requested.",
        "type": "string",
        "enum": [
          "num_managed_workloads",
          "ransomware_protection_coverage_percent",
          "num_workloads_by_exposure"
        ]
      }
    }
}

```

### **reports\_time\_series\_statistics\_post\_response**

This API specifies the time series data about the protected workloads.

This API gives the percentage of the end date of the counted period.

It is referencing the schema num\_workloads\_by\_exposure\_time\_series.

```

"data": {
  "oneOf": [
    {
      "$ref": "../../../../agent/schema/v2/num_workloads_by_exposure_
              time_series.schema.json"
    },
    {
      "count": {
        "description": "The integer count on the end date of this period.",
        "type": "integer"
      }
    },
    {
      "percentage": {
        "description": "The percentage on the end date of this period.",
        "type": "number",
        "minimum": 0,
        "maximum": 100
      }
    }
  ]
}

```

## VEN Statistics APIs

The Dashboard uses the following API to aggregate various data from the system and help you focus on the data you are interested in:

```
POST api/v2/orgs/:xorg_id/vens/statistics
```

You can obtain summary statistics for VENs by specifying which statistics you are interested in from a set of options. The API also supports obtaining a count for a specific property value (such as a count of VENs from a specific product version).

VEN information can include: "status", "version", "health", "condition", "os\_id", "enforcement\_mode", and "ven\_type".

## VEN Dashboard Reference

This topic contains properties and examples for the VEN dashboard.

### Examples of VEN Dashboard APIs

vens\_statistics\_post

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "property_counts"
  ],
  "properties": {
    "property_counts": {
      "type": "array",

```

```
        "items": {
          "type": "object",
          "required": [
            "property"
          ],
          "properties": {
            "property": {
              "type": "string",
              "enum": [
                "status",
                "version",
                "health",
                "condition",
                "os_id",
                "enforcement_mode",
                "ven_type"
              ]
            },
            "values": {
              "type": "array",
              "items": {
                "type": "string"
              }
            },
            "filters": {
              "type": "array",
              "items": {
                "type": "object",
                "required": [
                  "filter_property"
                ],
                "properties": {
                  "filter_property": {
                    "type": "string",
                    "enum": [
                      "status",
                      "version",
                      "health",
                      "condition",
                      "os_id",
                      "enforcement_mode",
                      "ven_type"
                    ]
                  },
                  "values": {
                    "type": "array",
                    "items": {
                      "type": "string"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

    }
}
```

### Sample Request

```
{
  "property_counts": [
    {
      "property": "version",
      "values": [
        "19.3",
        "18.3"
      ],
      "filters": [
        {
          "filter_property": "status",
          "values": [
            "active",
            ""
          ]
        },
        {
          "filter_property": "containerized",
          "values": [
            "true"
          ]
        }
      ]
    },
    {
      "property": "version",
      "filters": [
        {
          "filter_property": "status",
          "values": [
            "active"
          ]
        }
      ],
      {
        "property": "health"
      }
    ]
  }
}
```

Sample Response for version and health

```
{
  "property_counts": [
    {
      "property": "version",
      "counts": [
        {
          "value": "18.1",
          "count": 1
        },
        {
          "value": "19.3",
          "count": 1
        }
      ]
    }
  ]
}
```

```
{
    "value": "18.2",
    "count": 1
},
{
    "value": "18.3",
    "count": 2
},
{
    "value": "19.1",
    "count": 2
}
]
},
{
    "property": "health",
    "counts": [
        {
            "value": "healthy",
            "count": 3
        },
        {
            "value": "warning",
            "count": 3
        },
        {
            "value": "err",
            "count": 2
        }
    ]
}
]
```

Sample Response for ven\_type

```
[ {
    property: "ven_type",
    total_filtered_count: 205,
    counts: [
        filtered_count:422,
        unfiltered_count: 424,
        value: "active",
    ]
}]
```

## Vulnerabilities

Vulnerabilities are defined as entries based on the possible risk of allowing traffic on a port/protocol combination, and a vulnerability instance is the existence of a vulnerability.

This Public Experimental API lists, creates, updates, and deletes vulnerabilities.

**NOTE**

The Illumio Core Vulnerability Maps license is required to import Qualys report data into the Illumio PCE. For information about obtaining the Illumio Core Vulnerability Maps license, contact Illumio Support. When you obtain your license, you also receive information about how to install it.

## Delete the Vulnerability License

To delete the vulnerability license, use the following CURL command from your CLI environment:

```
export API_KEY=api_key_username:api_key_secret

curl -i -H "Content-Type: application/json" https://pce_fqdn:8443/api/v2/orgs/org_id/licenses/9df01357-93cf-4f33-b720-e47bba783c55 -X DELETE -u $API_KEY
```

Replace the variables, which are entered in blue bold.

## Vulnerability API Methods

Functionality	HTTP	URI
Get vulnerabilities	GET	[api_version][org_href]vulnerabilities
Get an individual vulnerability.	GET	[api_version][org_href][vulnerabilities_href]
Create an individual vulnerability.	POST	[api_version][org_href][vulnerabilities_href]
Modify an individual vulnerability.	PUT	[api_version][org_href][vulnerabilities_href]
Delete an individual vulnerability.	DELETE	[api_version][org_href][vulnerabilities_href]

## Vulnerability Reports

This Public Experimental API creates, updates, and deletes vulnerability reports.

**NOTE**

An Illumio Core Vulnerability Maps license is required to import Qualys report data into the Illumio PCE. For information about obtaining the Illumio Core Vulnerability Maps license, contact Illumio Support. When you obtain your license, you also receive information about how to install it.

## Vulnerability Reports API Methods

HTTP	Functionality	URI
GET	Get a collection of vulnerability reports.	[api_version][org_href]/vulnerability_reports
GET	Get an individual vulnerability report.	[api_version][vulnerability_reports_href]
POST	Create an individual vulnerability report.	[api_version][vulnerability_reports_href]
PUT	Update an individual vulnerability report.	[api_version][vulnerability_reports_href]
DELETE	Delete an individual vulnerability report.	[api_version][vulnerability_reports_href]

### Get a Collection of Vulnerability Reports

This method gets a collection of all vulnerability reports in your organization.

By default, the maximum number of vulnerability reports returned by a GET collection is 500. For more than 500 vulnerability reports, use Asynchronous GET Collections.

### Delete a Vulnerability Report

To delete an individual vulnerability report, specify the last element of its HREF, which can be obtained from the response from `GET /vulnerabilities`.

### Delete a Vulnerability

To delete an individual vulnerability, specify its HREF, which can be obtained from the response from `GET /vulnerabilities`.

## Vulnerabilities API Reference

This topic contains properties and examples for vulnerability APIs.

### Examples of Vulnerability APIs

#### Get a Collection of all Vulnerabilities

This example sets the maximum number of vulnerability reports to 2. Not using this query parameter in this GET method would return all the vulnerability reports up to a maximum of 500.

Parameter	Description	Data Type
<code>org_id</code>		Integer
<code>max_results</code>	The maximum number of vulnerabilities returned by a call to  <code>GET /vulnerabilities</code> .  (Optional. If not specified, all vulnerabilities are returned up to a maximum of 500.)	Integer

**Curl Command to Get Collection of Vulnerabilities**

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/
vulnerabilities -H 'Accept: application/json' -u $KEY:$TOKEN
```

**Response Body**

```
[
  {
    "href": "/orgs/2/vulnerabilities/qualys-xxxxxebe7e17",
    "name": "Host Scan Time",
    "score": 37,
    "description": "{\"severity\":\"1\"}",
    "cve_ids": [],
    "created_at": "2017-12-21T19:15:48.000Z",
    "updated_at": "2017-12-21T19:17:26.000Z",
    "created_by": null,
    "updated_by": null
  },
  ...
]
```

**Get an Individual Vulnerability****Parameters**

Parameter	Description	Parameter Type
org_id	Organization	Integer
reference_id	The ID of the vulnerability to return by GET /vulnerabilities/{reference_id}.	String

**Curl Command to Get an Individual Vulnerability**

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/
vulnerabilities/qualys-xxxxxebe7e18 -H 'Accept: application/json' -u
$KEY:$TOKEN
```

**Response Body**

```
{
  "href": "/orgs/2/vulnerabilities/qualys-xxxxxebe7e18",
  "name": "Host Scan Time",
  "score": 37,
  "description": "{\"severity\":\"1\"}",
  "cve_ids": [],
  "created_at": "2017-12-21T19:15:48.000Z",
  "updated_at": "2017-12-21T19:17:26.000Z",
  "created_by": null,
  "updated_by": null
}
```

## Create or Update a Vulnerability

### Parameters

Parameter	Description	Parameter Type	Data Type
reference_id	The ID of the vulnerability. The reference_id is the last element of the href property returned by a call to GET /vulnerabilities.	Path	String
score	The normalized score of the vulnerability in the range of 0 to 100 inclusive. CVSS Score can be used here with a 10x multiplier.	Body	Integer
name	The title/name of the vulnerability.	Body	String
cve_ids	The cve_ids for the vulnerability.	Body	[String]
description	An arbitrary field to store details about the vulnerability class.	Body	String

### Curl Command to Create or Update Vulnerability

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/7/vulnerabilities/qualys-xxxxxebe7e18 -H 'Content-Type: application/json' -u $KEY:$TOKEN -d '{"score": 50, "cve_ids": ["CVE-2012-xxxx", "CVE-2017-xxxx"], "description": "My vulnerability test."}'
```

### Example Request Body

```
{
  "score": 50,
  "cve_ids": ["CVE-2012-xxxx", "CVE-2017-xxxx"],
  "description": "My vulnerability test."
}
```

### Response

On success, the system displays HTTP/1.1 204 No Content.

### Request Parameter to delete a vulnerability

Parameter	Description	Parameter Type	Data Type
reference_id	The reference ID of the vulnerability. The last element of the href property of a vulnerability returned by a call to GET /vulnerabilities.	Path	String

## Curl Command to Delete Vulnerability

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/7/vulnerabilities/qualys-xxxxxebe7e18 -u $KEY:$TOKEN
```

## Curl Command to Get Collection of Vulnerability Reports

In this example, the maximum number of vulnerability reports is set to 2. Not using this query parameter in this GET method would return all the vulnerability reports up to a maximum of 500.

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/vulnerability_reports -H 'Accept: application/json' -u $KEY:$TOKEN
```

## Query Parameter to Get a Collection of Vulnerability Reports

Parameter	Description	Parameter Type	Data Type
max_results	<p>The maximum number of vulnerability reports returned by a call to GET /vulnerability_reports.</p> <p>Optional. If not specified, by default, all vulnerability reports are returned up to a maximum of 500.</p>	Query	Integer

## Response Body

```
[
  {
    "href": "/orgs/2/vulnerability_reports/qualys-report-12345",
    "report_type": "qualys",
    "name": "my-report-2017-12-21-19-15-47",
    "created_at": "2017-12-21T19:15:48.000Z",
    "updated_at": "2017-12-21T19:15:48.000Z",
    "num_vulnerabilities": 4887,
    "created_by": null,
    "updated_by": null
  },
  {
    "href": "/orgs/2/vulnerability_reports/qualys-report-12346",
    "report_type": "qualys",
    "name": "my-report-2017-12-21-19-17-15",
    "created_at": "2017-12-21T19:17:15.000Z",
    "updated_at": "2017-12-21T19:17:15.000Z",
    "num_vulnerabilities": 1776,
    "created_by": null,
    "updated_by": null
  }
]
```

## Get a Vulnerability Report

### Curl Command to Get Vulnerability Report

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/7/
vulnerability_reports/qualys-report-123456 -H 'Accept: application/json' -u
$KEY:$TOKEN
```

### Request Parameter to Get an Individual Vulnerability Report

The following required path parameter restricts the results of the GET command to the specified vulnerability report.

Parameter	Description	Parameter Type	Data Type
reference_id	The ID of the vulnerability report (this is the last element in the vulnerability report HREF returned by a call to <code>GET /vulnerability_reports</code> ).	Path	String

### Response Body

```
{
  "href": "/orgs/2/vulnerability_reports/qualys-report-123456",
  "report_type": "qualys",
  "name": "my-report-2017-12-21-19-17-15",
  "created_at": "2017-12-21T19:17:15.000Z",
  "updated_at": "2017-12-21T19:17:15.000Z",
  "num_vulnerabilities": 1776,
  "created_by": null,
  "updated_by": null
}
```

### Create or Update a Vulnerability Report

#### Curl Command to Update a Vulnerability Report

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/7/
vulnerability_reports/qualys-report-123456 -H 'Content-Type: application/
json' -u $KEY:$TOKEN -d '{"name": "My vulnerability report", "report_type": "qualys"}'
```

#### Response Properties

Property	Description	Data Type
name	User generated the name of the vulnerability report.	Integer
report_type	A string representing the type of the report.	String
authoritative	Boolean value specifies whether a report is authoritative or not.	[String]
scanned_ips	The ips on which the scan was performed.  Enforced 100K maxitem limit.	String
detected_vulnerabilities	An array of parameters, of which ip_address, workload, and vulnerability are required.  Enforced 100K maxitem limit.  ip_address: (Required) The IP address of the host where the vulnerability is found (string)  port: The port associated with the vulnerability (integer)  proto: The protocol that is associated with the vulnerability (integer)  workload: (Required) The URI of the workload associated with this vulnerability (string)  vulnerability: (Required) The URI of the vulnerability class associated with this vulnerability (string)	
external_data_reference	(PUT only) This parameter supports third-party reference data	
state	(PUT only) Enables deletion, addition, or updating of vulnerabilities	
exported_at	(PUT only) Saves the timestamp for the next delta pull.	

## Example Request Body

```
{
  "name": "My vulnerability report",
  "report_type": "qualys",
  "authoritative": true
}
```

## Response

On success, the system displays HTTP/1.1 204 No Content.

## Curl Command to Delete Vulnerability Report

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/7/vulnerability_reports/qualys-report-2017-12-21-19-17-15 -u $KEY:$TOKEN
```

### Request Parameter

Parameter	Description	Parameter Type	Data Type
reference_id	The ID of the vulnerability report (this is the last element in the vulnerability report HREF returned by a call to GET /vulnerability_reports).	Path	String

## Vulnerability Exposure per Enforcement Mode

Before release 24.4, the vulnerability APIs allowed users to calculate vulnerability exposure only for the full enforcement mode.

The two new and several changed common schemas now support multiple calculated values for vulnerability exposure for each enforcement mode.

The UI is updated to allow users to see the exposure scores for different enforcement modes without changing the workload's enforcement mode.

These two new schemas are referenced from the other schemas.

### **common\_vulnerability\_summary\_exposure**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "description": "Vulnerability exposure details",
    "properties": {
        "vulnerable_port_exposure": {
            "description": "The aggregated vulnerability port exposure score of the workload in the specified mode across all the vulnerable ports",
            "type": [ "integer", "null" ]
        },
        "vulnerability_exposure_score": {
            "description": "The aggregated vulnerability exposure score of the workload in the specified mode across all vulnerable ports",
            "type": [ "integer", "null" ]
        }
    }
}
```

### **common\_workloads\_detected\_vulnerabilities\_exposure**

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
```

```

"type": "object",
"description": "Vulnerability exposure details for workloads",
"properties": {
    "vulnerable_port_exposure": {
        "description": "The exposure of the port based on the current policy for the specified enforcement mode",
        "type": ["integer", "null"]
    },
    "port_vulnerability_exposure_score": {
        "description": "The vulnerability exposure score calculated for the port, based on the port exposure and vulnerability for the specified enforcement mode",
        "type": ["integer", "null"]
    }
}
}

```

## APIs Affected

The following table shows the affected vulnerability APIs:

**Table 8. Vulnerability API Changes**

Meth-od	URL	Description
GET	/orgs/:xorg_id//workloads/<:id>/detected_vulnerabilities	Four new columns are being added to the workload vulnerabilities tables in the UI to support comparing exposure and ve-scores based on different enforcement types.
GET	/orgs/:xorg_id//workloads?representation=workload_labels_vulnerabilities	Four new columns are being added to the workload list tables in the UI to support comparing exposure and ve-scores for different enforcement types.
GET	/orgs/:xorg_id//workloads/:workload_id?representation=workload_labels_vulnerabilities	
GET	/orgs/:xorg_id/aggregated_detected_vulnerabilities	The vulnerability scores and summary scores are now in additional tables, and scores from them are added to the response.
GET	/orgs/:xorg_id/app_groups	
GET	/orgs/:xorg_id/workloads/detailed_vulnerabilities	The vulnerability data in response is not computed at runtime but is taken from the database generated by the proper stats processor.

## Vulnerability Exposure per Enforcement Mode Reference

This topic contains some examples and code changes for the vulnerability APIs that regulate exposure per enforcement mode.

### API Changes for Vulnerability APIs

Changes in the existing schemas are shown below.

#### **common aggregated\_detected\_vulnerability**

```
{
  "properties": {
    "aggregated_detected_vulnerabilities": {
      "items": {
        "properties": {
          "full_enforcement_vulnerability_exposure_added": {
            "$ref": "#/workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for full enforcement mode"
          },
          "selective_enforcement_vulnerability_exposure_added": {
            "$ref": "#/workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for selective enforcement mode"
          },
          "visibility_enforcement_vulnerability_exposure_added": {
            "$ref": "#/workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for visibility-only enforcement mode"
          },
          "current_enforcement_vulnerability_exposure_added": {
            "$ref": "#/workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for the current enforcement mode"
          }
        }
      }
    }
  }
}
```

### **common\_vulnerability\_summary**

```
{
  "properties": {
    "full_enforcement_vulnerability_exposure_added": {
      "$ref": "#/vulnerability_summary_exposure.schema.json",
      "description": "vulnerability exposure details for full enforcement mode"
    },
    "selective_enforcement_vulnerability_exposure_added": {
      "$ref": "#/vulnerability_summary_exposure.schema.json",
      "description": "vulnerability exposure details for selective enforcement mode"
    },
    "visibility_enforcement_vulnerability_exposure_added": {
      "$ref": "#/vulnerability_summary_exposure.schema.json",
      "description": "vulnerability exposure details for visibility-only enforcement mode"
    },
    "current_enforcement_vulnerability_exposure_added": {

```

```

"$ref": "vulnerability_summary_exposure.schema.json",
"description": "vulnerability exposure details for the current
enforcement mode"
},
"max_vulnerability_exposure_score__added": {
    "description": "The maximum vulnerability exposure score of the
workload in its current enforcement state across all vulnerable ports",
    "type": [
        "integer",
        "null"
    ]
},
"last_updated_at__added": {
    "description": "Indicates when the vulnerability data was last
updated",
    "type": "string",
    "format": "date-time"
},
"vulnerable_port_exposure": {
    "description": {
        "__old": "The aggregated vulnerability port exposure score of the
workload across all the vulnerable ports",
        "__new": "The aggregated vulnerability port exposure score of the
workload in full enforcement mode across all the vulnerable ports"
    }
},
"vulnerability_exposure_score": {
    "description": {
        "__old": "The aggregated vulnerability exposure score of the
workload across all the vulnerable ports.",
        "__new": "The aggregated vulnerability exposure score of the
workload in full enforcement mode across all the vulnerable ports."
    }
}
}
}

```

### **common workload\_detected\_vulnerabilities**

```

{
    "properties": {
        "last_updated_at__added": {
            "description": "Indicates when the vulnerability data was last
updated",
            "type": "string",
            "format": "date-time"
        },
        "workload_detected_vulnerabilities": {
            "items": {
                "properties": {
                    "full_enforcement_vulnerability_exposure__added": {
                        "$ref":
                            "workloads_detected_vulnerabilities_exposure.schema.json",
                        "description": "vulnerability exposure details for full
enforcement mode"
                    }
                }
            }
        }
    }
}

```

```
        },
        "selective_enforcement_vulnerability_exposure__added": {
            "$ref": "workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for selective enforcement mode"
        },
        "visibility_enforcement_vulnerability_exposure__added": {
            "$ref": "workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for visibility-only enforcement mode"
        },
        "current_enforcement_vulnerability_exposure__added": {
            "$ref": "workloads_detected_vulnerabilities_exposure.schema.json",
            "description": "vulnerability exposure details for the current enforcement mode"
        }
    }
}
```

## Workloads APIs

The Workloads APIs allow you to get information about workloads and network interfaces and to identify unauthorized traffic to or from workloads. Use the Workloads APIs to perform workload-related operations, such as pair workloads, configure pairing profiles, and obtain pairing keys.

Configure pairing profiles to apply properties to workloads as they pair with the PCE, such as what labels to apply. Configuring a pairing profile gives you a unique pairing key that identifies the VEN. Pair workloads to install VENs on them. The VEN reports detailed workload information to the PCE, such as which services run on the workload.

## Workload Operations

This Public Stable API allows you to perform workload operations, such as creating an unmanaged workload, updating workload information, unpairing a workload, and deleting a workload.

## Workload Methods

Functionality	HTTP	URI
Get a collection of all workloads.	GET	[api_version][org_href]/workloads
Get a specified workload.	GET	[api_version][org_href]/workloads/workload_id
Create an unmanaged workload.	POST	[api_version][org_href]/workloads
Update a workload or mark it as suspended.	PUT	[api_version]/workloads/workload_id

## Vulnerability Computation State

The new field `vulnerability_computation_state` is added to `vulnerability_summary` for all APIs that return the namespace. It defines three computation states:

- `not_applicable` (N/A) indicates that the vulnerability exposure score cannot be calculated and happens in the following cases:
  - Unmanaged workloads
  - Idle workloads
  - Vulnerabilities that have no port associated with them.
- `syncing`: For managed workloads, when the vulnerability exposure score hasn't been calculated yet, and the value is not available.
- `in_sync`: For managed workloads, when the workload with the VES value is calculated and available.

The following APIs have been updated to return `vulnerability_computation_state`:

- `workloads`(get collection) API
- `workloads/detailed_vulnerability`
- `workloads` (get instance)
- `workloads/:uuid/detected_vulnerabilities`
- `aggregated_detected_vulnerabilities`

## Vulnerability Exposure Score (VES) Filters

The `workloads` GET collection API includes query parameters to filter returned workloads based on their Vulnerability Exposure Score.

These vulnerability filters are considered experimental and might be changed in the future.

Specify these parameters to get all the workloads that have a specific score.



