Trend Micro™ TippingPoint™

Security Management System (SMS) Web API Guide

Trend Micro Incorporated reserves the right to make changes to this document and to the product described herein without notice. Before installing and using the product, review the readme files, release notes, and/or the latest version of the applicable documentation, which are available from the Trend Micro website at:

https://docs.trendmicro.com/en-us/tippingpoint/security-management-system.aspx

Trend Micro, the Trend Micro t-ball logo, TippingPoint, and Digital Vaccine are trademarks or registered trademarks of Trend Micro Incorporated. All other product or company names may be trademarks or registered trademarks of their owners.

Copyright © 2025. Trend Micro Incorporated. All rights reserved.

Document Part No.: TPEM09844/230927

Release Date: September 2025

Protected by U.S. Patent No.: Pending

This documentation introduces the main features of the product and/or provides installation instructions for a production environment. Read through the documentation before installing or using the product.

Detailed information about how to use specific features within the product may be available at the Trend Micro Online Help Center and/or the Trend Micro Knowledge Base.

Privacy and Personal Data Collection Disclosure

Certain features available in Trend Micro products collect and send feedback regarding product usage and detection information to Trend Micro. Some of this data is considered personal in certain jurisdictions and under certain regulations. If you do not want Trend Micro to collect personal data, you must ensure that you disable the related features.

The following link outlines the types of data that TippingPoint Security Management System collects and provides detailed instructions on how to disable the specific features that feedback the information.

https://success.trendmicro.com/data-collection-disclosure

Data collected by Trend Micro is subject to the conditions stated in the Trend Micro Privacy Notice:

https://www.trendmicro.com/privacy

Table of Contents

Chapter 1: SMS Web API Guide

API 1-2

Authentication 1-2

Errors 1-3

Chapter 2: Profile management

Export a profile 2-3

Import a profile 2-4

Distribute a profile 2-6

Get distribution status 2-8

Create a traffic management filter 2-9

Delete a traffic management filter 2-12

Get current filter override settings (with POST) 2-13

Get all filter settings (with GET) 2-19

Update filter settings 2-20

Get Digital Vaccine information 2-26

Chapter 3: Device administration

Get fallback status 3-2

Set fallback status 3-2

Chapter 4: SMS administration

Backup SMS database 4-2

SMS software version 4-3

Restore backup file 4-4

Chapter 5: Virtual segment management

Create a virtual segment 5-3
Update a virtual segment 5-3
Delete a virtual segment 5-4
Get list of virtual segments 5-4

Chapter 6: Reputation database management

Import a Reputation entry 6-3
Add a Reputation entry 6-9
Query the Reputation database 6-10
Delete a Reputation entry 6-13

Chapter 7: Packet trace

Device-based packet trace 7-2
Events-based packet trace 7-2

Chapter 8: Responder

Quarantine 8-2
Unquarantine 8-3

Chapter 9: Enterprise Vulnerability Remediation (eVR)

Import a vulnerability scan 9-3
Convert a vulnerability scan 9-3

Chapter 10: STIX/TAXII

Data format 10-4
Bundle 10-4
Indicators 10-5
Pattern 10-6
Comparison expression 10-6
Labels 10-7

TAXII 2.0 APIs 10-8
Server discovery 10-8
Get API root information 10-9
Get collections 10-10
Add objects 10-11
Get status 10-12
TAXII 2.1 APIs 10-13
Server discovery 10-13
Get API root information 10-14
Get collections 10-15
Add objects 10-16
Get status 10-17

Chapter 11: Database access

Usage sequence 11-2
DataDictionary 11-2
ACTIONSET table 11-3
ALERT_TYPE table 11-4
DEVICE table 11-4
POLICY table 11-5
PRODUCT_CATEGORY table 11-6
PROFILE table 11-6
PROFILE_INSTALL_INVENTORY table 11-7
QUARANTINE_NETWORK_DEVICES table 11-7
SEGMENT table 11-8
SEGMENT_GROUP table 11-8
SIGNATURE table 11-9
TAXONOMY_MAJOR table 11-10
TAXONOMY_MINOR table 11-10
TAXONOMY_PLATFORM table 11-11
TAXONOMY_PROTOCOL table 11-11
THRESHOLD_UNITS table 11-12
VIRTUAL_SEGMENT table 11-12
GetData - Events data 11-13
ALERTS table 11-14

DDOS_STATS table 11-19
FIREWALL_BLOCK_ALERTS table 11-19
FIREWALL_TRAFFIC_ALERTS table 11-21
PORT_TRAFFIC_STATS table 11-22
QUARANTINE_HOSTS table 11-23
RATELIMIT_STATS table 11-24
GetNewestRecord 11-24
GetOldestRecord 11-25
Schema 11-26
Status 11-26
Version 11-26

Chapter 12: External database

Configure the SMS for external access 12-3
ALERTS table – ExternalAccess 12-4
Configure the SMS for replication 12-6
Replication – database schema 12-8
Configure the SMS to enable restricted access 12-8

Chapter 13: MIB files for the SMS

SMS MIBs 13-2
Public MIB files 13-2
Health monitoring 13-2

Chapter 14: Event Taxonomy

Event Taxonomy 14-2
Taxonomy Event ID 14-2
Data detail examples 14-2
Major categories 14-4
Minor categories 14-4
Protocol type 14-7
Platform type 14-11

Chapter 1

SMS Web API

The Trend Micro™ TippingPoint™ Security Management System (SMS) Web API provides access to the following set of SMS features:

- *Profile management on page 2-1*
- *Device administration on page 3-1*
- *SMS administration on page 4-1*
- *Virtual segment management on page 5-1*
- *Reputation database management on page 6-1*
- *Packet trace on page 7-1*
- *Responder on page 8-1*
- *Enterprise Vulnerability Remediation (eVR) on page 9-1*
- *STIX/TAXII on page 10-1*
- *Database access on page 11-1*
- *External database on page 12-1*
- *MIB files for the SMS on page 13-1*
- *Event Taxonomy on page 14-1*

API

The *SMS Web API Guide* describes HTTP APIs you can use to access multiple SMS features if you have HTTPS service to the SMS. To ensure that you have the latest versions of product documentation, visit the Online Help Center.

Authentication

Access to the SMS Web API requires that you authenticate by using HTTP authentication or the API key.

- HTTP authentication: `-u {username}:{password}`

```
curl -k -u {username}:{password} "https://<sms_server>/ipsProfileMgmt/
exportProfile?profileName=MyProfile"
```

- API key: authentication mechanism that does not require a username and password. Use the API key as part of the header for HTTP requests. `X-SMS-API-KEY: <string>`

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/ipsProfileMgmt/
exportProfile?profileName=MyProfile"
```

Note

HTTP authentication is supported; however this authentication mechanism is deprecated and not recommended. We recommend that you authenticate by using the API key.

Keep in mind the following considerations:

- To view the API key on the SMS, go to **Admin** > **Authentication and Authorization** > **Users** > **Edit** > **Authentication** > **API Key**.
- To customize or replace the default SMS SSL X509 certificate on the SMS, go to **Admin** > **Certificate Management**.
- Only superusers should have web access for full authorization. On the SMS, go to **Admin** > **Authentication and Authorization** > **Roles**.

- HTTPS service is required to send API requests to the SMS. On the SMS, go to **Admin** > **Server Properties** > **Services**.

Errors

The SMS web API returns one of the following HTTP status codes if the request is unsuccessful.

Code	Description
400	Bad request – malformed parameter or request.
401	Unauthorized – missing or incorrect credentials.
403	No web access capability. If you receive this message, check the user role capabilities, and enable the `Access SMS Web Services` capability. On the SMS, go to **Admin** > **Authentication and Authorization** > **Roles** > **Edit** > **Capabilities** > **Admin** > **Access SMS Web Services**.
404	Not found – invalid or nonexistent requested source.
412	Preconditioned fail – unexpected error. Check the SMS System Log. On the SMS, go to **Admin** > **General** > **SMS System Log**.
500	Internal server error – server-side exception. Check the SMS System Log. On the SMS, go to **Admin** > **General** > **SMS System Log**.

Chapter 2

Profile management

Use this API to export, import, and distribute an SMS profile, and to create and update filters. In addition, you can retrieve profile distribution status and data about the TippingPoint Digital Vaccine (DV) on the SMS.

Note

API calls do not support profiles with special characters. Use only alphanumeric characters in profile names.

This API includes:

- *Export a profile on page 2-3*
- *Import a profile on page 2-4*
- *Distribute a profile on page 2-6*
- *Get distribution status on page 2-8*
- *Create a traffic management filter on page 2-9*
- *Delete a traffic management filter on page 2-12*
- *Get current filter override settings (with POST) on page 2-13*
- *Update filter settings on page 2-20*

- *Get Digital Vaccine information on page 2-26*

Export a profile

Get and export a profile package from the SMS.

- Profile packages typically remain unchanged.
- If you want to change the files within a profile package, update the md5sum in the `sms-security-manifest` file before you import the profile back to the SMS.

Definition

```
ipsProfileMgmt/exportProfile
```

Parameters

Parameter	Type	Description
exportMethod	string	Optional. Export destination. • SMS HTTPS server (default) • SMB • NFS
profileName	string	Required. Profile name to export.
profileVersion	string	Optional. Profile version to export. The latest version of the profile is used if this parameter is not specified.
remoteDirectory	string	Required. • SMB • NFS
remoteDirectory/SMB/userid	string	SMB user ID.
remoteDirectory/SMB/password	string	SMB password.
remoteDirectory/SMB/domain	string	SMB domain.

Parameter	Type	Description
remoteFilename	string	Optional. Remote filename (default: "profile_name.pkg")
remoteServer	string	SMB or NFS server.

Example

```
curl -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/ipsProfileMgmt/exportProfile?
exportMethod=SMB&profileName=Default&remoteDirectory=MyExportDirectory
&remoteServer=MyRemoteServer&userid=guest&password=guestpass&domain=Domain"
```

Import a profile

Post and import an exported profile package to the SMS. Profiles include shared settings such as action sets, notification contacts, and services.

- If the imported profile includes policies or category settings that use a particular action set, the action set is added to the SMS. The SMS does not overwrite an existing action set with the same name. Instead, the SMS renames the new action set by appending a number to the end of the file name, for example, “My Quarantine_2”.
- A notification contact that is used by an action set is also imported and renamed, if necessary.
- Existing port definitions for services on the SMS remain the same. If an imported profile includes a service with a port definition that differs from the existing service on the SMS, the service is added to the SMS service list. Review services any time a profile is imported from a different user or from a different environment.

Definition

```
ipsProfileMgmt/importProfile
```

Parameters

Parameter	Type	Description
importAction	string	Required. • add: Adds a completely new profile; must have an unused name or import fails. • combine_add: Adds new settings and merges non-conflicting changes into an existing profile. • combine_change: Adds new settings to and overwrites existing settings of an existing profile with settings of the new profile. • replace: Overwrites contents of SMS profile with those of the profile being imported; name and UUID remain the same; snapshot of replaced profile occurs and updated profile gets new version.
targetProfileName	string	Name of the existing profile on the SMS. Required for all replace and combine actions. • The profile must exist on the SMS. • If the specified profile does not exist or is not specified in the request, the operation fails and updates the audit log.
replacedProfileName	string	Name of the imported profile that will have its contents applied to the existing profile on the SMS. • Required for all replace and combine actions. • The profile must be specified in the request. • If the specified profile does not exist or is not specified in the request, the operation fails and updates the audit log.

Example

```
curl -k --header "X-SMS-API-KEY: <string>"-F "file=@</path/to/import.pkg>"
"https://<sms_server>/ipsProfileMgmt/importProfile?importAction=replace&
targetProfileName=<name>&replacedProfileName=<name>"
```

Distribute a profile

Get and initiate a profile distribution to a single segment target or to a segment group.

Definition

```
ipsProfileMgmt/distributeProfile
```

Parameters

Parameter	Type	Description
profile		• id • name • version
priority	string	Distribution priority. • high • low
segmentGroup	string	Segment group.
virtualSegment	string	Virtual segment.
device/id	string	Internal ID assigned to the device.
device/shortID	integer	Internal number assigned to the device.
device/name	string	Device name.
device/ipAddress	string	Device IP address.
device/ virtualSegment	string	Virtual segment on the device.

Example

```
curl -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/ipsProfileMgmt/distributeProfile?"
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
     <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:simpleType name="uuid">
          <xs:restriction base="xs:string">
          <xs:pattern value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
                        {4}-[0-9a-f]{12}"/>
                  </xs:restriction>
     </xs:simpleType>

    <xs:complexType name="idname">
          <xs:choice>
              <xs:element name="id" type="uuid"/>
              <xs:element name="name" type="xs:string"/>
          </xs:choice>
 </xs:complexType>

    <xs:element name="distribution">
             <xs:complexType>
                 <xs:sequence>
                     <xs:element name="profile" minOccurs="1"maxOccurs="1">
             <xs:complexType>
                  <xs:attribute name="id" type="uuid"/>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="version" type="xs:string" use="required"/>
          </xs:complexType>
 </xs:element>

    <xs:element name="priority" minOccurs="0">
         <xs:simpleType>
             <xs:restriction base="xs:string">
                     <xs:enumeration value="high"/>
                     <xs:enumeration value="low"/>
             </xs:restriction>
      </xs:simpleType>
 </xs:element>

<xs:element name="segmentGroup" type="idname" minOccurs="0"
            maxOccurs="unbounded"/>
 <xs:element name="virtualSegment" minOccurs="0"
                maxOccurs="unbounded">
            <xs:complexType>
                 <xs:sequence>
                  <xs:element name="id" type="uuid"/>
             </xs:sequence>
          </xs:complexType>
 </xs:element>

    <xs:element name="device" minOccurs="0" maxOccurs="unbounded">
```

```
    <xs:complexType>
        <xs:sequence>
            <xs:choice>
             <xs:element name="id" type="uuid"/>
             <xs:element name="shortID" type="xs:positiveInteger"/>
             <xs:element name="name" type="xs:string"/>
             <xs:element name="ipAddress" type="xs:string"/>
         </xs:choice>
             <xs:element name="virtualSegment" type="idname"
                         maxOccurs="unbounded"/>
     </xs:sequence>
     </xs:complexType>

</xs:element>
    </xs:sequence>
  </xs:complexType>
  </xs:element>
</xs:schema>
```

Get distribution status

Get distribution status. Actual percent-complete progress and predicted end-time are not available.

Definition

```
ipsProfileMgmt/distributionStatus
```

Parameters

Parameter	Type	Description
distribution/id	string	Internal ID assigned to the distribution session.
device/id	string	Internal ID assigned to the device.
device/shortID	integer	Internal number assigned to the device.
device/name	string	Device name.
device/ipAddress	string	Device IP address.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/ipsProfileMgmt/
distributionStatus?"
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:simpleType name="uuid">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]
                {4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:element name="distributions">
<xs:complexType>
    <xs:sequence>
        <xs:element name="distribution" minOccurs="1"
                maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
          <xs:element name="device" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                  <xs:choice>
                      <xs:element name="name" type="xs:string"/>
                      <xs:element name="id" type="uuid"/>
                      <xs:element name="shortID" type="xs:positiveInteger"/>
                      <xs:element name="ipAddress" type="xs:string"/>
              </xs:choice>
          </xs:complexType>
     </xs:element>
 </xs:sequence>
<xs:attribute name="id" type="uuid"/>
 </xs:complexType>
  </xs:element>
      </xs:sequence>
         </xs:complexType>
     </xs:element>
</xs:schema>
```

Create a traffic management filter

Create a traffic management filter.

Definition

```
ipsProfileMgmt/createTrafficMgmt
```

Parameters

Parameter	Type	Description
name	string	Required. Name of the traffic management filter; must be unique for each profile.
profile	string	Required. Name of the profile that contains the traffic management filter; the profile must already exist on the SMS.
srcAddr	IP address	Required. Source address for the filter. Valid values: any or IP address.
destAddr	IP address	Required. Destination address for the filter. Valid values: any or an IP address.
direction	string	Optional. Filter direction. If a parameter is not specified, the default value is used. • AtoB (default) • BtoA • both
action	string	Optional. Filter action set. For rate limiting, use the rate-limit parameter. • block (default) • allow • trust
rate-limit	string	Optional. Filter rate-limit action set; the action set must already be defined and set to rate-limit.

