

Configure and Troubleshoot SNMP on Firepower FDM

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[SNMP v3](#)

[SNMP v2c](#)

[SNMP Configuration Removal](#)

[Verify](#)

[SNMP v3 Verification](#)

[SNMP v2c Verification](#)

[Troubleshoot](#)

[Q&A](#)

[Related Information](#)

Introduction

This document describes how to enable Simple Network Management Protocol (SNMP) on Firepower Device Management on version 6.7 with REST API.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Firepower Threat Defense (FTD) managed by Firepower Device Management (FDM) on version 6.7
- Knowledge of REST API
- Knowledge of SNMP

Components Used

Firepower Threat Defense (FTD) managed by Firepower Device Management (FDM) on version 6.7.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

What's New on 6.7

FTD Device REST API supports configuration and management of SNMP server, users, host, and host-groups. With the SNMP FTD Device REST API support in FP 6.7:

- A user can configure SNMP via FTD Device REST API to manage the network
- SNMP server, users, and host/host-groups can be added/updated or managed via FTD Device REST API.

The examples included in the document describe the configuration steps taken by FDM API Explorer.

 **Note:** SNMP can only be configured via REST API when FTD run version 6.7 and managed by FDM

Feature Overview – SNMP FTD Device REST API Support

- This feature adds new FDM URL endpoints specific to SNMP.
- These new APIs can be used to configure SNMP for polls and traps to monitor systems.
- Post SNMP configuration via APIs, the Management Information Bases (MIBs) on the Firepower devices, are available for polls or for trap notification on NMS/ SNMP Client.

SNMP API/URL Endpoints

URL	Methods	Models
/devicesettings/default/snmpservers	GET	SNMPServer
/devicesettings/default/snmpservers/{objId}	PUT, GET	SNMPServer
/object/snmphosts	POST, GET	SNMPHost
/object/snmphosts/{objId}	PUT, DELETE, GET	SNMPHost
/object/snmpusergroups	POST, GET	SNMPUserGroup
/object/snmpusergroups/{objId}	PUT, DELETE, GET	SNMPUserGroup
/object/snmpusers	POST, GET	SNMPUser
/object/snmpusers/{objId}	PUT, DELETE, GET	SNMPUser

Configure

- The SNMP host has 3 primary versions

- SNMP V1

- SNMP V2C

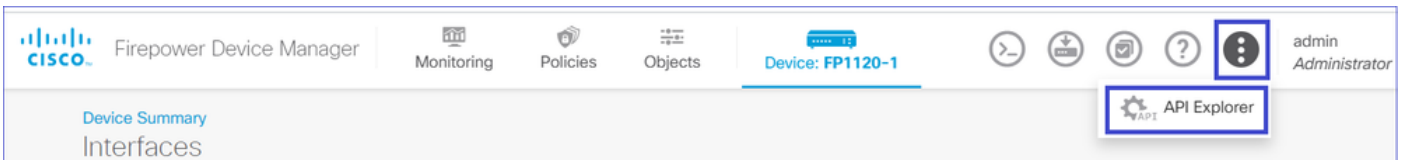
- SNMP V3

- Each of these has a specific format for “securityConfiguration”.
- For V1 and V2C: It contains a “Community String” and a “type” field that identifies the config as V1 or V2C.
- For SNMP V3: It contains a valid SNMP V3 user and a “type” field that identifies the config as V3.

SNMP v3

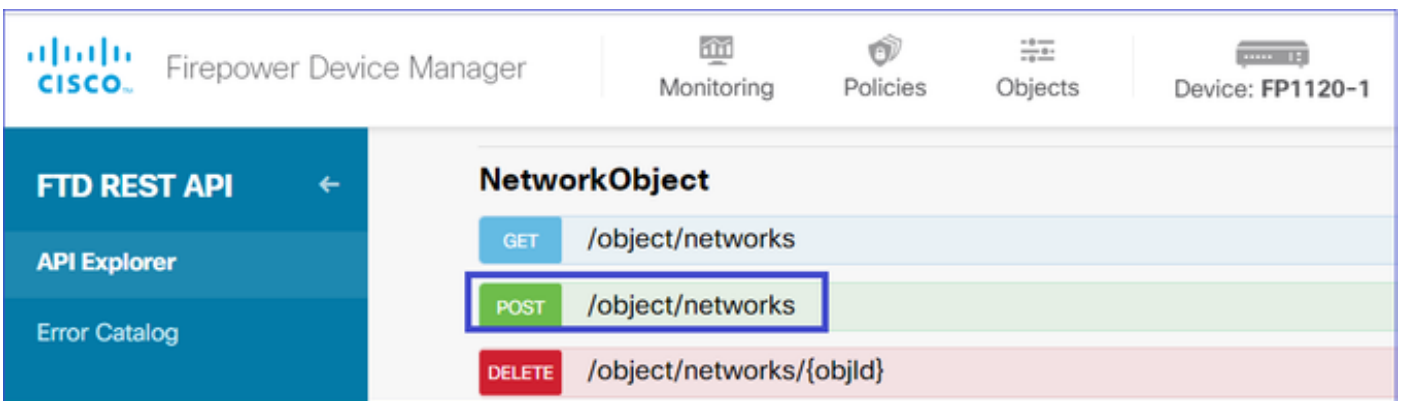
1. Access the FDM API Explorer

To access the FDM REST API Explorer from the FDM GUI select the 3 dots and then **API Explorer**. Alternatively, navigate to URL https://FDM_IP/#/api-explorer:



2. Network Object Config

Create a new network object for the SNMP host: on FDM API Explorer select NetworkObject and then POST `/object/networks`:



The SNMP Host JSON format is this. Paste this JSON into the body section and change the IP address on "value" to match the SNMP host IP address:

```
{
  "version": "null",
  "name": "snmpHost",
  "description": "SNMP Server Host",
  "subType": "HOST",
  "value": "192.168.203.61",
  "isSystemDefined": false,
  "dnsResolution": "IPV4_ONLY",
  "type": "networkobject"
}
```