### NOTE

To use these new query parameters, you must also include the query parameter `representation=workload_labels_vulnerabilities`; otherwise, the PCE won't perform any vulnerability calculations.

Some examples of using the filters are:

```
GET api/v2/orgs/:xorg_id/workloads?
representation=workload_labels_vulnerabilities&vulnerability_summary.vulnerability_exposure_score%5Blte%5D=50
```

```
GET api/v2/orgs/:xorg_id/workloads?
representation=workload_labels_vulnerabilities&vulnerability_summary.vulnerability_exposure_score%5Bgte%5D=50&vulnerability_summary.vulnerability_exposure_score%5Blte%5D=999
```

## **Update Workload Information**

This API allows you to update information about a workload. To make this call, you need the URL of the workload you want to update, which is returned as an HREF path when you get either a single or a collection of workloads in an organization.

### **URI to Update an Individual Workload's Information**

```
PUT [api_version][workload_href]
```

#### **Example Payload**

This example shows what the JSON payload looks like for changing a workload's policy state (called mode in the API) from its current state to enforced.

```
{"agent": {"config": {"mode": "enforced", "log_traffic": true}}}
```

## **Mark Workload as Suspended**

You can use this API to mark a workload VEN as suspended or unsuspended.

### **URI to Mark a Workload VEN as Suspended or Unsuspended**

```
PUT [api_version][workload_href]
```

#### **Create an Unmanaged Workload**

The Unmanaged Workload API enables you to create a workload without installing the VEN on it. This API is commonly used for Kerberos authentication between the VEN and the PCE.

### **URI to Create an Unmanaged Workload**

```
POST [api_version][org_href]/workloads
```

#### **Delete a Workload Record**

If you have unpaired a workload, you can use this API to delete the workload's record from the PCE.

### **URI to Delete a Workload Record**

```
DELETE [api_version][workload_href]
```

## Workloads Going Offline

Three new properties are now available to describe `LOG_INFO` level notification, `LOG_WARNING` level notification, and `LOG_ERR` level notification for workloads going offline.

When a VEN does not contact the PCE within a set time interval, it is marked offline. Before that happened, a notification was created when the VEN was AWOL (missing) for 25% of the offline time.

These three new optional settings generate different levels of notifications to varying intervals so the user can customize the timing and levels of notification.

They are described in the schema `resource_canonical_representations`:

## Unpair Workloads

This API allows you to unpair workloads from the PCE by uninstalling the Illumio VEN from each workload. You can unpair up to 1,000 workloads at a time.

Pairing a workload installs the Illumio VEN on a workload. Unpairing a workload uninstalls the VEN from the workload so that the workload no longer reports any information to the PCE and can no longer receive any policy information.

When you unpair workloads with this API, you can set the state for the workload's iptables (Linux) or WFP (Windows) configuration.

### URI to Unpair a Workload

```
PUT [api_version][org_href]/workloads/unpair
```



#### IMPORTANT

The endpoint `workloads/unpair` is DEPRECATED. Use VEN Unpair instead.

- See Curl Commands in [VEN API Methods \[381\]](#).

## Workload Operations Reference

Learn about parameters, properties, and examples of workload operations.

## Workload Operations Parameters

Parameter	Description	Type	Re-required
org_id	Organization	Integer	Yes
agent.active_pce_fqdn	FQDN of the PCE	String	No
container_clusters	List of container cluster URIs, encoded as a JSON string	String	No
datacenter_net_lrol	Used to enforce the creation of pseudo-public interfaces	Boolean	No
enforcement_mode	Enforcement mode of workload(s) to return.	String	No
external_data_set	The data source from which a resource originates		
external_data_reference	A unique identifier within the external data source	String	No
include_deleted	Include deleted workloads	Boolean	No
ip_address	IP address of workload(s) to return. Supports partial matches	String	No
labels	List of lists of label URIs, encoded as a JSON string.	String	No
	From release 22.3.0, this API is not referencing <code>labels.schema.json</code> and it lists labels associated with this workload. Required properties are: <code>href</code> , <code>key</code> , and <code>value</code> .		
last_heartbeat_on[gte]	Greater than or equal to value for last heartbeat on timestamp	Integer	No
last_heartbeat_on[lte]	Less than or equal to value for last heartbeat on timestamp	Integer	No
log_traffic	Whether we want to log traffic events from this workload	Boolean	No
max_results	Maximum number of workloads to return.	Integer	No
mode	Management mode of workload(s) to return. DEPRECATED AND REPLACED (Use enforcement_mode)	String	No
name	Name of workload(s) to return. Supports partial matches	String	No
online	Return online/offline workloads using this filter	Boolean	No
os_id	Operating System of workload(s) to return. Supports partial matches	String	No
policy_health	Policy of health of workload(s) to return. Valid values: active, warning, error, suspended	String	No
security_policy_sync_state	Advanced search option for workload based on policy sync state	String	No
security_policy_update_mode	Advanced search option for workload based on security policy update mode	String	No

Parameter	Description	Type	Re- quired
soft_deleted	DEPRECATED WITH NO REPLACEMENT:  Only soft-deleted workloads	Boolean	No
ven	URI of the VEN to filter by. From release 22.3.0,in addition to providing the VENs HREF, it is required to give its <code>hostname</code> , <code>name</code> , <code>ven_type</code> , and <code>status</code> . The VEN properties are now clearly displayed, without a need to use expanded representations.  The <code>ven_type</code> property is introduced through the reference to a common schema <code>ven_type.schema.json</code> :	String	No
	<pre>{   "properties": {     "ven_type": {       "\$ref": "../common/ven_type.schema.json"     } }</pre>		
visibility_level	Filter by visibility level		No
vulnerability_summary. vulnerability_exposure_score[gte]	Greater than or equal to value for <code>vulnerability_exposure_score</code>	Integer	No
vulnerability_summary. vulnerability_exposure_score[lte]	Less than or equal to value for <code>vulnerability_exposure_score</code>	Integer	No

## Workload Operations Properties

Property	Description	Type	Re- quired
deleted	This workload has been deleted	Boolean	Yes
name	Interface name	String	Yes
managed	Return managed or unmanaged workloads using this filter. <code>True</code> if the workload is managed, else <code>false</code>	Boolean	Yes
hostname	Hostname of workload(s) to return. Supports partial matches	String	Yes
service_principal_name	The Kerberos Service Principal Name (SPN)	String	Yes
public_ip	The public IP address of the workload.	String	Yes
		Null	
interfaces	ref": "workloads_interfaces_get.schema.json		
service_provider	Name of the service source that is hosting the workload.	String	Yes
data_center	The name of the data center where the workload is being hosted.		Yes
data_center_zone	The zone inside the data center hosting the workload.	String	Yes
os_id	Unique OS identifier given by Illumio to the workload.	String	Yes
os_detail	Additional descriptive information about the workload OS	String	Yes
online	Indicates whether the workload is online and can communicate with the PCE.	Boolean	Yes
labels	Labels that are attached to the workload: <code>href</code> , <code>key</code> , and <code>value</code>	Array	Yes
services	This field contains the following data:  <ul style="list-style-type: none"> <li>• <code>uptime_seconds</code></li> <li>• <code>created_at</code></li> <li>• <code>open_service_ports</code>: with the following data: <code>protocol</code>, <code>address</code>, <code>port</code>, <code>process_name</code>, <code>user</code>, <code>package</code>, <code>win_service_name</code></li> </ul>		Yes
agent	DEPRECATED AND REPLACED (USE 'ven' INSTEAD).  Information about the agent that manages this workload.		Yes
created_at	The time (rfc3339 timestamp) at which this workload was created	String  date/ time	Yes
updated_at	The time (rfc3339 timestamp) at which this workload was last updated	String	Yes

Property	Description	Type	Re- quired
		date/ time	
vulnerabilities_summary	Reference to common/vulnerability_summary.schema.json		No
detected_vulnerabilities	Reference to common/workloads_detected_vulnerabilities.schema.json		No
ven	This section of the response returns the following data:  <ul style="list-style-type: none"> <li>• href</li> <li>• hostname</li> <li>• name</li> <li>• status</li> </ul>		No
container_cluster	Reference to common/compact_container_cluster.schema.json		No
ike_authentication_certificate	IKE authentication certificate for certificate-based Secure Connect and Machine Auth connections		No
risk_summary	Risk Summary for this workload. These properties have been added to the workload-get API to manage the features of the Ransomware Dashboard introduced in release 23.5.  <p>See also <a href="#">Ransomware Protection Dashboard APIs [298]</a>.</p> <p>For <code>risk_summary</code> object , the required object is <code>ransomware</code>. and it has three required properties:</p> <ul style="list-style-type: none"> <li>• <code>workload_exposure_severity</code>: This property is referencing /common/workload_exposure_severity.schema.json</li> <li>• <code>ransomware_protection_percent</code>: description: "Ransomware protection percentage for this workload", type: number</li> <li>• <code>last_updated_at</code>: description: "The time at which the ransomware stats are last computed at", type: string format: date-time</li> </ul>	Object	No

## Properties for Workload Disconnection

Property	Description	Type
workload_disconnected_timeout_second	Disconnected timeout in seconds	Integer
workload_goodbye_timeout_seconds	Goodbye timeout in seconds	Integer
workload_disconnect_notification_info	Threshold in seconds for LOG_INFO level notification of a workload going offline	Integer
workload_disconnect_notification_warning	Threshold in seconds for LOG_WARNING level notification of a workload going offline	Integer
workload_disconnect_notification_error	Threshold in seconds for LOG_ERR level notification of a workload going offline	Integer

## Example of Computation States:

**syncing:** Workload is in syncing state (VES is calculable but hasn't been calculated yet):

```
"vulnerability_summary": {
    "num_vulnerabilities": 30,
    "max_vulnerability_score": 88,
    "vulnerability_score": 1248,
    "vulnerable_port_exposure": null,
    "vulnerable_port_wide_exposure": {
        "any": null,
        "ip_list": null
    },
    "vulnerability_exposure_score": null,
    "vulnerability_computation_state": "syncing"
},
```

## Curl Command to Update a Workload

A workload state can be build, test, or enforced. This example shows how to use curl to update a workload policy state from its current state to enforced.

This example assumes that you want to update the state of a single workload in an organization. You can obtain an organization ID when you use the Users API to log in a user to Illumio.

```
curl -i -X PUT https://pce.my-company.com/api/v2/orgs/3/workloads/043902c883d133fa -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{"agent":{"config":{"mode":"enforced","log_traffic":true}}}'
```

## Example Payload for marking a workload as suspended

This example shows what the JSON payload looks like for marking a workload VEN as suspended, with the status property for the agent (the VEN) set to suspended.

To mark a workload VEN as unsuspended, use the same JSON body but replace `suspend` with `unsuspend`.

```
{
  "agent": {
    "status": {
      "status": "suspended"
    }
  }
}
```

### **Curl Command to Mark Workload as Suspended**

This example shows you how to use curl to mark a workload VEN as suspended.

This example assumes that you want to mark a single workload VEN as suspended. You can obtain an organization ID when you use the Users API to log in a user to Illumio.

```
curl -i -X PUT https://pce.my-company.com/api/v2/orgs/3/workloads/
043902c883d133 -H "Content-Type:application/json" -u $KEY:$TOKEN -d
'{"agent": {"status": {"status": "suspended"}}}'
```

### **Example payload for unmanaged workload**

For example, to create an unmanaged workload by providing a name, hostname, public IP address, and its Kerberos Service Principal Name, construct the JSON payload as follows:

```
{
  "name": "web_tier1",
  "hostname": "web_workload1.example.com",
  "public_ip": "10.10.10.10",
  "service_principal_name": "my_company-device-auth/
web_workload1.example.com",
}
```

### **Curl Command to Create an Unmanaged Workload**

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/4/
workloads -H "Content-Type: application/json" -u $KEY:$TOKEN -d
'{"name": "web_tier1", "hostname": "web_workload1.example.com", "public_ip": "10.10.10.10", "service_principal_name": "my_company-device-auth/web_workload1.example.com"}'
```

### **Parameters to unpair a workload**

Parameter	Description	Type	Required
org_id	Organization	Integer	Yes
workloads	Defines the list of workloads you want to unpair. You must specify at least one workload to unpair by defining the workload HREF. You can define up to 1,000 workloads to unpair with this API.  Required property:  <code>href</code> : URI of the workload to unpair.	Array	Yes
ip_table_restore	<p><b>IMPORTANT</b></p> <p>Use <code>/vens/unpair</code> and the parameter <code>firewall_restore</code> instead.</p>	String	Yes

This property allows you to determine the state of the workload iptables (Linux) or WFP (Windows) configuration after the workload is unpaired.

Options include:

- **saved**: Revert the iptables on the workload to the configuration before the VEN was installed. However, depending on how old the iptables or WFP configuration was on the workload, VEN removal could adversely impact security.
- **default**: Apply the recommended policy, which is to uninstall the VEN and allow only port 22 SSH connections to the workload. Safest from a security viewpoint, but if this workload is running a production application, it could break because this workload will no longer allow any connections to it.
- **disable**: Uninstall the VEN and leave all port connections on the workload open. This is the least safe from a security viewpoint. If iptables or WFP configuration or Illumio were the only security being used for this workload, the workload would be opened up to anyone and become vulnerable to attack.

## Example Payload for Unpairing Workloads

```
{
  "workloads": [
    {"href": "/orgs/7/workloads/XXXXXXXXX-9611-44aa-ae06-fXXX8903db65"},
    {"href": "/orgs/7/workloads/xxxxxxxx-9611-xxxx-ae06-f7bXXX03db71"}
  ],
  "firewall_restore": "saved"
}
```

## Curl Command for Unpairing Workload

```
curl -i -X PUT https://pce.my-company.com/api/v2/orgs/3/workloads/unpair -H "Content-Type:application/json" -u $KEY:$TOKEN -d '{"workloads": [{"href": "/orgs/7/workloads/XXXXXXXXX-9611-44aa-ae06-fXXX8903db65", "href": "/orgs/7/workloads/xxxxxxxx-9611-xxxx-ae06-f7bXXX03db71"}], "firewall_restore": "default"}'
```

## Workload Settings

This Public Stable API lets you get network interface information from a workload for all interfaces on a workload or an individual interface. You can also configure (create) or delete a network interface configuration.

### Workload Settings Methods

Functionality	HTTP	URI
Get agent timeout notifications.	GET	[api_version][org_href]/settings/workloads
Update agent timeout notifications.	PUT	[api_version][org_href]/settings/workloads

### Endpoint Offline Timer

The Endpoint Offline Timer was introduced to overcome the 24-hour limitation that was hard-coded for the endpoint's heartbeating.

If the endpoints did not heartbeat for 24 hours, they were marked as offline, and the endpoint timer was hard-coded to 24 hours. However, the 24-hour limit was found to be limiting, so it was adjusted to allow for endpoint mobility and usability.

The existing two APIs have been changed:

- **GET /api/v2/orgs/:xorg\_id/settings/workloads**: Added properties to reflect the endpoint timeout values: disconnect.
- **PUT /api/v2/orgs/:xorg\_id/settings/workloads**: Updated offline endpoint, heartbeat, disconnect, and quarantine warning timeout values.

The three workload timeout setting fields have been updated:

## Workload Timeout Setting Fields

Field	Description	Re- quired
workload_disconnected_timeout_seconds	Timer setting triggered if the server or endpoint has not heartbeaten to the PCE.  Referencing the schema <code>settings_workload_detailed.schema.json</code>	Yes
workload_goodbye_timeout_seconds	The timer setting is triggered if the server or endpoint operation is performed (stop, disable, etc.).  Referencing the schema <code>settings_workload_detailed.schema.json</code>	Yes
workload_disconnected_notification_seconds	Period to wait with no heartbeat before a warning is emitted.  Referencing the schema <code>settings_workload_notifications.schema.json</code>	Yes
ven_uninstall_timeout_hours	Defines the period (in hours) to wait before uninstalling a VEN.  Referencing the schema <code>settings_workload.schema.json</code>	Yes

## Schemas that Support the Endpoint Offline Timer

### **`settings_workload_notifications`**

This schema file was updated and now has an additional property `ven_type` to support the VEN type by the referenced timeout fields.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "scope",
      "warning"
    ],
    "properties": {
      "scope": {
        "$ref": "labels.schema.json"
      },
      "warning": {
        "description": "Workload disconnect warning timeout",
        "type": "integer",
        "minimum": -1,
        "maximum": 2147483647
      },
      "ven_type": {
        "description": "The ven type that this property is applicable to",
        "type": [
          "string",
          "null"
        ]
      }
    }
  }
}
```

```

        ],
        "enum": [
            "server",
            "endpoint"
        ]
    }
},
"uniqueitems": true
}

```

### **settings\_workload\_detailed**

The new schema `settings_workload_detailed` is expanded from the previous schema `settings_workload` so that additional information about the `ven_type` was added.

To ensure backend compatibility, the new field `ven_type` is specified as optional. If it is missing in the request, the parameter is considered to be of a server type.

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "array",
    "items": {
        "type": "object",
        "additionalProperties": false,
        "required": [
            "scope",
            "value"
        ],
        "properties": {
            "scope": {
                "$ref": "labels.schema.json"
            },
            "value": {
                "description": "Property value associated with the scope",
                "type": "integer",
                "minimum": -1,
                "maximum": 2147483647
            },
            "ven_type": {
                "description": "The ven type that this property is applicable to",
                "type": [
                    "string",
                    "null"
                ],
                "enum": [
                    "server",
                    "endpoint",
                    null
                ]
            }
        },
        "uniqueItems": true
    }
}

```

## Workload Settings Reference

This reference provides examples of workload settings.

### Examples

The example below represents the complete JSON string returned by the `GET /api/v2/orgs/:xorg_id/settings/workloads` request:

```
{
  "href": "/orgs/1/settings/workloads",
  "workload_disconnected_timeout_seconds": [
    {
      "scope": [ ],
      "value": 10800,
      "ven_type": "server"
    },
    {
      "scope": [ ],
      "value": 3600,
      "ven_type": "endpoint"
    },
    ],
    "workload_goodbye_timeout_seconds": [
      {
        "scope": [ ],
        "value": 12000,
        "ven_type": "server"
      },
      {
        "scope": [ ],
        "value": 7200,
        "ven_type": "endpoint"
      },
      ],
      "workload_disconnected_notification_seconds": [
        {
          {
            "scope": [ ],
            "info": 1800,
            "warning": 3600,
            "error": 5400,
            "ven_type": "server"
          },
          {
            "scope": [ ],
            "info": 1801,
            "warning": 3602,
            "error": 5403,
            "ven_type": "server"
          }
        }
      ],
      "ven_uninstall_timeout_hours": [
        {
          "scope": [ ],

```

```

        "value"=>300
    }
]
}
}
```

The following example shows how to set all four workload timeout setting properties via the `PUT /api/v2/orgs/:xorg_id/settings/workloads` request:

```
{
  "workload_disconnected_timeout_seconds": [
    {
      "scope": [ ],
      "value": 10800,
      "ven_type": "server"
    },
    {
      "scope": [ ],
      "value": 3600,
      "ven_type": "endpoint"
    },
  ],
  "workload_goodbye_timeout_seconds": [
    {
      "scope": [ ],
      "value": 12000,
      "ven_type": "server"
    },
    {
      "scope": [ ],
      "value": 7200,
      "ven_type": "endpoint"
    }
  ],
  "workload_disconnected_notification_seconds": [
    {
      {
        "scope": [ ],
        "info": 1800,
        "warning": 3600,
        "error": 5400,
        "ven_type": "server"
      },
      {
        "scope": [ ],
        "info": 1801,
        "warning": 3602,
        "error": 5403,
        "ven_type": "endpoint"
      }
    ],
    "ven_uninstall_timeout_hours": [
      {
        "scope": [ ],
        "value"=>300
      }
    ]
}
```

```

        }
    ]
}

```

## Workload Interfaces

This Public Stable API allows you to get network interface information from a workload for all interfaces on a workload or an individual interface. You can also configure (create) or delete an individual network interface configuration.

## Workload Interfaces Methods

Functionality	HTTP	URI
Request the list of the <b>workload_interfaces</b> (outside of the workloads or VENs scope). The <code>href</code> property in the API response is deprecated.	GET	<code>[api_version][workload_href]/interfaces</code>
Get an instance for the workload interface using the name. (DEPRECATED)	GET	<code>[api_version][workload_href]/interfaces/:name</code>
Directly creates a workload interface. The request payload was not changed. However, the <code>href</code> field in the API response is deprecated.	POST	<code>[api_version][workload_href] /interfaces</code>
Delete the workload interface with the name.(DEPRECATED)	DELETE	<code>[api_version][workload_href]/interfaces/:name</code>
Set the network manually and update the automatic network detection. (DEPRECATED)	PUT	<code>[api_version][workload_href]/interfaces/ :name/network</code>

## Get Workload Network Interface

This API allows you to get information about one or all of the interfaces on a workload. You can retrieve workload interface information, including its connectivity status (up, down, or unknown), interface IP address, subnet mask size, default gateway IP address, and associated network.

### URI to Get a Collection of a Workload's Network Interfaces

```
GET [api_version][workload_href]/interfaces
```

### Create Workload Network Interface

Directly creates a workload interface. The request payload was not changed. However, the `href` field in the API response is deprecated.

### URI to Create a Workload Network Interface Configuration

```
POST [api_version][workload_href]/interfaces
```

## Workload Interfaces Reference

This topic contains properties and examples for workload interfaces.

## GET Properties for workload interfaces

Property	Description	Type	Required
name	Interface name.	String	Yes
address	The IP address is assigned to the interface.	String	Yes
cidr_block	The number of bits in the subnet (for example, /24 is 255.255.255.0).	Integer, Null	Yes
default_gateway_address	The default IP address of the default gateway.	String, Null	Yes
link_state	State of the interface connection, which is one of three values: <ul style="list-style-type: none"><li>• up: Interface is communicating.</li><li>• down: Interface is not communicating.</li><li>• unknown: The state of the interface is unknown.</li></ul>	String, Null	Yes
network_detection_mode	Network Detection Mode	String, Null	Yes
friendly_name	A user-friendly name is given to the interface.	String, Null	Yes
network	A network that the interface belongs to	Object, Null	Yes
href	DEPRECATED WITH NO REPLACEMENT	String	No

## POST properties for workload interfaces

Properties	Description	Type	Re-required
name	The short, friendly name of the workload	String	Yes
link_state	State of the interface connection, which is one of three values: <ul style="list-style-type: none"><li>• up: Interface is communicating.</li><li>• down: Interface is not communicating.</li><li>• unknown: The state of the interface is unknown.</li></ul>	String	Yes
address	The IP address assigned to the interface.  Reference to common schema ip_address_format_validation.schema.json	String	No
cidr_block	The number of bits in the subnet (for example, /24 is 255.255.255.0).	Integer	No
default_gateway_address	The default IP address of the default gateway.  Reference to common schema ip_address_format_validation.schema.json	String	No
friendly_name	User-friendly name given to the interface.	String	No
href	DEPRECATED WITH NO REPLACEMENT	String	No

### Request Body

