SecurityCenter API Best Practices Guide

Last Revised: September 21, 2018

Table of Contents

Welcome to SecurityCenter API Best Practices

The REST API for SecurityCenter allows you to integrate SecurityCenter with other standalone or web applications by scripting interactions with the SecurityCenter server.

This document describes recommended approaches to common tasks using the SecurityCenter API. For descriptions of all available endpoints for the SecurityCenter API, see the SecurityCenter API reference guide.

This document contains recommendations for the following tasks:

- Launch a Remediation Scan on SecurityCenter
- Retrieve Vulnerability Data for a Specific Time Range
- Retrieve Asset Data from SecurityCenter
- Add Asset Data to SecurityCenter

Note: This documentation provides examples in JavaScript Object Notation (JSON).

Launch a Remediation Scan on SecurityCenter

A remediation scan evaluates a specific plugin against a specific target or targets where the related vulnerability was present in an earlier scan. A remediation scan is a type of active scan.

Remediation scans allow you to validate whether your vulnerability remediation actions on specific targets have been successful. If a remediation scan can no longer identify the vulnerability on targets where it was previously identified, the system changes the status of the vulnerability instances to mitigated. For more information about the methodology the system uses in remediation scans, see "Mitigation Logic" in the SecurityCenter User Guide.

To launch a remediation scan on the SecurityCenter, Tenable recommends the following approach:

1. Authenticate, if you have not already done so. Be sure to include the authentication token in your request headers. For more information on authentication, see the description of the **/token** endpoint in the SecurityCenter API reference guide.

2. Identify the plugin ID and plugin family ID for the vulnerability you want to remediate.

   You can determine the plugin ID and plugin family ID from the vulnerability results of a standard scan. For more information on retrieving these results, see Retrieve Vulnerability Data for a Specific Time Range.

   Alternatively, if you know the plugin ID but not the plugin family ID, use the **/pluginFamily/{id}** endpoint to retrieve the plugin family ID. For more information about this endpoint, see the description of the **/pluginFamily** endpoint in the SecurityCenter API reference guide.

   HTTP Request Syntax:

   ```
   GET /rest/pluginFamily/pluginID
   ```

   HTTP Request Example:

   ```
   GET /rest/pluginFamily/97833
   ```

   HTTP Response Example:

   ```
   {
     "type" : "regular",
     "response" : {
   ```

```
    "id" : "1000030",
    "name" : "Malware",
    "type" : "passive",
    "plugins" : [],
    "count" : 0
  },
  "error_code" : 0,
  "error_msg" : "",
  "warnings" : [],
  "timestamp" : 1408728549
}
```


The **id** field in this response is the plugin family ID. Retain this value for use in the next HTTP request (specifically, in the **families** parameter of the **/policy** request).

3. Create a scan policy for one-time use.

   The policy you create in this step cannot be retrieved in a GET request to the **/policy** endpoint. This policy can be used only for this particular remediation scan.

   > **Note:** If you launch a remediation scan from the SecurityCenter user interface, the system automatically creates a default policy, so there is no action in the SecurityCenter user interface that is equivalent to this step.

   HTTP Request Syntax:

   `POST /rest/policy{parameters}`

   Parameters must include:

Parameter	Value
`context`	`scan`
`families`	Both of the following: • the plugin for the vulnerability you want to remediate (specifically, the plugin ID and plugin family ID you identified in Step 2)

	• the Nessus Scan Information plugin (plugin ID 19506; plugin family ID 41) **Note:** If you omit plugin 19506, the remediation scan returns incomplete scan information, if any.
`policyTemplate {id}`	The code for the Advanced Template (1).
`preferences {parameter,...)`	Specifies parameters corresponding to custom scan policy options. The HTTP request example below includes the following parameters with their default settings: • `portscan_range`—equivalent to **Port Scanning > Ports > Port scan range** • `tcp_scanner`—equivalent to **Port Scanning > Network Port Scanners > TCP** • `syn_scanner`—equivalent to **Port Scanning > Network Port Scanners > SYN** • `udp_scanner`—equivalent to **Port Scanning > Network Port Scanners > UDP** • `syn_firewall_detection`—equivalent to **Port Scanning > Network Port Scanners > Override** automatic firewall detection For more information on these parameters, see “Custom Scan Policy Options” in the SecurityCenter User Guide.