The screenshot shows the Cisco Firepower Device Manager interface. The top navigation bar includes 'Monitoring', 'Policies', 'Objects', and 'Device: FP1120-1'. The left sidebar has 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'Response Content Type application/json'. Under 'Parameters', a table lists a parameter 'body' with a value of a JSON object:

```
{
  "version": "null",
  "name": "snmpHost",
  "description": "SNMP Server Host",
  "subType": "HOST",
  "value": "192.168.203.61",
  "isSystemDefined": false,
}
```

. To the right, a 'Data Type' section shows a 'Model' and an 'Example Value' which is a more detailed JSON object:

```
{
  "version": "string",
  "name": "string",
  "description": "string",
  "subType": "HOST",
  "value": "string",
  "isSystemDefined": true,
  "dnsResolution": "IPV4_ONLY",
  "id": "string",
  "type": "networkobject"
}
```

Scroll down and select the TRY IT OUT! button to execute the API call. A successful call returns Response code 200.

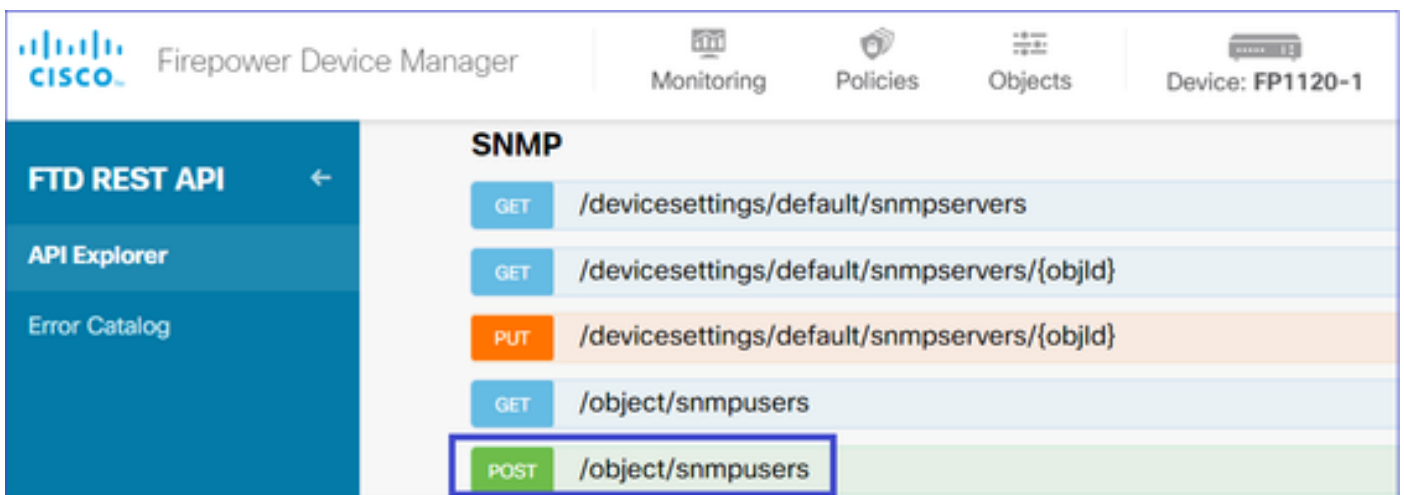


Copy the JSON data from the response body to a notepad. Later, you need to fill out the information about the SNMP host.



3. Create a new SNMPv3 user

On FDM API Explorer select SNMP and then POST /object/snmpusers

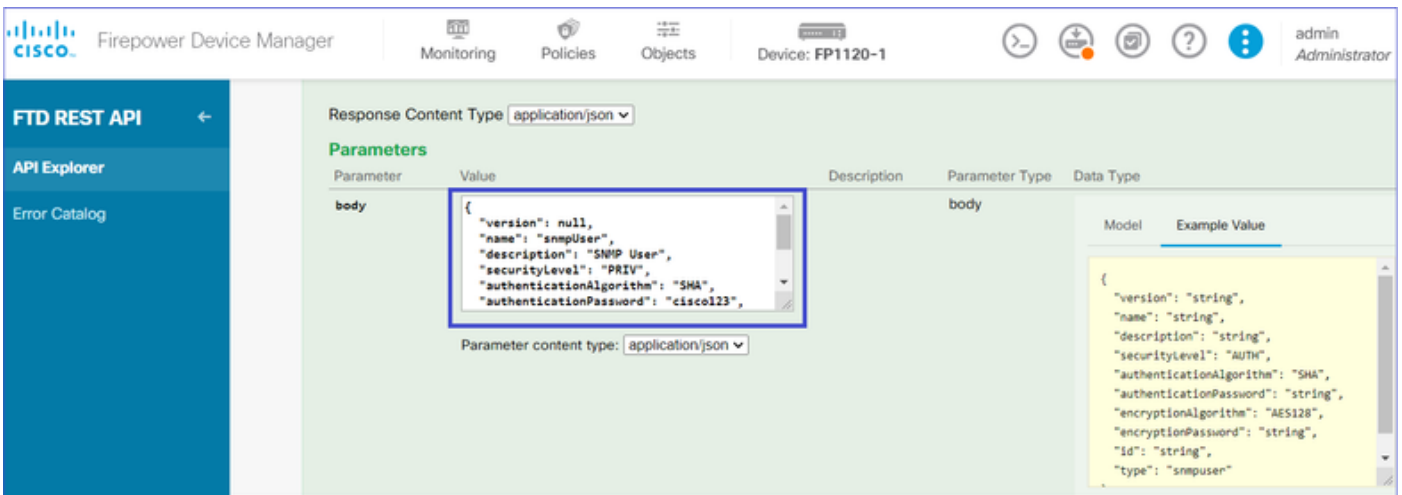


Copy this JSON data to a notepad and modify the sections that you are interested (for example, 'authenticationPassword', 'encryptionPassword' or the algorithms):

```
{
  "version": null,
  "name": "snmpUser",
  "description": "SNMP User",
  "securityLevel": "PRIV",
  "authenticationAlgorithm": "SHA",
  "authenticationPassword": "cisco123",
  "encryptionAlgorithm": "AES128",
  "encryptionPassword": "cisco123",
  "id": null,
  "type": "snmpuser"
}
```