```
{
  "name": "eth0.public",
  "address": "192.0.2.0",
  "cidr_block": 32,
  "default_gateway_address": 255.255.255.0,
  "link_state": "up",
}
```

### Curl Command Create Network Interface

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/workloads/
xxxxxxxxx-c4e9-44e7-8a31-e86acf6b276c/interfaces -H "Content-Type:
application/json" -u $KEY:$TOKEN -d '{"name": "eth0.public", "address":
"192.0.2.0", "cidr_block": "32", "default_gateway_address": "255.255.255.0",
"link_state": "up"}'
```

### Examples

Request for all workload interfaces with a specific name

**Request:** GET /api/v2/orgs/:org\_id/workloads/:workload\_id/interfaces?name=eth0.public

The response includes the deprecated href field in the response:

```
[
  {
    "href": "/orgs/1/workloads/561bd65e-136c-4005-8aa2-bdc8af1b3600/
interfaces/eth0.public",
    "name": "eth0.public",
    "cidr_block": null,
    "link_state": null,
    "network_detection_mode": null,
    "friendly_name": null,
    "network": {
      "href": "/orgs/1/networks/366ff4c1-ec60-49be-a05f-3a5ccab09c2f"
    },
    "loopback": false,
    "address": "1.1.1.1",
    "default_gateway_address": null
  },
  {
    "href": "/orgs/1/workloads/561bd65e-136c-4005-8aa2-bdc8af1b3600/
interfaces/eth0.public",
    "name": "eth0.public",
    "cidr_block": null,
    "link_state": null,
    "network_detection_mode": null,
    "friendly_name": null,
    "network": {
      "href": "/orgs/1/networks/366ff4c1-ec60-49be-a05f-3a5ccab09c2f"
    },
    "loopback": false,
    "address": "2.2.2.2",
    "default_gateway_address": null
  }
]
```

API request/response creating new workload interface

**Request:** POST /api/v2/orgs/:org\_id/workloads/:workload\_id/interfaces

```
{
  "name": "eth1.private",
  "cidr_block": 32,
  "link_state": "up",
  "address": "99.99.99.7"
}
```

The Response body (with the href deprecated):

```
{
  "href": "/orgs/1/workloads/561bd65e-136c-4005-8aa2/interfaces/
eth1.private",
    "name": "eth1.private",
    "cidr_block": 32,
    "link_state": "up",
    "network_detection_mode": "single_private_brn",
    "friendly_name": null,
    "network": {
```

```

        "href": "/orgs/1/networks/5b25c11d-4e95-42d3-abd2-488506e48b02"
    },
    "loopback": false,
    "address": "99.99.99.7",
    "default_gateway_address": null
}

```

API request deleting multiple workload interfaces (bundle delete)

**Request:** PUT /api/v2/orgs/:org\_id/workloads/:workload\_id/interfaces/delete

Successful delete

Payload - all interfaces with the name `eth0.public` and only one interface with the name `eth1.private` are deleted.

Response code - 200

```

{
    "name": "eth0.public"
},
{
    "name": "eth1.private",
    "address": "10.10.10.1"
}

```

## Workload Bulk Operations

This Public Stable API supports “bulk” operations on collections of workloads. These operations create, update, or delete a collection of workloads using a single API call, instead of requiring separate API calls on individual workloads one at a time.



### IMPORTANT

Any tooling that parses the HTTP headers should be changed to allow case-insensitive header name matching to retain compatibility with future PCE releases. Refer to RFC 7230, section 3.2, “Header Fields,” which states that field names should be case insensitive.

## About Bulk Operations

When you use a bulk operations API to create a collection of workloads, the workload record is created in the PCE in the “unmanaged” state, which means that a VEN has not been installed on the workload, and no policy can be applied to the workload. To apply a policy to unmanaged workloads, you can do one of the following:

- Pair the workloads using the pairing script located in the PCE web console.

- Install and activate the VEN on the workload using the VEN control interface.

When you use this API to *update* a collection of workloads, those workloads can be either **managed** or **unmanaged**.

When you use this API to *delete* a collection of workloads, those workloads can only be **unmanaged**.

## Workload Bulk Operations Methods

Functionality	HTTP	URI
Create a collection of workloads.	PUT	[api_version][org_href]/workloads/bulk_create
Update a collection of workloads.	PUT	[api_version][org_href]/workloads/bulk_update
Delete a collection of workloads.	PUT	[api_version][org_href]/workloads/bulk_delete

### Caveats for Workload Bulk Operations

Bulk operations are rate-limited to 1,000 items per operation. When you create, update, or delete a collection of workloads (also called “bulk” operations), you can only execute one such operation at a time. This means you must wait for the first bulk operation to finish before executing a new one. If you execute a new bulk operation before the currently running operation has been completed, the second operation will fail with an `HTTP 429` error.

Additionally, when you perform a bulk workload operation, any policy changes caused by the operation are applied to the affected VENs after the next VEN heartbeat to the PCE.

After a bulk operation completes, *all* of your PCE VEN connectivity states show as `Syncing` until the next VEN heartbeat. Only affected VENs receive a policy update. VENs that are not affected by policy changes transition from `Syncing` to `In Sync` after the VENs heartbeat. This process can take up to 5 minutes.



### NOTE

Bulk operations are rate-limited to 1,000 items per operation.

## External Data Reference IDs for Workloads

External data references can add your own internal identifiers to new workloads independent of Illumio PCE-created workload HREFs. You can add these entities when you create a collection of workloads using the `bulk_create` method or create an individual workload using the public API.

External data references are useful if you want to keep a set of PCE resources in sync with your internal representation of the resources, such as a configuration management database (CMDB) that holds the “source of truth” for your workloads. Once workloads are created with these identifiers added to their properties, you can run an asynchronous query to get all workloads through an offline job, which includes the external data references in the response.

The schema for creating and updating External data references includes two properties:

- `external_data_set`: Identifies the original data source of the resource.
- `external_data_reference`: A unique identifier within that data source.

These properties are UTF-8 strings with a maximum length of 255 characters each. The contents must form a unique composite key, meaning that both values of these properties are treated as a unique key. Together, these two properties are recognized as unique keys, even if one of them is left blank or set to zero.

## Create a Collection of Workloads

URI to Create a Collection of Workloads

```
PUT [api_version][org_href]/workloads/bulk_create
```

## Update Collection of Workloads

This method allows you to update information about a collection of workloads. To update workload information, construct the JSON object for each workload exactly as you would for modifying one workload, but modify the properties as needed.

## URI to Update a Collection of Workloads

```
PUT [api_version][org_href]/workloads/bulk_update
```

## Delete a Collection of Workloads

You can use this Public Experimental API to delete a collection of unmanaged workloads.

When you call this method, you identify each unmanaged workload to delete by its HREF. For example:

```
/orgs/7/workloads/XXXXXXXX-9611-44aa-ae06-fXXX8903db65
```

If an unmanaged workload has the following two properties:

- `external_data_set=cmdb`
- `external_data_reference=25`

you can construct the HREF as a query parameter that matches the values of these two properties as they are defined on the target workload. For example:

```
/orgs/1/workloads?external_data_set=cmdb&external_data_reference=25
```



### NOTE

Both query parameters must match the exact same parameters on the workload for the delete operation to succeed.

## Bulk Import using a CSV File

### **workloads/bulk\_import**

This new API is used to update workloads using a CSV file, and the only allowed input type is 'text/csv'.

We recommend that users export a CSV file from the workloads page before they use this import function, so that they can just modify the CSV file they exported with the labels they would like to assign to the workloads.

- `PUT /api/v2/orgs/:xorg_id/workloads/bulk_import?delete_token`  
If the value in the CSV file for the `label_dimension` is the same as the delete token passed in the request, the label in that label dimension will be deleted for the workload. When users use CSV to update workload labels, they can pass in the delete token in the request to specify the labels to be deleted.
- `PUT /api/v2/orgs/:xorg_id/workloads/bulk_import?create_labels=true/false` (default is false)  
Provides an option in the CSV labels update to create new labels if they don't exist. If the option is `false`, rows with non-existent labels will be skipped entirely.
- `PUT /api/v2/orgs/:xorg_id/workloads/bulk_import?dry_run=true/false` (default is false)  
If users set this parameter to be `true`, the API will only return the potential changes and error tokens without making actual changes to the workloads.

## Workload Bulk Operations Reference

This topic contains examples of bulk workload operations.

### **Request Body for collection of workloads**

When creating a collection of workloads, you must pass a JSON object request body that describes the workload details.

This example illustrates the request body for a single managed workload, you can add as many workloads as you want.

For example:

```
{
  "name": "workload_0",
  "description": "workload_desc_0",
  "service_principal_name": "spn_0",
  "hostname": "workload-0.example.com",
  "public_ip": "24.1.1.1",
  "external_data_set": "cmdb",
  "external_data_reference": "0",
  "interfaces": [
    {
      "name": "eth0",
      "link_state": "up",
      "address": "10.0.0.2",
      "cidr_block": 24,
      "mac_address": "00:0C:29:AB:CD:EF"
    }
  ]
}
```

```

        "ip_version": 4,
        "default_gateway_address": "10.0.0.1",
        "friendly_name": "wan"
    }
],
"labels": [
{
    "href": "/orgs/2/labels/1"
},
{
    "href": "/orgs/2/labels/9"
}
],
"service_source": "service source",
"data_center": "central data center",
"os_id": "os id 0",
"os_detail": "os detail 0",
"online": true,
"agent": {
    "config": {
        "enforcement_mode": "full",
        "visibility_level": "flow_summary"
    }
}
}
]
}

```

## Curl Commands

### Bulk Create

This curl example illustrates how to create two workloads using the `bulk_create` API.

```

curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/workloads/
bulk_create -H "Content-Type:application/json" -u $KEY:$TOKEN -d
'[{ "name": "web_app1", "description": "workload desc
0", "service_principal_name": "spn 0",
"hostname": "workload-0.example.com", "public_ip": "24.1.0.1", "external_data_se
t": "cmdb", "external_data_reference": "0", "interfaces":
[ { "name": "eth0", "link_state": "up", "address": "10.0.0.2",
"cidr_block": 24, "ip_version": 4, "default_gateway_address": "10.0.0.1", "friendl
y_name": "wan" } ], "labels": [{ "href": "/orgs/2/labels/1"}, {"href": "/orgs/2/
labels/9"}], "service_source": "service source", "data_center": "central data
center", "os_id": "os id 0", "os_detail": "os detail 0", "online": true, "agent": {
"config": { "enforcement_mode": "visibility_only",
"visibility_level": "flow_summary" }}}]'

```

### Bulk Update Workloads

This example shows how to update two workloads with the Bulk Update API.

```

curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/workloads/
bulk_update -H "Content-Type:application/json" -u $KEY:$TOKEN -d
'[{ "name": "web_app1", "description": "workload desc
0", "service_principal_name": "spn
0", "hostname": "workload-0.example.com", "public_ip": "24.1.2.1", "external_data

```

```

_set": "cmdb", "external_data_reference": "0", "interfaces": [
  { "name": "eth0", "link_state": "up", "address": "10.0.0.2", "cidr_block": 24,
    "ip_version": 4, "default_gateway_address": "10.0.0.1", "friendly_name": "wan" } ],
  "labels": [ { "href": "/orgs/2/labels/1" }, { "href": "/orgs/2/labels/9" } ],
  "service_source": "service source", "data_center": "central data center",
  "os_id": "os id 0", "os_detail": "os detail 0", "online": true, "agent": {
    "config": {
      "enforcement_mode": "visibility_only", "visibility_level": "flow_summary" } } },
  { "name": "web_app2 0", "description": "workload desc 0",
    "service_principal_name": "spn 0", "hostname": "workload-0.example.com",
    "public_ip": "24.1.3.1", "external_data_set": "cmdb", "external_data_reference": "0", "interfaces": [
      { "name": "eth0", "link_state": "up", "address": "10.0.0.2", "cidr_block": 24, "ip_version": 4,
        "default_gateway_address": "10.0.0.1", "friendly_name": "wan" } ],
      "labels": [ { "href": "/orgs/2/labels/1" }, { "href": "/orgs/2/labels/9" } ],
      "service_source": "service source", "data_center": "central data center",
      "os_id": "os id 0", "os_detail": "os detail 0", "online": true, "agent": {
        "config": { "enforcement_mode": "full", "visibility_level": "flow_summary" } } } ] }

```

## Delete a Collection of Workloads

PUT [api\_version][org\_href]/workloads/bulk\_delete

### Request Properties

Property	Description	Type	Required
href	The HREF of a specific workload or unmanaged workload using the external_data_set and external_data_reference query parameters.	String	Yes

### Request Body

```
[
  { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ec551r" },
  { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ecd642ix" },
  { "href": "/orgs/1/workloads?external_data_set=cmdb&external_data_reference=25" },
  { "href": "/orgs/1/workloads/92f4a252-74d1-40ef-5af0-b46a4acd333dt" }
]
```

## Delete Collection of Workloads

```
curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/3/workloads/bulk_delete -H "Accept: application/json" -u $KEY:$TOKEN '[ { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ec551rse" }, { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ecd642ix" }, { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ecd5421q" }, { "href": "/orgs/1/workloads/92f4a252-68d1-40ef-8cf0-b46e4ecd214dt" }, { "href": "/orgs/1/workloads/c20efa40-c615-4fa7-b8a1-badbadbadbad" } ]'
```

### Response

A successful response returns an HTTP 200 message and an array of status objects, one for each workload and one that failed to be deleted as requested. If all unmanaged workloads are successfully deleted, an empty array is returned.

For example, two errors are shown—one where the operation was not allowed (due to lack of permissions) and one where the workload did not exist.

```
[  
  {  
    "href": "/orgs/1/workloads/c20efa40-c615-4fa7-b8a1-c3af4d34c5f5",  
    "errors": [ {"token": "method_not_allowed_error", "message": "Not  
allowed"} ]  
  },  
  {  
    "href": "/orgs/1/workloads/c20efa40-c615-4fa7-b8a1-badbadbadbad",  
    "errors": [ {"token": "not_found_error", "message": "Not found"} ]  
  }  
]
```

## Pairing Profiles and Pairing Keys

The Public Stable API for pairing profiles gets, creates, updates, and deletes pairing profiles.

### About Pairing Profiles and Keys

Pairing Profiles apply specific properties to workloads as they pair with the PCE, such as labels and the workload policy state.

When you configure a pairing profile, the pairing script contains a unique pairing key at the end of the script (activation code) that identifies the VEN securely so it can authenticate with the PCE. You can configure a pairing key for one-time use or more and set time and use limits.

The Pairing Key API can generate a new pairing key from a specified pairing profile.

### Pairing Profile Methods

Functionality	HTTP	URI
Get a collection of pairing profiles.	GET	[api_version][org_href]/pairing_profiles
Get the specified pairing profile.	GET	[api_version][org_href]/pairing_profile_href
Create an individual pairing profile.	POST	[api_version][org_href]/pairing_profiles
Update an individual pairing profile.	PUT	[api_version][pairing_profile_href]
Delete an individual pairing profile.	DELETE	[api_version][pairing_profile_href]

## Get Pairing Profiles

This method enables you to retrieve a collection of all pairing profiles in your organization or a specific pairing profile.

By default, the maximum number of pairing profiles returned in a GET collection is 500. For more than 500 pairing profiles, use Asynchronous GET Collection.

## Create a Pairing Profile

This method creates an individual pairing profile. The only required parameter for the POST method is enabled; others are optional.

### URI to Create a Pairing Profile

```
POST [api_version][org_href]/pairing_profiles
```

## Update a Pairing Profile

To update a pairing profile, specify its HREF, which can be obtained from getting a collection of pairing profiles.

### URI to Update a Pairing Profile

```
PUT [api_version][pairing_profile_href]
```

## Delete a Pairing Profile

To delete an individual pairing profile, specify its HREF, which you can obtain from a collection of pairing profiles.

### URI to Delete a Pairing Profile

```
DELETE [api_version][pairing_profile_href]
```

## Pairing Key API Method

Functionality	HTTP	URI
Create a pairing key	POST	[api_version][org_href]/pairing_profiles[pairing_profile_href] /pairing_key

## Create a Pairing Key

You need a pairing profile HREF to pass as a parameter to create a pairing key. You can obtain the pairing profile HREF from the pairing profile page in the PCE web console.

A pairing key is governed by the parameters configured in the pairing profile.

### URI to Create a Pairing Key

Obtain the pairing key HREF from the response body returned by an API call to get a collection of pairing keys.

```
POST [api_version][pairing_key_href]/pairing_key
```

## Pairing Profiles and Pairing Keys Reference

This topic contains properties and examples for pairing profiles and keys.

### GET a collection of Pairing Profiles

#### URI to Get a Collection of Pairing Profiles

```
GET [api_version][org_href]/pairing_profiles
```

#### Curl Command to Get Collection of Pairing Profiles

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/1/pairing_profiles -H 'Accept: application/json' -u $KEY:'TOKEN'
```

#### Parameters for Pairing Profiles

Parameter	Description	Type
org_id	Organization ID	
agent_software_release	The agent software release for pairing profiles.	
description	The long description of the pairing profile.	String
external_data_set	The data source from which the resource originates.  For example, if the pairing profile information is stored in an external database.	String NULL
external_data_reference	External data reference identifier	String Null
name	The short, friendly name of the pairing profile. Supports partial matches.	String
labels[]	Return only pairing profiles with all these labels specified in the pairing profile. Labels are structured in JSON as a list of lists of label HREFs.	Array
ven_type	Specifies the pairing profile by the VEN type: server, endpoint, or specified_during_activation	String

#### Properties for Pairing Profiles

All properties are required:

Property	Description	Type
name	The short, friendly name of the Pairing Profile	String
description	The long description of the pairing profile. Supports partial matches.	String
mode	Reference to the common schema <code>legacy_workload_modes.schema.json</code>	String
enabled	The enabled flag of the pairing profile	Boolean
total_use_count	The number of times the Pairing Profile has been used	Integer
allowed_uses_per_key	The number of times the pairing profile can be used.  Minimum: 1	String
key_lifespan	The number of seconds pairing profile keys will be valid.  Minimum: 1	Integer (min 1)
last_pairing_at	Timestamp when this pairing profile was last used for pairing a workload	String  NULL
created_at	Timestamp when this pairing profile was first created	String  date-time
updated_at	Timestamp when this pairing profile was last updated	String  date-time
created_by	The user who originally created this pairing profile  Reference by common schema <code>href_object.schema.json</code>	Object  String
updated_by	The user who last updated this pairing_profile  Reference by common schema <code>href_object.schema.json</code>	
last_pairing_key_generated_at	Timestamp of when the last pairing key was generated  • Null • Referenced by the common shema <code>href_object.schema.json</code>	String,  Null  date/time
last_pairing_key_generated_by	The user who generated the last pairing key	
is_default	Flag indicating this is the default auto-created Pairing Profile.	Boolean
labels	Reference to <code>labels.schema.json</code>	
env_label_lock	A flag that controls whether the <code>env</code> label can be overridden from the pairing script	Boolean

Property	Description	Type
loc_label_lock	A flag that controls whether the <b>loc</b> label can be overridden from the pairing script	Boolean
app_label_lock	A flag that controls whether the <b>app</b> label can be overridden from the pairing script	Boolean
enforce- ment_mode_lock	A flag that controls whether enforcement mode can be overridden from the pairing script	Boolean
mode_lock	DEPRECATED AND REPLACED    (USE /enforcement_mode_lock INSTEAD)	Boolean
	A flag that controls whether mode can be overridden from the pairing script.	
log_traffic	DEPRECATED AND REMOVED. Alerting status	Boolean
log_traffic_lock	DEPRECATED AND REMOVED.	Boolean
	A flag that controls whether log_traffic can be overridden from the pairing script	
visibility_level_lock	A flag that controls whether visibility_level can be overridden from the pairing script	Boolean
status_lock	A flag that controls whether status can be overridden from the pairing script	Boolean
external_data_set	External data set identifier	String, Null
external_data_refer- ence	External data reference identifier	String, Null
agent_software_re- lease	Agent software release associated with this paring profile	String, Null
ven_type	Referenced to common/pairing_profile_ven_type.schema.json	

## Examples of query parameters for filtering pairing profiles

Filter by Name

```
/api/v2/orgs/1/pairing_profiles?name=prod_app
```

Filter by Description

```
/api/v2/orgs/1/pairing_profiles?description="some descriptionstring"
```

Filter by software release

```
/api/v2/orgs/1/pairing_profiles?agent_software_release="xx.x.x"
```

## Response Body

The response includes generated pairing keys:

```
{
  "href": "/orgs/4002/pairing_profiles/4101",
  "name": "org 3 pp 1",
  "description": "org 3 pp 1",
  "total_use_count": 0,
  "enabled": true,
  "is_default": false,
  "created_at": "2022-01-21T00:44:16.863Z",
  "updated_at": "2022-01-21T00:44:16.863Z",
  "created_by": { "href": "/users/0" },
  "updated_by": { "href": "/users/0" },
  "mode": "illuminated",
  "enforcement_mode": "visibility_only",
  "key_lifespan": "unlimited",
  "allowed_uses_per_key": "unlimited",
  "last_pairing_at": null,
  "last_pairing_key_generated_at": "2022-01-21T00:49:13.841Z",
  "last_pairing_key_generated_by": { "href": "/users/6" },
  "labels": [ { "href": "/orgs/4002/labels/4104" } ],
  "env_label_lock": true,
  "loc_label_lock": true,
  "role_label_lock": true,
  "app_label_lock": true,
  "mode_lock": true,
  "enforcement_mode_lock": true,
  "log_traffic": false,
  "log_traffic_lock": true,
  "visibility_level": "flow_summary",
  "visibility_level_lock": true,
  "agent_software_release": "Default (19.3.0)",
  "caps": [ "write", "generate_pairing_key" ]
}
```

### **Example Request Body for a Pairing Profile**