Parameter	Type	Description
protocol	string	Optional. Filter protocol. • ip (default) • ipv6 • tcp • tcpv6 • udp • udpv6 • icmp • icmpv6
protocol/ip/ ipFragments	boolean	Optional. Applies only to IP fragments; valid only when protocol is IP. • false (default) • true
protocol/icmp/ icmptype	integer	Optional. ICMP type; valid only when protocol is ICMP. • 0-255 (default is 0)
protocol/icmp/ icmpcode	integer	Optional. ICMP code; valid only when protocol is ICMP. • 0-255 (0 is default)
protocol/tcp or udp/ srcPort	integer	Optional. Source port; valid only when protocol is TCP or UDP. • any • 0-65535 (default is 0, and all ports)
protocol/tcp or udp/ destPort	integer	Optional. Destination port; valid only when protocol is TCP or UDP. • any • 0-65535 (default is 0, and all ports)

Parameter	Type	Description
position	integer	Optional. Filter precedence. • 0-200 (default is 0, which uses the lowest unused value)
comment	string	Optional. Filter comments.
state	boolean	Optional. Filter state. • enable (default) • disable

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/ipsProfileMgmt/
createTrafficMgmt?name=<name>&profile=<p_name>&srcAddr=<ip_address>&destAddr=<ip_address>"
```

Delete a traffic management filter

Delete a traffic management filter.

Definition

```
ipsProfileMgmt/deleteTrafficMgmt
```

Parameters

Parameter	Type	Description
name	string	Required. Name of the traffic management filter to be deleted; must be unique for each profile.
profile	string	Required. Name of the profile that contains the traffic management filter; the profile must already exist on the SMS.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/ipsProfileMgmt/
deleteTrafficMgmt?name=<name_1>,<name_2>,<name_3>&profile=<p_name>"
```

Get current filter override settings (with POST)

Retrieve current filter override settings for a profile. This is a POST request, and requires you to provide an XML file that identifies the profile and the filter(s). When the SMS receives a current filter settings service request, it:

- Validates the filter ID using the DV metadata.
- Finds the category the filter ID belongs to.
- Finds the setting of the category from the profile specified by the Profile ID and version.
- Sets the filter ID in the response XML.

The setting of a given filter might be changed by IPS administrators. The changes are defined in the POLICY response XML defined by the existing service interface.

Definition

```
ipsProfileMgmt/getFilters
```

Request Schema

The Remote Profile Management API uses the following XML schema for current filter settings status requests.

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:simpleType name="uuid">
    <xs:restriction base="xs:string">
     <xs:pattern value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]
      {4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:element name="getFilters">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="profile">
            <xs:complexType>
              <xs:attribute name="id" type="uuid"/>
              <xs:attribute name="name" type="xs:string"/>
           </xs:complexType>
         </xs:element>
         <xs:element name="filter" maxOccurs="unbounded">
```

```
          <xs:complexType>
           <xs:sequence>
            <xs:element name="number" type="xs:positiveInteger"minOccurs="0"/>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="signature-id" type="uuid" minOccurs="0"/>
            <xs:element name="policy-id" type="uuid" minOccurs="0"/>
          </xs:sequence>
         </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Request parameters

Parameter	Type	Description
profile	string	Empty element with these attributes: • id • name • version
number	integer	Unique filter number.
name	string	Filter name.
signature-id	string	Internally assigned filter ID.
policy-id	string	Internally assigned policy ID.

Response parameters

Parameter	Type	Description
profile	string	Empty element with these attributes: • id • name • version
name	string	Filter name.

Parameter	Type	Description
policy-id	string	Internally assigned policy ID.
version	integer	IPS TOS version for the filter.
locked	boolean	Indicates whether the filter is locked. You cannot remotely change a locked filter.
useParent	boolean	Indicates whether the action set on the filter is inherited from a parent profile.
comment	string	User comments.
severity	string	Filter severity. • Low • Minor • Major • Critical
enabled	boolean	• enabled • disabled
actionset	string	• refid • name
control	string	Controlling element of the filter `actionset` setting. • category: controlled by the category action set. • filter: controlled by the overriding default action set.
afc	boolean	Indicates whether the filter is managed by the Adaptive Filter Configuration (AFC). If a filter is managed by AFC, then the filter is automatically disabled when the IPS device is under heavy load and the given filter is triggered without an actual filter match.
policyGroup		Profile group identified by a `refid`, expressed in UUID format. This parameter is never used by a filter.

Parameter	Type	Description
trigger		Trigger frequency detection parameter for the filter. Used only for scan/sweep filters. • threshold: specify the number of filter triggers. • timeout: specify the time period under which the number of triggers are being counted (in seconds).
capability		Element with a device name attribute having these child elements: • enabled • actionset: specifies the filter setting. • refid: maps to the action set ID for the capability.

Example

```
curl -X POST -k --header "X-SMS-API-KEY: <string>" --form name=@getFilters.xml
https://<sms_server>/ipsProfileMgmt/getFilters
```

The following sample shows how the getFilters XML file is composed according to the schema. You must provide values for the profile name and at least one of the filter search terms: number, signature ID, policy ID, or name.

```
<?xml version="1.0"?>
<getFilters>
    <profile name="Default"/>
    <filter>
        <number>3295</number>
    </filter>
    <filter>
        <signature-id>00000001-0001-0001-0001-000000000027</signature-id>
    </filter>
    <filter>
        <policy-id>00000002-0002-0002-0002-000000000051</policy-id>
    </filter>
    <filter>
        <name>0050: IP Options: Unknown Code</name>
    </filter>
</getFilters>
```

Response Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs"http://www.w3.org/2001/XMLSchema">
 <xs:simpleType name="uuid">
   <xs:restriction base="xs:string">
    <xs:pattern
value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
  <xs:restriction>
 <xs:simpleType>
  <xs:element name="filters">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="profile">
  <xs:compleType>
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="id" type="xs:string"/>
      <xs:attribute name="version" type="xs:string"/>
  </xs:complexType>
           </xs:element>
    <xs:element name="filter" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
         <xs:element name="name" type="xs:string"/>
            <xs:element name="policy-id" type="uuid"/>
            <xs:element name="version" type="xs:string"/>
            <xs:element name="locked" type="xs:boolean"/>
            <xs:element name="useParent" type="xs:boolean"/>
            <xs:element name="comment" type="xs:string" minOccurs="0"/>
            <xs:element name="description" type="xs:string" minOccurs="0"/>
            <xs:element name="severity" minOccurs="0">
        <xs:simpleType>
         <xs:restriction base="xs:string">
              <xs:enumeration value="Low"/>
              <xs:enumeration value="Minor"/>
              <xs:enumeration value="Major"/>
              <xs:enumeration value="Critical"/>
         </xs:restriction>
        </xs:simpleType>
      </xs:element>
        <xs:element name="enabled" type="xs:boolean"/>
        <xs:element name="actionset" minOccurs="0">
       <xs:complexType>
        <xs:attribute name="refid" type="uuid"/>
        <xs:attribute name="name" type="xs:string"/>
       </xs:complexType>
     </xs:element>
    <xs:element name="control">
     <xs:simpleType>
         <xs:restriction base="xs:string">
```

```
        <xs:enumeration value="Category"/>
        <xs:enumeration value="Filter"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:element>
  <xs:element name="afc" type="xs:boolean"/>
  <xs:element name="policyGroup" minOccurs="0">
    <xs:complexType>
      <xs:attribute name="refid" type="uuid"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="trigger" minOccurs="0">
   <xs:complexType>
     <xs:attribute name="threshold">
       <xs:simpleType>
         <xs:restriction base="xs:integer">
           <xs:minInclusive value="2"/>
           <xs:maxInclusive value="10000"/>
         </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="timeout">
      <xs:simpleType>
        <xs:restriction base="xs:long">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="999999"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="capability" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="enabled" type="xs:boolean"/>
       <xs:element name="actionset" minOccurs="0">
         <xs:complexType>
              <xs:attribute name="refid" type="uuid"/>
              <xs:attribute name="name" type="xs:string"/>
         </xs:complexType>
        </xs:element>
       </xs:sequence>
           <xs:attribute name="name" type="xs:string"/>
          </xs:complexType>
         </xs:element>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:complexType>
```

```
    </xs:element>
  </xs:schema>
```

Get all filter settings (with GET)

Retrieve all filter settings for a profile. The All Filter Settings Request is a GET request. When the SMS receives an all filter settings service request, it:

- Validates that the profile exists.
- Finds the list of filter IDs that can be overriden.
- Finds the the category that the filter ID belongs to.
- Finds the setting of the category from the profile specified by the Profile ID and version.
- Sets the filter ID in the response XML.

The setting of a given filter might be changed by IPS administrators. The changes are defined in the POLICY response XML defined by the existing service interface.

Note

This API might take longer to return results depending on the workload of the SMS.

Definition

```
ipsProfileMgmt/getFilters
```

Parameters

Parameter	Type	Description
profileName	string	Required. Profile name that contains filter settings to retrieve.

Example

```
curl -k --header "X-SMS-API-KEY: <string>"
     "https://<sms_server>/ipsProfileMgmt/getFilters?profileName=Default"
```

Response

The response for this API resembles the *response to the Get current filter override settings API on page 2-17*, except that this API returns all the profile filter settings, not just the filter overrride settings.

Update filter settings

Apply policy changes, such as profile and filter details, to a profile. This API setting is a POST request. The posted XML request identifies both the filter to be updated and the filter detail that is being set. Note that when making update filter settings requests:

- Filters can be identified by name, number, policy-ID, or signature-ID, but only one of these identifications are required inside a filter schema type.
- If a filter is not identified within the filter XML tags, then that filter update section will fail.
- If an XML request contains more than one update and one of them fails, the other updates can still succeed and get applied successfully.

Definition

```
ipsProfileMgmt/setFilters
```

Parameters

Parameter	Type	Description
actionset	string	• refid • name

Parameter	Type	Description
afc	boolean	Indicates whether the filter is managed by the IPS Adaptive Filter Configuration (AFC). If a filter is managed by AFC, then the filter will be automatically disabled when the device is under heavy load and the given filter is being triggered without actual filter match.
comment	string	Filter comments.
control	string	• category: action set is controlled by the category action set. • filter: action set is controlled by overriding the default action set.
enabled	boolean	• enabled • disabled
filter		Read-only parent element.
locked	boolean	Boolean variable indicating if the filter is locked. Locked filters cannot be remotely changed.
number	integer	Read-only internal assigned number for the filter.
name	string	Read-only filter name.
policy-id	string	Read-only internal ID assigned to the policy, expressed in UUID format.
profile	string	• id • name
signature-id	string	Read-only internal ID assigned to the filter, expressed in UUID format.

Parameter	Type	Description
trigger		Trigger frequency detection parameter for the filter. Used only for scan/sweep filters. • threshold: specify the number of filter triggers. • timeout: specify the time period under which the number of triggers are being counted (in seconds).
useParent	boolean	Indicates whether the action set setting for the filter is inherited from a parent profile.

Example

```
curl -X POST -k --header "X-SMS-API-KEY: <string>" -form name=@setfilters.xml
       "https://<sms_server>/ipsProfileMgmt/setFilters?"
```

Request Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="uuid">
    <xs:restriction base="xs:string">
<xs:pattern value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="setFilters">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="profile">
          <xs:complexType>
            <xs:attribute name="name" type="xs:string"/>
            <xs:attribute name="id" type="uuid"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="filter" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:choice>
                <xs:element name="policy-id" type="uuid"/>
                <xs:element name="signature-id" type="uuid"/>
                <xs:element name="number" type="xs:positiveInteger"/>
                <xs:element name="name" type="xs:string"/>
              </xs:choice>
              <xs:element name="locked" type="xs:boolean" minOccurs="0"/>
```

```
            <xs:element name="comment" type="xs:string" minOccurs="0"/>
            <xs:element name="control" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="Category"/>
                  <xs:enumeration value="Filter"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="actionset" minOccurs="0">
              <xs:complexType>
                <xs:attribute name="refid" type="uuid"/>
                <xs:attribute name="name" type="xs:string"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="enabled" type="xs:boolean" minOccurs="0"/>
            <xs:element name="afc" type="xs:boolean" minOccurs="0"/>
            <xs:element name="useParent" type="xs:boolean" minOccurs="0"/>
            <xs:element name="trigger" minOccurs="0">
              <xs:complexType>
                <xs:attribute name="threshold">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="2"/>
                      <xs:maxInclusive value="10000"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="timeout">
                  <xs:simpleType>
                    <xs:restriction base="xs:long">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="999999"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

The following example shows an update filter serttings request XML:

```
<setFilters>
        <profile name="ProfileNameHere"/>
        <filter>
```

```
                <number>7001</number>
                <actionset name="Block + Notify"/>
                <trigger threshold="10" timeout="5000"/>
        </filter>
        <filter>
                <number>3295</number>
                <actionset name="Block + Notify"/>
         </filter>
         <filter>
                <signature-id>00000001-0001-0001-0001-000000000027</signature-id>
                <enabled>false</enabled>
         </filter>
         <filter>
                <policy-id>00000002-0002-0002-0002-000000000051</policy-id>
                <comment>this is a comment</comment>
         </filter>
         <filter>
                <name>0050: IP Options: Unknown Code</name>
                <actionset refid="57ec4769-ca05-4dc5-8e79-a34c182adc48"/>
         </filter>
</setFilters>
```

Update Filters Settings XML Response

Response Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs"http://www.w3.org/2001/XMLSchema">
 <xs:simpleType name="uuid">
   <xs:restriction base="xs:string">
    <xs:pattern
value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
  <xs:restriction>
 <xs:simpleType>
  <xs:element name="filters">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="profile">
  <xs:compleType>
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="id" type="xs:string"/>
      <xs:attribute name="version" type="xs:string"/>
  </xs:complexType>
            </xs:element>
    <xs:element name="filter" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
         <xs:element name="name" type="xs:string"/>
            <xs:element name="policy-id" type="uuid"/>
            <xs:element name="version" type="xs:string"/>
```

```
          <xs:element name="locked" type="xs:boolean"/>
          <xs:element name="useParent" type="xs:boolean"/>
          <xs:element name="comment" type="xs:string" minOccurs="0"/>
          <xs:element name="description" type="xs:string" minOccurs="0"/>
          <xs:element name="severity" minOccurs="0">
       <xs:simpleType>
        <xs:restriction base="xs:string">
             <xs:enumeration value="Low"/>
             <xs:enumeration value="Minor"/>
             <xs:enumeration value="Major"/>
             <xs:enumeration value="Critical"/>
        </xs:restriction>
       </xs:simpleType>
     </xs:element>
       <xs:element name="enabled" type="xs:boolean"/>
       <xs:element name="actionset" minOccurs="0">
      <xs:complexType>
       <xs:attribute name="refid" type="uuid"/>
       <xs:attribute name="name" type="xs:string"/>
      </xs:complexType>
    </xs:element>
  <xs:element name="control">
   <xs:simpleType>
       <xs:restriction base="xs:string">
       <xs:enumeration value="Category"/>
       <xs:enumeration value="Filter"/>
   </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="afc" type="xs:boolean"/>
<xs:element name="policyGroup" minOccurs="0">
  <xs:complexType>
    <xs:attribute name="refid" type="uuid"/>
 </xs:complexType>
</xs:element>
<xs:element name="trigger" minOccurs="0">
 <xs:complexType>
   <xs:attribute name="threshold">
     <xs:simpleType>
       <xs:restriction base="xs:integer">
         <xs:minInclusive value="2"/>
         <xs:maxInclusive value="10000"/>
       </xs:restriction>
     </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="timeout">
    <xs:simpleType>
      <xs:restriction base="xs:long">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="999999"/>
      </xs:restriction>
```

```
      </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="capability" minOccurs="0" maxOccurs="unbounded">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="enabled" type="xs:boolean"/>
       <xs:element name="actionset" minOccurs="0">
         <xs:complexType>
              <xs:attribute name="refid" type="uuid"/>
              <xs:attribute name="name" type="xs:string"/>
         </xs:complexType>
        </xs:element>
       </xs:sequence>
           <xs:attribute name="name" type="xs:string"/>
          </xs:complexType>
         </xs:element>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:schema>
```

Get Digital Vaccine information

Get the active Digital Vaccine and all Digital Vaccines on the SMS.

Definition

```
ipsProfileMgmt/dvInfo
```

Parameters

Parameter	Type	Description
request	string	Required. • active • all

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/ipsProfileMgmt/dvInfo?"
```

Chapter 3

Device administration

Use this API to retrieve the Layer-2 Fallback status, and to place a device or device group into or out of Layer-2 Fallback.

This API includes:

- *Get fallback status on page 3-2*
- *Set fallback status on page 3-2*

Get fallback status

Retrieve the Layer-2 Fallback status for any current device or device group on the SMS.

Definition

```
deviceAdmin/getFallback
```

Parameters

Parameter	Type	Description
deviceName	string	Required. Device that will return the Layer-2 Fallback status.
deviceGroupName	string	Optional. Device group name that will return a comma-delimitated list that shows the Layer-2 Fallback status for each device in the device group.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/deviceAdmin/
getFallback?deviceGroupName=exampleDeviceGroupName"
```

Set fallback status

Place a device or device group into or out of Layer-2 Fallback.