Caution: The passwords used in the examples are for demonstration purposes only. In a production environment ensure that you use strong passwords

Copy the modified JSON data to the body section:



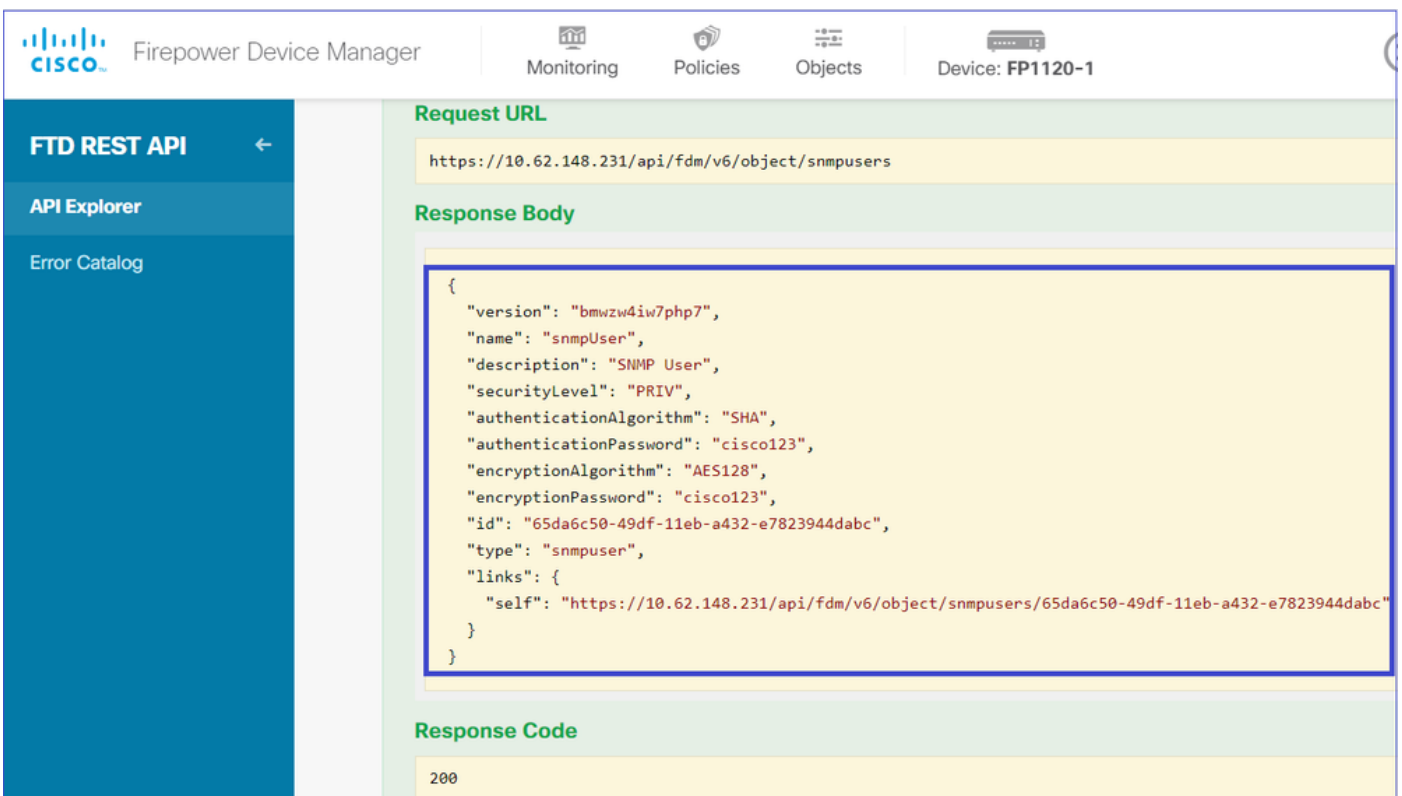
The screenshot shows the Firepower Device Manager interface. On the left, there is a sidebar with 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'Parameters' and shows a table with columns for 'Parameter', 'Value', 'Description', 'Parameter Type', and 'Data Type'. A single row is visible with 'body' as the parameter and a JSON object as the value. The JSON object is highlighted with a blue box and contains the following data:

```
{
  "version": null,
  "name": "snmpUser",
  "description": "SNMP User",
  "securityLevel": "PRIV",
  "authenticationAlgorithm": "SHA",
  "authenticationPassword": "cisco123",
}
```

Below the table, there is a 'Parameter content type' dropdown set to 'application/json'. To the right, there is a 'Data Type' section with a 'Model' and 'Example Value' tab. The 'Example Value' tab shows a JSON object with the following structure:

```
{
  "version": "string",
  "name": "string",
  "description": "string",
  "securityLevel": "AUTH",
  "authenticationAlgorithm": "SHA",
  "authenticationPassword": "string",
  "encryptionAlgorithm": "AES128",
  "encryptionPassword": "string",
  "id": "string",
  "type": "snmpuser"
}
```

Scroll down and select the **TRY IT OUT!** button to execute the API call. A successful call returns Response code 200. Copy the JSON data from the response body to a notepad. Later, you need to fill out the information about the SNMP user.



The screenshot shows the Firepower Device Manager interface. On the left, there is a sidebar with 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'Request URL' and shows the URL 'https://10.62.148.231/api/fdm/v6/object/snmpusers'. Below this, there is a 'Response Body' section with a JSON object highlighted by a blue box. The JSON object contains the following data:

```
{
  "version": "bmwz4iw7php7",
  "name": "snmpUser",
  "description": "SNMP User",
  "securityLevel": "PRIV",
  "authenticationAlgorithm": "SHA",
  "authenticationPassword": "cisco123",
  "encryptionAlgorithm": "AES128",
  "encryptionPassword": "cisco123",
  "id": "65da6c50-49df-11eb-a432-e7823944dabc",
  "type": "snmpuser",
  "links": {
    "self": "https://10.62.148.231/api/fdm/v6/object/snmpusers/65da6c50-49df-11eb-a432-e7823944dabc"
  }
}
```

Below the response body, there is a 'Response Code' section showing the value '200'.