```
{
  "href": "/orgs/2/pairing_profiles/12375",
  "name": "Limited Pairing",
  "description": "",
  "total_use_count": 0,
  "enabled": true,
  "is_default": false,
  "created_at": "2015-11-01T01:20:06.135Z",
  "updated_at": "2015-11-01T01:20:06.135Z",
  "created_by": {
    "href": "/users/18"
  },
  "updated_by": {
    "href": "/users/18"
  },
  "enforcement_mode": "visibility_only",
  "key_lifespan": "unlimited",
  "allowed_uses_per_key": "unlimited",
```

```

"last_pairing_at": null,
"labels": [
  {
    "href": "/orgs/2/labels/6"
  },
  {
    "href": "/orgs/2/labels/14"
  },
  {
    "href": "/orgs/2/labels/8"
  },
  {
    "href": "/orgs/2/labels/12"
  }
],
"env_label_lock": false,
"loc_label_lock": false,
"role_label_lock": false,
"app_label_lock": false,
"mode_lock": true,
"visibility_level": "flow_summary",
"visibility_level_lock": true
}
}

```

### Curl Command to Create Pairing Profiles

```

curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/2/
pairing_profiles -H "Content-Type:application/json" -u $KEY:'TOKEN' -d
'{"href":"/orgs/2/pairing_profiles/12375","name":"Limited
Pairing","description":"","total_use_count":0,"enabled":true,"is_default":fa
lse,"created_at":"2015-11-01T01:20:06.135Z","updated_at":"2015-11-01T01:20:0
6.135Z","created_by": {"href":"/users/18"}, "updated_by": {"href":"/users/
18"}, "enforcement_mode": "visibility_only", "key_lifespan": "unlimited", "allowe
d_uses_per_key": "unlimited", "last_pairing_at":null, "labels": [{"href":"/
orgs/2/labels/6"}, {"href": "/orgs/2/labels/14"}, {"href": "/orgs/2/labels/
8"}], "href": "/orgs/2/labels/
12"}],"env_label_lock":false,"loc_label_lock":false,"role_label_lock":false,
"app_label_lock":false,"visibility_level": "flow_summary", "visibility_level_l
ock":true}'

```

### Curl Command to Update Pairing Profiles

```

curl -i -X PUT https://pce.my-company.com:8443/api/v2/orgs/2/
pairing_profiles -H "Accept: application/json" -u $KEY:'TOKEN'-d '{
"href":"/orgs/2/pairing_profiles/12375","name":"Limited
Pairing","description":"","total_use_count":0,"enabled":true,"is_default":fa
lse,"created_at":"2015-11-01T01:20:06.135Z","updated_at":"2015-11-01T01:20:0
6.135Z","created_by": {"href":"/users/18"}, "updated_by": {"href":"/users/
18"}, "enforcement_mode": "visibility_only", "key_lifespan": "unlimited", "allowe
d_uses_per_key": "one_use", "last_pairing_at":null, "labels": [{"href":"/
orgs/2/labels/6"}, {"href": "/orgs/2/labels/14"}, {"href": "/orgs/2/labels/8"}],
{"href": "/orgs/2/labels/
12"}],"env_label_lock":false,"loc_label_lock":false,"role_label_lock":false,
"app_label_lock":false,"visibility_level": "flow_summary", "visibility_level_l
ock":true}'

```

## Curl Command to Delete Pairing Profile

```
curl -i -X DELETE https://pce.my-company.com:8443/api/v2/orgs/2/
pairing_profiles/12375 -H "Accept: application/json" -u $KEY:'TOKEN'
```

## Request Body for a Pairing Key

The request body is an empty JSON object.

```
{}
```

## Curl Command to Create Pairing Key

```
curl -i -X POST https://pce.my-company.com:8443/api/v2/orgs/3/
pairing_profiles/34/pairing_key -H 'Content-Type: application/json' -u
$KEY:'TOKEN' -d "{}"
```

## Filtering and Aggregating Traffic

This Public Stable API method allows you to handle broadcast and multicast traffic better, save storage in the traffic database, and reduce the stress of the whole data pipeline.

Windows-heavy environments can have a large amount of broadcast or multicast traffic, which can be as much as 50% in syslog data and 30% in traffic data. Because some broadcast and multicast data might not be useful for writing policies, this API provides a function to filter out or aggregate the broadcast and multicast traffic that is not useful.



### NOTE

This API is implemented in Supercluster.



### NOTE

Only Global Organization Owners can create/update/delete traffic collector settings, but more roles can read them.

## Traffic Collector API Methods

Use these methods to get, create, update, or delete a traffic collector.

Functionality	HTTP	URI
Get a traffic collector collection.	GET	[api_version][org_href]/settings/traffic_collector
Get a specific collector instance.	GET	[api_version][org_href]/settings/traffic_collector/:uuid
Create a traffic collector.	POST	[api_version][org_href]/settings/traffic_collector
Update a specific traffic collector instance.	PUT	[api_version][org_href]/settings/traffic_collector/:uuid
Delete a specific traffic collector instance.	DELETE	[api_version][org_href]/settings/traffic_collector/:uuid

## Filtering and Aggregating Traffic Reference

This topic contains properties and examples for filtering and aggregating traffic.

### Parameters for Traffic Collector Methods

Parameters	Description	Type
org_id	Org ID	Integer
traffic_collector_setting_id	traffic_collector setting UUID	String

## Properties for Traffic Collector Methods

Property	Description	Type
href	URI of the destination	String
transmis-sion	(For the transmission type, choose broadcast, multicast or unicast)	String
action	Drop or aggregate the target traffic:	String
	<ul style="list-style-type: none"> <li>If you select "drop," the PCE drops all the traffic that matches the filters you supply. The data will be lost forever.</li> <li>If you select "aggregate," the PCE aggregates broadcast and multicast traffic. If multiple workloads receive one broadcast or multicast traffic flow, all reported flows on the same traffic are aggregated into one record in the traffic database, and the destination workload information will be lost.</li> <li>The PUT method will fail if you change the aggregator from "broadcast" to "multicast" because the default port and protocol will not pass the validation step.</li> </ul>	
target	(PUT, POST) The target object has the following properties: <ul style="list-style-type: none"> <li>dst_port: Single destination ip address or CIDR . Can be an Integer or NULL</li> <li>proto: Port is required for POST</li> <li>dst_ip: Single destination ip address or CIDR</li> <li>src_port: Single source ip address or CIDR. Allows users to filter traffic based on the source port.</li> <li>src_ip: Single source ip address or CIDR</li> </ul> <p>If dst_port and dst_ip are not specified for the target session, traffic is dropped on "all ips" and "all ports" by default.</p> <p>The PUT method will fail If the traffic filter you want to modify has "ANY" in port or protocol field, and you want to modify other fields in this filter. The change will fail because the default port and protocol will not pass the validation step.</p> <p>Oracle flows are currently filtered via a runtime src_ip/dst_ip (CIDR) setting and this feature is not available in SaaS. Runtime changes also require a PCE restart, while API settings do not.</p> <p>The collector filters now support src_ip (CIDR) so that various filters can be created per organization without restarting the PCE.</p>	Object Integer Integer String String
data_source	Flow summary data source to support more granular filters, particularly for endpoints.	String
network	The Flow summary network supports more granular filters, for endpoints in particular.	String

## Examples of Traffic Collector Methods

### Curl Command for settings\_traffic\_collector\_post

```
curl -i -u
api_10415cd5bcc0e14cc:'2ac31cbee8cd3e8fa7ca79d32d39a0249636624ada675965dd2ec
239e3ea8af0' --request POST --data
'{"action":"drop","transmission":"unicast","target":
```

```
{"proto":6,"src_ip":"10.1.2.3"}' https://2x2testvc360.ilabs.io:8443/api/v2/orgs/2/settings/traffic_collector --header "Content-Type: application/json"
```

### Broadcast Transmission and Drop Action

```
curl 'https://pce.my-company.com:8443/api/v2/orgs/1/settings/traffic_collector' -H 'Origin: https://pce.my-company.com:8443' -H 'Accept-Encoding: gzip,deflate, br' -H 'content-type: application/json' -H 'accept: application/json' -H 'Referer: https://pce.my-company.com:8443/' -i -u api_1dfe2432a7b314ee6:'21c10ea1a4ad38d76ef22977e8ac45bc10839c5cc6ebffd650eae4f95dc5b364' --data-binary '{"transmission": "broadcast", "action": "drop", "target": {"proto": 17, "dst_port": 20, "dst_ip": "10.255.255.255"} }' --compressed
```

### Multicast Transmission and Aggregate Action

```
curl 'https://pce.my-company.com:8443/api/v2/orgs/1/settings/traffic_collector' -H 'Origin: https://pce.my-company.com:8443' -H 'Accept-Encoding: gzip, deflate, br' -H 'content-type: application/json' -H 'accept: application/json' -H 'Referer: https://pce.my-company.com:8443/' -i -u api_1dfe2432a7b314ee6:'21c10ea1a4ad38d76ef22977e8ac45bc10839c5cc6ebffd650eae4f95dc5b364' --data-binary '{"transmission": "multicast", "action": "aggregate"}' --compressed
```

### Example Response

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [ "href", "transmission", "action" ],
  "properties": {
    "href": {
      "description": "URI of the destination",
      "type": "string"
    },
    "transmission": {
      "description": "transmission type: broadcast/multicast",
      "type": "string",
      "enum": [
        "broadcast",
        "multicast"
      ]
    },
    "target": {
      "type": "object",
      "required": [
        "proto"
      ],
      "properties": {
        "dst_port": {
          "type": "integer"
        },
        "proto": {
          "type": "integer"
        }
      }
    }
  }
}
```

```

        },
        "dst_ip": {
            "type": "string",
            "description": "single ip address or CIDR"
        }
    }
},
"action": {
    "description": "drop or aggregate the target traffic",
    "type": "string",
    "enum": [
        "drop",
        "aggregate"
    ]
}
}
}

```

## VEN Operations

### Overview of VEN Suspension

The VEN Update API (`PUT [api-version][ven-href]`) allows you to mark a VEN as either suspended or unsuspended in the PCE. It does not, however, actually suspend or unsuspend the VEN.

To suspend a VEN, use the `illumio-ven-ctl` command-line tool, which isolates a VEN on a workload so that you can troubleshoot issues and determine if the VEN is the cause of any anomalous behavior.

When you suspend a VEN, the VEN informs the PCE that it is in suspended mode.

However, if the PCE does not receive this notification, you must mark the VEN as "Suspended" in the PCE web console (select the VEN and click `Edit`), or you can use this API to mark the VEN as suspended.

When you don't mark the VEN as suspended in the PCE, the PCE assumes that the workload is offline and removes it from the policy after one hour. When you mark the VEN as suspended, that VEN's workload is still included in the policy of other workloads.

When a VEN is suspended:

- The VEN still appears in the PCE on the VEN list page.
- The VEN's host cannot be unpaired.
- An organization event (`server_suspended`) is logged. This event is exportable to CEF/LEEF and has the severity of WARNING.
- Heartbeats or other communications are not expected, but when received, all communications are logged by the PCE.
- If the PCE is rebooted, the VEN remains suspended.
- Any custom iptables rules are removed, and you need to reconfigure them manually.

When a VEN is unsuspended:

- The PCE is informed that the VEN is no longer suspended and can receive policy from the PCE. If existing rules affect the unsuspended workload, the PCE reprograms those rules.
- An organization event (`server_unsuspended`) is logged. This event is exportable to CEF/LEEF and has a WARNING severity.
- The workload reverts to the policy state it had before it was suspended.
- Custom iptables rules are configured back into the iptables.

VEN upgrade APIs allow you to specify an array of VEN HREFs to upgrade to a specific version instead of a list of agent IDs.

Rules when validating with the VEN upgrade APIs are as follows:

- No downgrades are allowed.
- Users cannot upgrade to a VEN version higher than the PCE version.
- No AIX, Solaris, or C-VEN are allowed.
- Users can only upgrade VENs paired to the same region.
- Only workload managers can upgrade VENs, and they can only upgrade VENs within their scope.

## **VEN API Methods**

In addition to the page in the PCE web console that lists all VENs and shows the details of a single VEN, there is a Public Experimental API for getting VEN collections and VEN instances. Other new APIs support VEN filtering in the PCE web console and update and unpair VENs.

VEN Methods	HTTP	URI
Get the collection of all VENs (The href property in each interface in the ven interfaces array is dropped from the response).	GET	[api_version][org_href]/vens/
Get details on a VEN instance (The href property in each interface in the VEN interfaces array is dropped from the response)	GET	[api_version][org_href]/vens/ven_id
Use to get the default release without iterating through the whole collection.	GET	[api_version][org_href]/software/vens/default
Support VEN filtering in the PCE web console.	GET	[api_version][org_href]/vens/autocomplete
		[api_version][org_href]/vens/facets
To set the target_pce_fqdn on a VEN	PUT	[api_version][org_href]/vens/ven_id
Update a VEN	PUT	[api_version][org_href]/vens/update
Upgrade a VEN. This API accepts a list of hrefs instead of agent_ids. The upgrade endpoint falls under /vens/resource instead of the /software/resource.	PUT	[api_version][org_href]vens/upgrade
Lists the VEN releases available to the org, one per VEN version, along with metadata such as whether it is the default version, whether that release supports servers and/or endpoints, and so on. The list of images is longer than the list of releases; multiple images belong to the same version.	GET	[api_version]/software/ven/releases
Shows the complete list of VEN images. There is one image for each Linux distribution we support (such as RHEL and Ubuntu), plus images for Windows and macOS.	GET	[api_version]/software/ven/releases-images
Unpair a VEN: trigger the unpairing of one or more VENs.	PUT	[api_version][org_href]/vens/unpair
<b>NOTE:</b> This endpoint replaces <a href="#">/workloads/unpair [336]</a> , which is deprecated.		
Provided so that users can set the default version for VEN pairing.	PUT	[api_version][org_href]/software/vens/default

## Software VEN Releases

### release\_ven\_types

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Supported ven types in this release",
  "type": "array",
  "items": {
    "type": "string",
    "enum": [ "server", "endpoint" ]
  }
}
```

The new common schema `release_ven_types` is introduced to show `ven_types` for each release and to filter releases by `ven_type`.

Previously, the `ven_type` was not stored for the release, and database records looked as follows:

<b>Release</b>	<b>Distribution</b>
22.5.1	CentOS
22.5.1	MacOS
22.5.1	Windows

With the property `ven_type` added, the database records are expanded with an additional `ven_types` column:

<b>Release</b>	<b>Distribution</b>	<b>ven_types</b>
22.5.1	CentOS	server + endpoint
22.5.1	MacOS	server + endpoint
22.5.1	Windows	server + endpoint

Note that in release 22.5.1, the code supports the type "server+endpoint". However, Centos (Linux) supports a server-only VEN image, MacOS supports an endpoint-only image, and Windows supports both server and endpoint:

<b>Release</b>	<b>Distribution</b>	<b>ven_types</b>
22.5.1	CentOS	server
22.5.1	MacOS	endpoint
22.5.1	Windows	server + endpoint

When a user opens the list of release images via UI and looks for the type `server + endpoint`, only the Windows image will appear as a complete match.

To fix this issue, the `ven_type` is now based on release and distribution:

- All releases before 21.2.2 were just `server` (there was no endpoint)
- Any release with 22.3.x was an `endpoint` (there was no server)
- Any other releases were `server + endpoint`, but instead of setting it to all the images (database records), the `ven_types` are set in a specific way for the Os.

## GET VENs

To get a collection of VENs with a specific label applied, take a label string that was returned when you got a collection of VENs, and then add a query parameter to GET all VENs with that specific label.

## Network Enforcement Nodes (NEN) APIs

### Network Enforcement Node Reassignment

`network_enforcement_nodes_put`

This API is used to change an agent's target PCE FQDN.

It updates the `target_pce_fqdn` property so that a different PCE can manage the NEN in a Supercluster.

### Change Target PCE

When you have the NEN HREF, you can update the target PCE with the PCE FQDN the NEN will use. In your JSON request body, pass the following data:

```
"target_pce_fqdn": "new-pce-fqdn.example.com"
}
```

The URI for this operation is:

`PUT [api_version][nen_href]/update`

This curl example shows how you can pass the `target_pce_fqdn` property containing the FQDN of the new PCE:

```
curl -u
api_xxxxxxxxxx64fce809:'xxxxxxxx5048a6a85ce846a706e134ef1d4bf2ac1f253b84c1bf8df
6b83c70d95' -H "Accept: application/json" -H "Content-Type:application/
json" -X PUT -d '{"target_pce_fqdn":"new-pce-fqdn.example.com"}' https://
my.pce.supercluster:443/api/v1/orgs/3/network_enforcement_nodes/f67d35d5-
ea71-42da-b40d-8dcc3b1420c2/update
```

### Authorization and Exposure Changes

Release 23.5.0 changes some of the existing experimental APIs to facilitate the creation of fully scripted endpoint management system integrations with the PCE using the Network Enforcement Nodes (NEN) Switch integration capabilities.

### Exposure Changes

Exposure of the listed NEN APIs was changed in release 23.5.0 from Public Experimental to Public Stable.

### Authorization Changes

In release 23.5.0, the authorization of some NEN APIs was changed from the default ("Global Administrator" and "Global Organization Owner") to authorize additional users, as listed in the table.

API	Exposure Change	New Authorization Change
network_device_config	YES	NO
network_device_get	YES	NO
network_device_network_end-point_get	YES	NO
network_devices_enforcement_instructions_applied_post	YES	"Global Policy Object Provisioner" and "Ruleset Provisioner"
network_devices_enforcement_instructions_request_post	YES	"Global Policy Object Provisioner" and "Ruleset Provisioner"
network_devices_get	YES	"Global Policy Object Provisioner", "Global Read Only", "Limited Ruleset Manager", "Ruleset Provisioner", "Ruleset Viewer", "Workload Manager"
network_devices_multi_enforcement_instructions_applied_post	YES	"Global Policy Object Provisioner" and "Ruleset Provisioner"
network_devices_multi_enforcement_instructions_request_post	YES	"Global Policy Object Provisioner" and "Ruleset Provisioner"
network_devices_network_end-points_get	YES	NO
network_devices_network_end-points_post	YES	"Workload Manager"
network_devices_network_end-points_put	YES	"Workload Manager"
network_devices_put	YES	"Workload Manager"
network_endpoint_config	YES	NO
network_enforcement_node_get	YES	NO
network_enforcement_nodes_get	YES	"Full Ruleset Manager", "Global Policy Object Provisioner", "Global Read Only", "Limited Ruleset Manager", "Ruleset Provisioner", "Ruleset Viewer", "Workload Manager"
network_enforcement_nodes_network_devices_post	YES	"Workload Manager"
network_enforcement_nodes_put	YES	NO

## VEN Operations Reference

This topic contains parameters and examples for VEN operations.

## VEN Parameters

Parameter	Description	Type	Re-required
org_id	Organization ID	Integer	Yes
activation_type	The method by which the VEN was activated	String	No
active_pce_fqdn	FQDN of the PCE	String	No
activation_recovery	Return VENs in or not in authentication recovery.	Boolean	No
condition	A specific error condition to filter by	String	No
container_clusters	The array of container cluster URIs, encoded as a JSON string	Object	No
disconnected_before	Return VENs that have been disconnected since the given time.	date/time	No
health	The overall health (condition) of the VEN	String	No
hostname	The hostname of VEN(s) to return. Supports partial matches.	String	No
ip_address1	IP address of VEN(s) to return. Supports partial matches	String	No
last_goodbye_at	The time (rfc3339 timestamp) of the last goodbye from the VEN.	String, Null	
os_platform	OS platform of the host managed by the VEN	String, Null	
version	Software version of the VEN.	String	
status	The current status of the VEN. Options are:  "active", "suspended", "uninstalled."	String	
activation_type	The method in which the VEN was activated. Options are:  "pairing_key", "kerberos", "certificate."	String, Null	No
active_pce_fqdn	The FQDN of the PCE that the VEN last connected to	String, Null	No
target_pce_fqdn	cluster FQDN for target PCE	String, Null	
labels	Labels assigned to the host that are managed by the VEN.	Array	
interfaces	Network interfaces of the host that are managed by the VEN.	Array	
workloads	The only required property is <code>Href</code> ; the others are optional:  name, managed, hostname,  <code>os_id</code> , <code>os_detail</code> , <code>labels</code> ,	Array	

Parameter	Description	Type	Re- quired
	interfaces, etc.		
description	Description of VEN(s) to return. Supports partial matches	String, Null	
last_heartbeat_at	The last time (rfc3339 timestamp) a heartbeat was received from this VEN.	String, Null	
status	VEN Status:  • "active" • "suspended"	String	
ven_type	The <code>ven_type</code> property is introduced through the reference to a common schema <code>ven_type.schema.json</code> :	String	No

## VEN Properties

Parameter	Description	Type	Re- quired
ven_type	The type of the release marked as default:  "server", "endpoint"	String	No
default_release_ven_types	The type of the release marked as default	String	
name	Friendly name for the VEN	String, Null	
hostname	The hostname of the host managed by the VEN	String, Null	Yes
uid	The unique ID of the host managed by the VEN	String, Null	
os_id	OS identifier of the host managed by the VEN	String, Null	
os_detail	Additional OS details from the host managed by the VEN	String, Null	
os_platform	OS platform of the host managed by the VEN	String, Null	
version	Software version of the VEN.	String	
status	The current status of the VEN. Options are:  "active", "suspended", "uninstalled"	String	
activation_type	The method in which the VEN was activated. Options are:  "pairing_key", "kerberos", "certificate"	String, Null	No
active_pce_fqdn	The FQDN of the PCE that the VEN last connected to	String, Null	No
target_pce_fqdn	cluster FQDN for target PCE	String, Null	
labels	Labels assigned to the host managed by the VEN.	Array	
interfaces	Network interfaces of the host managed by the VEN.	Array	
workloads	The only required property is <code>HREF</code> , the others are optional:  name, managed, hostname,  os_id, os_detail, labels,  interfaces, etc.  managed: <code>True</code> if the workload is managed, else <code>false</code> .	Array	

Parameter	Description	Type	Required
container_clusters	The array of container cluster URIs, encoded as a JSON string	Object	No
secure_connect	The issuer name match the criteria for the certificate used when establishing secure connections.	Object, Null	
last_heartbeat_at	The last time (rfc3339 timestamp), a heartbeat was received from this VEN.	String, Null	
last_goodbye_at	The time (rfc3339 timestamp) of the last goodbye from the VEN.	String, Null	
status	VEN Status: <ul style="list-style-type: none"><li>• "active"</li><li>• "suspended"</li></ul>	String	
disconnected_before	Return VENs that have been disconnected since the given time.	date/time	
health	The overall health (condition) of the VEN	String	
ip_address	IP address of VEN(s) to return. Supports partial matches	String	
firewall_restore	The strategy to use to restore the firewall state after the VEN is uninstalled.  The strategy to use to restore the firewall state after the VEN is uninstalled:  Options are: <code>saved</code> , <code>default</code> , and <code>disable</code> .  The default is: <code>default</code> .  Works with <code>vens_unpair_put</code> .	String	
ven_id	VEN ID (works with <code>GET /api/v2/orgs/{org_id}/vens/{ven_id}</code> )	String	
vens	VENs to unpair (works with <code>PUT /api/v2/orgs/{org_id}/vens/unpair</code> )  Required property: <code>href</code>	Array	Yes
secure_connect	Property: <code>matching_issuer_name</code> .  Issuer name match criteria for certificate used during establishing secure connections.  <code>matching_issuer_name</code> : Issuer name match criteria for certificate used while establishing secure connections.	Object	
security_policy_applied_at	Last reported time when policy was applied to the workload (UTC),  only present in expanded representations.	date-time	

Parameter	Description	Type	Required
security_policy_received_at	Last reported time when policy was received by the workload (UTC), only present in expanded representations.	date-time	Null
enforcement_mode	Policy enforcement mode, only present in expanded representations.  Options are: "idle", "visibility_only", "full", "selective"	String	
visibility_level	The amount of data the VEN collects and reports to the PCE from a resource demands on workloads.  The higher levels of detail are useful for visualizing traffic flows in the Illumination map inside the PCE web console.  If this parameter is not set, then VEN visibility level is set to <code>flow_summary</code> . <ul style="list-style-type: none"> <li>• <code>flow_summary</code>: ("High Detail" in the PCE web console) The VEN collects traffic connection details (source IP, destination IP, protocol, and source and destination port) for both allowed and blocked connections. This option creates traffic links in the Illumination map and is typically used during your security policy's building and testing phase.</li> <li>• <code>flow_drops</code>: ("Less Detail" in the PCE web console.) The VEN only collects traffic connection details (source IP, destination IP, protocol, and source and destination port) for blocked connections. This option provides less detail for Illumination but demands fewer system resources from a workload and is typically used for policy enforcement.</li> <li>• <code>flow_off</code>: ("No Detail" in the PCE web console.) The VEN does not collect any details about traffic connections.  This option provides no Illumination detail and demands the least resources from workloads. This mode is useful when you are satisfied with the rules that have been created and do not need additional overhead from observing workload communication.</li> </ul>	String	
upgrade_pending	Only return VENs with/without a pending upgrade.	Boolean	No
ven_type	The <code>ven_type</code> property is introduced through the reference to a common schema <code>ven_type.schema.json</code> :	String	No
upgrade_expires_at	The time (rfc3339 timestamp) at which the PCE stops attempting VEN upgrade	String, Null	No
upgrade_target_version	The software release to upgrade to	String, Null	No
upgrade_timeout_seconds	Number of seconds during which the PCE tries to trigger the agent upgrade:  "minimum": 900,  "maximum": 15552000	Integer	

**Curl Command to Get VENs with a Specific Label**

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/vens?  
labels="[[/orgs/2/labels/1642]]" -H "Accept: application/json" -u  
$KEY:$TOKEN
```