HTTP Request Example:

```
{
  "name": "",
  "description": "",
  "context": "scan",
```

```
"createdTime": 0,
"modifiedTime": 0,
"groups": [],
"policyTemplate": {
  "id": 1
},
"auditFiles": [],
"preferences": {
  "portscan_range": "default",
  "tcp_scanner": "no",
  "syn_scanner": "yes",
  "udp_scanner": "no",
  "syn_firewall_detection": "Automatic (normal)"
},
"families": [
  {
    "id": "20",
    "plugins": [
      {
        "id": "97833"
      }
    ]
  },
  {
    "id": "41",
    "plugins": [
     {
        "id": "19506"
      }
    ]
  }
]
```

```
}
```

4. When you receive an HTTP response to your POST **/policy** request, retain the **id** element for use in the next HTTP request (that is, the POST **/scan** request).

   HTTP Response Example:

```
"type": "regular",
"response":
  {
    "id": "1000007",
    "name": "_1513112172_0.87691100_1_1_172.20.128.186_9070_scan_
e448b4e5c935ed3738b1d06527975e73",
    "description": "",
    "tags": ""
  }
```

   In this example, the policy ID is `1000007`.

5. Use the **/scan** endpoint to launch a remediation scan.

   HTTP Request Syntax:

   `POST /rest/scan{parameters}`

   Recommended parameters include:

Parameter	Value
`assets`	Specifies the asset or assets you want the remediation scan to target. **Note:** You can use either the `assets` parameter or the `ipList` parameter to specify assets, but you cannot use both parameters in a single request. Use an array of objects with the `id` attribute to specify individual assets. You can obtain asset IDs from the original scan

	results. For example: `"assets": [ { "id": 24 }, { "id": 20 }, { "id": 19 } ]` This parameter corresponds to the **Targets** section of an active scan in the SecurityCenter user interface.
`classifyMitigatedAge`	Specifies the number of days the system waits to remove vulnerabilities from the cumulative database if the related hosts do not reply to the scan. If this parameter is set to `0`, the system removes the vulnerabilities immediately. If this parameter is set to any other valid value, the system waits that number of days before removing the vulnerabilities. Valid values include: `1`, `2`, `3`, `4`, `5`, `6`, `30`, `60`, `90`, and `365` (default). This parameter corresponds to the following parameters in the **Settings** section of an active scan in the SecurityCenter user interface: • **Immediately remove vulnerabilities from scanned hosts that do not reply** • **Number of days to wait before removing dead hosts**
`dhcpTracking`	Specifies whether the system uses tracks hosts associated with

	changed IP addresses. The default value is `false`. Networks using DHCP require that this parameter be set to `true` to properly track hosts. This parameter corresponds to the **Track hosts which have been issued new IP address** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
`emailOnLaunch`	Specifies whether the system emails you a notification when the scan launches. If you set this parameter to `true`, the system uses the email associated with the user account making the API request. This parameter corresponds to the **E-mail me on Launch** parameter in the **Post Scan** section of an active scan in the SecurityCenter user interface.
`emailOnFinish`	Specifies whether the system emails you a notification when the scan completes. If you set this parameter to `true`, the system uses the email associated with the user account making the API request. This parameter corresponds to the **E-mail me on Completion** parameter in the **Post Scan** section of an active scan in the SecurityCenter user interface.
`ipList`	Specifies the IP address or addresses of the asset or assets you want to scan. **Note:** You can use either the `assets` parameter or the `ipList` parameter to specify assets, but you cannot use both parameters in a single request. This value can be an IPv4 or IPv6 address (depending on the repository to be scanned), a CIDR address, or a DNS name. For example: `"ipList": "198.168.1.1"` This parameter corresponds to the **Targets** section of an active scan in the SecurityCenter user interface.
`maxScanTime`	Specifies the number of hours after which the scan stops run-