4. Get interface information

On FDM API Explorer select Interface and then GET **/devices/default/interfaces**. You need to collect information from the interface that connects to the SNMP server.

Scroll down and select the **TRY IT OUT!** button to execute the API call. A successful call returns Response code 200. Copy the JSON data from the response body to a notepad. Later, you need to fill out information about the interface.

```

https://10.62.148.231/api/fdm/v6/devices/default/interfaces

Response Body

{
  "version": "kkpkibjlu6qro",
  "name": "inside",
  "description": null,
  "hardwareName": "Ethernet1/2",
  "monitorInterface": true,
  "ipv4": {
    "ipType": "STATIC",
    "defaultRouteUsingDHCP": false,
    "dhcpRouteMetric": null,
    "ipAddress": {
      "ipAddress": "192.168.203.71",
      "netmask": "255.255.255.0",
      "standbyIpAddress": null,
      "type": "haipv4address"
    },
    "dhcp": false,
    "addressNull": false,
    "type": "interfaceipv4"
  },
  "ipv6": {
    "enabled": false,
  }
}

Response Code

200

```

Note down the interface "version", "name", "id", and "type" from the JSON data. Example of a JSON data from interface inside:

```

<#root>

{
"version": "kkpkibjlu6qro",
"name": "inside",
"description": null,
"hardwareName": "Ethernet1/2",
"monitorInterface": true,
"ipv4": {

```

```
"ipType": "STATIC",
"defaultRouteUsingDHCP": false,
"dhcpRouteMetric": null,
"ipAddress": {
  "ipAddress": "192.168.203.71",
  "netmask": "255.255.255.0",
  "standbyIpAddress": null,
  "type": "haipv4address"
},
"dhcp": false,
"addressNull": false,
"type": "interfaceipv4"
},
"ipv6": {
  "enabled": false,
  "autoConfig": false,
  "dhcpForManagedConfig": false,
  "dhcpForOtherConfig": false,
  "enableRA": false,
  "dadAttempts": 1,
  "linkLocalAddress": {
    "ipAddress": "",
    "standbyIpAddress": "",
    "type": "haipv6address"
  },
  "ipAddresses": [
    {
      "ipAddress": "",
      "standbyIpAddress": "",
      "type": "haipv6address"
    }
  ],
  "prefixes": null,
  "type": "interfaceipv6"
},
"managementOnly": false,
"managementInterface": false,
"mode": "ROUTED",
"linkState": "UP",
"mtu": 1500,
"enabled": true,
"macAddress": null,
"standbyMacAddress": null,
"pppoe": null,
"speedType": "AUTO",
"duplexType": "AUTO",
"present": true,
"tenGigabitInterface": false,
"gigabitInterface": false,

"deviceId": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",

"type": "physicalinterface",

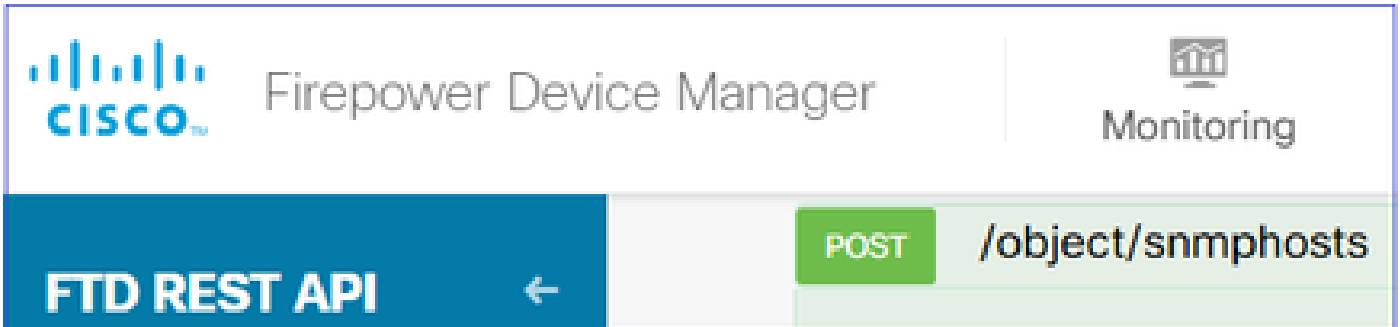
"links": {
  "self": "https://10.62.148.231/api/fdm/v6/devices/default/interfaces/fc3d07d4-49d2-11eb-85a8-65aec636a0"
}
},
```


From the JSON data, you can see interface 'inside' has this data that needs to be associated with the SNMP server:

- "version": "kkpkibjlu6qro"
- "name": "inside",
- "id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
- "type": "physicalinterface",

5. Create a new SNMPv3 host

On FDM API Explorer select SNMP and then POST `/object/snmphosts/` under SNMP



Use this JSON as a template. Copy and paste data from previous steps to the template accordingly:

```
{
"version": null,
"name": "snmpv3-host",
"description": null,
"managerAddress": {
"version": "bsha3bhghu3vmk",
"name": "snmpHost",
"id": "1d10ce6d-49de-11eb-a432-e320cd56d5af",
"type": "networkobject"
},
"pollEnabled": true,
"trapEnabled": true,
"securityConfiguration": {
"authentication": {
"version": "bmwzw4iw7php7",
"name": "snmpUser",
"id": "65da6c50-49df-11eb-a432-e7823944dabc",
"type": "snmpuser"
},
"type": "snmpv3securityconfiguration"
},
"interface": {
"version": "kkpkibjlu6qro",
"name": "inside",
"id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
"type": "physicalinterface"
},
"id": null,
"type": "snmphost"
}
```

Note:

- Replace the value in managerAddress id, type, version, and name with the information you received from Step 1
- Replace the value in authentication with the information you received from Step 2
- Replace the value in interface with the data you received from Step 3
- For SNMP2, there is no authentication, and the type is snmpv2csecurityconfiguration instead of snmpv3securityconfiguration

Copy the modified JSON data to the body section

The screenshot shows the Cisco Firepower Device Manager (FDM) REST API configuration page. The page is titled "FTD REST API" and has a sidebar with "API Explorer" and "Error Catalog". The main content area shows the "Parameters" section with a "body" parameter. The "body" parameter value is a JSON object, which is highlighted with a blue box. The JSON object is: {"version": null, "name": "snmpv3-host", "description": null, "managerAddress": {"version": "bsha3bhghu3vmk", "name": "snmpHost"}, "type": "snmpv3securityconfiguration"}. The page also shows the "Response Content Type" dropdown set to "application/json" and the "Parameter content type" dropdown set to "application/json".