Definition

```
deviceAdmin/setFallback
```

Parameters

Parameter	Type	Description
deviceName	string	Required. Name of the device that will be put into or out of Layer-2 Fallback status.

Parameter	Type	Description
deviceGroupName	string	Optional. Comma-delimitated list that contains the names of the devices within the device group that will be put into or out of Layer-2 Fallback.
L2FB	boolean	Required. Represents the Layer-2 Fallback status that the device or device group will be set to. • true • false

Example

```
curl -X POST -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/deviceAdmin/setFallback?
deviceName=exampleTpsDevice&L2FB=true"
```

Chapter 4

SMS administration

Use this API to create a backup the SMS database, and to retrieve SMS software version information.

This API includes:

- *Backup SMS database on page 4-2*
- *SMS software version on page 4-3*
- *Restore backup file on page 4-4*

Backup SMS database

Create a backup of the SMS database.

Definition

```
smsAdmin/backup
```

Parameters

Parameter	Type	Description
type	string	Destination type. • smb • nfs • scp • sftp • sms (Stored locally on the SMS. Only one backup allowed at a time.)
location	string	Destination path for backup file. Does not apply for `type/sms`.
username	string	Type-specific username. Required for `type/smb`, `scp, or sftp`.
password	string	Type-specific password. Required for `type/smb`, `scp, or sftp .`
domain	string	Type-specific domain; only used for destination type `smb`
tos	integer	Number of most recent TOS packages to include. Default is 0.
dv	integer	Number of most recent DV packages to include. Default is 1.

Parameter	Type	Description
events	boolean	Whether to include events data. • true • false (default)
sslPrivateKeys	boolean	Whether to include SSL private key. • true • false (default)
notify	boolean	Whether to send email notifications when a backup has completed or failed. • true (default) • false
timestamp	boolean	Whether to include the timestamp. • true (default) • false
encryptionPass	string	Encrypt backup using supplied password. Default is `null`, do not encrypt.

Example

```
curl -k --header "X-SMS-API-KEY: <string>""https://<sms_server>/smsAdmin/backup?
type=<smb>&location=<//198.51.100.100/backups/sms.bak>&username=<smb_user>&
password=<smb_pwd>&domain=<dom00>&tos=<1>&dv=<1>&events=<false>&notify=<false>&
timestampName=<true>"
```

SMS software version

Retrieve the SMS software version.

Definition

```
smsAdmin/info
```

Parameters

Parameter	Type	Description
request	string	Returns a version number.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/smsAdmin/info?request=version"
```

Restore backup file

Restore the backup file.

Definition

```
smsAdmin/restore
```

Parameters

Parameter	Type	Description
encryptionPassword	string	Password. Default is `null`.
restoreAdminSetting	boolean	Whether to restore the backup file. • true • false (default)

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/smsAdmin/restore?
encryptionPassword=<password>&restoreAdminSetting=true"
```

Chapter 5

Virtual segment management

Use this API to create, update, and delete virtual segments. You can retrieve a list of virtual segments from a device.

- You can create a virtual segment that does not initially contain a physical segment.
- IPS devices with virtual segments that were configured locally on an IPS device and then added to the SMS are merged to the global virtual segment listing.
- A virtual segment must include at least one VLAN ID, source IP address, or destination IP address.
- Named resources must already exist on the SMS.

This API includes:

- *Create a virtual segment on page 5-3*
- *Update a virtual segment on page 5-3*
- *Delete a virtual segment on page 5-4*
- *Get list of virtual segments on page 5-4*

Response codes

The API captures a response code for virtual segment operations.

Web API response code	HTTP response code	Description
0	200	Successful completion.
100	401	Authentication error.
200	400	Missing parameter error.
205	400	Operation error.
300	400	Input XML file error.
305	500	Output result file error.
310	400	Validation error.
320	400	Resource error.
500	500	Unexpected error.

Create a virtual segment

Add a virtual segment to the SMS database by using a file.

Definition

```
virtualSegment/create
```

Parameters

Parameter	Type	Description
file		Name of the file that contains the virtual segment XML.

Example

```
curl -v -k --header "X-SMS-API-KEY: <string>" -F "file=@Name.xml"
"https://<sms_server>/virtualSegment/create?"
```

Update a virtual segment

Update a virtual segment on the SMS by using a file.

Definition

```
virtualSegment/update
```

Parameters

Parameter	Type	Description
file		Name of the file that contains the virtual segment XML.
vs	string	Virtual segment name.

Example

```
curl -v -k --header "X-SMS-API-KEY: <string>" -F "file=@update.xml"
"https://<sms_server>/virtualSegment/update?&vs=<name>"
```

Delete a virtual segment

Delete a virtual segment.

Definition

```
virtualSegment/delete
```

Parameters

Parameter	Type	Description
vs		Name of the virtual segment to be deleted from the device and from the SMS.

Example

```
curl -v -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/virtualsegment/delete?
&vs="NamedResourceExample"
```

Get list of virtual segments

Retrieve a list of all of the virtual segments on the SMS in XML format. The request also returns the device NAME from the DEVICE table. See *DEVICE table on page 11-4*.

Note

Use the following link examples to download the XML schema from the SMS: "https://<sms_ip_or_hostname>/xsds/VirtualSegment.xsd" or "https://<sms_ip_or_hostname>/xsds/sms/response/xsd".

Definition

```
virtualsegment/get
```

Parameters

Parameter	Type	Description
name	string	Name of the virtual segment
description (optional)	string	Description for the virtual segment
virtualSegPosition		Indicates where in the list virtual segment is placed. You define the priority order for a virtual segment so that any overlapping definitions are resolved. Attempting to define an overlapping virtual segment on a device which does not allow it will produce an error.
virtualSegPosition/positionType	ORDINAL_POSITION, FIRST, LAST	Attribute; must be one of the three values
virtualSegPosition/ordinalPosition	positive integer	Must be provided when positionType is ORDINAL_POSITION
vlanIdList (optional)		Used to assign a list of VLAN IDs, and/or VLAN ranges or a named object referencing a named VLAN group
vlanIdList/vlanList		Used when assigning a list of VLAN IDs and/or VLAN ranges to the virtual segment
vlanIdList/vlanList/vlan		Single element for either a VLAN ID or VLAN range
vlanIdList/vlanList/vlan/vlanID	integer (1 to 4094)	VLAN ID

Parameter	Type	Description
vlanIdList/vlanList/vlan/vlanID/ vlanRange		Element containing a VLAN range
vlanIdList/vlanList/vlan/vlanID/ vlanRange/start	integer (1 to 4094)	VLAN ID start of the range
vlanIdList/vlanList/vlan/vlanID/ vlanRange/end	integer (1 to 4094)	VLAN ID end of the range
vlanIdList/namedVlanGroup	string	Named VLAN group identifier
sourceAddressList (optional)		Used to assign a list of IP addresses and/or IP address blocks or a named object referencing a named address group for the source address
sourceAddressList/cidrList		Used when providing a list of IP addresses and/or IP address blocks
sourceAddressList/cidrList/cidr		IP address or IP address block
sourceAddressList/namedAddrGroup	string	Named address group identifier
destinationAddressList (optional)		Used to assign a list of IP addresses, and/or IP address blocks or a named object referencing a named address group for the destination address
destinationAddressList/cidrList		Used when providing a list of IP addresses and/or IP address blocks
destinationAddressList/cidrList/cidr		IP address or IP address block
destinationAddressList/ namedAddrGroup	string	Named address group identifier
segmentGroup		Used when assigning a virtual segment to a segment group

Parameter	Type	Description
segmentGroup/segmentGroupID		Identifier element for the segment group
segmentGroup/segmentGroupID/name	string	Name of the segment group
segmentGroup/segmentGroupID/id	string	ID of the segment group
physicalSegments (optional)		Used for assigning the virtual segment to one or more segments on one or more devices
physicalSegments/physicalSegment		Identifies the device and the segments to assign the virtual segment to
physicalSegments/physicalSegment/device		Identifies the device
physicalSegments/physicalSegment/device/uuid	string	UUID of the device
physicalSegments/physicalSegment/device/shortID	positive integer	Short ID of the device
physicalSegments/physicalSegment/device/name	string	Name of the device
physicalSegments/physicalSegment/device/ipAddress	string	IP Address of the device
physicalSegments/physicalSegment/segmentNameList		Element containing a list of the segment names
physicalSegments/physicalSegment/segmentNameList/segmentNames	string	Name of the segment

Example

```
curl -v -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/virtualsegment/
get?"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >


<xs:simpleType name="uuid">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]
            {4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="vs_name">
    <xs:restriction base="xs:string">
     <xs:maxLength value="127"/>
 </xs:restriction>
</xs:simpleType>

<xs:simpleType name="vlan_Constraint">
    <xs:restriction base="xs:int">
     <xs:minInclusive value="0"/>
     <xs:maxInclusive value="4095"/>
 </xs:restriction>
</xs:simpleType>

<xs:simpleType name="vs_description">
 <xs:restriction base="xs:string">
  <xs:maxLength value="250"/>
  </xs:restriction>
</xs:simpleType>


<xs:simpleType name="positionType">
    <xs:restriction base="xs:string">
     <xs:annotation>
      <xs:documentation>Placement of the object in the list, first, last,
               or somewhere in between</xs:documentation>
  </xs:annotation>
      <xs:enumeration value="FIRST" />
      <xs:enumeration value="LAST" />
      <xs:enumeration value="ORDINAL_POSITION" />
 </xs:restriction>
</xs:simpleType>

<xs:complexType name="messageList">
    <xs:sequence>
     <xs:element type="xs:string" name="message"
           minOccurs="1"
```

```
            maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>


<xs:complexType name="deviceResult">
    <xs:all>
     <xs:element name="device" type="deviceType"/>
  <xs:element name="success" type="xs:boolean"/>
  <xs:element name="messages" type="messageList"
            minOccurs="0" maxOccurs="1"/>
 </xs:all>
</xs:complexType>

<xs:complexType name="deviceResultList">
    <xs:sequence>
     <xs:element type="deviceResult" name="deviceResult"
            minOccurs="1"    maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="rangeType">
    <xs:all>
     <xs:annotation>
      <xs:documentation>Range (i.e. 5 - 90)</xs:documentation>
   </xs:annotation>
       <xs:element type="vlan_Constraint" name="start"/>
    <xs:element type="vlan_Constraint" name="end"/>
 </xs:all>
</xs:complexType>


<xs:complexType name="idName">
    <xs:choice>
     <xs:element name="id"   type="xs:string"/>
  <xs:element name="name" type="xs:string"/>
 </xs:choice>
</xs:complexType>

<xs:complexType name="cidrListType">
    <xs:sequence>
     <xs:element type="xs:string" name="cidr" maxOccurs="unbounded">
      <xs:annotation>
       <xs:documentation>1 or more repetitions:1
                   or more repetitions:</xs:documentation>
   </xs:annotation>
  </xs:element>
 </xs:sequence>
</xs:complexType>

<xs:element name="virtualSegment"  type="virtualSegmentType"
```

```
        nillable="false" />
    <xs:element name="virtualSegmentList" type="virtualSegmentListType"
        nillable="false"/>

 <xs:complexType name="segmentGroupType">
     <xs:sequence>
      <xs:element type="segmentGroupIDType" name="segmentGroupID"/>
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="sourceAddressListType">
     <xs:choice>
      <xs:annotation>
       <xs:documentation>You have a CHOICE of the next
                    2 items at this level</xs:documentation>
   </xs:annotation>
       <xs:element type="cidrListType" name="cidrList">
        </xs:element>
         <xs:element type="xs:string" name="namedAddrGroup">
     </xs:element>
  </xs:choice>
</xs:complexType>

<xs:complexType name="vlanIdListType">
    <xs:sequence>
     <xs:annotation>
      <xs:documentation>VLAN can either be a 1 named resource
                or a list of integer/ranges</xs:documentation>
  </xs:annotation>
  <xs:choice>
      <xs:element type="vlanListType" name="vlanList" >
   </xs:element>
   <xs:element type="xs:string" name="namedVlanGroup">
   </xs:element>
  </xs:choice>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="virtualSegmentType" >
        <xs:annotation>
            <xs:documentation>Definition of the virtual segment</xs:documentation>
      <xs:documentation>Any optional fields should be omitted,
                no empty elements</xs:documentation>
      <xs:documentation>Required: Name, segmentGroup, one,
                two or all of: [vlanIdList,sourceAddressList,
                destinationAddressList]</xs:documentation>
      <xs:documentation>Optional: description, and physicalSegments.
                If physicalSegments is not provided no devices will be updated with the
                virtual segment</xs:documentation>
        </xs:annotation>
```

```
    <xs:all>
     <xs:element type="vs_name" name="name" />
  <xs:element type="vs_description" name="description"
            nillable="false" minOccurs="0"/>
  <xs:element type="virtualSegPositionType" name="virtualSegPosition"/>
  <xs:element type="vlanIdListType" name="vlanIdList"
            nillable="false" minOccurs="0">
  </xs:element>
  <xs:element type="sourceAddressListType" name="sourceAddressList"
            nillable="false" minOccurs="0">
  </xs:element>
  <xs:element type="destinationAddressListType" name="destinationAddressList"
            nillable="false" minOccurs="0">
  </xs:element>
  <xs:element type="segmentGroupType" name="segmentGroup" />
  <xs:element type="physicalSegmentsType" name="physicalSegments"
            nillable="false" minOccurs="0">
  </xs:element>
 </xs:all>
</xs:complexType>

<xs:complexType name="virtualSegmentListType">
    <xs:sequence>
     <xs:element type="virtualSegmentType" name="virtualSegment"
            nillable="false" minOccurs="1" maxOccurs="unbounded">
  </xs:element>
 </xs:sequence>
</xs:complexType>
<xs:complexType name="destinationAddressListType">
    <xs:choice>
     <xs:annotation>
      <xs:documentation>You have a CHOICE of the next
                2 items at this level</xs:documentation>
  </xs:annotation>
  <xs:element type="cidrListType" name="cidrList">
  </xs:element>
  <xs:element type="xs:string" name="namedAddrGroup">
  </xs:element>
 </xs:choice>
</xs:complexType>

<xs:complexType name="segmentGroupIDType">
    <xs:choice>
     <xs:annotation>
      <xs:documentation>You have a CHOICE of the next
                2 items at this level</xs:documentation>
  </xs:annotation>
  <xs:element type="xs:string" name="id">
  </xs:element>
  <xs:element type="xs:string" name="name"/>
```

```
 </xs:choice>
</xs:complexType>

<xs:complexType name="virtualSegPositionType">
    <xs:sequence>
     <xs:element  nillable="true" type="xs:positiveInteger"
           minOccurs="0" name="ordinalPosition">
  </xs:element>
 </xs:sequence>
 <xs:attribute type="positionType" name="positionType"/>
</xs:complexType>

<xs:complexType name="deviceType">
    <xs:choice>
     <xs:annotation>
      <xs:documentation>You have a CHOICE of the next
                4 items at this level</xs:documentation>
  </xs:annotation>
  <xs:element type="uuid" name="id"/>
  <xs:element type="xs:positiveInteger" name="shortID"/>
  <xs:element type="xs:string" name="name"/>
  <xs:element type="xs:string" name="ipAddress"/>
 </xs:choice>
</xs:complexType>


<xs:complexType name="segmentNameListType">
    <xs:sequence>
     <xs:element type="xs:string" name="segmentNames"
           minOccurs="1" maxOccurs="unbounded">
      <xs:annotation>
       <xs:documentation>1 or more device segment names</xs:documentation>
   </xs:annotation>
  </xs:element>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="vlanIdRangeType" >
    <xs:choice>
     <xs:element name="vlanID" type="vlan_Constraint"/>
  <xs:element name="vlanRange" type="rangeType"/>
 </xs:choice>
</xs:complexType>


<xs:complexType name="vlanListType" >
    <xs:sequence>
     <xs:element name="vlan" type="vlanIdRangeType"
           minOccurs="1" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

```
    <xs:complexType name="physicalSegmentsType">
        <xs:sequence>
         <xs:annotation>
          <xs:documentation>1 or more repetitions:</xs:documentation>
      </xs:annotation>
      <xs:element type="deviceSegmentsType" name="physicalSegment"
                maxOccurs="unbounded">
      </xs:element>
     </xs:sequence>
    </xs:complexType>

    <xs:complexType name="deviceSegmentsType">
        <xs:sequence>
         <xs:element type="deviceType" name="device"/>
      <xs:element type="segmentNameListType" name="segmentNameList"/>
     </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Chapter 6

Reputation database management

Use this API to manage the SMS Reputation database. The following factors can affect performance levels:

- Method used for the Reputation entries submission – `import` or `add`. Use `import` with a large number of entries to reduce the number of distributions.
- Number of files to be imported into the Reputation database and the number of entries in each file.
- Number of entries on the SMS. A bigger reputation database takes longer to copy and distribute, resulting in less frequent distributions. For improved performance, limit the entries in the Reputation database to 6,000,000.
- Number and type of devices that the SMS manages. Newer models load the entries faster. If you have a large number of devices, increase the interval of entry submission so that the SMS is not overloaded with frequent distributions.