	ning. The default value is `unlimited`. This parameter corresponds to the **Max scan duration** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
`name`	Specifies the display name of the remediation scan. To preserve consistency with remediation scans launched from the SecurityCenter user interface, use the following name format: `Remediation Scan of Plugin #pluginID` This parameter corresponds to the **Name** parameter in the **General** section of an active scan in the SecurityCenter user interface.
`policy`	Specifies the ID of the policy you created earlier. See Step 4 in this procedure.
`pluginID`	Specifies the ID of the plugin you want to remediate. **Note:** The plugin family ID is *not* required in this request.
`repository`	Specifies the ID of the repository where you want to import the remediation scan results. The repository for the remediation scan must be the same repository as the original scan. You can obtain the repository ID from the vulnerability results of the original scan. For more information on retrieving these results, see Retrieve Vulnerability Data for a Specific Time Range. This parameter corresponds to the **Import Repository** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
`rolloverType`	Specifies how the system schedules the rollover scan it creates. This field is required if the `timeoutAction` is set to rollover. Valid values are: • `nextday`—Create a rollover scan scheduled to launch

	the next day at the same start time as the just completed scan. • `template`—Create a rollover scan as a template for users to launch manually. This parameter corresponds to the **Rollover Schedule** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
`scanningVirtualHosts`	Specifies whether the system treats a new DNS entry for an IP address as a virtual host as opposed to a DNS name update. The default value is `false`. This parameter corresponds to the **Scan Virtual Hosts** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
`schedule`	Specifies the schedule for the scan. For a remediation scan, this parameter must include the following fields: • `start`—The time you want the scan to launch. Use the following format: `TZID=`*`timezone`* The time zone must be in the iCalendar date-time format. • `repeatRule`—The frequency of the scan. Use the following value for this field: `FREQ=NOW;INTERVAL=1` • `type`—The type of scan. Use the following value for this field: now This value causes the system to delete the related one-time scan policy after the scan completes. For example:

	<pre>"schedule": {
	  "start": "TZID=America/New_
	York:20171212T160900",
	  "repeatRule": "FREQ=NOW;INTERVAL=1",
	  "type": "now"
	},</pre>

timeoutAction	Specifies how the system responds if it cannot complete the scan. Valid values are: • discard—Do not import any of the results obtained by the scan to the database. • import—Import the results of the current scan and discard the information for any unscanned targets. • rollover—Import the results from the scan into the database and create a rollover scan that may be launched at a later time to complete the scan. This parameter corresponds to the **Scan Timeout Action** parameter in the **Settings** section of an active scan in the SecurityCenter user interface.
type	Specifies the scan type. To use the scan policy you created earlier, this value must be policy.

HTTP Request Example:

```
{
  "name": "Remediation Scan of Plugin #97833",
  "description": "",
  "context": "",
  "createdTime": 0,
  "modifiedTime": 0,
```

```
"groups": [],
"repository": {
  "id": 1
},
"schedule": {
  "start": "TZID=America/New_York:20171212T160900",
  "repeatRule": "FREQ=NOW;INTERVAL=1",
  "type": "now"
},
"dhcpTracking": "true",
"emailOnLaunch": "false",
"emailOnFinish": "false",
"reports": [],
"type": "policy",
"policy": {
  "id": 1000007
},
"pluginID": "97833",
"timeoutAction": "rollover",
"rolloverType": "template",
"scanningVirtualHosts": "false",
"classifyMitigatedAge": 0,
"assets": [
  {
    "id": 24
  },
  {
    "id": 20
  },
  {
    "id": 19
  }
```

```
  ],
  "ipList": "",
  "credentials": [],
  "maxScanTime": "unlimited"
}
```

Retrieve Vulnerability Data for a Specific Time Range

To retrieve vulnerability data using the SecurityCenter API, Tenable recommends the following approach:

1. Authenticate, if you have not already done so. Be sure to include the authentication token in your request headers. For more information on authentication, see the description of the **/token** endpoint in the SecurityCenter API reference guide.