Scroll down and select the **TRY IT OUT!** button to execute the API call. A successful call returns Response code 200.

FTD REST API ←

API Explorer

Error Catalog

Request URL

https://10.62.148.231/api/fdm/v6/object/snmphosts

Response Body

```
{
  "version": "gneswdadd3isp",
  "name": "snmpv3-host",
  "description": null,
  "managerAddress": {
    "version": "bsha3bhghu3vm",
    "name": "snmpHost",
    "id": "1d10ce6d-49de-11eb-a432-e320cd56d5af",
    "type": "networkobject"
  },
  "udpPort": 162,
  "pollEnabled": true,
  "trapEnabled": true,
  "securityConfiguration": {
    "authentication": {
      "version": "bmwzw4iw7php7",
      "name": "snmpUser",
      "id": "65da6c50-49df-11eb-a432-e7823944dabc",
      "type": "snmpuser"
    }
  },
}
```

Response Code

200

Navigate to FDM GUI and Deploy the changes. You can see most of the SNMP configuration:

Pending Changes
? ×

✔ Last Deployment Completed Successfully
 29 Dec 2020 02:32 PM. [See Deployment History](#)

Deployed Version (29 Dec 2020 02:32 PM)	Pending Version
+ Network Object Added: snmpHost	
-	subType: Host value: 192.168.203.61 isSystemDefined: false dnsResolution: IPV4_ONLY description: SNMP Server Host name: snmpHost
+ snmpHost Added: snmpv3-host	
-	udpPort: 162 pollEnabled: true trapEnabled: true name: snmpv3-host
snmpInterface:	inside
managerAddress:	snmpHost
securityConfiguration.authentication:	snmpUser

MORE ACTIONS ▾
CANCEL
DEPLOY NOW ▾

SNMP v2c

For v2c you don't need to create a user but you still need to:

1. Create a Network Object Config (same as described in the SNMPv3 section)
2. Get interface information (same as described in the SNMPv3 section)
3. Create a new SNMPv2c host object

This is a sample of a JSON payload that creates an SNMPv2c object:

```
{
  "version": null,
  "name": "snmpv2-Host",
  "description": null,
  "managerAddress": {
    "version": "bsha3bhghu3vmk",
    "name": "snmpv4hostgrp",
    "id": "1d10ce6d-49de-11eb-a432-e320cd56d5af",
    "type": "networkobject"
  },
  "pollEnabled": true,
  "trapEnabled": true,
  "securityConfiguration": {
    "community": "cisco123",
    "type": "snmpv2csecurityconfiguration"
  },
}
```

```

"interface": {
"version": "kkpkibjlu6qro",
"name": "inside",
"id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
"type": "physicalinterface"
},
"id": null,
"type": "snmpHost"
}

```

Use the POST method to deploy the JSON payload:

The screenshot shows the Cisco Firepower Device Manager REST API interface. The left sidebar contains 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'Parameters' and shows a table with 'Parameter' and 'Value' columns. The 'body' parameter is highlighted with a blue box and contains the following JSON payload:

```

{
  "version": null,
  "name": "snmpv2-Host",
  "description": null,
  "managerAddress": {
    "version": "bsha3bhghu3vmk",
    "name": "snmpv4hostgrp",
  }
}

```

The 'Response Content Type' is set to 'application/json' and the 'Parameter content type' is also set to 'application/json'.

Scroll down and select the TRY IT OUT! button to execute the API call. A successful call returns Response code 200.

The screenshot shows the Cisco Firepower Device Manager REST API interface displaying the results of an API call. The 'Request URL' is 'https://10.62.148.231/api/fdm/v6/object/snmpHosts'. The 'Response Body' is a JSON object:

```

{
  "udpPort": 162,
  "pollEnabled": true,
  "trapEnabled": true,
  "securityConfiguration": {
    "community": "*****",
    "type": "snmpv2csecurityconfiguration"
  },
  "interface": {
    "version": "kkpkibjlu6qro",
    "name": "inside",
    "hardwareName": "Ethernet1/2",
    "id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
    "type": "physicalinterface"
  },
  "id": "1bfd1f0-4ac6-11eb-a432-e76cd376bca7",
  "type": "snmpHost",
  "links": {
    "self": "https://10.62.148.231/api/fdm/v6/object/snmpHosts/1bfd1f0-4ac6-11eb-a432-e76cd376bca7"
  }
}

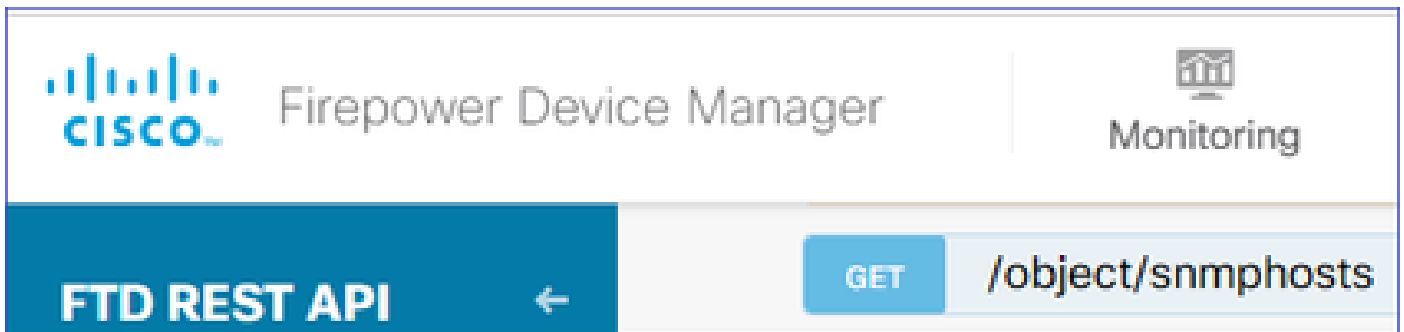
```

The 'Response Code' is 200.