Note

Monitor the device distribution queue to identify the appropriate time interval for submitting the Reputation Management API requests in your environment.

This API includes:

- *Import a Reputation entry on page 6-3*
- *Add a Reputation entry on page 6-9*
- *Delete a Reputation entry on page 6-13*
- *Query the Reputation database on page 6-10*

Import a Reputation entry

Post and upload a file with one or more Reputation entries.

- **IP, domain name, and file hash entries** — Import file must be in a comma-separated value (CSV) format with each line representing a Reputation entry without any blank lines. Each line is made up of one or more fields separated by commas. The SMS can upload one file at a time, and each file can contain multiple entries.
- **URL entries** — Import file must be delimited by a pipe (|) instead of a comma with each entry representing *URLs only* or *URLs associated with one or more tags*. Each line is made up of one or more fields separated by pipes. For more information about the URL import guidelines, see the *URL Reputation Filtering Deployment and Best Practices Guide*.
- Comment lines are discarded during import.
- Each request results in a distribution and a sync time to the managed devices.
- For improved performance, limit the number of entries in a file to between 1,000 and 10,000.
- CSV files are imported using HTTP POST requests.

Definition

```
repEntries/import
```

Parameters

Parameter	Type	Description
type	string	Optional. Address type of the Reputation entry. Only one type is allowed within a file. • ipv4 (default) • ipv6 • dns • url • filehash-sha1 • filehash-sha256

Examples

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-ipv4.csv" "https://<sms_server>/repEntries/import?type=ipv4"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-ipv6.csv" "https://<sms_server>/repEntries/import?type=ipv6"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-dns.csv" "https://<sms_server>/repEntries/import?type=dns"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-url.csv" "https://<sms_server>/repEntries/import?type=url"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-sha1.csv" "https://<sms_server>/repEntries/import?type=filehash-sha1"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"
-F "requestFile=@./repEntries-sha256.csv"   "https://<sms_server>/repEntries/import?type=filehash-sha256"
```

Tip

When you import large files through the SMS or a Web API, the SMS uses an import process to queue the files that will be imported. This process begins after the Web API has returned. An HTTP Status code of `200` indicates that a file was queued for import. View the SMS audit log for the results of the import request.

If a file contains duplicate values for the first (address) field, the row with the first of the duplicates found is the one that gets imported. A duplicate record found in a file will not keep the remaining records in the file from being processed.

Syntax rules for import files

Except for files with URL entries, an import file must be in a comma-separated value (CSV) format with each line representing a reputation entry without any blank lines. Each line is made up of one or more fields separated by commas. The import discards any comment lines.

For URL entries, the import file must be delimited by pipe (|) instead of commas, and entries can be URLs only or URLs associated with one or more tags. Each line is made up of one or more fields separated by pipes. For more information about the URL import guidelines, refer to the *URL Reputation Filtering Deployment and Best Practices Guide*.

Use the fields described in the following table for the entries in your file.

Field	Required	Description
Address	Yes	The first field on each line must be the IPv4 address, IPv6 address, domain name, URL, or file hash for that entry. The remaining fields on a line are optional. If present, remaining fields are processed as tag category/tag value pairs. • Only one type of address (IPv4, IPv6, domain name, URL, file hash SHA-1, or file hash SHA-256) can be contained in a file. • File hash CSV files should have either SHA-1 or SHA-256 records, but never both in the same file. The entire file will be expected to have either SHA-1 or SHA-256 file hashes. • SHA-1 files can have entries prefixed with the string `sha1:`. SHA-256 files can have entries prefixed with the string `sha256:`. Prefixing will still not enable a file to mix file hash types. To import both file types, import them in different files. SHA-1 hashes are expected to be 40 characters long in the appropriate range. SHA-256 hashes are expected to be 64 characters long in the appropriate range. • A domain name entry matches any look-ups that contain the specified string. For example, `foo.com` matches `foo.com`, `www.foo.com`, and `images.foo.com`. To specify an exact domain name entry match, enclose the domain name in square brackets. For example, `[foo.com]`. • CIDR values are normalized. Any bits outside the portion of the address specified by the prefix length are changed to zero. For example, `192.168.66.127/24` is stored as `192.168.66.0/24`.

Field	Required	Description
Tag category/tag value pair	No	If the reputation entry within the file does not have tags, the imported entry merges with the values of the existing entry. If the reputation entry within the file does have tags, the imported entry merges and overwrites the values of the existing entry. • Any tag categories in the file must exist on the SMS prior to import. • Tag category/value pairs do not have to be listed in the same order on each line. The entries in the file do not have to list all the tag categories or specify the ones shared with other entries in the file. • Empty pairs of fields are ignored. If a tag category field is empty, an error occurs, and the entry is not imported. If a tag value field is empty, the corresponding tag category is discarded, and the next field of the entry is processed; the net result is equivalent to the tag category not appearing on that line at all. • Except for yes/no tag categories, character case is significant in all tag category names and tag values. • For yes/no tag categories, the text `yes`, regardless of case, denotes a yes value. Any other text is considered a no value. • For list categories, the list values must be separated by `~~~` (three tildes). • A field can be enclosed in double-quotes; this is mandatory when a value contains a comma that should not be treated as a field separator. • To represent a double-quote character within a quoted value, use two double-quotes. For example, `"This comment ""contains"" quotes."`

Import file example

For this example, the following tag categories are defined:

- Country (List)
- Approved (Yes/No)
- Comment (Text)

For the Country tag category, the following tag values are defined:

- China
- Mexico
- United States

The following example shows a file with IPv4 Reputation entries.

```
1.2.3.0/24,Country,United States,Approved,yes
2.3.0.0/16,Country,Mexico,Approved,no
3.4.5.0/24,Approved,yes,Country,China
1.2.3.0/24,Country,United States,Approved,yes,Comment,"This
           comment, contains a comma"
1.2.3.0/24,Country,United States,Approved,yes,Comment,"This
           comment ""contains"" quotes"
2.3.0.0/16
3.4.5.0/24,,,,
```

The following example shows a file with URL Reputation entries. It contains a user-defined tag category, `Source`, which must be created before attempting to import the file.

```
https://www.cnn.com|Source|URL
https://www.google.com|Source|URL
https://www.facebook.com|Source|URL
https://www.trendmicro.com|Source|URL
https://www.ebay.com|Source|URL
https://www.amazon.com|Source|URL
```

The following example shows a file with SHA-1 file hash entries that has no defined tags. Note that SHA-1 file hashes should be exactly 40 characters in

length in the appropriate range. (A file with SHA-256 file hash entries would have hashes 64 characters long in the appropriate character range.)

```
0047ec317c28baa9b6a4b5ee3e8c78abfe2d0c61
0dfc9caaef84a33795541655ae0a0bb76c27f9b3
0fe995653b91dc4760d4b12f1464414b5dbe1747
11bb0c64cccba49ed249b526d9c3563087248625
2b91418b6f9e75231db1f4a8a0046dbf6ec9d9ab
2fed31e9687a49127676ea2f60a1bda38538cca8
4474703263bc73015ba83d39dbd3211baee37b4c
```

Note

When you request back-to-back imports with files that have 10 or less Reputation entries, the SMS groups those entries to use the `add` method instead to reduce the number of distributions.

Add a Reputation entry

Create a Reputation entry.

- Each request can result in a distribution and a sync time to the managed devices.
- For improved performance, send requests in bursts up to 1,000 entries in time intervals that allow distributions to complete in a timely manner.
- This Web API is performed using an HTTP GET request.

Definition

```
repEntries/add
```

Parameters

Only one of the following parameters can be used in the request.

Parameter	Type	Description
ip	IP address	IPv4 or IPv6 address of the Reputation entry.
dns	string	DNS address of the Reputation entry.
url	url	Reputation URL entry.

filehash	File hash string	Prefix the file hash string with either `sha1:` or `sha256:` in order to indicate the type of SHA algorithm the file hash string represents. For example:
		`sha1:b7c74f7fbde596ba87ac98ff4a9c8235d437ebce`
		`sha256:779d0a008cb96a61ad6b075e29dad85d97ccf2d5c528d3292ccb9535079b6b40`

TagData	string	Optional. One or more tag categories and their values. Must be UTF-8 encoded and separated by a comma (,). Reputation entries with a list tag category can include multiple values only when the **Allow Multiple Values?** check box is selected from the **Edit Tag Category** box on the SMS. The list values must be separated by ~~~ (three tildes). `MalwareIpType,malwareSource~~~cncHost`

Example

```
curl -v -k  --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/add?&ip=1.1.1.1&TagData=MalwareIpType,
infectedHost,CreatedDate,%22Jan%2022,%202014%22"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/add?dns=www.baddomain.com"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>" "
https://<sms_server>/repEntries/add?url=http://badurl.com/this/is/nice"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/add?filehash=sha1:b7c74f7fbde596ba87ac98ff4a9c8235d437ebce"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/add?filehash=sha256:1c17d07a2762f6baab41004b57eec113
774bc3f745d4c1d64c0fe464720a2126&TagData=Color,Blue"
```

Query the Reputation database

Search the Reputation database for one or more user Reputation entries.

- Specify up to 10,000 entries in a single request.
- The SMS returns all matching entries in the query in UTF-8 encoding.
- Returned entries are ordered from lowest to highest address, regardless of the order in which they are specified in the query.
- Each entry is terminated by a newline character.

Definition

```
repEntries/query
```

Parameters

A request can use the ip, dns, url, or filehash parameter, but not a mix of them.

Parameter	Type	Description
ip	IP address	IPv4 or IPv6 address of the Reputation entry.
dns	string	DNS address of the Reputation entry.
url	url	Reputation URL entry.

filehash	File hash string	The file hash string can be a full file hash or a portion of a file hash to query for file hashes that contain the string. You can also query all file hashes for a specific SHA type by using one of the two supported SHA hash prefixes in the query: `sha1:` or `sha256:`.
		
		Prefix the file hash string with either `sha1:` or `sha256:` in order to indicate the type of SHA algorithm the file hash string represents. For example:
		`sha1:b7c74f7fbde596ba87ac98ff4a9c8235d437ebce`
		`sha256:779d0a008cb96a61ad6b075e29dad85d97ccf2d5c528d3292ccb9535079b6b40`

punycode	N/A	Used in conjunction with the dns parameter, punycode converts Unicode characters in domain names to ASCII before displaying them.

Examples

The following example includes multiple IPs in a single query request:

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/query?&ip=1.1.1.1&ip=1.1.1.2"
```

The following example includes a specific URL:

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>//repEntries/query?url=http://badurl.com/this/is/nice"
```

The following example includes a less specific URL so that more records can be matched:

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>//repEntries/query?url=http://badurl.com"
```

The following examples query file hashes:

```
curl -v -k --header "X-SMS-API-KEY: <string>”
"https://<sms_server>/repEntries/query?filehash=b7c74f7fbde596ba87ac98ff4a9c8235d437ebce"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/query?filehash=sha256:1c17d07a2762f6baab41004b57eec1
13774bc3f745d4c1d64c0fe464720a2126 "
```

The following example query will match all SHA-1 file hash entries:

```
curl -v -k --header "X-SMS-API-KEY: <string>”
"https://<sms_server>/repEntries/query?filehash=sha1"
```

Response

All responses will include matched user reputation entries with their tag values.

In most cases, the returned queries will be in a format that, when saved to a file, could be used in an import/POST request, or a delete/POST request. An exception would be where the results return both SHA-1 and SHA-256 file hashes. These would need to be separated before importing, since SHA hash types cannot be combined in the same CSV file.

```
1.1.1.1,AtaHost,myata.device.com,MalwareIpType,infectedHost
1.1.1.2,AtaHost,myata.device.com,ThreatScore,28,MalwareIpType,cncHost~~~infectedHost
```

```
https://www.cnn.com/,Source,URL
https://www.google.com/,Source,URL
```

```
https://www.facebook.com/,Source,URL
https://www.trendmicro.com/,Source,URL
https://www.ebay.com/,Source,URL
https://www.amazon.com/,Source,URL
```

```
sha1:0047ec317c28baa9b6a4b5ee3e8c78abfe2d0c61
sha1:0dfc9caaef84a33795541655ae0a0bb76c27f9b3
sha1:0fe995653b91dc4760d4b12f1464414b5dbe1747
sha1:11bb0c64cccba49ed249b526d9c3563087248625
sha1:2b91418b6f9e75231db1f4a8a0046dbf6ec9d9ab
sha1:2fed31e9687a49127676ea2f60a1bda38538cca8
sha1:4474703263bc73015ba83d39dbd3211baee37b4c
sha1:4634aa9544f9c33d69edde732bb4c9bf91479213
```

Note

DNS queries return data in UTF-8 character encoding so unicode domain names can be displayed.

Delete a Reputation entry

This Web API has an HTTP POST and a GET version.

- The GET request enables you to delete individual user Reputation entries, but it can also be used to delete all user-defined reputation entries, RepDV entries, or the ThreatDV URL package.
- The POST request enables you to use a CSV file to delete multiple user Reputation entries of the same type. This is convenient when you want to use the same CSV file that you used to import the user Reputation entries.
- You cannot set a query string to delete the user-defined Reputation entries. You must specify them individually in the GET request, and you must already have them in a CSV file to use the POST request.
- You can create a query for the records you want to delete, then save the response in a CSV file. After verifying the file contains the records you want deleted, you can use the file in a delete POST request.

- Each request can result in a distribution and a sync time to the managed devices. For optimal performance, when deleting multiple user defined reputation entries, delete the reputation entries with a file.

Definition

```
repEntries/delete
```

Using POST to delete multiple user Reputation entries with a CSV file

Follow these guidelines when using POST requests for deleting entries:

- The POST request can only delete user-defined entries.
- The CSV file can contain user Reputation entries in the same format that was used to import them.
- All user entries defined in the CSV file must be of the same type. Whatever type parameter you specify as a parameter is applied to the entire CSV file posted.

Parameter	Type	Description
type	Any of the supported values	Import a file with Reputation entries to delete on the SMS. This enables you to delete many reputation entries by using a file. Only one of the following types of entry is permitted within a file:IPv4 or IPv6 address of the Reputation entry. • ipv4 (default) • ipv6 • dns • url • filehash-sha1 • filehash-sha256 This parameter applies only to the POST request.

Examples

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-ipv4.csv" "https://<sms_server>/repEntries/delete?type=ipv4"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-ipv6.csv" "https://<sms_server>/repEntries/delete?type=ipv6"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-dns.csv" "https://<sms_server>/repEntries/delete?type=dns"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-url.csv" "https://<sms_server>/repEntries/delete?type=url"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-sha1.csv" "https://<sms_server>/repEntries/delete?type=
filehash-sha1"
```

```
curl -v  -X POST -k  --header "X-SMS-API-KEY: <string>"  -F
"requestFile=@./repEntries-filehash-sha256.csv" "https://<sms_server>/repEntries/delete?type=
filehash-sha256"
```

Using GET to delete Reputation entries (

Follow these guidelines when using GET requests for deleting entries:

- Use the GET request to delete not only user-defined entries but the ThreatDV URL package and all RepDV entries as well.
- You can use the GET request to delete individual user Reputation entries or user-defined entries across all types with a single request.

Parameter	Type	Description
criteria	One of the supported values	Required. This parameter applies only to the GET request. The following values are supported: **all** Deletes all Reputation entries, including user-defined, RepDV, and the ThreatDV URL feed. **user** Deletes all user-defined entries. **repdv** Deletes all RepDV entries. **entry** Deletes specified entries. **threat-url** Deletes the ThreatDV URL package. This parameter applies only to the POST request.

When the preceding criteria parameter is set specifically to entry, the following parameters can also be provided.

Parameter	Type	Description
ip	IP address	IPv4 or IPv6 address of the Reputation entry.
dns	string	Domain name of the Reputation entry.
url	URL	Reputation URL entry.
filehash	A prefixed file hash	An existing file hash entry with either a `sha1:` or `sha256:` prefix.

Examples

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/delete?criteria=entry&ip=1.1.1.1&ip=1.1.1.2&
dns=malware.source1.com&dns=malware.source2.com"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/delete?criteria=entry&url=http://badurl.com/this/is/nice"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/delete?criteria=entry&filehash=sha1:b7c74f7fbde596ba87ac98ff4
a9c8235d437ebce"
```

```
curl -v -k --header "X-SMS-API-KEY: <string>"
"https://<sms_server>/repEntries/delete?criteria=entry&filehash=sha256:1c17d07a2762f6baab41004b57
eec113774bc3f745d4c1d64c0fe464720a2126"
```

Chapter 7

Packet trace

The SMS Packet Trace feature compiles information about packets that have triggered a filter. Packet trace encapsulates the information according to requirements set for the filter in the SMS.