To restrict the type of VENs you want to be returned and set a limit on how many results you want to be returned, use the relevant query parameters. For example, you might want to get a collection of no more than 50 VENs running CentOS 6.3 with active status.

### Curl Command to Get VENS Using Other Query Parameters

```
curl -i -X GET https://pce.my-company.com:8443/api/v2/orgs/2/  
vens?os_id=centos-x86_64-6.3&max_results=50&status=active -H "Accept:  
application/json"-u $KEY:$TOKEN
```

### Unpairing and Suspending VENs

Instead of unpairing and suspending workloads, use the new VEN APIs to unpair and suspend VENs.



#### NOTE

The endpoint `workloads/unpair` is DEPRECATED. Use `/vens/unpair` instead.

### Curl Command for Unpairing VENs

```
curl -i -X PUT https://pce.my-company.com/api/v2/orgs/3/vens/unpair -H  
"Content-Type:application/json" -u $KEY:$TOKEN -d '{"vens": [{"href": "/  
orgs/7/vens/xxxxxxxx-9611-44aa-ae06-fXXX8903db65"}, {"href": "/orgs/7/vens/  
xxxxxxxx-9611-xxxx-ae06-f7bXXX03db71"}], "firewall_restore": "default"}'
```

### Curl Command to Mark VEN as Suspended

```
curl -i -X PUT https://pce.my-company.com/api/v2/orgs/3/vens/xxxxxxxx-9611-  
xxxx-ae06-f7bXXX03db71 -H "Content-Type:application/json" -u $KEY:$TOKEN  
-d '{"status": "suspended"}'
```

# Events Administration and REST APIs

## Abstract

Events Administration guide provides information on how to administer your PCE deployment: an overview of events and SIEM integration, event setup, event record formats, and event types by resource.

## About this guide

This guide provides the following information to administer your PCE deployment:

- An overview of events and SIEM integration
- Events setup considerations
- Event record formats, types, and common fields
- Event types by resource
- SIEM integration considerations and recommendations

See also the following related documentation:

- U.S. National Institute for Standards and Technology's [NIST 800-92 Guide to Computer Security Log Management](#)
- U.S. Department of Homeland Security [National Cybersecurity Center](#)

## Before reading this guide

Illumio recommends that you be familiar with the following technology:

- Solid understanding of the Illumio Core
- Familiarity with syslog
- Familiarity with your organization's Security Information and Event Management (SIEM) systems

## Notational conventions in this guide

- Newly introduced terminology is italicized. Example: *activation code* (also known as pairing key)
- Command-line examples are monospace. Example: `illumio-ven-ctl --activate`
- Arguments on command lines are monospace italics. Example: `illumio-ven-ctl --activate activation_code`
- In some examples, the output might be shown across several lines but is actually on one single line.
- Command input or output lines not essential to an example are sometimes omitted, as indicated by three periods in a row. Example:

```
...
some command or command output
...
```

## Events Framework

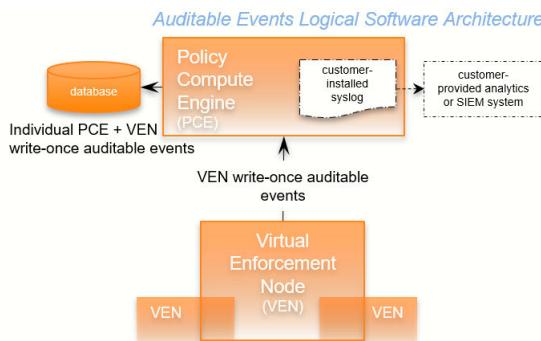
The Illumio events framework provides an information-rich, deep foundation for actionable insights into the operations of the Illumio Core.

### Overview of the Framework

*Auditable* events are records of transactions collected from the following management interfaces:

- PCE web console
- REST API
- PCE command-line tools
- VEN command-line tools

All actions that change the configuration of the PCE, security policy, and VENs are recorded, including workload firewall tampering.



As required by auditing standards, every recorded change includes a reference to the program that made the change, the change's timestamp, and other fields. After recording, the auditable events are read-only.

Auditable events comply with the [Common Criteria Class FAU Security Audit](#) requirements standard for auditing.

### Auditing Needs Satisfied by Framework

Need	Description	See topic...
<b>Audit and Compliance</b>	Evidence to show that resources are managed according to rules and regulatory standards.	<a href="#">Events Record Information [396]</a>
<b>Resource Lifecycle Tracking</b>	All information is necessary to track a resource through creation, modification, and deletion.	<a href="#">Events Lifecycle for Resources [384]</a>
<b>Operations</b>	Trace of recent changes to resources.	<a href="#">Events Lifecycle for Resources [384]</a>
<b>Security</b>	Evidence to show which changes failed, such as incorrect user permissions or failed authentication.	<a href="#">User Password Update Failed (JSON) [401]</a>

## Benefits of Events Framework

The events framework in Core provides the following benefits:

- Exceeds industry standards
  - Delivers complete content
    - Comprehensive set of event types
    - Includes more than 200 events
    - Additional notable system events are generated.
  - Easily accessible interfaces to capture events:
    - Event Viewer in the PCE web console
    - REST API with filtering
    - SIEM integration
    - Events are the same across all interfaces.
  - Designed for customer ease of use
    - Flattened, common structure for all events
    - Eliminates former duplicate or multiple events for single actions
    - Streamed via syslog in JSON, CEF, or LEEF format
    - Create/Update/Delete REST APIs recorded as events.
- Read APIs/GET requests are not recorded because they do not change the Illumio Core.

## Events Lifecycle for Resources

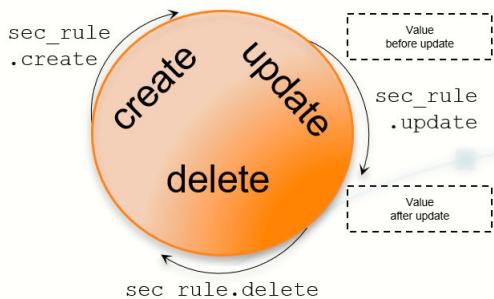
Illumio resources progress through the lifecycle stages (creation, updating, deletion) and the Illumio Core records them with the appropriate event types.

### About the Lifecycle

Many resources have a lifecycle from creation, through update, to deletion. For example, the events related to a security policy rule (identified by the resource name `sec_rule`) are recorded with the following event types.

- `sec_rule.create`
- `sec_rule.update`: Update events record with the values of the resource object both before and after the event for a lifecycle audit trail.
- `sec_rule.delete`

#### *Auditable Events: Lifecycle of a Resource*



## Other Resource Lifecycles

Some resources have unique characteristics and do not follow the create-update-delete pattern. For example, workloads have the following event types:

- workload.update
- workload.upgrade
- workload.redetect\_network
- workload.recalc\_rules
- workload.soft\_delete
- workload.delete
- workload.undelete

## Events Described

This section describes the concepts and types of PCE events.

### List of Event Types

The following table provides the types of JSON events generated and their description. For each of these events, the CEF/LEEF success or failure events generated are the event name followed by .success or .failure.

For example, the CEF/LEEF success event for agent.activate is agent.activate.success and the failure event is agent.activate.failure.

Each event can generate a variety of notification messages. See [Notification Messages in Events \[394\]](#).

JSON Event Type	Description
access_restriction.create	Access restriction created
access_restriction.delete	Access restriction deleted
access_restriction.update	Access restriction updated
agent.activate	Agent paired
agent.activate_clone	Agent clone activated
agent.clone_detected	Agent clone detected
agent.deactivate	Agent unpaired
agent.goodbye	Agent disconnected
agent.machine_identifier	Agent machine identifiers updated
agent.refresh_token	Agent refreshed token
agent.refresh_policy	Success or failure to apply policy on VEN
agent.request_upgrade	VEN upgrade requested
agent.service_not_available	Agent reported a service not running
agent.suspend	Agent suspended
agent.tampering	Agent firewall tampered
agent.unsuspend	Agent unsuspended
agent.update	Agent properties updated.
agent.update_interactive_users	Agent interactive users updated
agent.update_iptables_href	Agent updated existing iptables href
agent.update_running_containers	Agent updated existing containers
agent.upload_existing_ip_table_rules	Agent existing IP tables uploaded
agent.upload_support_report	Agent support report uploaded
agent_support_report_request.create	Agent support report request created
agent_support_report_request.delete	Agent support report request deleted
agents.clear_conditions	Condition cleared from a list of VENs
agents.unpair	Multiple agents unpaired
api_key.create	API key created
api_key.delete	API key deleted

JSON Event Type	Description
api_key.update	API key updated
auth_security_principal.create	RBAC auth security principal created
auth_security_principal.delete	RBAC auth security principal deleted
auth_security_principal.update	RBAC auth security principal updated
authentication_settings.update	Authentication settings updated
cluster.create	PCE cluster created
cluster.delete	PCE cluster deleted
cluster.update	PCE cluster updated
container_workload.update	Container workload updated
container_cluster.create	Container cluster created
container_cluster.delete	Container cluster deleted
container_cluster.update	Container cluster updated
container_cluster.update_services	Container cluster services updated as Kubelink
container_workload_profile.create	Container workload profile created
container_workload_profile.delete	Container workload profile deleted
container_workload_profile.update	Container workload profile updated
database.temp_table_autocleanup_started	DB temp table cleanup started
database.temp_table_autocleanup_completed	DB temp table cleanup completed
domain.create	Domain created
domain.delete	Domain deleted
domain.update	Domain updated
enforcement_boundary.create	Enforcement boundary created
enforcement_boundary.delete	Enforcement boundary deleted
enforcement_boundary.update	Enforcement boundary updated
event_settings.update	Event settings updated
firewall_settings.update	Global policy settings updated
group.create	Group created
group.update	Group updated

JSON Event Type	Description
ip_list.create	IP list created
ip_list.delete	IP list deleted
ip_list.update	IP list updated
ip_lists.delete	IP lists deleted
ip_tables_rule.create	IP tables rules created
ip_tables_rule.delete	IP tables rules deleted
ip_tables_rule.update	IP tables rules updated
job.delete	Job deleted
label.create	Label created
label.delete	Label deleted
label.update	Label updated
label_group.create	Label group created
label_group.delete	Label group deleted
label_group.update	Label group updated
labels.delete	Labels deleted
ldap_config.create	LDAP configuration created
ldap_config.delete	LDAP configuration deleted
ldap_config.update	LDAP configuration updated
ldap_config.verify_connection	LDAP server connection verified
license.delete	License deleted
license.update	License created or updated
login_proxy_ldap_config.create	Interservice call to login service to create LDAP config
login_proxy_ldap_config.delete	Interservice call to login service to delete LDAP config
login_proxy_ldap_config.update	Interservice call to login service to update LDAP config
login_proxy_ldap_config.verify_connection	Interservice call to login service to verify connection to the LDAP server
logout_from_jwt	User logged out
lost_agent.found	Lost agent found
network.create	Network created

JSON Event Type	Description
network.delete	Network delete
network.update	Network updated
network_device.ack_enforcement_instructions_applied	Enforcement instruction applied to a network device
network_device.assign_workload	Existing or new unmanaged workload assigned to a network device
network_device.create	Network device created
network_device.delete	Network device deleted
network_device.update	Network device updated
network_devices.ack_multi_enforcement_instructions_applied	Enforcement instructions applied to multiple network devices
network_endpoint.create	Network endpoint created
network_endpoint.delete	Network endpoint deleted
network_endpoint.update	Network endpoint updated
network_enforcement_node.activate	Network enforcement node activated
network_enforcement_node.clear_conditions	Network enforcement node conditions cleared
network_enforcement_node.deactivate	Network enforcement node deactivated
network_enforcement_node.network_devices_network_endpoints_workloads	Workload added to network endpoint
network_enforcement_node.policy_ack	Network enforcement node acknowledgment of policy
network_enforcement_node.request_policy	Network enforcement node policy requested
network_enforcement_node.update_status	Network enforcement node reports when switches are not reachable
nfc.activate	Network function controller created
nfc.delete	Network function controller deleted
nfc.update_discovered_virtual_servers	Network function controller virtual servers discovered
nfc.update_policy_status	Network function controller policy status
nfc.update_slb_state	Network function controller SLB state updated
org.create	Organization created
org.recalc_rules	Rules for organization recalculated
org.update	Organization information updated

JSON Event Type	Description
pairing_profile.create	Pairing profile created
pairing_profile.create_pairing_key	Pairing profile pairing key created
pairing_profile.delete	Pairing profile deleted
pairing_profile.update	Pairing profile updated
pairing_profile.delete_all_pairing_keys	Pairing keys deleted from pairing profile
pairing_profiles.delete	Pairing profiles deleted
password_policy.create	Password policy created
password_policy.delete	Password policy deleted
password_policy.update	Password policy updated
permission.create	RBAC permission created
permission.delete	RBAC permission deleted
permission.update	RBAC permission updated
request.authentication_failed	API request authentication failed
request.authorization_failed	API request authorization failed
request.internal_server_error	API request failed due to internal server error
request.service_unavailable	API request failed due to unavailable service
request.unknown_server_error	API request failed due to unknown server error
resource.create	Login resource created
resource.delete	Login resource deleted
resource.update	Login resource updated
rule_set.create	Rule set created
rule_set.delete	Rule set deleted
rule_set.update	Rule set updated
rule_sets.delete	Rule sets deleted
saml_acs.update	SAML assertion destination services updated
saml_config.create	SAML configuration created
saml_config.delete	SAML configuration deleted
saml_config.update	SAML configuration updated

JSON Event Type	Description
saml_sp_config.create	SAML Service Provider created
saml_sp_config.delete	SAML Service Provider deleted
saml_sp_config.update	SAML Service Provider updated
sec_policy.create	Security policy created
sec_policy_pending.delete	Pending security policy deleted
sec_policy.restore	Security policy restored
sec_rule.create	Security policy rules created
sec_rule.delete	Security policy rules deleted
sec_rule.update	Security policy rules updated
secure_connect_gateway.create	SecureConnect gateway created
secure_connect_gateway.delete	SecureConnect gateway deleted
secure_connect_gateway.update	SecureConnect gateway updated
security_principal.create	RBAC security principal created
security_principal.delete	RBAC security principal bulk deleted
security_principal.update	RBAC security principal bulk updated
security_principals.bulk_create	RBAC security principals bulk created
service.create	Service created
service.delete	Service deleted
service.update	Service updated
service_binding.create	Service binding created
service_binding.delete	Service binding created
service_bindings.delete	Service bindings deleted
service_bindings.delete	Service binding deleted
services.delete	Services deleted
slb.create	Server load balancer created
slb.delete	Server load balancer deleted
slb.update	Server load balancer updated
support_report_request.create	Support report requested

JSON Event Type	Description
support_report_request.delete	Deleted a request for a support report
support_reports	Support report added
syslog_destination.create	syslog remote destination created
syslog_destination.delete	syslog remote destination deleted
syslog_destination.update	syslog remote destination updated
system_task.agent_missed_heartbeats_check	Agent missed heartbeats
system_task.agent_offline_check	Agents marked offline
system_task.prune_old_log_events	Event pruning completed
traffic_collector_setting.create	Traffic collector setting created
traffic_collector_setting.delete	Traffic collector setting deleted
traffic_collector_setting.update	Traffic collector setting updated
trusted_proxy_ips.update	Trusted proxy IPs created or updated
user.accept_invitation	User invitation accepted
user.authenticate	User authenticated
user.create	User created
user.delete	User deleted
user.invite	User invited
user.login	User logged in
user.login_session_terminated	User login session terminated
user.logout	User logged out
user.pce_session_terminated	User session terminated
user.reset_password	User password reset
user.sign_in	User session created
user.sign_out	User session terminated
user.update	User information updated
user.update_password	User password updated
user.use_expired_password	User entered expired password
user_local_profile.create	User local profile created

JSON Event Type	Description
user_local_profile.delete	User local profile deleted
user_local_profile.reinvite	Invitation email resent for local user
user_local_profile.update_password	User local password updated
ven_settings.update	VEN settings updated
ven_software.upgrade	VEN software release upgraded
ven_software_release.create	VEN software release created
ven_software_release.delete	VEN software release deleted
ven_software_release.deploy	VEN software release deployed
ven_software_release.update	VEN software release updated
ven_software_releases.set_default_version	Default VEN software version set
virtual_server.create	Virtual server created
virtual_server.delete	Virtual server created
virtual_server.update	Virtual server updated
virtual_service.create	Virtual service created
virtual_service.delete	Virtual service deleted
virtual_service.update	Virtual service updated
virtual_services.bulk_create	Virtual services created in bulk
virtual_services.bulk_update	Virtual services updated in bulk
vulnerability.create	Vulnerability record created
vulnerability.delete	Vulnerability record deleted
vulnerability.update	Vulnerability record updated
vulnerability_report.delete	Vulnerability report deleted
vulnerability_report.update	Vulnerability report created or updated
workload.create	Workload created
workload.delete	Workload deleted
workload.online	Workload online
workload.recalc_rules	Workload policy recalculated
workload.redetect_network	Workload network redetected

JSON Event Type	Description
workload.undelete	Workload undeleted
workload.update	Workload settings updated
workload.upgrade	Workload upgraded
workload_interface.create	Workload interface created
workload_interface.delete	Workload interface deleted
workload_interface.update	Workload interface updated
workload_interfaces.update	Workload interfaces updated
	For example, IP address changes, new interface added, and interface shut down.
workload_service_report.update	Workload service report updated
workload_settings.update	Workload settings updated
workloads.apply_policy	Workloads policies applied
workloads.bulk_create	Workloads created in bulk
workloads.bulk_delete	Workloads deleted in bulk
workloads.bulk_update	Workloads updated in bulk
workloads.remove_labels	Workloads labels removed
workloads.set_flow_reporting_frequency	Workload flow reporting frequency changed
workloads.set_labels	Workload labels applied
workloads.unpair	Workloads unpaired
workloads.update	Workloads updated

## Notification Messages in Events

Events can generate a variety of notifications that are appended after the event type:

- agent.clone\_detected
- agent.fw\_state\_table\_threshold\_exceeded
- agent.missed\_heartbeats
- agent.missing\_heartbeats\_after\_upgrade
- agent.policy\_deploy\_failed
- agent.policy\_deploy\_succeeded
- agent.process\_failed
- agent.service\_not\_available
- agent.upgrade\_requested
- agent.upgrade\_successful
- agent.upgrade\_time\_out

- container\_cluster.duplicate\_machine\_id
- container\_cluster.region\_mismatch
- container\_workload.invalid\_pairing\_config
- container\_workload.not\_created
- database.temp\_table\_autocleanup\_completed
- database.temp\_table\_autocleanup\_started
- hard\_limit.exceeded
- pce.application\_started
- pce.application\_stopped
- remote\_syslog.reachable
- remote\_syslog.unreachable
- request.authentication\_failed
- request.authorization\_failed
- request.internal\_server\_error
- request.invalid
- request.service\_unavailable
- request.unknown\_server\_error
- sec\_policy.restore
- soft\_limit.exceeded
- system\_task.event\_pruning\_completed
- system\_task.hard\_limit\_recovery\_completed
- user.csrf\_validation\_failed
- user.login\_failed
- user.login\_failure\_count\_exceeded
- user.login\_session\_created
- user.login\_session\_terminated
- user.pce\_session\_created
- user.pce\_session\_terminated
- user.pw\_change\_failure
- user.pw\_changed
- user.pw\_complexity\_not\_met
- user.pw\_reset\_completed
- user.pw\_reset\_requested
- virtual\_service.not\_created
- workload.duplicate\_interface\_reported
- workload.nat\_rules\_present
- workload.offline\_after\_ven\_goodbye
- workload.online
- workload.oob\_policy\_changes
- workload.partial\_policy\_delivered
- workload.update\_mismatched\_interfaces
- workloads.flow\_reporting\_frequency\_updated

## Event Types, Syntax, and Record Format

When working with events, it is important to recognize their type, REST API schema, syntax, and record information.