SNMP Configuration Removal

Step 1.

Get the SNMP host information (SNMP > /object/snmphosts):



Scroll down and select the TRY IT OUT! button to execute the API call. A successful call returns Response code 200.

You get a list of objects. Note down the id of the snmpHost object that you want to remove:

```
<#root>
{
  "items": [
    {
      "version": "ofaasthu26ulx",
      "name": "snmpv2-Host",
      "description": null,
      "managerAddress": {
        "version": "bsha3bhghu3vm",
        "name": "snmpHost",
        "id": "1d10ce6d-49de-11eb-a432-e320cd56d5af",
        "type": "networkobject"
      },
      "udpPort": 162,
      "pollEnabled": true,
      "trapEnabled": true,
      "securityConfiguration": {
        "community": "*****",
        "type": "snmpv2csecurityconfiguration"
      },
      "interface": {
        "version": "kkpkibjlu6qro",
        "name": "inside",
        "hardwareName": "Ethernet1/2",
        "id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
        "type": "physicalinterface"
      },
      "id": "
1bfbd1f0-4ac6-11eb-a432-e76cd376bca7
",
      "type": "snmpHost",
      "links": {
        "self": "https://10.62.148.231/api/fdm/v6/object/snmpHosts/1bfbd1f0-4ac6-11eb-a432-e76cd376bca7"
      }
    }
  ]
}
```

},

Step 2.

Choose the DELETE option in **SNMP > /object/snmphosts{objId}**. Paste the id you collected in step 1:

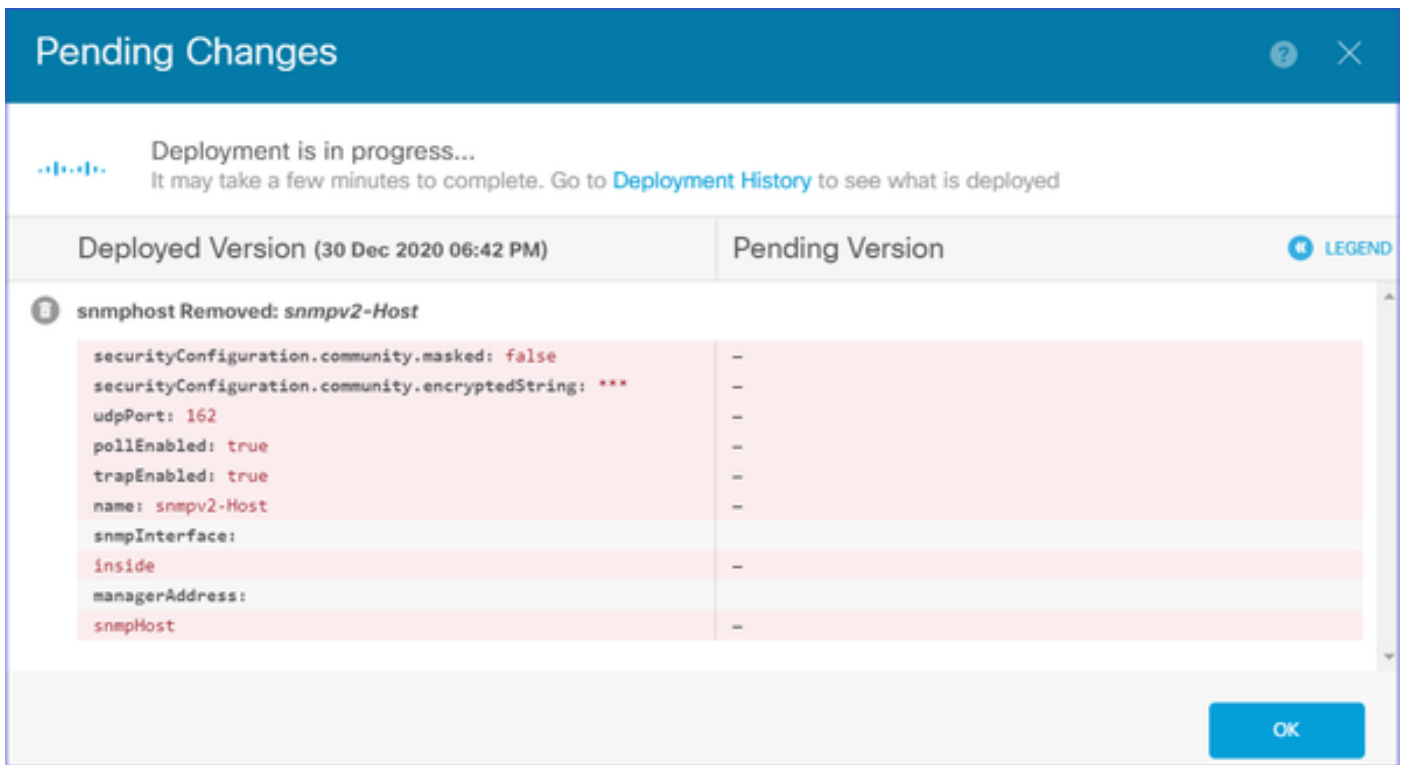
The screenshot shows the FTD REST API interface. On the left is a blue sidebar with 'FTD REST API' at the top, a back arrow, and menu items for 'API Explorer' and 'Error Catalog'. The main area has a red header with 'DELETE' and the endpoint '/object/snmphosts/{objId}'. Below this, there are sections for 'Implementation Notes' (stating the call is not allowed on the standby unit in an HA pair) and 'Parameters'. A table with two columns, 'Parameter' and 'Value', shows 'objId' with the value '1bfd1f0-4ac6-11eb-a432-e76cd376bca7' entered in a text box.