Packet trace options are configured for an action set, and an action set is specified for each filter. Filters are distributed to devices according to profiles. If a filter uses an action set for which packet trace logging is enabled, then you can view the compiled and stored packet trace information for events that triggered the filter.

The SMS saves packet trace information to a PCAP file. Two retrieval options are available for a packet trace:

- *Device-based packet trace on page 7-2*
- *Events-based packet trace on page 7-2*

Device-based packet trace

Device-based packet trace compiles PCAP information for a particular device from the SMS database. For more information, see *DEVICE table on page 11-4*.

Definition

```
pcaps/getByDevice
```

Parameters

Parameter	Type	Description
device/id	string	Internal ID assigned to the device. This is the SHORT_ID for the device.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/pcaps/
getByDevice?deviceId=<SHORT_ID>"
```

Events-based packet trace

To obtain all the PCAP information from the SMS for a group of events, you must know the event IDs. Event IDs are included in data sent to a remote syslog server.

Definition

```
pcaps/getByEventIds
```

Set up event-based packet trace

1. Set up a remote syslog server.
2. Add all the event IDs to a file as a comma separated list. New line breaks are allowed. The result outputs to STDOUT and can be redirected to a file with a '>' operator.

```
curl -k --header "X-SMS-API-KEY: <string>" -F "file=@<path/to/file.txt>"
"https://<sms_server>/pcaps/getByEventIds?"
```

Chapter 8

Responder

Responder is a policy-based service in SMS that reacts to inputs to perform a set of actions. Its reactions, and the set of actions it takes, are based on the Responder policies that have been configured.

By default, no policies can be externally triggered. To enable external triggering, configure the active response policy to allow an SNMP trap or web service to invoke the policy.

This API includes the following:

- *Quarantine on page 8-2*
- *Unquarantine on page 8-3*

Quarantine

Quarantine an IP address and create a response.

Definition

```
quarantine/quarantine
```

Parameters

Parameter	Type	Description
ip	IP address	IP address for the target host. Required to create or close a response.
id	integer	Response History ID that is displayed in the Response History table on the SMS. • To close a response, either IP or ID must be specified.
policy	string	Specific Active Response Policy to implement. • The policy name is case sensitive and must match an existing SMS Active Response policy name. • The **Allow an SNMP Trap or Web Service call to invoke this Policy** initiation setting must be enabled for this policy. • This argument is not necessary to close a response and, if provided, is ignored.
timeout	long	Optional argument to specify the duration of response. • The specified value overrides the default already in the policy. • If no parameter is specified, the timeout value from the policy is used. • This argument is not necessary to close a response and, if provided, is ignored.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/quarantine/
quarantine?ip=<target_ip>&policy=<policy_name>&timeout=<minutes_to_quarantine>"
```

Unquarantine

Unquarantine an IP address and close a response.

Definition

```
quarantine/unquarantine
```

Parameters

Parameter	Type	Description
ip	IP address	IP address for the target host. Required to create or close a response.
id	integer	Response History ID that is displayed in the Response History table on the SMS. • To close a response, either IP or ID must be specified.
policy	string	Specific Active Response Policy to implement. • The policy name is case sensitive and must match an existing SMS Active Response policy name. • The **Allow an SNMP Trap or Web Service call to invoke this Policy** initiation setting must be enabled for this policy. • This argument is not necessary to close a response and, if provided, is ignored.

Parameter	Type	Description
timeout	long	Optional argument to specify the duration of response. • The specified value overrides the default already in the policy. • If no parameter is specified, the timeout value from the policy is used. • This argument is not necessary to close a response and, if provided, is ignored.

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/quarantine/
unquarantine?ip=<target_ip>"
```

Chapter 9

Enterprise Vulnerability Remediation (eVR)

Use this API to import vulnerability scan (eVR) files to the SMS. After you import a vulnerability scan, you review the following on the SMS:

- Vulnerabilities (listed by CVE) that have been discovered in your network.
- Which assets impacted by those vulnerabilities.
- Which DV filters can defend those assets from the discovered vulnerabilities.

This API includes:

- *Import a vulnerability scan on page 9-3*
- *Convert a vulnerability scan on page 9-3*

eVR specifications

The minimum data required for a native SMS-standard vulnerability scan is:

- **IP Address** - (host IP addresses) The maximum number of host IP address and vulnerability combinations that you can import on the SMS is 10 million. When the SMS reaches the maximum limit, it displays an error message, and you must delete vulnerability scans on the SMS before you can import a new scan using this API.

- **CVE IDs** - CVE must be in the format CVE-YYYY-NNNN where YYYY is a 4 digit year and NNNN is a sequence number.
- **Severity** - Vulnerabilities are assigned a severity levels to define the urgency associated with remediating each vulnerability. Rankings are based on a variety of industry standards including CVE.

CSV file specifications

Vulnerability scans must be in a native, comma-separated value (CSV) format before they can be used on the SMS. If you use a supported vulnerability management product, custom converters are available for Qualys®, Nexpose®, and Nessus®.

- The first line in the CSV file must be the column headers for each of the columns.
- Each row after the header must contain the same number of columns that are in the header.
- Each column must be delimited with a comma.
- The value within each column must be wrapped in double quotes; however, embedded double quotes are not permitted ("This is "invalid" data").
- Each row in a CSV file must be less than 65536 bytes.

Import a vulnerability scan

Import a vulnerability scan (eVR) file that is in native SMS-standard format.

Definition

```
vulnscanner/import
```

Parameters

Parameter	Type	Description
vendor	string	Required. Name of the vulnerability management vendor. • Native SMS-standard format: Use `SMS-Standard`. • For other values, see *Convert a vulnerability scan on page 9-3*.
product	string	Required. Product name associated with the vulnerability scanner, and can be any value.
version	string	Required. Version of the vulnerability scanning file format, and can be any value.
runtime	date	Required. • Scan start time and end time, and can be a single date or a date range. • When entering a date range, you must use a forward slash (/) to separate the scan start and scan end dates. • Date format must be yyyy-MM-dd'T'HH:mm:ss.SSS'Z

Examples

```
curl -k --header "X-SMS-API-KEY: <string>" -F "file=@ScanSample.csv"
"https://<sms_server>/vulnscanner/import?&vendor=Example&product=VulnScanner&version=2.2
&runtime=2018-12-15T13:01:15.255Z/"
```

Convert a vulnerability scan

Convert a vulnerability scan (eVR) file that is not in native SMS-standard format to import to the SMS.

Definition

```
vulnscanner/convert
```

Parameters

Parameter	Type	Description
vendor	string	Required. Name of the vulnerability management vendor. • `Nexpose` • `Qualys-CSV` • `Nessus`
product	string	Required. Product name associated with the vulnerability scanner, and can be any value.
version	string	Required. Version of the vulnerability scanning file format, and can be any value.
runtime	date	Required. • Scan start time and end time, and can be a single date or a date range. • When entering a date range, you must use a forward slash (/) to separate the scan start and scan end dates. • The date format must be yyyy-MM-dd'T'HH:mm:ss.SSS'Z

Examples

Import a vulnerability scan (eVR) in the Nexpose format:

```
curl -v -k --header "X-SMS-API-KEY: <string>" -F "file=@vulnScanSampleNexpose.xml"
"https://<sms_server>/vulnscanner/convert?&vendor=Nexpose&product=Nexpose&version=1.0
&runtime=2014-01-20T13:01:15.255Z/2014-01-20T13:22:14.333Z"
```

Import a vulnerability scan (eVR) in the Qualys-CSV format:

```
curl -v -k --header "X-SMS-API-KEY: <string>" -F "file=@vulnScanSampleQualys.csv"
"https://<sms_server>/vulnscanner/convert?&vendor=Qualys-CSV&product=Qualys&version=1.0
&runtime=2014-01-20T13:01:15.255Z/2014-01-20T13:22:14.333Z"
```

Import a vulnerability scan (eVR) in the Nessus format:

```
curl -v -k --header "X-SMS-API-KEY: <string>" -F "file=@vulnScanSampleNessus.nessus"
"https:<sms_server>/vulnscanner/convert?&vendor=Nessus&product=Nessus-Sample&version=1.0
&runtime=2014-01-20T13:01:15.255Z/2014-01-20T13:22:14.333Z"
```

Chapter 10

STIX/TAXII

Trusted Automated Exchange of Intelligence Information (TAXII) is an application protocol for exchanging Cyber Threat Intelligence (CTI) over HTTPS. The SMS contains embedded TAXII 2.0 and 2.1 Servers. TAXII clients can push Structured Threat Information eXpression (STIX™) 2.0/2.1 data to the SMS TAXII Servers. The advanced threat intelligence provided in tag categories keeps the Reputation Database updated, and enables robust reputation filters for enhanced protection of your system. You can use STIX/TAXII for IPS enforcement of IP, DNS, URL, and file hash Indicators of Compromise (IoCs).

Reputation database

The SMS automatically includes the following predefined tag categories for STIX/TAXII data. Use the following table to map STIX objects with user-provided Reputation tag categories.

Reputation tag	STIX object property	Description
STIX - ID	id	Identifies the STIX **Indicator** object, which is the only STIX 2.0 Domain Object the SMS imports. Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity. For example, an indicator may be used to represent a set of malicious IP addresses, domains, or URLs. To be imported to the Reputation database, an indicator STIX object must: • Only contain a single comparison expression. • Object path pattern must be domain, URL, IPv4, IPv6, SHA-1, or SHA-256.
STIX - Severity	labels	Identifies the severity for the discovered threat, based on rules that match severity. This is not a standard property for STIX 2.0.
STIX - Confidence	labels	Identifies the confidence for the discovered threat, based on rules that match a confidence score. This is not a standard property for STIX 2.0.
Reputation Entries TTL	valid_until	Identifies the date SMS will remove the entry.
-	revoked	The SMS deletes the entry when it is tagged `true`.

Versions

This feature implements STIX/TAXII 2.

Import rules

- To automatically send STIX data to the SMS, enable the TAXII service. The TAXII service is enabled by default. For more information, see "Enable SMS Services" in the *SMS User Guide*.
- Only STIX Indicator objects can be added to the Reputation database.
- STIX Indicator objects must only contain a single comparison expression.
- You cannot export STIX objects from the SMS.

Data format

Bundle

Collection of STIX objects grouped together in a single container.

Properties

Parameter	Description
type	Bundle type.
id	Bundle identifier.
spec_version	STIX specification version used to represent the content in the bundle.
objects	(Optional). Specifies a set of one or more STIX Objects.

Example

```
{
   "id":"bundle--eac5fcf6-e5a4-40d9-8721-f0e79efdadf6",
   "objects":[
      {
         "created":"2016-02-26T18:24:18.396Z",
         "id":"indicator--a6f43caf-be25-4335-bfa1-badfc13b0bae",
         "labels":[
            "malicious-activity",
            "sms-severity-high",
            "sms-confidence-75"
         ],
         "modified":"2016-02-26T18:24:18.396Z",
         "pattern":"[domain-name:value = 'example.com']",
         "type":"indicator",
         "valid_from":"2016-02-26T18:24:18.396Z"
      }
   ],
   "spec_version":"2.0",
```

```
   "type":"bundle"
}
```

Indicators

Pattern that can be used to detect suspicious or malicious cyber activity.

Properties

Parameter	Type	Description
type		Value, must be indicator.
id		Object ID.
created	timestamp	The time that the first version of the object was created.
modified	timestamp	The time that this particular version was created.
labels	One or multiple open vocabulary	Values that comes from the indicator-label-ov vocabulary.
pattern	valid pattern string	Detection pattern.
valid_from	timestamp	The time when the indicator will not be valid.
valid_until	timestamp	The time when the indicator will not be valid.
revoked	boolean	Indicates whether the object has been revoked.

Example

```
{
   "id":"bundle--eac5fcf6-e5a4-40d9-8721-f0e79efdadf6",
   "objects":[
      {
         "created":"2016-02-26T18:24:18.396Z",
```

```
         "id":"indicator--a6f43caf-be25-4335-bfa1-badfc13b0bae",
         "labels":[
            "malicious-activity",
            "sms-severity-high",
            "sms-confidence-75"
         ],
         "modified":"2016-02-26T18:24:18.396Z",
         "pattern":"[domain-name:value = 'example.com']",
         "type":"indicator",
         "valid_from":"2016-02-26T18:24:18.396Z"
      }
   ],
   "spec_version":"2.0",
   "type":"bundle"
}
```


Pattern

STIX Patterns are composed of multiple building blocks, ranging from simple key-value comparisons to more complex, context-sensitive expressions. The SMS only supports a pattern with a single comparison expression.

```
"pattern":"[domain-name:value='example.com']"
```

Comparison expression

Object path

SMS only receives the following paths:

- domain-name:value
- ipv6-addr:value
- ipv4-addr:value
- url:value

- file:hashes.SHA-1
- file:hashes.SHA-256

Comparison operator

The SMS Web API only supports the "=" comparison operator.

Labels

Labels come from the indicator-label-ov vocabulary.

Indicator label vocabulary

If an object contains a "benign" label, it is not added into the Reputation database.

- anomalous-activity
- anonymization
- benign
- compromised
- malicious-activity
- attribution

STIX - Severity

The SMS tags the severity level as either low, medium, or high.

Label	Severity
-severity-high	-
a-b-severity-low	low
severity-low	low
severity-LOW	low
severity-low-aaa	-

Label	Severity
threatstream-severity-high	high
threatstream-severity-highba	-
threatstream-severity-very-high	high

STIX - Confidence

The following table includes examples of how the SMS tags STIX - Confidence labels.

Label	Confidence
confidence-99	99
aaa-confidence-99	99
confidence-50	50
confidence-101	-
-confidence-99	-

TAXII 2.0 APIs

The following APIs are used by TAXII 2.0 servers.

Server discovery

Provides general information about the TAXII 2.0 server.

- Common entry point for TAXII clients into the data and services provided by a TAXII server.
- API Roots are logical groupings of TAXII channels, collections, and related functionality.

Definition

```
taxii
```

Parameters

Parameter	Description
title	Server name.
api_roots	List of URLs that identify known API roots.
default	Default API root.

Example

```
{
    "title": "TippingPoint Security Management System",
    "default": "https://1.2.3.4/taxii/feeds/",
    "api_roots":[
        "https://1.2.3.4/taxii/feeds/"
    ]
}
```

Get API root information

Provides general information about the API Root.

Definition

```
taxii/feeds
```

Parameters

Parameter	Description
title	Name.
versions	List of compatible TAXII versions.
max_content_length	Maximum size of the request body in octets (8-bit bytes).

Example

```
{
    "title": "TAXII feeds",
    "versions": ["taxii-2.0"],
    "max_content_length": 2097152
}
```

Get collections

Provides information about the collections.

Request

```
taxii/feeds/collections
```

Parameters

Parameter	Description
id	Collection ID.
title	Name used to identify the collection.
can_read	Indicates if you can read (GET) objects from the collection.
can_write	Indicates if you can write (POST) objects to the collection.

Response

```
{
  "collections": [
    {
      "id": "00000000-0000-0000-0000-000000000001",
      "title": "User Reputation Entries",
      "can_read": true,
      "can_write": true
    }
```

```
  ]
}
```

Add objects

Adds objects to a collection.

Definition

```
taxii/feeds/collections/00000000-0000-0000-0000-000000000001
```

Example

```
{
   "id":"bundle--eac5fcf6-e5a4-40d9-8721-f0e79efdadf6",
   "objects":[
      {
         "created":"2016-02-26T18:24:18.396Z",
         "id":"indicator--a6f43caf-be25-4335-bfa1-badfc13b0bae",
         "labels":[
            "malicious-activity",
            "sms-severity-high",
            "sms-confidence-75"
         ],
         "modified":"2016-02-26T18:24:18.396Z",
         "pattern":"[domain-name:value = 'example.com']",
         "type":"indicator",
         "valid_from":"2016-02-26T18:24:18.396Z"
         "valid_until":2016-02-27T18:24:18.396Z"
      }
   ],
   "spec_version":"2.0",
   "type":"bundle"
}
```

Get status

Provides information about the status of a previous request. In TAXII 2.0, the only request that can be monitored is one to add objects to a Collection.

Definition

```
taxii/feeds/status/<id>
```

Parameters

Parameter	Description
id	ID
status	Status of a previous POST request; the value of this property is `complete` or `pending`.
total_count	Total number of objects in the request.
success_count	Number of objects that were successfully created.
successes	List of object IDs that were successfully processed.
failure_count	Number of objects that failed to be created.
failures	List of status failures including object ID and message.
pending_count	Number of objects that have not been processed.
pendings	List of objects that have not been processed.

Response

```
{
    "id": "2d086da7-4bdc-4f91-900e-d77486753710",
    "status": "pending",
    "total_count": 3,
    "success_count": 1,
```