### Types of Events

The Illumio Core includes the following general categories of auditable events:

- Organizational events: Organizational events are further grouped by their source:
  - API-related events: Events occurring from a use of the REST API, including the PCE web console
  - System-related events: Events caused by some system-related occurrence
- Traffic events

## Anonymized Database Dumps

To troubleshoot customer-reported issues, Illumio Customer Support sometimes requests that you supply an anonymized dump of the PCE database.

To safeguard your organization's privacy, the event information is not included in the anonymized database dump.

## REST API Events Schema

The Events schema in JSON is downloadable from this documentation portal in the zipfile of the REST API schemas. From the documentation portal Home page, go to the **Develop** category > **REST API Public Schemas (Archive File)**.

## Event Syntax

The names of recorded auditable events have the following general syntax:

```
resource.verb[ .success_or_failure]
```

Where:

- **resource** is a PCE and VEN object, such as PCE `user` or VEN agent component.
- **verb** describes the action of the event on that resource.
- In CEF and LEEF formats, the success or failure of the verb is included in the recorded event type. This indicator is not needed in the JSON format.

## Events Record Information

The following information is included in an event record, which answers the who, what, where, how, and when:

Type of information	Description
Who	<ul style="list-style-type: none"> <li>• VEN identified by hostname and agent href</li> <li>• User identified by username and href</li> <li>• PCE system identified by “system”</li> </ul>
What	<p>The action that triggered the event, including the following data:</p> <ul style="list-style-type: none"> <li>• Resource type + operation + success or failure</li> <li>• Application Request ID</li> <li>• Status of successful events and failed events: <ul style="list-style-type: none"> <li>• In case of failure, exception type and exception message.</li> <li>• All failures related to security, such as authentication and authorization.</li> <li>• Severity as INFO, WARNING, or ERROR.</li> </ul> </li> <li>• The pre-change and post-change values of the affected resources.</li> </ul>
Where	<p>The target resource of the action, composed of the following data:</p> <ul style="list-style-type: none"> <li>• Identifier of the target resource (primary field).</li> <li>• Friendly name for the target resource. For example: <ul style="list-style-type: none"> <li>• workload/VEN: hostname</li> <li>• user.username</li> <li>• ruleset, label, service, etc: name, key/value</li> </ul> </li> </ul>
How	API endpoint, method, HTTP status code, and source IP address of the request.
When	Timestamp of the event's occurrence. This timestamp is <i>not</i> the time the event was recorded.

## Event Record Structure

Regardless of export format (JSON, CEF, or LEEF), the records and fields for all events share a common structure. This common structure of composite events makes post-processing of event data easier.

Bulk change operations on many resources simultaneously are recorded as individual operations on the resource within a single composite event. Failed attempts to change a configuration, such as incorrect authentication, are also collected.

## Common Fields

Field Name	Description
href	Unique event identifier; contains a UUID.
timestamp	The exact time that the event occurred was in RFC 3339 format, which was fractional seconds.
pce_fqdn	The fully qualified domain name of the PCE; especially useful for Supercluster deployments or if multiple PCEs send data to the SIEM server.
created_by	Identifies creator of the event; could be a user, the system, or a workload.
event_type	Name of the event; see the <a href="#">List of Event Types [385]</a> table for more information.
status	“Success” or “failure;” if the status is null, the event is for information only and doesn’t indicate success or failure.
severity	“Informational,” “warning,” or “error” indicating the severity of the event.
version	Schema version for events.

## Events Displayed in PCE Web Console

The PCE web console provides an ongoing log of all Organization events in the PCE. For example, Organization events capture actions such as users logging in and logging out, and failed login attempts; when a system object is created, modified, deleted, or provisioned; when a workload is paired or unpaired; and so on.

From the platform and API perspective, Organization events are referred to internally as `auditable_events` and are generated by the `auditable_events_service`.

You can use the filter at the top of the page to search for events by type of event, event severity level, and when the event occurred.

## Cross-Site Request Forgery Protection

A cross-site request forgery (CSRF) is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent to act as the victim. The underlying cause is an application functionality using predictable URL or form actions in a repeatable way. The nature of the attack is that CSRF exploits a website’s trust for a user.

For more details on this attack, see the [CSRF article](#) on the Web Application Security Consortium website.

Illumio Core can notify you of this type of attack in the following ways:

- The PCE web console logs the attack as an Organization Event called “CSRF token validation failure.”
- The event is logged in the Illumio Core REST API as `authz_csrf_validation_failure` in the `audit_log_events_get.schema`.
- The event `authz_csrf_validation_failure` appears in the PCE syslog output if you have deployed the PCE as a software.

**IMPORTANT**

When you see this event occur, you should immediately investigate the issue because the request might not have originated from a valid user.

## View and Export Events

You can view events by default in the PCE web console or the PCE command line. You can then export Organization events using the PCE web console.

### View Events in PCE Web Console

By default, the PCE web console shows events in your organization, such as when a workload is paired, if a pairing failed, when a user logs in or logs out, when a user fails to authenticate, and so on.

If you want to see only certain events, you can filter by event type to see those that interest you most. You can also search for Organization events by their universally unique identifier (UUID) and filter events by their severity.

You can also export the list of organization events as a CSV file.

To view Organization events:

1. From the PCE web console menu, choose **Troubleshooting > Events**.
2. You can filter events by several criteria, such as severity, status, timestamp, or who generated them.

**NOTE**

The suggested values for the filters are generated from all possible values. For example, the “Generated By” filter shows all users on the system. However, the actual results displayed by that filter might not contain any data.

### VEN Event Not Displayed in PCE Web Console

The following events related to VENs are not currently viewable in the PCE web console. This is a two-column list of event names.

### VEN Events not shown in PCE Web Console

fw_tampering_revert_failure	lost_agent
fw_tampering_reverted	missing_os_updates
fw_tampering_subsystem_failure	pce_incompat_api_version
invoke_powershell_failure	pce_incompat_version
ipsec_conn_state_change	pce_reachable
ipsec_conn_state_failure	pce_unreachable
ipsec_monitoring_failure	proc_config_failure
ipsec_monitoring_started	proc_envsetup_failure
ipsec_monitoring_stopped	proc_init_failure
ipsec_subsystem_failure	proc_malloc_failure
ipsec_subsystem_started	proc_restart_failure
ipsec_subsystem_stopped	proc_started
refresh_token_failure	proc_stopped
refresh_token_success	

## View Events Using PCE Command Line

Run this command at any runlevel to display:

- The total number of events
- The average number of events per day

```
sudo -u ilo-pce illumio-pce-db-management events-db events-db-show
```

Run this command at any runlevel to display:

- The amount of disk space used by events
- The total number of events

```
sudo -u ilo-pce illumio-pce-db-management events-db disk-usage-show
```

## Export Events Using the PCE Web Console

You can export all Organization events or export a filtered list of organization events to a CSV file.

<https://product-docs-repo.illumio.com/Tech-Docs/Animated+GIFs/Export+Events+using+PCE+Console.mp4>

1. From the PCE web console menu, choose **Troubleshooting > Events**.

You see a list of events based on the activities performed.

2. Click **Export > Export All** to export all Organization events.
3. To export a filtered list of events, filter the list and then click **Export > Export Filtered** to export only the filtered view.
4. Use the search filter for events based on event type, severity, status, timestamp, and who generated them.

## Examples of Events

This section presents examples of recorded events in JSON, CEF, and LEEF for various auditing needs.

### User Password Update Failed (JSON)

This example event shows a user password change that failed validation. Event type `user.update_password` shows "status": "failure", and the notification shows that the user's attempted new password did not meet complexity requirements.

```
{
    "href": "/orgs/1/events/xxxxxxxx-39bd-43f1-a680-cc17c6984925",
    "timestamp": "2018-08-29T22:07:00.978Z",
    "pce_fqdn": "pcel.bigco.com",
    "created_by": {
        "system": {}
    },
    "event_type": "user.update_password",
    "status": "failure",
    "severity": "info",
    "action": {
        "uuid": "xxxxxxxx-a5f7-4975-a2a5-b4dbd8b74493",
        "api_endpoint": "/login/users/password/update",
        "api_method": "PUT",
        "http_status_code": 302,
        "src_ip": "10.3.6.116"
    },
    "resource_changes": [],
    "notifications": [
        {
            "uuid": "xxxxxxxx-7b8e-4205-a62a-1f070d8a0ee2",
            "notification_type": "user.pw_complexity_not_met",
            "info": null
        },
        {
            "uuid": "xxxxxxxx-9721-4971-b613-d15aa67a4ee7",
            "notification_type": "user.pw_change_failure",
            "info": {
                "reason": "Password must have minimum of 1 new character(s)"
            }
        }
    ],
    "version": 2
}
```

### Resource Updated (JSON)

This example shows the before and after values of a successful update event `rule_set.update`. The name of the ruleset changed from "before": "rule\_set\_2" to "after": "rule\_set\_3".

```
{ "href": "/orgs/1/events/xxxxxxxx-8033-4f1a-83e9-fde57c425807",
  "timestamp": "2018-08-29T22:04:04.733Z",
  "pce_fqdn": "pcel.bigco.com",
  "created_by": {
    "user": {
      "href": "/users/1",
      "username": "albert.einstein@bigco.com"
    }
  },
  "event_type": "rule_set.update",
  "status": "success",
  "severity": "info",
  "action": {
    "uuid": "xxxxxxxx-7488-480b-9ef9-0cd2a8496004",
    "api_endpoint": "/api/v2/orgs/1/sec_policy/draft/rule_sets/6",
    "api_method": "PUT",
    "http_status_code": 204,
    "src_ip": "10.3.6.116"
  },
  "resource_changes": [
    {
      "uuid": "xxxxxxxx-1d13-4e5e-8f0b-e0e8bcc44e0",
      "resource": {
        "rule_set": {
          "href": "/orgs/1/sec_policy/draft/rule_sets/6",
          "name": "rule_set_3",
          "scopes": [
            {
              "label": {
                "href": "/orgs/1/labels/19",
                "key": "app",
                "value": "app2"
              }
            },
            {
              "label": {
                "href": "/orgs/1/labels/20",
                "key": "env",
                "value": "env2"
              }
            },
            {
              "label": {
                "href": "/orgs/1/labels/21",
                "key": "loc",
                "value": "loc2"
              }
            }
          ]
        }
      }
    },
    {
      "changes": {
        "name": {
          "before": "rule_set_2",
          "after": "rule_set_3"
        }
      },
      "change_type": "update"
    }
  ]
}
```

```
    } ] ,
  "notifications": [ ],
  "version": 2
}
```

## Security Rule Created (JSON)

In this example of a successful `sec_rule` composite event, a new security rule is created. Because this is a creation event, the `before` values are null.

```
{
  "href": "/orgs/1/events/xxxxxxxxx-6d29-4905-ad32-ee863fb63697",
  "timestamp": "2018-08-29T21:48:28.954Z",
  "pce_fqdn": "pce24.bigco.com",
  "created_by": {
    "user": {
      "href": "/users/1",
      "username": "albert.einstein@bigco.com"
    }
  },
  "event_type": "sec_rule.create",
  "status": "success",
  "severity": "info",
  "action": {
    "uuid": "xxxxxxxxx-165b-4e06-aaac-60e4d8b0b9a0",
    "api_endpoint": "/api/v2/orgs/1/sec_policy/draft/rule_sets/1/sec_rules",
    "api_method": "POST",
    "http_status_code": 201,
    "src_ip": "10.6.1.156"
  },
  "resource_changes": [
    {
      "uuid": "9fcf6feb-bf25-4de8-a68a-a50598df4cf6",
      "resource": {
        "sec_rule": {
          "href": "/orgs/1/sec_policy/draft/rule_sets/1/sec_rules/5"
        }
      },
      "changes": {
        "rule_list": {
          "before": null,
          "after": {
            "href": "/orgs/1/sec_policy/draft/rule_sets/1"
          }
        },
        "description": {
          "before": null,
          "after": "WinRM HTTP/HTTPS and RDP"
        },
        "type": {
          "before": null,
          "after": "SecRule"
        },
        "resolve_labels": {
          "before": null,
          "after": "1010"
        },
        "providers": {

```

```

"created": [ {
  "source": true,
  "actors": "ams"
} ]
},
"destinations": {
  "created": [ {
    "source": false,
    "actors": "ams"
  }, {
    "source": false,
    "ip_list": {
      "href": "/orgs/1/sec_policy/draft/ip_lists/1"
    }
  }
],
"ingress_services": {
  "created": [ {
    "href": "/orgs/1/sec_policy/draft/services/7",
    "name": "WinRM HTTP/HTTPS and RDP"
  }]
},
"change_type": "create"
},
"notifications": [ ],
"version": 2
}

```

## User Logged In (JSON)

```

[
{
  "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "timestamp": "2019-06-25T23:34:12.948Z",
  "pce_fqdn": "someFullyQualifiedDomainName",
  "created_by": {
    "user": {
      "href": "/users/1",
      "username": "someUser@someDomain"
    }
  },
  "event_type": "user.sign_in",
  "status": "success",
  "severity": "info",
  "action": {
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "api_endpoint": "/login/users/sign_in",
    "api_method": "POST",
    "http_status_code": 302,
    "src_ip": "xxx.xxx.xx.x"
  },
  "resource_changes": [
    {
      "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "resource": {
        "name": "someResource"
      }
    }
  ]
}
]

```

```

"user": {
    "href": "/users/1",
    "type": "local",
    "username": "someUser@someDomain"
},
"changes": {
    "sign_in_count": {
        "before": 4,
        "after": 5
    }
},
"change_type": "update"
}
],
"notifications": [
{
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "notification_type": "user.login_session_created",
    "info": {
        "user": {
            "href": "/users/1",
            "type": "local",
            "username": "someUser@someDomain"
        }
    }
}
]
},
{
    "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "timestamp": "2019-06-25T23:34:15.147Z",
    "pce_fqdn": "someFullyQualifiedDomainName",
    "created_by": {
        "user": {
            "href": "/users/1",
            "username": "someUser@someDomain"
        }
    },
    "event_type": "user.login",
    "status": "success",
    "severity": "info",
    "action": {
        "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "api_endpoint": "/api/v2/users/login",
        "api_method": "GET",
        "http_status_code": 200,
        "src_ip": "xxx.xxx.xx.x"
    },
    "resource_changes": [
    ],
    "notifications": [
    {
        "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "notification_type": "user.login_session_created",
        "info": {
            "user": {
                "href": "/users/1",
                "type": "local",
                "username": "someUser@someDomain"
            }
        }
    }
]
}
]
}

```

```

    "notification_type": "user.pce_session_created",
    "info": {
        "user": {
            "href": "/users/1",
            "username": "someUser@someDomain"
        }
    }
}
]
}
]

```

## User Logged Out (JSON)

```

[
{
    "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "timestamp": "2019-06-25T23:35:16.636Z",
    "pce_fqdn": "someFullyQualifiedDomainName",
    "created_by": {
        "user": {
            "href": "/users/1",
            "username": "someUser@someDomain"
        }
    },
    "event_type": "user.sign_out",
    "status": "success",
    "severity": "info",
    "action": {
        "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "api_endpoint": "/login/logout",
        "api_method": "GET",
        "http_status_code": 302,
        "src_ip": "xxx.xxx.xx.x"
    },
    "resource_changes": [
        ],
    "notifications": [
        {
            "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "notification_type": "user.login_session_terminated",
            "info": {
                "reason": "user_logout",
                "user": {
                    "href": "/users/1",
                    "username": "someUser@someDomain"
                }
            }
        }
    ],
    "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "timestamp": "2019-06-25T23:35:16.636Z",
    "pce_fqdn": "someFullyQualifiedDomainName",

```

```

"created_by": {
    "user": {
        "href": "/users/1",
        "username": "someUser@someDomain"
    }
},
"event_type": "user.sign_out",
"status": "success",
"severity": "info",
"action": {
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "api_endpoint": "/login/logout",
    "api_method": "GET",
    "http_status_code": 302,
    "src_ip": "xxx.xxx.xx.x"
},
"resource_changes": [
],
"notifications": [
{
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "notification_type": "user.login_session_terminated",
    "info": {
        "reason": "user_logout",
        "user": {
            "href": "/users/1",
            "username": "someUser@someDomain"
        }
    }
}
]
}
]
]

```

## Login Failed – Incorrect Username (JSON)

```

{
    "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "timestamp": "2019-06-25T23:35:41.560Z",
    "pce_fqdn": "someFullyQualifiedDomainName",
    "created_by": {
        "system": {}
    },
    "event_type": "user.sign_in",
    "status": "failure",
    "severity": "info",
    "action": {
        "uuid": "someFullyQualifiedDomainName",
        "api_endpoint": "/login/users/sign_in",
        "api_method": "POST",
        "http_status_code": 200,
        "src_ip": "xxx.xxx.xx.x"
    },
    "resource_changes": [
]
}
]
```

```
],
"notifications": [
  {
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "notification_type": "user.login_failed",
    "info": {
      "associated_user": {
        "supplied_username": "invalid_username@someDomain"
      }
    }
  }
]
```

## Login Failed – Incorrect Password (JSON)

```
{
  "href": "/orgs/1/events/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "timestamp": "2019-06-25T23:35:27.649Z",
  "pce_fqdn": "someFullyQualifiedDomainName",
  "created_by": {
    "system": {}
  },
  "event_type": "user.sign_in",
  "status": "failure",
  "severity": "info",
  "action": {
    "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "api_endpoint": "/login/users/sign_in",
    "api_method": "POST",
    "http_status_code": 200,
    "src_ip": "xxx.xxx.xx.x"
  },
  "resource_changes": [
    ],
  "notifications": [
    {
      "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
      "notification_type": "user.login_failed",
      "info": {
        "associated_user": {
          "supplied_username": "someUser@someDomain"
        }
      }
    }
  ]
}
```

## User Log Out (CEF)

This example of an event record in CEF shows a successful user log out.

```
CEF:0|Illumio|PCE|19.3.0|user.logout.success|User Logout Success|1|rt=Mar
06 2020
```

```

18:38:59.900 +0000 dvchost=mypce.com duser=system dst=10.6.5.4
outcome=success
cat=audit_events request=/api/v2/users/logout_from_jwt requestMethod=POST
reason=204
cs2= cs2Label=resource_changes
cs4=[ { "uuid" :"b5ba8bf0-7ca8-47fc-870f-6c61ddc1648d",
"notification_type" :"user.pce_session_terminated", "info" :
{ "reason" :"user_logout",
"user" :{ "href" :"#/users/1", "username" :"testuser@mypce.com" } } }]
cs4Label=notifications
cn2=2 cn2Label=schema-version cs1Label=event_href cs1=/system_events/
e97bd255-4316-4b5e-a885-5b937f756f17

```

## Workload Security Policy Updated (LEEF)

This example of an event record in LEEF shows a successful update of security policy for a workload's Ethernet interfaces.

```

LEEF:2.0|Illumio|PCE|18.2.0|interface_status.update.success|
src=xx.xxx.xxx.xxx
cat=organizational devTime=someUTCdatetime devTimeFormat=yyyy-mm-
dd'T'HH:mm:ss.ttttttZ
sev=1
usrName=albert.einstein url=/orgs/7/agents/someUUID version=2
pce_fqdn=someFQDN
created_by={ "agent" :{ "href" :"#/orgs/7/agents/
someUUID", "hostname" :"someHostname" } }
action={ "uuid" :"someUUID",
"api_endpoint" :"#/api/v6/orgs/7/agents/xxxxxx/interface_statuses/update",
"api_method" :"PUT", "http_status_code":200, "src_ip" :"someIP" }
resource_changes=[ { "uuid" :"someUUID",
"resource" :{ "workload" :{ "href" :"#/orgs/7/workloads/someUUID", "name" :null,
"hostname" :"someHostname",
"labels" :[ { "href" :"#/orgs/7/labels/
xxxxxx", "key" :"loc", "value" :"test_place_1" },
{ "href" :"#/orgs/7/labels/xxxxxx", "key" :"env", "value" :"test_env_1" },
{ "href" :"#/orgs/7/labels/xxxxxx", "key" :"app", "value" :"test_app_1" },
{ "href" :"#/orgs/7/labels/xxxxxx", "key" :"role", "value" :"test_access_1" } ] },
"changes" :{ "workload_interfaces" :
{ "updated" :[ { "resource" :
{ "href" :"#/orgs/7/workloads/someUUID/interfaces/eth1", "name" :"eth0",
"address" :{ "family" :2, "addr" :xxxxxxxxx, "mask_addr" :someMask },
"changes" :{ "address" :{ "before" :null, "after" :
{ "family" :2, "addr" :xxxxxxxxx, "mask_addr" :someMask } },
"cidr_block" :{ "before" :null, "after" :16 }, "default_gateway_address" :
{ "before" :null, "after" :
{ "family" :2, "addr" :someGateway, "mask_addr" :someMask } },
"link_state" :{ "before" :"unknown", "after" :"up" },
"network" :{ "before" :null, "after" :{ "href" :"#/orgs/7/networks/xx" } },
"network_detection_mode" :{ "before" :null, "after" :"single_private_brn" } } },
"resource" :{ "href" :"#/orgs/7/workloads/someUUID/interfaces/eth1",
"name" :"eth1", "address" :
{ "family" :2, "addr" :someAddress, "mask_addr" :someMask } },
"changes" :{ "address" :{ "before" :null, "after" :{ "family" :2, "addr" :someAddress,
"mask_addr" :someMask } },
"cidr_block" :{ "before" :null, "after" :16 }, "link_state" :

```

```
{
  "before": "unknown", "after": "up" },
  "network": { "before": null, "after": { "href": "/orgs/7/networks/xx" } },
  "network_detection_mode": { "before": null, "after": "single_private_brn" } } ] },
  "change_type": "update" } ] notifications=[ ] event_href=/orgs/7/events/someUUID
```

## Differences from Previous Releases

The following table indicates which event names changed in the Illumio Core 18.2 release. If you are upgrading from a release prior to 18.2, be sure to use the current event name in your alert monitoring system.