Scroll down and select the TRY IT OUT! button to execute the API call. The call returns Response code 400.

The screenshot shows the API response details. It has a 'Response Code' section with the value '400'. Below it is a 'Response Headers' section containing a JSON object with various headers: 'accept-ranges', 'cache-control', 'connection', 'content-type', 'date', 'expires', 'pragma', 'server', 'strict-transport-security', 'transfer-encoding', 'x-content-type-options', 'x-frame-options', and 'x-xss-protection'.

Step 3.

Deploy the change:



The deployment removes the host information:

```
<#root>
```

```
FP1120-1#
```

```
show run snmp-server
```

```
snmp-server group AUTH v3 auth
snmp-server group PRIV v3 priv
snmp-server group NOAUTH v3 noauth
snmp-server location null
snmp-server contact null
snmp-server community *****
```

snmpwalk for v2c fails:

```
<#root>
```

```
root@kali2:~#
```

```
snmpwalk -v2c -c cisco123 -OS 192.168.203.71
```

```
Timeout: No Response from 192.168.203.71
```

For v3 you must delete the objects in this order.

1. SNMP host (the successful return code is 204)
2. SNMP user (the successful return code is 204)


```
snmp-server group NOAUTH v3 noauth
snmp-server user snmpUser PRIV v3
engineID 80000009febdf0129a799ef469aba2d5fcf1bfd7e86135a1f8
    encrypted auth sha ca:1b:18:f3:62:b1:63:7e:92:34:92:b3:cf:54:86:f9:8e:2a:4c:fd priv aes 128 ca:1b:18:f3:62:b1:63:7e:92:34:92:b3:cf:54:86:f9:8e:2a:4c:fd

snmp-server listen-port 161

snmp-server host inside 192.168.203.61 version 3 snmpUser udp-port 162

snmp-server location null
snmp-server contact null
snmp-server community *****
snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
no snmp-server enable traps syslog
no snmp-server enable traps ipsec start stop
no snmp-server enable traps entity config-change fru-insert fru-remove fan-failure power-supply power-supply-failure
no snmp-server enable traps memory-threshold
no snmp-server enable traps interface-threshold
no snmp-server enable traps remote-access session-threshold-exceeded
no snmp-server enable traps connection-limit-reached
no snmp-server enable traps cpu threshold rising
no snmp-server enable traps ikev2 start stop
no snmp-server enable traps nat packet-discard
no snmp-server enable traps config
no snmp-server enable traps failover-state
no snmp-server enable traps cluster-state
snmp-server enable oid mempool
snmp-server enable
```

snmpwalk test

<#root>

root@kali2:~#

```
snmpwalk -v3 -l authPriv -u snmpUser -a SHA -A cisco123 -x AES -X cisco123 192.168.203.71
```

```
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco Firepower Threat Defense, Version 6.7.0 (Build 65), ASA Version 9.12(1)K9"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.9.1.2663
iso.3.6.1.2.1.1.3.0 = Timeticks: (1616700) 4:29:27.00
iso.3.6.1.2.1.1.4.0 = STRING: "null"
iso.3.6.1.2.1.1.5.0 = STRING: "FP1120-1"
iso.3.6.1.2.1.1.6.0 = STRING: "null"
iso.3.6.1.2.1.1.7.0 = INTEGER: 4
...
```

SNMP v2c Verification

<#root>

```
FP1120-1#
```

```
show run snmp-server
```

```
snmp-server host inside 192.168.203.61 community ***** version 2c
```

```
snmp-server location null
snmp-server contact null
snmp-server community *****
```

snmpwalk for v2c:

```
<#root>
```

```
root@kali2:~#
```

```
snmpwalk -v2c -c cisco123 -OS 192.168.203.71
```

```
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco Firepower Threat Defense, Version 6.7.0 (Build 65), ASA Version 9.
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.9.1.2663
iso.3.6.1.2.1.1.3.0 = Timeticks: (10482200) 1 day, 5:07:02.00
iso.3.6.1.2.1.1.4.0 = STRING: "null"
iso.3.6.1.2.1.1.5.0 = STRING: "FP1120-1"
iso.3.6.1.2.1.1.6.0 = STRING: "null"
iso.3.6.1.2.1.1.7.0 = INTEGER: 4
```

Troubleshoot

Enable capture with trace on the firewall:

```
<#root>
```

```
FP1120-1#
```

```
capture CAPI trace interface inside match udp any any eq snmp
```

Use the snmpwalk tool and verify you can see the packets:

```
<#root>
```

```
FP1120-1#
```

```
show capture
```

```
capture CAPI type raw-data trace interface inside
```

```
[Capturing - 3137 bytes]
```

```
match udp any any eq snmp
```

The capture contents:

```
<#root>
```

```
FP1120-1#
```

```
show capture CAPI
```

```
154 packets captured
```

```
 1: 17:04:16.720131      192.168.203.61.51308 > 192.168.203.71.161:  udp 39
 2: 17:04:16.722252      192.168.203.71.161 > 192.168.203.61.51308:  udp 119
 3: 17:04:16.722679      192.168.203.61.51308 > 192.168.203.71.161:  udp 42
 4: 17:04:16.756400      192.168.203.71.161 > 192.168.203.61.51308:  udp 51
 5: 17:04:16.756918      192.168.203.61.51308 > 192.168.203.71.161:  udp 42
```

Verify that the SNMP server statistics counters show SNMP Get or Get-next requests and responses:

```
<#root>
```

```
FP1120-1#
```

```
show snmp-server statistics
```

```
62 SNMP packets input
```

```
0 Bad SNMP version errors
0 Unknown community name
0 Illegal operation for community name supplied
0 Encoding errors
```

```
58 Number of requested variables
```

```
0 Number of altered variables
0 Get-request PDUs
```

```
58 Get-next PDUs
```

```
0 Get-bulk PDUs
0 Set-request PDUs (Not supported)
```

```
58 SNMP packets output
```

```
0 Too big errors (Maximum packet size 1500)
0 No such name errors
0 Bad values errors
0 General errors
```

58 Response PDUs

0 Trap PDUs

Trace an ingress packet. The packet is UN-NAT to the internal NLP interface:

<#root>

FP1120-1#

show capture CAPI packet-number 1 trace

30 packets captured

1: 17:04:16.720131 192.168.203.61.51308 > 192.168.203.71.