```
    "successes": [
        "indicator--c410e480-e42b-47d1-9476-85307c12bcbf"
     ],
    "failure_count": 1,
    "failures": [
        {
            "id": "malware--664fa29d-bf65-4f28-a667-bdb76f29ec98",
            "message": "Malware is an unsupported type"
        }
    ],
    "pending_count": 1,
    "pendings": [
        "indicator--252c7c11-daf2-42bd-843b-be65edca9f61"
    ]
}
```


TAXII 2.1 APIs

The following APIs are used by TAXII 2.1 servers.

Server discovery

Provides general information about the TAXII 2.1 server.

- Common entry point for TAXII clients into the data and services provided by a TAXII server.
- API Roots are logical groupings of TAXII channels, collections, and related functionality.

Definition

```
taxii2
```

Parameters

Parameter	Description
title	Server name.
api_roots	List of URLs that identify known API roots.
default	Default API root.

Response

```
{
    "title": "TippingPoint Security Management System",
    "default": "https://1.2.3.4/taxii2api/",
    "api_roots":[
        "https://1.2.3.4/taxii2api/"
    ]
}
```

Get API root information

Provides general information about the API Root.

Definition

```
taxii2api
```

Parameters

Parameter	Description
title	Name.
versions	List of compatible TAXII versions.
max_content_length	Maximum size of the request body in octets (8-bit bytes).

Response

```
{
  "title": "TAXII 2.1 API",
  "versions": ["application/taxii+json;version=2.1"],
  "max_content_length": 16777216
}
```

Get collections

Provides information about the collections.

Definition

```
taxii2api/collections
```

Parameters

Parameter	Description
id	Collection ID.
title	Name used to identify the collection.
can_read	Indicates if you can read (GET) objects from the collection.
can_write	Indicates if you can write (POST) objects to the collection.

Response

```
{
  "collections": [
    {
      "id": "00000000-0000-0000-0000-000000000001",
      "title": "User Reputation Entries",
      "can_read": true,
      "can_write": true
    }
```

```
  ]
}
```

Add objects

Adds objects to a collection.

Definition

```
taxii2api/collections/00000000-0000-0000-0000-000000000001
```

Request

```
{
   "id":"bundle--eac5fcf6-e5a4-40d9-8721-f0e79efdadf6",
   "objects":[
      {
         "created":"2016-02-26T18:24:18.396Z",
         "id":"indicator--a6f43caf-be25-4335-bfa1-badfc13b0bae",
         "labels":[
            "malicious-activity",
            "sms-severity-high",
            "sms-confidence-75"
         ],
         "modified":"2016-02-26T18:24:18.396Z",
         "pattern":"[domain-name:value = 'example.com']",
         "type":"indicator",
         "valid_from":"2016-02-26T18:24:18.396Z"
         "valid_until":2016-02-27T18:24:18.396Z"
      }
   ],
   "spec_version":"2.0",
   "type":"bundle"
}
```

Response

```
{
  "id": "7c5f703c-7b72-4a3a-8a55-1941770ec0af",
  "status": "pending",
  "successes": [],
  "pendings": [
    {
      "id": "indicator--a6f43caf-be25-4335-bfa1-badfc13b0bae",
      "version": "2016-02-26T18:24:18.396Z"
    }
  ],
  "failures": [],
  "pending_count": 1,
  "success_count": 0,
  "failure_count": 0,
  "total_count": 1
}
```

Get status

Provides information about the status of a previous request. In TAXII 2.1, the only request that can be monitored is one to add objects to a Collection.

Definition

```
taxii2api/status/<id>
```

Parameters

Parameter	Description
id	ID
status	Status of a previous POST request; the value of this property is `complete` or `pending`.

Parameter	Description
total_count	Total number of objects in the request.
success_count	Number of objects that were successfully created.
successes	List of object IDs that were successfully processed.
failure_count	Number of objects that failed to be created.
failures	List of status failures including object ID and message.
pending_count	Number of objects that have not been processed.
pendings	List of objects that have not been processed.

Response

```
{
  "id": "6b5572af-c687-429b-b403-c00a087aac4d",
  "status": "complete",
  "successes": [
    {
      "id": "indicator--ff714d3c-1ebf-4dd4-bd53-000000000001",
      "version": "2025-01-31T19:51:15Z"
    }
  ],
  "pendings": [],
  "failures": [
    {
      "id": "indicator--ff714d3c-1ebf-4dd4-bd53-000000000005",
      "version": "2025-01-31T19:51:15Z",
      "message": "Invalid SHA-256 File Hash"
    }
  ],
```

```
  "pending_count": 0,
  "success_count": 1,
  "failure_count": 1,
  "total_count": 2
}
```

Chapter 11

Database access

Used to access the SMS Web API.

Definition

```
dbAccess/tptDBServlet
```

Parameters

Parameter	Description
`method`	Required. • *DataDictionary on page 11-2* • *GetData - Events data on page 11-13* • *GetNewestRecord on page 11-24* • *GetOldestRecord on page 11-25* • *Schema on page 11-26* • *Status on page 11-26* • *Version on page 11-26*

Usage sequence

Follow this sequence when accessing the SMS database:

1. Use the *schema method on page 11-26* to retrieve the schema definition. Apply the returned data to user-defined database.
2. Use the *DataDictionary method on page 11-2* to retrieve supporting data. Apply the returned data to database, and repeat as needed to create profiles or activate Digital Vaccines.

   When the SMS receives a valid request using this method, it can return an XML response. Specific response content depends on data you specify in the request.
3. Use the *GetData method on page 11-13* to receive event data that you can then import into the database.

DataDictionary

Obtain SMS data dictionary information related to profiles, devices, segments, and virtual segments.

Definition

```
dbAcess/tptDBServlet?method=DataDictionary
```

Parameters

Parameter	Description
`format`	Optional. • `sql` (default) • `csv` • `xml`

Parameter	Description
`mode`	Optional. • `insert` (default) – use with `sql` format. • `update` • `replace` – use with MySQL.
`table`	Optional. If you do not specify a table, all tables are included. • *ACTIONSET table on page 11-3* • *ALERT_TYPE table on page 11-4* • *DEVICE table on page 11-4* • *POLICY table on page 11-5* • *PRODUCT_CATEGORY table on page 11-6* • *PROFILE table on page 11-6* • *PROFILE_INSTALL_INVENTORY table on page 11-7* • *QUARANTINE_NETWORK_DEVICES table on page 11-7* • *SEGMENT table on page 11-8* • *SEGMENT_GROUP table on page 11-8* • *SIGNATURE table on page 11-9* • *TAXONOMY_MAJOR table on page 11-10* • *TAXONOMY_MINOR table on page 11-10* • *TAXONOMY_PLATFORM table on page 11-11* • *TAXONOMY_PROTOCOL table on page 11-11* • *THRESHOLD_UNITS table on page 11-12* • *VIRTUAL_SEGMENT table on page 11-12*

ACTIONSET table

Record defined by the user and applied to a POLICY. Used to determine the action that is taken when a POLICY is triggered.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=ACTIONSET
```

Parameters

Column	Description
ID	Unique identifier. use this column to join from other tables.
NAME	Descriptive name.
RATE	Rate limit value applied to the action set. Has a value specifying the RATE to be applied for rate limit action sets.
FLOW_CONTROL	Traffic flow indicator. • ALLOW • DENY • TRUST • RATE

ALERT_TYPE table

Descriptive name for alerts.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=ALERT_TYPE
```

Parameters

Column	Description
ID	Unique identifier.
NAME	Descriptive name.

DEVICE table

Record for each of the IPS devices managed on the SMS.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=DEVICE
```

Parameters

Column	Description
ID	Unique identifier.
SHORT_ID	Lookup identifier.
NAME	Descriptive name of the device provided during device installation.
MODEL	String that represents the model of the device.
SERIAL_NUMBER	Alpha-numeric TippingPoint serial number.
IP_ADDRESS	IP address for the management port for the device.
LOCATION	Descriptive location text entered during device installation.
DV_VERSION	Current version of the Digital Vaccine installed on the device; if the device is a Core Controller, this field is null.
OS_VERSION	Current version of the TOS installed on the device.
DEVICE_GROUP	Name of the group to which the device belongs.
MANAGED	Boolean to show if the device is currently managed on the SMS.

POLICY table

Holds objects that determine what actions to take for a SIGNATURE.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=POLICY
```

Parameters

Column	Description
ID	Unique identifier.
PROFILE_ID	Identifier of the PROFILE object that contained this POLICY.
SIGNATURE_ID	Identifier of the SIGNATURE this object is defining in a POLICY.
ACTIONSET_ID	Identifier for the ACTIONSET applied to this object.
NAME	Descriptive name for the POLICY, which is usually the same as the SIGNATURE referenced by SIGNATURE_ID; however, THRESHOLDS allow you to name the POLICY.

PRODUCT_CATEGORY table

Maintains the names used for SIGNATURE categories. The SIGNATURE table contains a number that is joined to the ID field in this PRODUCT_CATEGORY table.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=PRODUCT_CATEGORY
```

Parameters

Column	Description
ID	Unique identifier. Use this column to join from other tables.
NAME	Descriptive name.

PROFILE table

Container for your POLICY entries. Name the PROFILE, make changes to the POLICY objects, and then distribute to a segment group.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=PROFILE
```

Parameters

Column	Description
ID	Unique identifier.
VERSION	Current profile version.
NAME	Profile name.
DESCRIPTION	Profile description.

PROFILE_INSTALL_INVENTORY table

Container for items associated with PROFILE entries.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=PROFILE_INSTALL_INVENTORY
```

Parameters

Column	Description
VIRTUAL_SEGMENT_ID	Lookup identifier for the virtual segment where the profile was distributed.
PROFILE_ID	Lookup identifier for the profile details.
PROFILE_VERSION	Profile version.
DISTRIBUTE_ID	Lookup identifier for the distribution details.
COMPLETE_TIME	Time the profile distribution completed. • Value is in milliseconds since Jan. 1, 1970 00:00:00 GMT

QUARANTINE_NETWORK_DEVICES table

Contains the defined quarantine switches.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=QUARANTINE_NETWORK_DEVICES
```

Parameters

Column	Description
NAME	Descriptive name for the network device switch type.
IP_ADDRESS	IP address for the switch.

SEGMENT table

Represents a physical segment on a device.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=SEGMENT
```

Parameters

Column	Description
ID	Unique identifier.
DEVICE_ID	Device to which this segment belongs.
NAME	Descriptive name.
IP_ADDRESS	OBSOLETE IP Address that may be given to the segment.
SLOT_INDEX	Internal chassis slot number. • Physical segments: 3 • Virtual segments: 0
SEGMENT_INDEX	• Physical segments: Physical segment number • Virtual segments: 0

SEGMENT_GROUP table

Represents a group of physical segments.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=SEGMENT_GROUP
```

Parameters

Column	Description
ID	Unique identifier.
NAME	Descriptive name for the segment group when it was created.

SIGNATURE table

Details the currently active Digital Vaccine package on the SMS for use with devices. The table grows as new Digital Vaccines are released, downloaded, and activated.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=SIGNATURE
```

Parameters

Column	Description
ID	Unique identifier.
NUM	Integer used to reference the signature, which is assigned by Trend.
SEVERITY_ID	Identifier for the SEVERITY of the SIGNATURE. • Join to SEVERITY.ID to obtain a descriptive name of the SEVERITY.
NAME	Signature name.
CLASS	Descriptive classification for the SIGNATURE.
PRODUCT_CATEGORY_ID	Category ID from PRODUCT_CATEGORY table, provided by Trend.

Column	Description
PROTOCOL	Signature protocol.
TAXONOMY_ID	Taxonomy classification.
CVE_ID	Comma-separated list of CVE IDs that can be used to link to the CVE database.
BUGTRAQ_ID	Comma-separated list of BugTraq IDs that can be used to link to the BugTraq database.
DESCRIPTION	Signature description, which is provided by Trend.
MESSAGE	Message that can be filled in with ALERTS. • MESSAGE_PARMS values to create a dynamic message for this SIGNATURE.

TAXONOMY_MAJOR table

Details the TippingPoint signature taxonomy major classifications.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=TAXONOMY_MAJOR
```

Parameters

Column	Description
ID	Unique identifier.
NAME	Short name.
DESCRIPTION	Description.

TAXONOMY_MINOR table

Details the TippingPoint signature taxonomy minor classifications.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=TAXONOMY_MINOR
```

Parameters

Column	Description
ID	Unique identifier.
MAJOR_ID	Identifier of the major classification ID to which this minor classification relates.
DESCRIPTION	Description.

TAXONOMY_PLATFORM table

Details the TippingPoint signature platforms.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=TAXONOMY_PLATFORM
```

Parameters

Column	Description
ID	Unique identifier.
DESCRIPTION	Description.

TAXONOMY_PROTOCOL table

Details the TippingPoint signature protocols.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=TAXONOMY_PROTOCOL
```

Parameters

Column	Description
ID	Unique identifier.
DESCRIPTION	Description.

THRESHOLD_UNITS table

Defines the units in which thresholds can be specified.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=THRESHOLD_UNITS
```

Parameters

Column	Description
ID	Unique identifier.
NAME	Descriptive name for the unit entry.

VIRTUAL_SEGMENT table

Represents a virtual physical segment on a device.

Definition

```
dbAccess/tptDBServlet?method=DataDictionary&table=VIRTUAL_SEGMENT
```

Parameters

Column	Description
ID	Unique identifier.
DEVICE_ID	Device to which this segment belongs.
SEGMENT_GROUP_ID	Segment group to which this segment belongs.

Column	Description
NAME	Descriptive name.

GetData - Events data

Request data from specified tables.

Definition

```
dbAccess/tptDBServlet?method=GetData
```

Parameters

Parameter	Value	Description
`begin_time`	integer	Required. • Expressed as the number of milliseconds since 01-01-1970 00:00:00 GMT
`end_time`	integer	Required. • Expressed as the number of milliseconds since 01-01-1970 00:00:00 GMT
`format`		Optional. • `csv` (default) • `sql` • `xml`
`limit`	integer	Optional. • Maximum number of values returned. • All values are returned by default.

Parameter	Value	Description
`table`		Required. Events data: • *ALERTS table on page 11-14* • *DDOS_STATS table on page 11-19* • *FIREWALL_BLOCK_ALERTS table on page 11-19* • *FIREWALL_TRAFFIC_ALERTS table on page 11-21* • *PORT_TRAFFIC_STATS table on page 11-22* • *QUARANTINE_HOSTS table on page 11-23* • *RATELIMIT_STATS table on page 11-24*

ALERTS table

Contains information pertaining to the event that caused a POLICY to trigger.

- When an ACTIONSET is applied to a POLICY and it has a **Management Console** notification selected, it is put in the ALERTS table.
- The primary key, a unique key, is a four column index, DEVICE_ID, ALERT_TYPE_ID, SEQUENCE_NUM, and END_TIME.
- The table is expected to have a continuous growth pattern and contain millions of records.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=ALERTS
```

Parameters

Column	Description
SEQUENCE_NUM	Reference to a particular logs row entry counter. • The ALERT_TYPE column defines the log being referenced. • This sequence number is not reliable as far as counting on it behaving as an ever increasing sequential number. It can be reset on the device and repeated for new events.
DEVICE_ID	Identifier for the DEVICE entry that sent the notification. • Second part of the ALERTS table unique index. • A foreign key to the DEVICE table was left off for the purpose of performance and due to the possibility that a DEVICE entry may not have been yet stored in the DEVICE table for this external database.
ALERT_TYPE_ID	The TYPE column is the third and final primary key constraint on the ALERTS table. • This field can be joined to the ALERT_TYPE table for a descriptive name for this column.
POLICY_ID	Identifier used to map this alert to a POLICY table entry.
SIGNATURE_ID	Identifier used to map this alert to a SIGNATURE table entry.
BEGIN_TIME	Time at which the event was first started or previously logged. • Value is in milliseconds elapsed since Jan. 1, 1970 00:00:00 GMT • When using notification aggregation, this value and the END_TIME typically are off by the number of minutes specified in the aggregation setting. • The difference between BEGIN_TIME and END_TIME may be larger if a lot of time passes between attack events. • When aggregation is turned off, the BEGIN_TIME usually is the same as the END_TIME.