2. Use the POST version of the **/analysis** endpoint.

   HTTP Request Syntax:

   `POST /rest/analysis{parameters}`

   Recommended parameters are:

Parameter	Value
`query`	Specifies the parameters of the data you want to retrieve for analysis. This parameter encapsulates the functionality of the **/query** endpoint. For the **/query** parameters Tenable recommends in this specific case, see recommended query parameters below. For a full description of the available **/query** parameters, see the SecurityCenter API reference guide.
`sortDir`	Specifies the sort order for the data, using the field you specify in `sortField`. Requires companion parameter, `sortField`. Valid values are: • `asc`—Sorts data in ascending order (A-Z, 0-9). • `desc`—Sorts data in descending order (Z-A, 9-0).
`sortField`	Specifies the field the system uses to sort the data. Requires companion parameter, `sortDir`. For vulnerabilities data, Tenable recommends you sort by severity: `“sortField”: “severity”`

`sourceType`	Specifies the status of the vulnerabilities you want to analyze. This field corresponds to the **Options > Switch to** options on the **Vulnerability Analysis** page in the SecurityCenter user interface. Valid values are: `cumulative`—Analyzes cumulative vulnerabilities. This parameter corresponds to **Options > Switch to Cumulative** on the **Vulnerability Analysis** page. `patched`—Analyzes mitigated vulnerabilities. This parameter corresponds to **Options > Switch to Mitigated** on the Vulnerability Analysis page.
`type`	Specifies the type of data you want to analyze. For vulnerability data, use `vuln`. This field corresponds to the options available when you click **Analysis** in the top navigation bar of the SecurityCenter user interface. Specifying `vuln` is equivalent to clicking **Analysis > Vulnerabilities**.

For all parameters supported for the **/analysis** endpoint, see the SecurityCenter API reference guide

Recommended Query Parameters:

Tenable recommends that you use the following parameters for the `query` element of the **/analysis** endpoint when retrieving vulnerability data:

Parameter	Value
`endOffset`	Specifies the last record in the range you want to retrieve. For example, if this value is `50`, the retrieved data range stops at the 50th result that meets the query criteria. This parameter requires the presence of the `startOffset` parameter.
`filters`	Specifies the filter criteria for the data you want the system to retrieve. Filter parameters include: • `filterName`—The name of the field on which the query filters. To

	limit your query results to a specific date range, use the following: ◦ `firstSeen`—Equivalent to filtering on **Vulnerability Discovered** on the **Vulnerability Analysis** page of the SecurityCenter user interface. ◦ `lastSeen`—Equivalent to filtering on **Vulnerability Observed** on the **Vulnerability Analysis** page of SecurityCenter user interface. Requires that the `sourceType` parameter is `cumulative`. ◦ `lastMitigated`—Equivalent to filtering on **Vulnerability Mitigated** on the **Vulnerability Analysis** page of SecurityCenter user interface. Requires that the `sourceType` parameter is `patched`. • `operator`—An operator that specifies the relationship between the `filterName` and the `value` elements. • `value`—A code in the format `#:#` specifying a date range relative to the current day. The first number represents the starting day, and the second number represents the last day of the range. For example: `"filters": { "filterName": "firstSeen", "operator": "=", "value": "#:#" }` In this example, if the `sourceType` for the query is `cumulative`, and this filter value is set to `0:11`, the query retrieves all vulnerabilities first discovered between today and 11 days ago. To set this filter to a single date, use the same number for both elements. For example, a value of `0:0` retrieves all vulnerabilities first discovered today.
`sourceType`	Specifies the status of the vulnerabilities you want to retrieve.

	Valid values are: `cumulative`—Retrieves vulnerabilities from the cumulative database. Equivalent to **Options > Switch to cumulative** in the **Vulnerability Analysis** page. `patched`—Retrieves vulnerabilities from the mitigated database. Equivalent to **Options > Switch to Mitigated** in the **Vulnerability Analysis** page.
`startOffset`	Specifies the first record in the range you want to retrieve. For example, if this parameter is `0`, the retrieved data starts at the first result that meets the query criteria. This parameter requires the presence of the `endOffset` parameter.
`tools`	Specifies the level of vulnerability detail you want to retrieve. This field corresponds to the drop-down options on the **Vulnerability Analysis** page in the SecurityCenter user interface. Valid values include: • `listvuln`—Equivalent to the **Vulnerability List** option on the **Vulnerability Analysis** page. • `vulndetails`—Equivalent to the **Vulnerability Details List** option on the **Vulnerability Analysis** page.
`type`	Specifies type of data you want to retrieve. For vulnerability data, use `vuln`.