161

: udp 39

Phase: 1

Type: CAPTURE

Subtype:

Result: ALLOW

Config:

Additional Information:

MAC Access list

Phase: 2

Type: ACCESS-LIST

Subtype:

Result: ALLOW

Config:

Implicit Rule

Additional Information:

MAC Access list

Phase: 3

Type: UN-NAT

Subtype: static

Result: ALLOW

Config:

Additional Information:

NAT divert to egress interface nlp_int_tap(vrfid:0)

Untranslate 192.168.203.71/161 to 169.254.1.3/4161

Phase: 4

Type: ACCESS-LIST

Subtype:

Result: ALLOW

Config:

Implicit Rule

Additional Information:

Phase: 5

Type: NAT

Subtype: per-session

Result: ALLOW

Config:

Additional Information:

Phase: 6

Type: IP-OPTIONS

Subtype:

Result: ALLOW

Config:

Additional Information:

Phase: 7

Type: NAT

Subtype: rpf-check

Result: ALLOW

Config:

Additional Information:

Phase: 8

Type: NAT

Subtype: per-session

Result: ALLOW

Config:

Additional Information:

Phase: 9

Type: FLOW-CREATION

Subtype:

Result: ALLOW

Config:

Additional Information:

New flow created with id 1078, packet dispatched to next module

Phase: 10

Type: INPUT-ROUTE-LOOKUP-FROM-OUTPUT-ROUTE-LOOKUP

Subtype: Resolve Preferred Egress interface

Result: ALLOW

Config:

Additional Information:

Found next-hop 169.254.1.3 using egress ifc nlp_int_tap(vrfid:0)

Phase: 11

Type: ADJACENCY-LOOKUP

Subtype: Resolve Nexthop IP address to MAC

Result: ALLOW

Config:

Additional Information:

Found adjacency entry for Next-hop 169.254.1.3 on interface nlp_int_tap

Adjacency :Active

MAC address 3208.e2f2.b5f9 hits 0 reference 1

Result:

```
input-interface: inside(vrfid:0)
```

```
input-status: up  
input-line-status: up
```

```
output-interface: nlp_int_tap(vrfid:0)
```

```
output-status: up  
output-line-status: up
```

```
Action: allow
```

The NAT rule is deployed automatically as a part of the SNMP configuration:

```
<#root>
```

```
FP1120-1#
```

```
show nat
```

```
Manual NAT Policies (Section 1)
```

```
1 (nlp_int_tap) to (inside) source dynamic nlp_client_0_192.168.203.61_intf4 interface destination static  
translate_hits = 0, untranslate_hits = 0
```

```
Auto NAT Policies (Section 2)
```

```
...
```

```
2 (nlp_int_tap) to (inside) source static nlp_server_0_snmp_intf4 interface service udp 4161 snmp
```

```
translate_hits = 0, untranslate_hits = 2
```

In the backend port UDP 4161 listens for SNMP traffic:

```
<#root>
```

```
>
```

```
expert
```

```
admin@FP1120-1:~$
```

```
sudo netstat -an | grep 4161
```

```
Password:
```

```
udp 0 0 169.254.1.3:4161 0.0.0.0:*  
udp6 0 0 fd00:0:0:1::3:4161 :::*
```

In a case of incorrect/incomplete configuration the ingress SNMP packet is dropped since there is no UN-NAT phase:

<#root>

FP1120-1#

show cap CAPI packet-number 1 trace

6 packets captured

1: 18:36:35.868485 192.168.203.61.50105 > 192.168.203.71.

161

: udp 42

Phase: 1

Type: CAPTURE

Subtype:

Result: ALLOW

Config:

Additional Information:

MAC Access list

Phase: 2

Type: ACCESS-LIST

Subtype:

Result: ALLOW

Config:

Implicit Rule

Additional Information:

MAC Access list

Phase: 3

Type: ROUTE-LOOKUP

Subtype: No ECMP load balancing

Result: ALLOW

Config:

Additional Information:

Destination is locally connected. No ECMP load balancing.

Found next-hop 192.168.203.71 using egress ifc identity(vrfid:0)

Phase: 4

Type: NAT

Subtype: per-session

Result: ALLOW

Config:

Additional Information:

Phase: 5

Type: ACCESS-LIST

Subtype:

Result: DROP

Config:
Implicit Rule
Additional Information:

Result:
input-interface: inside(vrfid:0)
input-status: up
input-line-status: up
Action: drop

Drop-reason: (acl-drop) Flow is denied by configured rule, Drop-location: frame 0x0000557415b6347d flow

FTD LINA syslogs show that the ingress packet is discarded:

<#root>

FP1120-1#

show log | include 161

Dec 30 2020 18:36:38: %FTD-7-710005: UDP request discarded from 192.168.203.61/50105 to inside:192.168.
Dec 30 2020 18:36:39: %FTD-7-710005: UDP request discarded from 192.168.203.61/50105 to inside:192.168.

Q&A

Q. Can I use the FTD management interface to send SNMP messages?

No, this is not currently supported.

Related enhancement defect: <https://bst.cloudapps.cisco.com/bugsearch/bug/CSCvu48012>

Related Information

- [Cisco Firepower Threat Defense Configuration Guide for Firepower Device Manager, Version 6.7](#)
- [Cisco Firepower Threat Defense REST API Guide](#)
- [Cisco Firepower Release Notes, Version 6.7.0](#)