Column	Description
END_TIME	Time at which the notification was logged and sent to the Management Console. • Value is in milliseconds elapsed since Jan. 1, 1970 00:00:00 GMT • Subtract BEGIN_TIME from END_TIME to determine the length of an attack, if aggregation is being used. • Difference between BEGIN_TIME and END_TIME might be unexpectedly large if a lot of time passes between attack events.
HIT_COUNT	Counter displaying the number of times the event triggered before the notification was sent to the Management Console.
SRC_IP_ADDR	Source IP of the packet causing the notification. Numeric value of an IPv4 address, or the low-order 64 bits for an IPv6 address if SRC_IP_ADDR_HIGH is not NULL.
SRC_IP_ADDR_HIGH	Source IP of the packet causing the notification. Numeric value of high-order 64 bits for an IPv6 address.
SRC_PORT	Source port of the packet causing the notification.
DST_IP_ADDR	Destination IP of the packet causing the notification. Numeric value of an IPv4 address, or the low-order 64 bits for an IPv6 address if DST_IP_ADDR_HIGH is not NULL.
DST_IP_ADDR_HIGH	Destination IP of the packet causing the notification. Numeric value of high-order 64 bits for an IPv6 address.
DST_PORT	Destination port of the packet causing the notification.
VIRTUAL_SEGMENT_INDEX	Identifier for which device segment this alert was seen on.
PHYSICAL_PORT_IN	Device port on which the event was detected.
VLAN_TAG	VLAN identifier contained in the event.
SEVERITY	SEVERITY of the event. Usually corresponds to the SIGNATURE.SEVERITY column, joined by the SIGNATURE_ID column. A foreign key constraint to the SEVERITY table has been applied here.
PACKET_TRACE	Indicates if a packet trace is available on the device.

Column	Description
DEVICE_TRACE_BUCKET	Part of the device packet trace identifier.
DEVICE_TRACE_BEGIN_SEQ	Part of the device packet trace identifier.
DEVICE_TRACE_END_SEQ	Part of the device packet trace identifier.

MESSAGE_PARMS	Variable list of message parameters.
	• Value can be tokenized and combined with the SIGNATURE.MESSAGE data to display a dynamic ALERT message.
	• Join SIGNATURE_ID with SIGNATURE.ID to retrieve the SIGNATURE.MESSAGE data. The MESSAGE_PARMS string is a delimited string, the delimiter is the "|" character.
	• The SIGNATURE.MESSAGE string contains place holders for these strings, the place holders are %1, %2, ..., %n.
	• The tokenized MESSAGE_PARMS replaces the %n values based on their location in the string.
	Example
	`MESSAGE_PARMS=Austin|Texas SIGNATURE.MESSAGE=%1 is in %2.`
	The preceding parameters and message generates the following message:
	`Austin is in Texas.`

QUARANTINE_ACTION	Quarantine action taken, either Added or Removed; used only in quarantine logs.
FLOW_CONTROL	Action taken by the action set: Permit, Rate Limit, or Trust.
ACTION_SET_UUID	Action set UUID; used only in rate limit logs.
ACTION_SET_NAME	Rate limit action; used only in rate limit logs.
RATE_LIMIT_RATE	Rate for rate limit logs; a numerical value followed by a unit. The unit can be Kbps or Mbps.

Column	Description
CLIENT_IP_ADDR	Long value of the Client IP address (**Capture Additional Event Information** must be enabled).
CLIENT_IP_ADDR_HIGH	Long value of the Client IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.
XFF_IP_ADDR	Long value of the X-Forwarded-For IP address (**Capture Additional Event Information** must be enabled).
XFF_IP_ADDR_HIGH	Long value of the X-Forwarded-For IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.
TCIP_IP_ADDR	Long value of the True-Client-IP address (**Capture Additional Event Information** must be enabled).
TCIP_IP_ADDR_HIGH	Long value of the True-Client-IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.
URI_METHOD	Method of the URI.
URI_HOST	Host of the URI.
URI_STRING	URI string.
SRC_USER_NAME	User name on the source machine. • User ID IP Correlation must be configured on the SMS to retrieve this information. • User ID IP Correlation is a feature that enables the SMS to collect user authentication data directly and continuously from an Identity Agent device.
SRC_DOMAIN	Name of the source domain.
SRC_MACHINE	Name of the source machine.
DST_USER_NAME	User name on the destination machine.
DST_DOMAIN	Name of the destination domain
DST_MACHINE	Name of the destination machine.

DDOS_STATS table

Data accumulated from the device for Advanced DDoS policies.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=DDOS_STATS
```

Parameters

Column	Description
POLICY_ID	POLICY ID.
STAT_TIME	Time the data was collected. • Time is stored in milliseconds since Jan. 1, 1970 00:00:00 GMT
REJECT_SYNS	Number of rejected SYN requests for the stat period.
PROXIED_CXNS	Number of proxied connections for the stat period.
CPS_CXNS	Number of Connections Per Second over stat period.
BLOCKED_CPS_CXNS	Number of blocked CPS in stat period.
CFLOOD_CXNS	Number of Connection Flood connections in stat period.
BLOCKED_CFLOOD_CXNS	Number of blocked Connection Flood connections in stat period.

FIREWALL_BLOCK_ALERTS table

Contains information pertaining to logs where traffic has been permitted by firewall rules that have logging enabled, including packets that were permitted by the content filtering configuration.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=FIREWALL_BLOCK_ALERTS
```

Parameters

Column	Description
SEQUENCE_NUM	Reference to a particular logs row entry counter.
DEVICE_ID	Identifier for the DEVICE entry that sent the notification.
BEGIN_TIME	Time in which the event was first started. • When using notification aggregation, this value and the TIME_END typically are off by the number of minutes specified in the aggregation setting. • When aggregation is turned off, the BEGIN_TIME usually is the same as the TIME_END. This value is in milliseconds since Jan. 1, 1970 00:00:00 GMT.
END_TIME	Time in which the notification was sent to the Management Console. • Subtracting BEGIN_TIME from TIME_END can determine the length of an attack if aggregation is being used. • This value is in milliseconds since Jan. 1, 1970 00:00:00 GMT
HIT_COUNT	Number of times the firewall rule was applied.
SRC_IP_ADDR	Source IP of the packet causing the notification.
SRC_PORT	Source port of the packet causing the notification.
DST_IP_ADDR	Destination IP of the packet causing the notification.
DST_PORT	Destination port of the packet causing the notification.
RULE_ID	Unique identifier for rule to monitor traffic between security zones.
PROTOCOL_NAME	Packet type.
PROTOCOL_NUMBER	Number associated with the protocol in the filter.
PROTOCOL_TYPE	Protocol that was used to respond to the event.
IN_ZONE_UUID	Security zone from which the attack originated.
OUT_ZONE_UUID	Security zone from which the attack was targeted.

Column	Description
PHYSICAL_PORT_IN	Device port on which the attack was detected.
VLAN	Local VLAN that was targeted.
CATEGORY	Type of traffic filter that was activated.
URL	URL associated with the attack.
URL_INFO	Additional information relevant to the URL.
SEVERITY_ID	Severity of the attack.

FIREWALL_TRAFFIC_ALERTS table

Contains information pertaining to logs where traffic has been permitted by firewall rules that have logging enabled, including packets that were permitted by the content filtering configuration.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=FIREWALL_TRAFFIC_ALERTS
```

Parameters

Column	Description
SEQUENCE_NUM	Reference to a particular logs row entry counter.
DEVICE_ID	Identifier for the DEVICE entry that sent the notification.
END_TIME	Time in which the notification was sent to the Management Console. • Value is in milliseconds (since Jan. 1, 1970 00:00:00 GMT) • Subtract BEGIN_TIME from TIME_END to determine the length of an attack, if aggregation is being used.
SRC_IP_ADDR	Packet source IP address.
SRC_PORT	Packet source port.

Column	Description
DST_IP_ADDR	Packet destination IP address.
DST_PORT	Packet destination port.
RULE_ID	Unique identifier to monitor traffic between security zones.
PROTOCOL_NAME	Packet type.
PROTOCOL_NUMBER	Protocol number in the filter.
IN_ZONE_UUID	Security zone from which the attack originated.
OUT_ZONE_UUID	Security zone from which the attack was targeted.
CATEGORY	Type of traffic filter that was activated.
DURATION	Duration of the attack.
URL	URL that was associated with the attack.
TRANSFER_BYTES	Number of bytes transferred for the event.
MESSAGE	Dynamic ALERT message.

PORT_TRAFFIC_STATS table

Contains information of traffic going through each IPS port.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=PORT_TRAFFIC_STATS
```

Parameters

Column	Description
DEVICE_ID	Identifier for the DEVICE entry that sent the notification.
PORT_ID	Identifier for the PORT entry that the traffic is going through.
SMS_TIME	SMS time in which the statistics get captured.

Column	Description
DEVICE_TIME	Device SMS time in which the statistics get captured.
IN_OCTETS	Device SMS time in which the statistics get captured.
OUT_OCTETS	Total traffic going out the port.

QUARANTINE_HOSTS table

Contains device and SMS quarantine actions.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=QUARANTINE_HOSTS
```

Parameters

Column	Description
ID	Unique identifier for the table entry.
QUARANTINED_IP	IP address of the quarantined host.
QUARANTINED_MAC	MAC address of the quarantined host.
POLICY_NAME	Descriptive name for the policy that triggered the host quarantine.
STATE	Current state of the host. • UNQUARANTINED • QUARANTINED • INITIAL • ERROR
AUTHORITY	Source of the quarantine state for the host.
CREATE_TIME	Time the initial quarantine state was set.
LAST_UPDATE	Time of the last quarantine state change.

RATELIMIT_STATS table

When using RATELIMIT ACTIONSETs, this data is accumulated from the DEVICE.

If you are using RATELIMIT ACTIONSETs, this table is expected to have a continuous growth pattern and contain millions of records.

Definition

```
dbAccess/tptDBServlet?method=GetData&table=RATELIMIT_STATS
```

Parameters

Column	Description
ACTIONSET_ID	Identifier of the ACTIONSET table entry for this record.
STAT_TIME	Time this stat was recorded. • Time is milliseconds since Jan. 1, 1970 00:00:00 GMT
DEVICE_ID	Identifier for the DEVICE.
RATE	RATE in kbps.
VALUE	Number of Bytes.

GetNewestRecord

Retrieve the newest record of a specific table.

Definition

```
/dbAccess/tptDBServlet?method=GetNewestRecord
```

Parameters

Parameter	Description
table	• *ALERTS table on page 11-14* • *DDOS_STATS table on page 11-19* • *FIREWALL_BLOCK_ALERTS table on page 11-19* • *FIREWALL_TRAFFIC_ALERTS table on page 11-21* • *PORT_TRAFFIC_STATS table on page 11-22* • *QUARANTINE_HOSTS table on page 11-23* • *RATELIMIT_STATS table on page 11-24*

GetOldestRecord

Retrieve the oldest record of a specific table.

Definition

```
/dbAccess/tptDBServlet?method=GetOldestRecord
```

Parameters

Parameter	Description
table	• *ALERTS table on page 11-14* • *DDOS_STATS table on page 11-19* • *FIREWALL_BLOCK_ALERTS table on page 11-19* • *FIREWALL_TRAFFIC_ALERTS table on page 11-21* • *PORT_TRAFFIC_STATS table on page 11-22* • *QUARANTINE_HOSTS table on page 11-23* • *RATELIMIT_STATS table on page 11-24*

Schema

Obtain SMS database schema information. The SMS returns the schema information in Oracle 8i or MySQL 4.0 compliant data definition language (DDL) statements.

Definition

```
dbAccess/tptDBServlet?method=Schema
```

Parameters

Parameter	Description
`database`	Only valid for sql format. • `MySQL` (default) • `Oracle`

Example

```
curl -k --header "X-SMS-API-KEY: <string>" "https://<sms_server>/dbAccess/tptDBServlet?method=Schema"
```

Status

Returns the status of the SMS web API support.

Definition

```
dbAccess/tptDBServlet?method=Status
```

Response

- `OK`: SMS web API support is enabled and running.
- `Not Found`: SMS web API support is not enabled.

Version

Returns the version number of the SMS.

Definition