### Changed VEN Event Names

This table lists the names of VEN-related events prior to the Illumio Core 18.2 release and the names they were changed to in the 18.2 release.

Old Name Prior to 18.2	New Name as of 18.2
fw_config_change	agent.firewall_config
activation_success	agent.activate
activation_failure	
deactivation_success	agent.deactivate
deactivation_failure	

## Events Monitoring Best Practices

The Illumio Core generates a rich stream of structured messages that provide the following information:

- Illumio PCE system health
- Illumio PCE notable activity
- Illumio VEN notable activity

Illumio Core events are structured and actionable. Using the event data, you can identify the severity, affected systems, and what triggered the event. Illumio Core sends the structured messages using the syslog protocol to remote systems, such as Splunk and QRadar. You can set up your remote systems to automatically process the messages and alert you.

## Monitoring Operational Practices

In addition to setting up an automated system, Illumio recommends implementing the following operational practices:

1. Determine the normal quantity of events from the Illumio Core and monitor the trend for changes; investigate spikes or reductions in the event generation rate.
2. Implement good operational practices to troubleshoot and investigate alerts and to recover from events.

3. Do not monitor Illumio Core events in isolation. Monitor them as part of your overall system. Understanding the events in the context of your overall system activity can provide as much information as the events themselves.

## **Recommended Events to Monitor**

As a best practice, Illumio recommends you monitor the following events at a minimum.

Events	Description
Program name = <code>illumio_pce/system_health</code>	<p>Provides multiple systems metrics, such as CPU and memory data, for each node in a PCE cluster. The PCE generates these events every minute. The Severity field is particularly important. When system metrics exceed thresholds, the severity changes to warning, error, or fatal.</p>
Severity = Warning, Error, or Fatal	<p>For more information about the metrics and thresholds, see the PCE Administration Guide.</p>
<b>Recommendation:</b> Monitor <code>system_health</code> messages with a severity of warning or higher and correlate the event with other operational monitoring tools to determine if administrative intervention is required.	
<code>event_type="lost_agent.found"</code>	<p>Contains the information necessary to identify workloads with lost agents. A lost agent occurs when the PCE deletes a workload from its database, but that workload still has a VEN running on it.</p>
<b>Recommendation:</b> Monitor <code>lost_agent.found</code> events and send alerts in case you need to pair the workloads' VENs with the PCE again.	
<code>event_type="system_task.agent_missed_heartbeats_check"</code>	<p>Lists the VENs that missed three heartbeats (default: total of 15 minutes). Typically, this event precedes the PCE taking the VENs offline to perform internal maintenance.</p>
<b>Recommendation:</b> For Server VENs, this event triggers an alert to be sent at 25% of the time configured in the offline timer. For example, if the offline timer is configured to 1 hour, an alert is sent after the Server VEN has not sent a heartbeat for 15 minutes; if the offline timer is configured to 4 hours, an alert is sent after the Server VEN hasn't sent a heartbeat for 1 hour. Alerts are disabled by default for Endpoint VENs.	
<b>Recommendation:</b> Monitor these events for high-value workloads. The PCE can take these workloads offline when the VENs miss 12 heartbeats (usually 60 minutes).	
<code>event_type="system_task.agent_offline_check"</code>	<p>This event lists VENs that the PCE has marked offline, usually because they missed 12 heartbeats. The VENs on these workloads haven't communicated with the PCE for an hour and it removed the workloads from policy.</p>
<b>Recommendation:</b> Monitor these events for high-value workloads because they indicate a change in the affected workloads' security posture.	
<code>event_type="agent.suspend"</code>	<p>This event indicates that the VEN is suspended and no longer protecting the workload. If you did not intentionally run the VEN suspend command on the workload, this event can indicate the workload is under attack.</p>
<b>Recommendation:</b> Monitor these events for high-value workloads.	
<code>event_type="agent.tampering"</code>	<p>This event indicates tampering with the workload's Illumio-managed firewall and that the VEN recovered the firewall. Firewall tampering is one of the first signs that a workload is compromised. During a tampering attempt, the VEN and PCE continue to protect the workload; however, you should investigate the event's cause.</p>
<b>Recommendation:</b> Monitor these events for high-value workloads.	
<code>event_type="agent.update"</code>	<p>Contains the state data that the VEN regularly sends to the PCE. Typically, these events contain routine information; however, the VEN can attach a notice indicating the following issues:</p>
	<ul style="list-style-type: none"> <li>• Processes not running</li> <li>• Policy deployment failure</li> </ul>

Events	Description
event_type="rule_set.create"	<b>Recommendation:</b> Monitor agent.update events that include notifications because they indicate workloads that might require administrative intervention.
event_type="rule_set.update"	Contains the labels indicating the scope of a draft ruleset. Illumio Core generates these events when you create, update, or delete a draft ruleset. When you include "All Applications," "All Environments," or "All Locations" in a ruleset scope, the PCE represents that label type as a null HREF. Ruleset scopes that are overly broad affect a large number of workloads. Draft rulesets do not take effect until they are provisioned.
event_type="rule_sets.delete"	<b>Recommendation:</b> Monitor these events to pinpoint ruleset scopes that are unintentionally overly broad.
event_type="sec_rule.create"	These events contain labels indicating when all workloads affected, all services, or a label/group are used as a rule source or destination. Illumio Core generates these events when you create, update, or delete a draft ruleset. The removed or added labels could represent high-value applications or environments.
event_type="sec_rule.update"	<b>Recommendation:</b> Monitor these events for high-value labels.
event_type="sec_rule.delete"	<b>[NEW in Illumio Core 19.3.0]</b> It contains the <code>workloads_affected</code> field, which includes the number of workloads affected by a policy. Illumio Core generates this event when you provision a draft policy that updates the policy on affected workloads. The number of affected workloads could be high or a significant percentage of your managed workloads.
event_type="sec_policy.create"	<b>Recommendation:</b> Monitor the <code>workloads_affected</code> field for a high number of affected workloads. If the number exceeds an acceptable threshold, investigate the associated policy.
event_type="agent.clone_detected"	The PCE detects cloned VENs based on clone token mismatch. This is a special alert from release 19.3.2 onwards, as clones have become a higher priority. The volume of these events makes the severity level important, not the fact that these events occurred.
	<b>Recommendation:</b> If severity is 1 or 'error', some intervention may be needed.
	 <b>NOTE</b> <b>Automatic Cloned VEN Remediation</b> <p>For on-prem domain joined Windows workloads, cloned VENs support automatic clone remediation by detecting changes to the workload's domain Security identifier (SID). After the VEN reports such changes to the PCE, the PCE tells the clone to re-activate itself, after which the cloned VEN is remediated and becomes a distinct agent from the original VEN.</p>

## Events Setup

This section describes PCE settings related to events and how to use them to configure PCE behavior.

## Requirements for Events Framework

To use the events framework, allocate enough disk space for event data and be familiar with the disk capacity requirements.

## Database Sizing for Events

Disk space for a single event is estimated at an average of 1,500 bytes.



### CAUTION

As the number of events increases, the increase in disk space is not a straight line. The projections below are rough estimates. Disk usage can vary in production and depends on the type of messages stored.

Number of Events	Disk Space
25 million	38GB
50 million	58GB

## Data and Disk Capacity for Events

For information about the default events data retention period, database dumps with and without events data, disk compacting, and more, see [Manage Data and Disk Capacity](#) in the PCE Administration Guide.

## Events Preview Runtime Setting

If you participated in the preview of Events in 18.1.0, you enabled it by configuring a setting in your PCE `runtime_env.yml` file.



### WARNING

**Remove preview parameter from `runtime_env.yml`**

Before you upgrade to the latest release, you must remove `v2_auditable_events_recording_enabled:true` from `runtime_env.yml`. Otherwise, the upgrade will not succeed.

Removing this preview parameter does not affect the ongoing recording of “organization events” records.

To remove the Events preview setting:

1. Edit the `runtime_env.yml` file and remove the line `v2_auditabile_events_recording_enabled`:

```
v2_auditabile_events_recording_enabled: true
```

If you are not participating in other previews, you can also remove the line `enable_preview_features`.

2. Save your changes.

## Events Settings

The following section describes configuring the Events Settings in the PCE web console.



### NOTE

Information about Event Settings applies only to the on-premises PCE.

## Events Are Always Enabled

Events are enabled by default in the PCE and cannot be disabled by [Common Criteria compliance](#).

Use the PCE web console to change event-related settings and the PCE `runtime_env.yml` for traffic flow summaries.

## Event Settings in PCE Web Console

From the PCE web console, you can change the following event-related settings:

- **Event Severity:** Sets the severity level of events to record. Only messages at the set severity level and higher are recorded. The default severity is “Informational.”
- **Retention Period:** The system retains event records for a specified number of days, ranging from 1 day to 200 days, with a default period of 30 days.
- **Event Pruning:** The system automatically prunes events based on disk usage and their age; events older than the retention period are pruned. When pruning is complete, the `system_task.prune_old_log_events` event is recorded.
- **Event Format:** This setting sets the message output to one of three formats. The selected message output format only applies to messages sent over Syslog to an SIEM. The REST API always returns events in JSON.
  - JavaScript Object Notation (JSON): The default; accepted by Splunk and QRadar SIEMs
  - Common Event Format (CEF): Accepted by ArcSight
  - Log Event Extended Format (LEEF): Accepted by QRadar

## Event Severity Levels

Severity	Description
Emergency	System is unusable
Alert	This should be corrected immediately.
Critical	Critical conditions
Error	Error conditions
Warning	Might indicate that an error will occur if action is not taken
Notice	Unusual events, but not error conditions
Informational	Normal operational messages that require no action
Debug	Information useful to developers for debugging the application

## Output Format Change

The output format can be changed in the PCE web console:

- JSON (default)
- CEF
- LEEF

Records are in JSON format until you change to one of the other formats. Then, the new events are recorded in the new format; however, the earlier events are not changed to the selected format and remain recorded in JSON.

## Set Event Retention Values

You can set the event retention values depending on the specific conditions described below.

If you use an SIEM, such as Splunk, as the primary long-term storage for events and traffic in a dynamic environment, consider setting the event retention period to 7 days. When setting it to 7 days, you can use the PCE Troubleshooting or Events Viewer to troubleshoot and diagnose events quickly. The benefit of setting it to 7 days is that if an issue occurs on a Friday, it can still be diagnosed the following Monday. Many events are generated in a dynamic environment, increasing the data stored (disk space used), backup size, etc. The period of 7 days provides a good balance between disk usage and the ability to troubleshoot.



### NOTE

A dynamic environment is when applications and infrastructure are subject to frequent changes, such as the usage of APIs, ETL, Containers, and so on.

If you use a SIEM in a non-dynamic environment, consider setting the event retention period to 30 days. In a non-dynamic environment, fewer events are generated, and less disk space is used.

If you are not using a SIEM such as Splunk and the PCE is the primary storage for the events data used for reporting, diagnosis, and troubleshooting, set the event retention period per the organization's record retention policy, such as 30 days. If you generate quarterly reporting using events, set the event retention period to 90 days.

SIEM	Consideration	Value
Yes: Primary storage for events	If the primary storage of events is not on the PCE	7 days (PCE troubleshooting) 1 day (minimum)
No: Not primary storage for events	If events are stored primarily on the PCE, consider the organization's record retention policy and the available disk and event growth pattern.	30 days (default)
No	<ul style="list-style-type: none"> <li>• If the organization's record retention is more than 30 days</li> <li>• If disk monitoring is not set up, it is required to set up disk monitoring.</li> </ul>	As per your record retention policy  200 days (maximum)
Not applicable	If events data is not needed for reporting or troubleshooting	1 day (minimum)

If disk space availability and event growth projections indicate that the desired retention period cannot be safely supported, consider using a SIEM because the PCE might not store events for the desired period.



### NOTE

Running the `illumio-pce-db-management events-db` command outputs the average number of events and the storage used.

## Configure Events Settings in PCE Web Console

- From the PCE web console menu, choose **Settings > Event Settings** to view your current settings.
- Click **Edit** to change the settings.
  - For Event Severity, select from the following options:
    - Error
    - Warning
    - Informational
  - For the Retention Period, enter the number of days you want to retain data.
  - For Event Format, select from the following options:
    - JSON
    - CEF
    - LIEEEF
- Click **Save** once you're done.

### Limits on Storage

The PCE will automatically limit the maximum number of events stored. The limits are set on the volume of events stored locally in the PCE database so that the events recorded in the database do not fill the disk. The limit is a percentage of the disk capacity, cumulative for all services that store events on the disk.

**IMPORTANT**

To change the default limits, contact Illumio Support.

The configuration limit includes both hard and soft limits.

- Soft limit: 20% of disk used by event storage

When the soft limit is reached, aggressive pruning is triggered. However, new events are still recorded while pruning.

On the Events list page of the PCE Web Console, the `system_task.prune_old_log_events` event is displayed with the "Object creation soft limit exceeded" message and 'Severity: Informational.'

- Hard limit: 25% of disk used by event storage.

More aggressive pruning is triggered when the hard limit is reached. New events are not recorded while pruning.

On the Events list page of the PCE Web Console, the `system_task.prune_old_log_events` event is displayed with the message "Object creation hard limit exceeded" and 'Severity: Error'. The pruning continues until the soft limit level of 20% is reached. When this occurs, a `system_task.hard_limit_recovery_completed` event occurs, and the PCE starts to behave as it did for the soft limit conditions.

## SIEM Integration for Events

Event data can be sent using syslog to your own analytics or SIEM systems for analysis or other needs.

### About SIEM Integration

This guide also explains how to configure the PCE to securely transfer PCE event data in the following message formats to some associated SIEM systems:

- JavaScript Object Notation (JSON) is needed for SIEM applications like Splunk®.
- Common Event Format (CEF) is needed for Micro Focus ArcSight®.
- Log Event Extended Format (LEEF) is needed for IBM QRadar®.

### Illumio Tools for SIEM Integration

Illumio offers other tools for SIEM integration.

Illumio App for Splunk:

- Software: [Technical Add-on for Illumio](#) and [Illumio App for Splunk](#)
- Documentation: [Illumio App for Splunk Guide 4.x](#)

Illumio App for QRadar:

- Software: [Illumio App for QRadar](#)
- Documentation: [Illumio App for QRadar Guide 1.4.0](#)

Illumio App for ServiceNow:

- Software: Illumio App for CMDB
- Documentation: [Illumio App for ServiceNow 2.1.0](#)

## Syslog Forwarding

The PCE can export logs to syslog. You can also use the PCE's own internal syslog configuration.

### Identify Events in Syslog Stream

Event records from the syslog stream are identified by the following string:

```
"version":2
```

AND

```
' "href":\s*/orgs/[0-9]*/events' OR ' "href":\s*/system_events/'
```

### Forward Events to External Syslog Server

The PCE has an internal syslog repository, “Local,” where all the events are stored. You can control and configure the relaying of Syslog messages from the PCE to multiple external Syslog servers.

To configure forwarding to an external Syslog server:

1. From the PCE web console menu, choose **Settings > Event Settings**.
2. Click **Add**.  
The Event Settings - Add Event Forwarding page opens.
3. Click **Add Repository**.

Add Repository	
* Description	Test123
* Address	1
* Protocol	TCP
* Port	
* TLS	Enabled
* Trusted CA Bundle	Choose File no file selected
* Verify TLS	<input checked="" type="checkbox"/> Ensure that TLS peer's server certificate is valid
<input type="button" value="Cancel"/> <input type="button" value="OK"/>	

**4.** In the Add Repository dialog:

- Description: Enter the name of the Syslog server.
- Address: Enter the IP address for the Syslog server.
- Protocol: Select TCP or UDP. If you select UDP, you only need to enter the port number and click **OK** to save the configuration.
- Port: Enter the port number for the syslog server.
- TLS: Select Disabled or Enabled. If you select Enabled, click “Choose File” and upload your organization’s “Trusted CA Bundle” file from the location where it is stored.

The Trusted CA Bundle contains all the certificates the PCE (internal syslog service) needs to trust the external syslog server. If you are using a self-signed certificate, that certificate is uploaded. If you are using an internal CA, the certificate of the internal CA must be uploaded as the “Trusted CA Bundle”.

- Verify TLS: Select the check-box to ensure the TLS peer’s server certificate is valid.

**5.** Click **OK**.

After ensuring that the events are being forwarded as configured to the correct external Syslog servers, you can stop using the “Local” server by editing the local server setting and deselecting all message types.



**NOTE**

You cannot delete the “Local” server.

## Disable Health Check Forwarding

PCE system health messages are helpful for PCE operations and monitoring. If they are needed at the remote destination, you can choose to forward them.

For example, IBM QRadar is usually used by security personnel who might not need to monitor the health of the PCE system. The Illumio App for QRadar does not process the PCE system health messages.

The PCE system health messages are only in key/value syslog format. They are not translatable into CEF, LEEF, or JSON formats. If your SIEM does not support processing key/value messages in Syslog format, do not forward system health messages to those SIEMs. For example, IBM QRadar and Micro Focus ArcSight do not automatically parse these system health messages.

<https://product-docs-repo.illumio.com/Tech-Docs/Animated+GIFs/Disable+Health+Check+Forwarding.mp4>

1. From the PCE web console menu, choose **Settings > Event Settings**.
2. Click the Event listed under the **Events** column.
3. Under the Events block, for the Status Logs entry, deselect **System Health Messages**. System health check is only available in key-value format. Selecting a new event format does not change the system health check format to CEF or LEEF.
4. Click **Save**.

**NOTE**

IBM QRadar and HP ArcSight do not support health messages in the system. If you are using either of these for SIEM, do not select the System Health Messages checkbox.

## Showing Rule ID in Syslog

For large customers handling 10K+ messages per second, including rule IDs in the Syslog events will substantially increase the volume of recorded data.

In release 25.1.0, an organization-level feature flag `rule_info_exposure_to_syslog` (disabled by default) was added. This flag controls whether rule ID information is included in the syslog messages:

```
rule_info_exposure_to_syslog
```

To add the rule IDs to the syslog events, the API `optional_features_put` was changed by adding the new property `rule_info_exposure_to_syslog`.

To provision the firewall settings via the PCE console, follow these steps:

1. In the PCE console, go to **Settings > Event Settings** .
2. In the Event Settings dialog, click Add next to Event Forwarding.
3. Select **Local**.
4. Select check boxes for all events: Organizational Events, System Events, Allowed, Potentially Blocked, Blocked, and System Health Messages.
5. Click **Save**.

## Enabling the Rule Data via API

To set the flag `enable_all_rule_hit_count_enabled` via API, use the following CURL command:

```
curl -u api_${ILO_API_KEY_ID}:${ILO_API_KEY_SECRET} -H "Content-Type: application/json" -X PUT -d '{"rule_hit_count_enabled_scopes": []}' https://$ILO_SERVER/api/v2/orgs/${ILO_ORG_ID}/sec_policy/draft/firewall_settings
```

For more details about using the Rule ID feature using the API, see [Showing Rule ID in Syslog](#).

## Traffic Flow Summaries

This section describes traffic flow summaries.

After you install a VEN on a workload and pair the VEN with the PCE, the VEN monitors each workload's traffic flows and sends the traffic flow summaries to the PCE.

Traffic summaries can be exported to syslog or Fluentd. If traffic data is configured for export, the PCE processes the received traffic flow summaries from each VEN and immediately sends them to syslog or Fluentd.

## Traffic Flow Types and Properties

The Illumio Core logs traffic flows based on the Visibility setting. Events have attributes that can be Allowed, Blocked, or Potentially Blocked and might not appear in the traffic flow summary.

### Visibility Settings

The table below indicates whether or not a traffic summary is logged as Allowed, Blocked, or Potentially Blocked according to a workload's visibility setting.



#### NOTE

Traffic from workloads in the “Idle” policy state is not exported to syslog from the PCE.

Visibility	Logged-in Traffic Flow Summary
<b>Off</b>	VEN does not log traffic connection information
<b>Blocked</b> - Low Detail	VEN logs connection information for blocked and potentially blocked traffic only
<b>Blocked + Allowed</b> - High Detail	VEN logs connection information for allowed, blocked, and potentially blocked traffic
<b>Enhanced Data Collection</b>	VEN logs byte counts in addition to connection details for allowed, blocked, and potentially blocked traffic

## Event Types

In a traffic flow summary, the event type is designated by Policy Decision (pd).



#### NOTE

An asterisk ( \* ) indicates that the attribute might not appear in the summary.