HTTP Request Example—Cumulative Vulnerabilities:

```
{
  "query": {
    "name": "",
    "description": "",
    "context": "",
    "createdTime": 0,
```

```
    "modifiedTime": 0,
    "groups": [],
    "type": "vuln",
    "tool": "vulndetails",
    "sourceType": "cumulative",
    "startOffset": 0,
    "endOffset": 50,
    "filters": [
      {
        "filterName": "firstSeen",
        "operator": "=",
        "value": "0:11"
      },
      {
        "filterName": "lastSeen",
        "operator": "=",
        "value": "0:12"
      }
    ]
  },
  "sourceType": "cumulative",
  "sortField": "severity",
  "sortDir": "desc",
  "columns": [],
  "type": "vuln"
}
```

HTTP Request Example—Mitigated Vulnerabilities:

```
{
  "query": {
    "name": "",
    "description": "",
```

```
    "context": "",
    "createdTime": 0,
    "modifiedTime": 0,
    "groups": [],
    "type": "vuln",
    "tool": "listvuln",
    "sourceType": "patched",
    "startOffset": 0,
    "endOffset": 50,
    "filters": [
      {
        "filterName": "lastMitigated",
        "operator": "=",
        "value": "0:25"
       },
    ]
  }
  "sourceType": "patched",
   "sortField": "severity",
  "sortDir": "desc",
  "columns": [],
  "type": "vuln"
}
```

Retrieve Asset Data from SecurityCenter

To retrieve asset data from SecurityCenter, Tenable recommends the following approach:

1. Authenticate, if you have not already done so. Be sure to include the authentication token in your request headers. For more information on authentication, see the description of the **/token** endpoint in the SecurityCenter API reference guide.

2. Use the GET version of the **/asset** endpoint.

   HTTP Request Syntax:

   `GET /rest/asset?filter=`*`filter`*`,...&fields=`*`field`*`,...`

   To retrieve data based on the default filters in the **Assets** page of the SecurityCenter user interface, Tenable recommends the following:

   ```
   filter=excludeAllDefined,usable
   ```

   For a full list of `filter` and `field` values for the **/asset** endpoint, see the SecurityCenter API reference guide.

   HTTP Request Example:

   ```
   GET /rest/asset?filter=excludeAllDefined,usable&fields=owner,groups,
   ownerGroup,status,name,type,template,description,createdTime,
   modifiedTime,ipCount,repositories,targetGroup,tags,creator
   ```

   Note: This example includes line breaks for readability. In actual use, the request cannot include line breaks.

Add Asset Data to SecurityCenter

Tenable recommends that you add asset data to SecurityCenter as a static IP list of assets.

To add assets:

1. Authenticate, if you have not already done so. Be sure to include the authentication token in your request headers. For more information on authentication, see the description of the **/token** endpoint in the SecurityCenter API reference guide.

2. Use the POST version of the **/asset** endpoint.

   HTTP Request Syntax:

   `POST /rest/asset{parameters}`

   Recommended parameters include:

Parameter	Value
`definedIPs`	Specifies the IP addresses for the assets you want to add. Use commas or newline characters (`\n`) to separate values in this field. This field can accept individual IP addresses, CIDR addresses, or IP address ranges. You can use multiple address types in a single entry. For example, the following valid value includes a single IP address, a CIDR address, and a range of IP addresses: `“definedIPs”: “127.0.0.1,172.36.48.0/24,198.168.1.1-198.168.1.11”` This parameter corresponds to the IP Addresses parameter in the custom asset list options in the SecurityCenter user interface.
`name`	Specifies the display name of the static IP list of assets. This parameter corresponds to the **Name** parameter in the custom asset list options in the SecurityCenter user interface.
`type`	Specifies the type of custom asset list you are adding. Tenable recommends `static`.

For a full list of parameters for the **/asset** endpoint, see the SecurityCenter API reference guide.

HTTP Request Example:

```
{
  "tags": "",
  "name": "static test list",
  "description": "",
  "context": "",
  "status": -1,
  "createdTime": 0,
  "modifiedTime": 0,
  "groups": [],
  "type": "static",
  "definedIPs": "127.0.0.1,198.168.1.1"
}
```

3. To verify that the system correctly added the assets, view the assets in the SecurityCenter user interface, or use the **/asset** endpoint to Retrieve Asset Data from SecurityCenter.