```
dbAccess/tptDBServlet?method=Version
```

Chapter 12

External database

The external database can be used for customized reporting. For custom reports, you can access the SMS database directly or replicate the SMS to your external server. If you require data that the SMS reports do not routinely provide, you can set up an SMS External Database with a reporting tool of your choice.

The SMS supports the following database options:

- External access - direct access to the database.
- External replication - remote replication of the database, which provides a copy of the database that can be edited, backed up, or used for offloading report functions. Data that you access remotely is read-only and cannot be changed.

External access

- *Set up the access service on page 12-3* to allow an external database tool to access data on the SMS. Do this before you configure the external application.
- Reboot the SMS to enable or disable this service.

External replication

- *Set up the replication service on page 12-6* to allow an external database server to replicate data from the SMS.

- Reboot the SMS to enable or disable this service.

Configure the SMS for external access

Open a MariaDB read-only database for any third-party access or reporting tool. The read-only database is named **ExternalAccess**.

Procedure

1. On the SMS, go to **Admin** > **Database** > **External Database Settings** > **Edit**.
2. Select **External Access Settings** > **Enable external database access**.
3. Enter the following:
   - **Username** – Provide the user name for an account with sufficient rights to read all the desired data from the SMS database.
   - **Password** – Enter and confirm the password.
4. If you changed the external access settings, click **Reboot** to restart the SMS server and initialize the service.

   Note

   Follow your company's server downtime policies, including notification to SMS clients of a pending reboot. Before you reboot the SMS, gracefully stop other client connections to the server.
5. Click **OK**.

   If verification fails:
   - Verify that the username/password on the database matches the SMS.
   - Reboot the SMS before you try to access the database.
   - Running a complex report against SMS server may slow down the SMS response time significantly.

ALERTS table – ExternalAccess

The database name is **ExternalAccess**.

Parameters

Column	Description
SEQUENCE_NUM	Reference to a particular logs row entry counter.
DEVICE_ID	Identifier for the DEVICE entry that sent the notification.
ALERT_TYPE_ID	This field can be joined to the ALERT_TYPE table for a descriptive name for this column.
POLICY_ID	Identifier used to map this alert to a POLICY table entry.
SIGNATURE_ID	Identifier used to map this alert to a SIGNATURE table entry.
BEGIN_TIME	Time at which the event was first started or previously logged.
END_TIME	Time at which the notification was logged and sent to the Management Console.
HIT_COUNT	Counter displaying the number of times the event triggered before the notification was sent to the Management Console.
SRC_IP_ADDR	Source IP of the packet causing the notification.
SRC_IP_ADDR_2	Represents the higher 64 bit for the IPv6 source addresses. For IPv4 address, this field has a NULL value.
SRC_PORT	Source port of the packet causing the notification.
DST_IP_ADDR	Destination IP of the packet causing the notification.
DST_IP_ADDR_2	Represents the higher 64 bit for the IPv6 destination addresses. For IPv4 address, this field has a NULL value.
DST_PORT	Destination port of the packet causing the notification.
VIRTUAL_SEGMENT_INDEX	Identifier for which device segment this alert was seen on.
PHYSICAL_PORT_IN	Device port on which the event was detected.

Column	Description
VLAN_TAG	VLAN identifier contained in the event.
SEVERITY	SEVERITY of the event. Usually corresponds to the SIGNATURE.SEVERITY column, joined by the SIGNATURE_ID column. A foreign key constraint to the SEVERITY table has been applied here.
PACKET_TRACE	Indicates if a packet trace is available on the device.
DEVICE_TRACE_BUCKET	Part of the device packet trace identifier.
DEVICE_TRACE_BEGIN_SEQ	Part of the device packet trace identifier.
DEVICE_TRACE_END_SEQ	Part of the device packet trace identifier.
MESSAGE_PARMS	Variable list of message parameters.
QUARANTINE_ACTION	Quarantine action taken, either Added or Removed; used only in quarantine logs.
FLOW_CONTROL	Action taken by the action set: Permit, Rate Limit, or Trust.
ACTION_SET_UUID	Action set UUID; used only in rate limit logs.
ACTION_SET_NAME	Rate limit action; used only in rate limit logs.
RATE_LIMIT_RATE	Rate for rate limit logs; a numerical value followed by a unit. The unit can be Kbps or Mbps.
CLIENT_IP_ADDR	Long value of the Client IP address (**Capture Additional Event Information** must be enabled).
CLIENT_IP_ADDR_HIGH	Long value of the Client IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.
XFF_IP_ADDR	Long value of the X-Forwarded-For IP address (**Capture Additional Event Information** must be enabled).
XFF_IP_ADDR_HIGH	Long value of the X-Forwarded-For IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.

Column	Description
TCIP_IP_ADDR	Long value of the True-Client-IP address (**Capture Additional Event Information** must be enabled).
TCIP_IP_ADDR_HIGH	Long value of the True-Client-IP address (**Capture Additional Event Information** must be enabled). For IPV6 only.
URI_METHOD	URI method.
URI_HOST	URI host.
URI_STRING	URI string.
SRC_USER_NAME	Source machine user name.
SRC_DOMAIN	Source domain name.
SRC_MACHINE	Source machine name.
DST_USER_NAME	Destination machine user name.
DST_DOMAIN	Destination domain name.
DST_MACHINE	Destination machine name.

Configure the SMS for replication

This service allows an external database server to replicate data from the SMS. Using an external database for data replication allows you to offload report processing to an external server which can provide performance gains to your existing system. Reboot the SMS to completely enable or disable this service.

Before you begin, make sure that your replication system has sufficient disk space to accommodate the database and any increase in size due to additional data or reporting.

Procedure

1. In the SMS, go to **Admin** > **Database**.
2. On the External Database Settings panel, click **Edit**.

3. In the Edit External Database Settings wizard, select **External Replication Settings**.

Note

To configure external database replication, you must create an SMS database snapshot, and then copy the snapshot to the target replication system and import it into a MariaDB database before the SMS server can replicate its data to the target system.

4. Select **Enable external database replication** to enable the service. (To disable the service, clear the check box.)
5. Provide the following:
   - **Username** – Provide the user name for an account with sufficient rights to read all the desired data from the SMS database.
   - **Password** – Provide the password for the user account. Retype the password in the Confirm Password field.
6. If you changed the replication settings, click **Reboot** to restart the SMS server and initialize the service.

Note

Follow your company's server downtime policies, including notification to SMS clients of a pending reboot. Before you reboot the SMS, gracefully stop other client connections to the server.

7. Click **Create Snapshot**, and select **Include Events in Snapshot** if you want the snapshot to include event data.

Note

The snapshot is saved locally on the SMS server. You must copy the snapshot to the target replication system and import it into a new or existing MariaDB database before the SMS server can replicate its data to the target system.

8. Click **OK**.

Note

External database replication and the SMS High Availability (HA) features both leverage the same functionality in the underlying MariaDB database. The SMS database does not support replication to multiple destinations; therefore, we do not recommend using SMS HA and external database replication at the same time.

Replication – database schema

Includes the following tables created when you dump the snapshot file to the replicated database server.

Some of the tables are for internal use only. The rest of tables are divided into two categories: *DataDictionary on page 11-2* and *Events Data on page 11-13*.

Configure the SMS to enable restricted access

This service allows access to the external database to be restricted to a set of IP addresses.

Procedure

1. In the SMS, go to **Admin** > **Database**.
2. On the External Database Settings panel, click **Edit**.
3. In the Edit External Database Settings wizard, select **Access Restrictions**.
4. Select **Enable restricted access** to enable the service. (To disable the service, clear the check box.)
5. Provide the following:

- **Named IP Address Group** – To restrict a set of IP addresses, click the arrow, and either select a Named IP Address Group or create a new one.

6. Click **OK**.

Chapter 13

MIB files for the SMS

A management information base (MIB) is a type of database that is used to manage devices in a communications network. Database entries are addressed through object identifiers (OIDs). MIB files are descriptions of network objects that can be managed using the Simple Network Management Protocol (SNMP). The format of the MIB is defined as part of the SNMP.

This information includes the following topics:

SMS MIBs on page 13-2

Public MIB files on page 13-2

Health monitoring on page 13-2

SMS MIBs

You can download TippingPoint SMS MIB files from the TMC at https://tmc.tippingpoint.com. On the TMC website, navigate to the Documentation area for this product release, and then select **SMS MIBS**.

The compressed file contains two MIB files:

- **TPT-SMSMIBS** defines monitoring functions
- **TPT-SMS-TRAP-MIB** defines the SMS traps

For more information about these MIBs, refer to the *TippingPoint MIB Guide for TOS v3.9.0*, available on the TMC.

Public MIB files

Publicly available UCD-SNMP-MIB and UCD-DISKIO-MIB definitions can be used to query SMS health values. These files can be downloaded from the following locations:

- http://net-snmp.sourceforge.net/docs/mibs/
- http://net-snmp.sourceforge.net/docs/mibs/UCD-SNMP-MIB.txt
- http://net-snmp.sourceforge.net/docs/mibs/UCD-DISKIO-MIB.txt

Note that only the SMS Health Section OIDs listed in *Health monitoring on page 13-2* are supported.

Health monitoring

The following table lists the OIDs that are used to graph and display values in the SMS Health section of the SMS client.

Section	Description	OID
CPU	CPU_USER	1.3.6.1.4.1.2021.11.50.0
	CPU_SYS	1.3.6.1.4.1.2021.11.52.0
	CPU_IDLE	1.3.6.1.4.1.2021.11.53.0

Section	Description	OID
Filesystem	FS_DSKPATH	1.3.6.1.4.1.2021.9.1.2
	FS_DEVPATH	1.3.6.1.4.1.2021.9.1.3
	FS_TOTAL	1.3.6.1.4.1.2021.9.1.6
	FS_AVAIL	1.3.6.1.4.1.2021.9.1.7
	FS_USED	1.3.6.1.4.1.2021.9.1.8
	FS_PERCENT	1.3.6.1.4.1.2021.9.1.9
	FS_IPERCENT	1.3.6.1.4.1.2021.9.1.10
High Availability	HA	1.3.6.1.4.1.2021.8.1.101.34
Memory	SWAP_TOTAL	1.3.6.1.4.1.2021.4.3.0
	SWAP_AVAIL	1.3.6.1.4.1.2021.4.4.0
	REALMEM_TOTAL	1.3.6.1.4.1.2021.4.5.0
	REALMEM_AVAIL	1.3.6.1.4.1.2021.4.6.0
Network Traffic	ETHO_RX_BYTES	1.3.6.1.4.1.2021.8.1.101.1
	ETHO_RX_PACKETS	1.3.6.1.4.1.2021.8.1.101.2
	ETHO_RX_ERRORS	1.3.6.1.4.1.2021.8.1.101.3
	ETHO_RX_DROPPED	1.3.6.1.4.1.2021.8.1.101.4
	ETHO_RX_FIFO_ERRORS	1.3.6.1.4.1.2021.8.1.101.5
	ETHO_RX_FRAME_ERRORS	1.3.6.1.4.1.2021.8.1.101.6
	ETHO_RX_COMPRESSED	1.3.6.1.4.1.2021.8.1.101.7
	ETHO_TX_BYTES	1.3.6.1.4.1.2021.8.1.101.8
	ETHO_TX_PACKETS	1.3.6.1.4.1.2021.8.1.101.9
	ETHO_TX_ERRORS	1.3.6.1.4.1.2021.8.1.101.10

Section	Description	OID
	ETH0_TX_DROPPED	1.3.6.1.4.1.2021.8.1.101.11
	ETH0_TX_FIFO_ERRORS	1.3.6.1.4.1.2021.8.1.101.12
	ETH0_TX_CARRIER_ERRORS	1.3.6.1.4.1.2021.8.1.101.13
	ETH0_TX_COMPRESSED	1.3.6.1.4.1.2021.8.1.101.14
	ETH0_MULTICAST	1.3.6.1.4.1.2021.8.1.101.15
	ETH0_COLLISIONS	1.3.6.1.4.1.2021.8.1.101.16
	ETH1_RX_BYTES	1.3.6.1.4.1.2021.8.1.101.17
	ETH1_RX_PACKETS	1.3.6.1.4.1.2021.8.1.101.18
	ETH1_RX_ERRORS	1.3.6.1.4.1.2021.8.1.101.19
	ETH1_RX_DROPPED	1.3.6.1.4.1.2021.8.1.101.20
	ETH1_RX_FIFO_ERRORS	1.3.6.1.4.1.2021.8.1.101.21
	ETH1_RX_FRAME_ERRORS	1.3.6.1.4.1.2021.8.1.101.22
	ETH1_RX_COMPRESSED	1.3.6.1.4.1.2021.8.1.101.23
	ETH1_TX_BYTES	1.3.6.1.4.1.2021.8.1.101.24
	ETH1_TX_PACKETS	1.3.6.1.4.1.2021.8.1.101.25
	ETH1_TX_ERRORS	1.3.6.1.4.1.2021.8.1.101.26
	ETH1_TX_DROPPED	1.3.6.1.4.1.2021.8.1.101.27
	ETH1_TX_FIFO_ERRORS	1.3.6.1.4.1.2021.8.1.101.28
	ETH1_TX_CARRIER_ERRORS	1.3.6.1.4.1.2021.8.1.101.29
	ETH1_TX_COMPRESSED	1.3.6.1.4.1.2021.8.1.101.30
	ETH1_MULTICAST	1.3.6.1.4.1.2021.8.1.101.31
	ETH1_COLLISIONS	1.3.6.1.4.1.2021.8.1.101.32

Section	Description	OID
Temperature	TEMPERATURE	1.3.6.1.4.1.2021.8.1.101.33

Chapter 14

Event Taxonomy

The following sections help you get started with the Event Taxonomy:

- *Taxonomy Event ID on page 14-2*
- *Major categories on page 14-4*
- *Minor categories on page 14-4*
- *Protocol type on page 14-7*
- *Platform type on page 14-11*

Event Taxonomy

This information provides details about the TippingPoint event taxonomy for use with the SMS Web Services API with SMS version 4.1 and later.

The event taxonomy provides further information for use with following taxonomy tables:

- TAXONOMY_MAJOR
- TAXONOMY_MINOR
- TAXONOMY_PROTOCOL
- TAXONOMY_PLATFORM

Taxonomy Event ID

The Taxonomy Event ID for a particular event is a 10-digit number constructed with the following components:

- Major Category (0-127)
- Minor Category (0-255)
- [Protocol Type optional] (0-255)
- [Platform Type optional] (0-255)

The number is then calculated much like a decimal IP address conversion: (Major * 16777216) + (Minor * 65536) + (Protocol * 256) + (Platform octet).

Note

The maximum value for a Taxonomy Event ID is 2,147,483,647.

Data detail examples

The following are data detail examples.

Example 1

TP ID - 17107965

Filter 2813: HTTP: HP Web Jetadmin Remote Command Injection Vulnerability

001 (Vulnerability) + **005** (Command Injection) + **011** (http protocol) + **253** (Multi-platform Server Application or Service) = 1*16777216 + 5*65536 + 11*256 + 253 = **17107965**

Example 2

TP ID - 67214080

Filter 1511: Kazaa: File Download/Upload

004 (Security Policy) + **001** (P2P) + **155** (FastTrack) + **001** (Windows Client Application) = 3*16777216 + 0*65536 + 112*256 + 252 = 4*16777216 + 1*65536 + 155*256 + 1 = **67214080**

Example 3

TP ID - 84151551

Filter 164: ICMP: Echo Request (Ping)

005 (Reconnaissance/ Suspicious Access) + **004** (Host Scan) + **012** (ICMP) + **255** (Other) = 5*16777216 + 4*65536 + 12*256 + 255 = **84151551**

Example 4

TP ID - 33693185

Filter 2785: POP/IMAP: Netsky-P Virus Propagation

002 (Malicious Code) + **002** (virus) + **030** (pop/imap) + **001** (Windows Client Application) = 2*16777216 + 2*65536 + 30*256 + 1 = **33693185**

Example 5

TP ID - 100750333

Filter 2824: SIP: From Field Anomaly

006 (Application/ Protocol Anomaly) + **001** (Protocol Anomaly) + **083** (sip) + **253** (Multi-platform Server Application or Service) = 6*16777216 + 1*65536 + 83*256 + 253 = **100750333**

Major categories

The following table gives the codes and descriptions for major categories.

Category code	Category	Description
001	Vulnerability	This category includes events triggered by an attempt to exploit a vulnerability in any application, operating system, or networked hardware device.
002	Malicious Code	This includes events triggered by viruses, worms, Trojans, backdoors, and all manner of blended malware threats.
003	Distributed Denial of Service (DDoS)	This category includes events triggered by traffic thresholds that indicate an attempt to make a resource unavailable.
004	Security Policy	This category includes events that indicate an attempt to violate an organization's security policy. It covers P2P, IM, email attachments, IRC, and other network communication types.
005	Reconnaissance or Suspicious Access	This category includes events that indicate network activity usually associated with common information gathering techniques used by attackers to launch more sophisticated attacks.
006	Application or Protocol Anomaly	This category includes events that indicate a violation of a protocol or application's RFC.
007	Traffic Thresholds	This category includes events triggered by predefined thresholds for specific applications or ports.
008	IP Filters	This category includes events triggered by predefined IP access control lists.

Minor categories

The following table gives the codes and descriptions for minor categories.

Category Code	Category	Description
001	Vulnerability	Buffer/Heap Overflow
002	Vulnerability	Denial of Service (Crash/Reboot)
003	Vulnerability	Configuration Error
004	Vulnerability	Race Condition
005	Vulnerability	Invalid Input (Command Injection, Cross-Site Scripting, SQL Injection, etc.)
006	Vulnerability	Access Validation
255	Vulnerability	Other
001	Malicious Code	Worm
002	Malicious Code	Virus
003	Malicious Code	Trojan/Backdoor
004	Malicious Code	IRC Botnet/Blended Threat
005	Malicious Code	Phishing
255	Malicious Code	Other
001	DDoS	SYN Flood Attack
002	DDoS	Other Flood Attack (e.g., ACK, CPS, etc.)
003	DDoS	Iterative Application Attack (Hammer)
255	DDoS	Other
001	Security Policy	P2P
002	Security Policy	Chat and Instant Messaging
003	Security Policy	Streaming Media
004	Security Policy	Email Attachments

Category Code	Category	Description
005	Security Policy	Forbidden Application Access or Service Request (Telnet, SMB Null Session, etc.)
006	Security Policy	Authentication Failure (Telnet login failed, brute force, etc.)
007	Security Policy	Spyware
255	Security Policy	Other
001	Reconnaissance or Suspicious Access	Port Scan
002	Reconnaissance or Suspicious Access	Suspicious Application Access
003	Reconnaissance or Suspicious Access	Suspicious Service Request
004	Reconnaissance or Suspicious Access	Host Scan
255	Reconnaissance or Suspicious Access	Other
001	Application or Protocol Anomaly	Protocol Anomaly
002	Application or Protocol Anomaly	Evasion Technique
003	Application or Protocol Anomaly	Application Anomaly
255	Application or Protocol Anomaly	Other Anomaly
001	Traffic Thresholds	Traffic Threshold
002	Traffic Thresholds	Application Threshold
255	Traffic Thresholds	Other

Category Code	Category	Description
001	IP Filters	Deny
002	IP Filters	Accept
255	IP Filters	Other

Protocol type

The following table lists the type codes for protocols.

Type code	Protocol
001	appletalk
002	auth
003	bgp
004	cdp
005	clns
006	dhcp
007	dns
008	finger
009	ftp
010	hsrp
011	http
012	icmp
013	igmp
014	igrp/eigrp
015	ipv6

Type code	Protocol
016	ipx
017	irc
018	is-is
019	isakmp/ike
020	ldap
021	mpls
022	ms-rpc
023	ms-sql
024	nat
025	netbios
026	nntp
027	ntp
028	oracle (sqlnet, etc.)
029	ospf
030	pop/imap
031	portmapper
032	qos
033	rip
034	rpc services
035	smb
036	smtp
037	snmp
038	sql

Type code	Protocol
039	ssh
040	ssl/tls
041	tacacs
042	tcp (generic)
043	telnet
045	udp (generic)
046	uucp
048	x-window
049	tftp
050	IP
051	nfs
052	wins
080	h.323 (voip)
081	megaco (voip)
082	mgcp (voip)
083	sip (voip)
084	rtp/rtcp (voip)
099	voip (other)
100	aim (IM)
101	msn (IM)
102	yahoo! (IM)
103	icq (IM)
119	IM (other)

Type code	Protocol
120	musicMatch
121	winamp
122	shoutcast
123	windows media
124	quicktime
125	rtsp
149	streaming media (other)
150	bittorrent
151	blubster/piolet/rockitnet
152	directconnect
153	earthstation5
154	edonkey/overnet/emule/mldonkey
155	fasttrack
156	gnutella
157	twister
158	winmx
180	p2p (other)
190	DNP3 (SCADA)
191	ICCP (SCADA)
192	IEC (SCADA)
193	MODBUS (SCADA)
194	OPC (SCADA)
199	SCADA (other)

Type code	Protocol
254	Multi-protocol
255	Other Protocol

Platform type

The following table lists the codes and descriptions for platforms.

Category code	Description
001	Windows Client Application
002	Mac OS Client Application
003	UNIX/Linux Client Application
004	Novell Client Application
075	Windows Server Application or Service
076	Mac OS Server Application or Service
077	UNIX/Linux Server Application or Service
078	Novell Server Application or Service
150	Networked Hardware Device (router, switch, printer, etc.) Application or Service
252	Multi-Platform Client Application
253	Multi-Platform Server Application or Service
254	Other Client Application
255	Other Service or Server Application

TREND MICRO INCORPORATED

225 E. John Carpenter Freeway, Suite 1500
Irving, Texas 75062 U.S.A.
Phone: +1 (817) 569-8900, Toll-free: (888) 762-8736
Email: support@trendmicro.com

www.trendmicro.com

Item Code: TPEM09844/230927