Event Attributes	Allowed (pd=0)	Potentially Blocked (pd=1)	Blocked (pd=2)
version	✓	✓	✓
count	✓	✓	✓
interval_sec	✓	✓	✓
timestamp	✓	✓	✓
dir	✓	✓	✓
src_ip	✓	✓	✓
dst_ip	✓	✓	✓
proto	✓	✓	✓
dst_prt	✓	✓	✓
state	✓	✓	✓
pd	✓	✓	✓
code*	✓	✓	✓
type*	✓	✓	✓
dst_vulns*	✓	✓	✓
fqdn*	✓	✓	✓
un*	✓	✓	✗
pn*	✓	✓	✗
sn*	✓	✓	✗
src_labels*	✓	✓	✓
dst_labels*	✓	✓	✓
src_hostname*	✓	✓	✓
dst_hostname*	✓	✓	✓
src_href*	✓	✓	✓
dst_href*	✓	✓	✓

## Showing the Data Transfer Amount

The JSON, CEF, and LEEF for the accurate byte count work events are related to the 'Show Amount of Data Transfer' preview feature, which is available with the 20.2.0 release.

The PCE now reports the amount of data transferred into and out of workloads and applications in a data center. The number of bytes sent by and received by an application's source is provided separately. These values can be seen in traffic flow summaries streamed from the

PCE. You can enable this capability on a per-workload basis in the Workload page. You can also enable it in the pairing profile so that workloads are directly paired into this mode.

The direction reported in the flow summary is from the viewpoint of the source of flow:

### **Destination Total Bytes Out (**

`dst_tbo`

): Number of bytes transferred out of source.

### **Destination Total Bytes In (**

`dst_tbi`

): Number of bytes transferred to source.

To activate the 'Show Amount of Data Transfer' capability on the PCE, contact your Illumio representative.

### **LEEF Mapping**

- LEEF field `x` contains JSON field `y`
- `srcBytes` contains `dst_tbo`
- `dstBytes` contains `dst_tbi`
- `dbi` contains `dst_db`
- `dbo` contains `dst_dbo`

### **CEF Mapping**

- CEF field `cn2` is `dst_db` with `cn2Label` is “`dbi`”
- CEF field `cn3` is `dst_dbo` with `cn3Label` is “`dbo`”
- CEF field “`in`” is `dst_tbi`
- CEF field “`out`” is `dst_tbo`

## **Traffic Flow Summary Examples**

The following topic provides examples of traffic flow summaries in JSON, CEF, and LEEF, and messages that appear in syslog.

### **JSON**

```
{
  "interval_sec": 600,
```

```

"count": 1,
"tbi": 73,
"tbo": 0,
"pn": "example-daemon",
"un": "example",
"src_ip": "xxx.xxx.xx.xxx",
"dst_ip": "xxx.x.x.xxx",
"timestamp": "2018-05-23T16:07:12-07:00",
"dir": "I",
"proto": 17,
"dst_port": 5353,
"state": "T",
"src_labels": {
    "app": "AppLabel",
    "env": "Development",
    "loc": "Cloud",
    "role": "Web"
},
"src_hostname": "test-ubuntu-3",
"src_href": "/orgs/1/workloads/xxxxxxxx-7741-4f71-899b-d6f495326b3f",
"dst_labels": {
    "app": "AppLabel",
    "env": "Development",
    "loc": "AppLocation",
    "role": "Database"
},
"dst_hostname": "test-ubuntu-2",
"dst_href": "/orgs/1/workloads/xxxxxxxx-012d-4651-b181-c6f2b269889e",
"pd": 1,
"dst_vulns": {
    "count": 8,
    "max_score": 8.5,
    "cve_ids": [
        "CVE-2016-2181",
        "CVE-2017-2241"
    ]
},
"fqdn": "xxx.ubuntu.com",
"version": 4
}

```

## Syslog

```

2019-02-11T22:50:15.587390+00:00 level=info host=detest01 ip=100.1.0.1
program=illumio_pce/collector| sec=925415.586 sev=INFO pid=9944
tid=30003240
rid=bb8ff798-1ef2-44b1-b74e-f13b89995520 {"interval_sec":1074,
"count":1,"tbi":3608,
"tbo":0,"pn":"company-daemon","un":"company","src_ip":"10.0.2.15",
"dst_ip":"211.0.0.232",
"class":"M","timestamp":"2019-02-11T14:48:09-08:00","dir":"I",
"proto":17,
"dst_port":5353,"state":"T","src_labels":{"app":"AppName",
"env":"Development","loc":"Cloud","role":"Web"},

"src_hostname": "dev-ubuntu-1",
"src_href": "/orgs/1/workloads/773f3e81-5779-4753-b879-35a1abe45838",

```

```
"dst_labels": { "app": "AppName", "env": "Development", "loc": "Cloud2",
"role": "Web" },
"dst_hostname": "dev-ubuntu-1", "dst_href": "/orgs/1/workloads/
773f3e81-5779-4753-b879-35a1abe45838", "pd": 0, "dst_vulns": { "count": 1,
"max_score": 3.7,
"cve_ids": [ "CVE-2013-2566", "CVE-2015-2808" ] }, "fqdn": "xxx.ubuntu.com",
"version": 4}
```

### **Allowed Flow Summary (pd = 0)**

```
2016-01-12T05:23:30+00:00 level=info host=myhost ip=127.0.0.1
program=illumio_pce/
collector| sec=576210.952 sev=INFO pid=25386 tid=16135120 rid=0
{ "interval_sec": 1244, "count": 3, "dbi": 180, "dbo": 180, "pn": "sshd", "un": "root",
"src_ip": "10.6.0.129", "dst_ip": "10.6.0.129", "timestamp": "2017-08-16T13:23:57
-07:00",
"dir": "I", "proto": 6, "dst_port": 22, "state": "A", "dst_labels":
{ "app": "test_app_1", "env":
"test_env_1", "loc": "test_place_1", "role": "test_access_1" }, "dst_hostname": "co
rp-vm-2",
"dst_href": "/orgs/1/workloads/5ddcc33b-b6a4-4a15-b600-64f433e4ab33", "pd": 0,
"version": 4}
```

### **Potentially Blocked Flow Summary (pd = 1)**

```
2016-01-12T05:29:21+00:00 level=info host=myhost ip=127.0.0.1
program=illumio_pce/
collector| sec=576561.327 sev=INFO pid=25386 tid=16135120 rid=0
sec=920149.541
sev=INFO pid=1372 tid=30276700 rid=136019d0-f9d8-45f3-ac99-f43dd8015675
{ "interval_sec": 600, "count": 1, "tbi": 229, "tbo": 0, "src_ip": "172.16.40.5",
"dst_ip": "172.16.40.255", "timestamp": "2017-08-16T14:45:58-07:00", "dir": "I",
"proto": 17, "dst_port": 138, "state": "T", "dst_labels": { "app": "test_app_1",
"env": "test_env_1", "loc": "test_place_1", "role": "test_access_1" }, "dst_hostname":
"corp-vm-2", "dst_href": "/orgs/1/workloads/5ddcc33b-b6a4-4a15-
b600-64f433e4ab33",
"pd": 1, "version": 4}
```

### **Blocked Flow Summary (pd = 2)**

```
2016-01-12T05:23:30+00:00 level=info host=myhost ip=127.0.0.1
program=illumio_pce/
collector| sec=576210.831 sev=INFO pid=25386 tid=16135120 rid=0
sec=915000.311
sev=INFO pid=1372 tid=30302280 rid=90a01be5-a3c1-44f9-84fd-3c3a5eaec1f8
{ "interval_sec": 589, "count": 1, "src_ip": "10.6.1.89", "dst_ip": "10.6.255.255",
"timestamp": "2017-08-16T13:22:09-07:00", "dir": "I", "proto": 17, "dst_port": 138,
"dst_labels": { "app": "test_app_1", "env": "test_env_1", "loc": "test_place_1",
"role": "test_access_1" }, "dst_hostname": "corp-vm-1", "dst_href": "/orgs/1/
workloads/
a83ba658-576b-4946-800a-b39ba2a2e81a", "pd": 2, "version": 4}
```

### **Unknown Flow Summary (pd = 3)**

```
2019-06-14T05:33:45.442561+00:00 level=info host=devtest0 ip=127.0.0.1
program=illumio_pce/collector| sec=490425.442 sev=INFO pid=12381
```

```

tid=32524120
rid=6ef5a6ac-8a9c-4f46-9180-c0c91ef94759
{"dst_port":1022,"proto":6,"count":20,
"interval_sec":600,"timestamp":"2019-06-06T21:03:57Z","src_ip":"10.23.2.7",
"dst_ip":"10.0.2.15","dir":"O","state":"S","pd":3,"src_href":"/orgs/1/
workloads/
a0d735ce-c55f-4a38-965f-bf6e98173598","dst_hostname":"workload1",
"dst_href":"/orgs/1/workloads/a20eb1b5-10a4-419e-
b216-8b35c795a01e","src_labels":
{"app":"app","env":"Development","loc":"Amazon","role":"Load Balancer"}
,"version":4}

```

## CEF

```

CEF:0|Illumio|PCE|2015.9.0|flow_potentially_blocked|Flow Potentially
Blocked|3|
act=potentially_blocked cat=flow_summary deviceDirection=0 dpt=137
src=someIPaddress
dst=someIPaddress proto=udp cnt=1 in=1638 out=0 rt=Jun 14 2018 01:50:14
cn1=120 cn1Label=interval_sec cs2=T cs2Label=state cs6=/orgs/1/workloads/
someID cs6Label=dst_href
cs4={"app":"CRM","env":"Development","loc":"AppLocation",
"role":"Web"} cs4Label=dst_labels dhost=connectivity-check.someDomainName
cs1={"count":1,"max_score":3.7,"cve_ids":
["CVE-2013-2566","CVE-2015-2808"]}
cs1Label=dst_vulns dvchost=someDomainName

```

## Unknown Flow Summary (pd = 3)

```

2019-06-14T21:02:55.146101+00:00 level=info host=devtest0 ip=127.0.0.1
program=illumio_pce/collector| sec=546175.145 sev=INFO pid=15416
tid=40627440
rid=f051856d-b9ee-4ac8-85ea-4cb857eefa82 CEF:0|Illumio|PCE|19.3.0|
flow_unknown|
Flow Unknown|1|act=unknown cat=flow_summary deviceDirection=0 dpt=22
src=10.0.2.2
dst=10.0.2.15 proto=tcp cnt=6 in=6 out=6 rt=Jun 14 2019 21:02:25
duser=root
dproc=sshd cn1=31 cn1Label=interval_sec cs2=S cs2Label=state
dhost=workload1
cs6=/orgs/1/workloads/a20eb1b5-10a4-419e-b216-8b35c795a01e
cs6Label=dst_href
dvchost=devtest0.ilabs.io msg=
{"trafcclass_code":"U"}

```

## LEEF

```

LEEF:2.0|Illumio|PCE|2015.9.0|flow_blocked|cat=flow_summary
devTime=2018-06-14T10:38:53-07:00 devTimeFormat=yyyy-MM-dd'T'HH:mm:ssX
proto=udp sev=5 src=someIPaddress dst=someIPaddress dstPort=5353 count=15
dir=I intervalSec=56728 dstHostname=someHostName dstHref=/orgs/1/workloads/
someID
dstLabels={"app":"CRM","env":"Development","loc":"Cloud","role":"Web"}
dstVulns={"count":2,"max_score":3.7} dstFqdn=someDomainName "cve_ids":
["CVE-2013-2566","CVE-2015-2808"]

```

## Unknown Flow Summary (pd = 3)

```
2019-06-14T19:25:53.524103+00:00 level=info host=devtest0 ip=127.0.0.1
program=illumio_pce/collector| sec=540353.474 sev=INFO pid=9960 tid=36072680
rid=49626dfa-d539-4cff-8999-1540df1a1f61 LEEF:2.0|Illumio|PCE|19.3.0|
flow_unknown|cat=flow_summary devTime=2019-06-06T21:03:57Z
devTimeFormat=yyyy-MM-dd'T'HH:mm:ssX proto=tcp sev=1 src=10.23.2.7
dst=10.0.2.15 dstPort=1022 count=20 dir=0 intervalSec=600 state=S
srcHref=/orgs/1/workloads/a0d735ce-c55f-4a38-965f-bf6e98173598 srcLabels=
{"app": "app", "env": "Staging", "loc": "Azure", "role": "API"}
dstHostname=workload1 dstHref=/orgs/1/workloads/a20eb1b5-10a4-419e-
b216-8b35c795a01e
```

## Manage Traffic Flows Using REST API

You can use the following properties to manage traffic flows using the REST API.



### NOTE

You should ignore and *not* use any extra properties that are not described in this document, such as tbi, tbo, dbi, and dbo.

Property	Description	Type	Required	Possible Values
version	The version of the flow summary schema.	Integer	Yes	4
timestamp	Indicates the time (RFC3339) when the first flow in the summary was created, represented in UTC.  Format: yyyy-MM-dd'T'HH:mm:ss.SSSSSZ	String	Yes	
interval_sec	Sample duration for the flows in the summary. The default is approximately 600 seconds (10 minutes), depending on the VEN's ability to report traffic and the PCE's current load.	Integer	Yes	
dir	Direction of the first packet: in or out (I, O).	String	Yes	I, O
src_ip	Source IP of the flows.	String	Yes	
dst_ip	Destination IP of the flows.	String	Yes	
proto	Protocol number (0-255).  Maximum=255	Integer	Yes	Minimum=0  Maximum=255
type	The ICMP message type is associated with the first flow in the summary. This value exists only if the protocol is ICMP (1).  Maximum=255	Integer	No	Minimum=0  Maximum=255
<div style="background-color: #f0f0f0; padding: 10px;">  <b>NOTE</b>            This information is included in blocked flows for VEN versions lower than 19.1.0. It is included in all flows for VEN version 19.1.0 and later.         </div>				
<p>Example: 3 for "Destination Unreachable."</p>				
code	The ICMP message code (subtype) is associated with the first flow in the summary. This value exists only if the protocol is ICMP (1).  Maximum=255	Integer	No	Minimum=0  Maximum=255
<div style="background-color: #f0f0f0; padding: 10px;">  <b>NOTE</b>            This information is included in blocked flows for VEN versions lower than 19.1.0. It is included in all flows for VEN version 19.1.0 and later.         </div>				
<p>Example: 1 for "Destination host unreachable."</p>				
dst_port	Destination port.	Integer	Yes	Minimum=0

Property	Description	Type	Required	Possible Values
	This value exists only if the protocol is not TCP (6) or UDP (17).			Maximun=65535
pd	<p>This is the Policy decision value, which indicates if the flow was allowed, potentially blocked (but allowed), blocked, or unknown.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <b>0</b> – Allowed traffic</li> <li>• <b>1</b> – Allowed traffic but will be blocked after policy enforcement</li> <li>• <b>2</b> – Blocked traffic</li> <li>• <b>3</b> – Unknown</li> </ul>	Integer	Yes	Minimum=0 Maximum=3
	 <b>NOTE</b> Policy decision is “unknown” in the following cases: <ul style="list-style-type: none"> <li>• Flows uploaded using existing bulk API (/orgs/&lt;org_id&gt;/agents/bulk_traffic_flows).</li> <li>• Flows uploaded using Network Flow Ingest Application (/orgs/&lt;org_id&gt;/traffic_data).</li> <li>• Traffic reported by idle VENs, specifically those reported with “s” state (snapshot).</li> </ul>			
count	Count of the number of flows in the flow summary.	Integer	Yes	
state	The session state for the traffic flows is in the flow summaries.	String	No	A, C, T, S, N
	<p>Possible values:</p> <ul style="list-style-type: none"> <li>• <b>Active (A):</b> The connection was still open when the flow summary was logged. This applies to allowed and potentially blocked flows.</li> <li>• <b>Closed (C):</b> (Linux only) The connection was closed when the flow summary was logged. Applies to allowed and potentially blocked flows.</li> <li>• <b>Timed out (T):</b> The Connection timed out when the flow summary was logged. Applies to allowed and potentially blocked flows. Due to a limitation of WFP, a Windows VEN will report “T” even when the connection is closed when the flow summary was logged.</li> <li>• <b>Snapshot (S):</b> Snapshot of current connections to and from the VEN, which applies only to workloads whose policy state is set to Idle. Applies to allowed and potentially blocked flows.</li> <li>• <b>New connection (N):</b> Dropped TCP packet contains a SYN associated with a new connection. Applies to blocked TCP flows. The value is empty for blocked UDP flows.</li> </ul>			

Property	Description	Type	Re-required	Possible Values
pn	The program name is associated with the first flow of the summary. It is supported on inbound flows for Linux and Windows VEN and outbound flows for only Windows VEN.	String	No	
	 <b>NOTE</b> This information might not be available on short-lived processes, which are Linux-specific.			
	Currently, flows are aggregated so that this value might represent only the first process detected across all aggregated flows.			
	No process is associated if network communication is done by an OS component (or a driver).			
un	The username is associated with the first flow of the summary. It is supported on inbound flows for Linux and Windows VEN and outbound flows for only Linux VEN.	String	No	
	On Windows, it can include the username of the user account that initiated the connection.			
	 <b>NOTE</b> This information might not be available on short-lived processes.			
sn	The service name associated with the first flow in the summary is supported only on inbound flows on Windows VEN.	String	No	
src_hostname	Hostname of the source workload that reported the flow.	String	No	
src_href	HREF of the source workload that reported the flow.	String	No	
src_labels	Labels applied to the source workload.	Object	No	
	 <b>NOTE</b> The <code>src_hostname</code> , <code>src_href</code> , and <code>src_labels</code> values are not included in a traffic summary if the source of the flow is not an Illumio-labeled workload, such as Internet traffic or a managed workload without any labels applied.			

Property	Description	Type	Required	Possible Values
dst_hostname	Hostname of the destination workload that reported the flow.	String	No	
dst_href	HREF of the destination workload that reported the flow.	String	No	
dst_labels	Labels applied to the destination workload.	Object	No	
<div style="background-color: #f0f0f0; padding: 10px;">  <b>NOTE</b>            The <code>dst_hostname</code>, <code>dst_href</code>, and <code>dst_labels</code> values are not included in a traffic summary if the flow's destination is not an Illumio-labeled workload, such as Internet traffic or a managed workload without any labels applied.         </div>				
dst_vulns	Information about the vulnerabilities on the destination of the traffic flow with the specific port and protocol.	Object	No	
<div style="background-color: #f0f0f0; padding: 10px;">  <b>NOTE</b>  <ul style="list-style-type: none"> <li>Vulnerabilities are defined by Common Vulnerabilities and Exposures (CVE), with identifiers and descriptive names from the U.S. Department of Homeland Security <a href="#">National Cybersecurity Center</a>.</li> <li>The vulnerability information is sent only when the Vulnerability Maps feature is turned on via a license and imported into the PCE from a Vulnerability Scanner, such as Qualys.</li> </ul> </div>				
fqdn	Fully qualified domain name	String	No	

The following table describes the subproperties for the `dst_vulns` property:

Sub-property	Description	Type	Required
count	The total number of existing vulnerabilities on the destination port and protocol.	Integer	No
max_score	The maximum of all the vulnerability scores on the destination port and protocol.	Number	No
cve_ids	The list of CVE-IDs associated with the vulnerabilities that have the maximum score. Up to 100 displayed.	Array	No

## Export Traffic Flow Summaries

Decide where to export the traffic flow summaries: Syslog or Fluentd.



### CAUTION

By default, the PCE generates all traffic flow summaries and sends them to Syslog.

If you have not configured Syslog, the Syslog data is written to a local disk by default. For example, it is written to `/var/log/messages`.

## Export to Syslog

To configure and export the traffic flow summaries to a remote syslog, follow these steps:

1. From the PCE web console menu, choose **Settings > Events Settings**.
2. Enable a remote Syslog destination.
3. Select specific traffic flow summaries to be sent to the remote syslog.

This filters the selected traffic flow summaries and sends those to the remote syslog.

To prevent the Syslog data from being written to a local disk based on your preference, deselect the Events checkboxes on the **Settings > Event Settings** > Local page in the PCE web console.



### NOTE

The generation of all traffic flow summaries is implemented to ensure that they can be controlled only from the PCE web console.

This example shows the `runtime_env.yml` configuration to generate all types of flow summaries.

### Export to Syslog

```
export_flow_summaries_to_syslog:
- accepted
- potentially_blocked
- blocked
```

This example shows the `runtime_env.yml` configuration if you do not want to generate any types of flow summaries.

### Export to Syslog

```
export_flow_summaries_to_syslog:
- none
```



### NOTE

Illumio does not currently support having a primary and secondary syslog configuration with disaster recovery and failover.

You can configure it on a system syslog (local) and use the internal syslog configuration to send messages to the local, which sends to the system syslog.

## Export to Fluentd

To generate and export the traffic flow summaries to Fluentd, follow these steps:

1. Set the `export_flow_summaries_to_fluentd` parameter in `runtime_env.yml`.
2. Set the `external_fluentd_aggregator_servers` parameter in `runtime_env.yml`.

This example shows the `runtime_env.yml` configuration to generate two flow summaries out of the three possible types.

### Export to Fluentd

```
external_fluentd_aggregator_servers:
- fluentd-server.domain.com:24224
export_flow_summaries_to_fluentd:
- accepted
- blocked
```

## Flow Duration Attributes

VENs send two attributes to the Syslog and fluentd output. These attributes describe the flow duration and are appended to the flow data.

- **Delta flow duration in milliseconds (ddms)**: The duration of the aggregate within the current sampling interval. This field lets you calculate the bandwidth between two applications in a given sampling interval. The formula is  $\text{dbo} / \text{delta\_duration\_ms}$  or  $\text{dbi} / \text{delta\_duration\_ms}$ .
- **Total flow duration in milliseconds (tdms)**: The duration of the aggregate across all sampling intervals. This field enables you to calculate the average bandwidth of a connection between two applications. The formula is  $\text{tbo} / \text{total\_duration\_ms}$ , or  $\text{tbo} / \text{total\_duration\_ms}$ . It also enables you to calculate the average volume of data in a connection between two applications. The formula is  $\text{tbo} / \text{count}$  (number of flows in an aggregate) or  $\text{tbi} / \text{count}$ .

## Legal Notice

Copyright © 2025 Illumio 920 De Guigne Drive, Sunnyvale, CA 94085. All rights reserved.

The content in this documentation is provided for informational purposes only and is provided "as is," without warranty of any kind, expressed or implied, of Illumio. The content in this documentation is subject to change without notice.

### Resources

- [Legal information](#)
- [Trademarks statements](#)
- [Patent statements](#)
- [License statements](#)

### Contact Information

- [Contact Illumio](#)
- [Contact Illumio Legal](#)
- [Contact Illumio Documentation](#)