

firepower FDM에서 SNMP 구성 및 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[SNMP v3](#)

[SNMP v2c](#)

[SNMP 컨피그레이션 제거](#)

[다음을 확인합니다.](#)

[SNMP v3 확인](#)

[SNMP v2c 확인](#)

[문제 해결](#)

[질문과 대답](#)

[관련 정보](#)

소개

이 문서에서는 REST API를 사용하는 버전 6.7의 Firepower 디바이스 관리에서 SNMP(Simple Network Management Protocol)를 활성화하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 버전 6.7의 FDM(firepower Device Management)에서 관리되는 FTD(Firepower Threat Defense)
- REST API 지식
- SNMP 지식

사용되는 구성 요소

버전 6.7의 FDM(firepower Device Management)에서 관리되는 FTD(Firepower Threat Defense).

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

6.7의 새로운 기능

FTD 디바이스 REST API는 SNMP 서버, 사용자, 호스트, 호스트 그룹의 컨피그레이션 및 관리를 지원합니다. FP 6.7에서 SNMP FTD 디바이스 REST API 지원:

- 사용자는 FTD Device REST API를 통해 SNMP를 구성하여 네트워크를 관리할 수 있습니다
- SNMP 서버, 사용자, 호스트/호스트 그룹은 FTD 디바이스 REST API를 통해 추가/업데이트하거나 관리할 수 있습니다.

이 문서에 포함된 예에서는 FDM API 탐색기에서 수행하는 구성 단계를 설명합니다.

 참고: FTD에서 버전 6.7을 실행하고 FDM에서 관리하는 경우 SNMP는 REST API를 통해서만 구성할 수 있습니다

기능 개요 - SNMP FTD 디바이스 REST API 지원

- 이 기능은 SNMP에 특정한 새 FDM URL 엔드포인트를 추가합니다.
- 이러한 새로운 API를 사용하여 시스템을 모니터링하기 위한 폴링과 트랩에 대한 SNMP를 구성할 수 있습니다.
- API를 통한 사후 SNMP 컨피그레이션, Firepower 디바이스의 MIB(Management Information Base)는 폴링 또는 NMS/SNMP 클라이언트의 트랩 알림에 사용할 수 있습니다.

SNMP API/URL 엔드포인트

URL	방법	모델
/devicesettings/default/snmpservers	가져오기	SNMP서버
/devicesettings/default/snmpservers/{objId}	퍼팅, 획득	SNMP서버
/object/snmphosts	게시물, 가져오기	스니프오스트
/object/snmphosts/{objId}	PUT, DELETE, GET	스니프오스트
/object/snmpusergroups	게시물, 가져오기	SNMPserGroup
/object/snmpusergroups/{objId}	PUT, DELETE, GET	SNMPserGroup
/object/snmpusers	게시물, 가져오기	SNMPUser

/object/snmpusers/{objId}	PUT, DELETE, GET	SNMPUser
---------------------------	------------------	----------

구성

- SNMP 호스트에는 3개의 기본 버전이 있습니다

- SNMP V1

- SNMP V2C

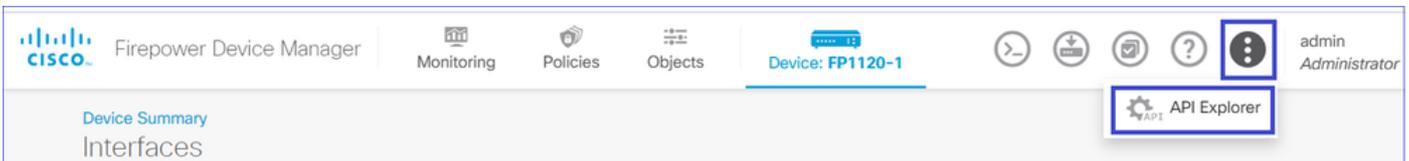
- SNMP V3

- 각 에는 "securityConfiguration"에 대한 특정 형식이 있습니다.
- V1 및 V2C의 경우: "Community String(커뮤니티 문자열)" 및 컨피그레이션을 V1 또는 V2C로 식별하는 "type(유형)" 필드가 포함됩니다.
- SNMP V3의 경우: 유효한 SNMP V3 사용자 및 구성을 V3으로 식별하는 "type" 필드가 포함되어 있습니다.

SNMP v3

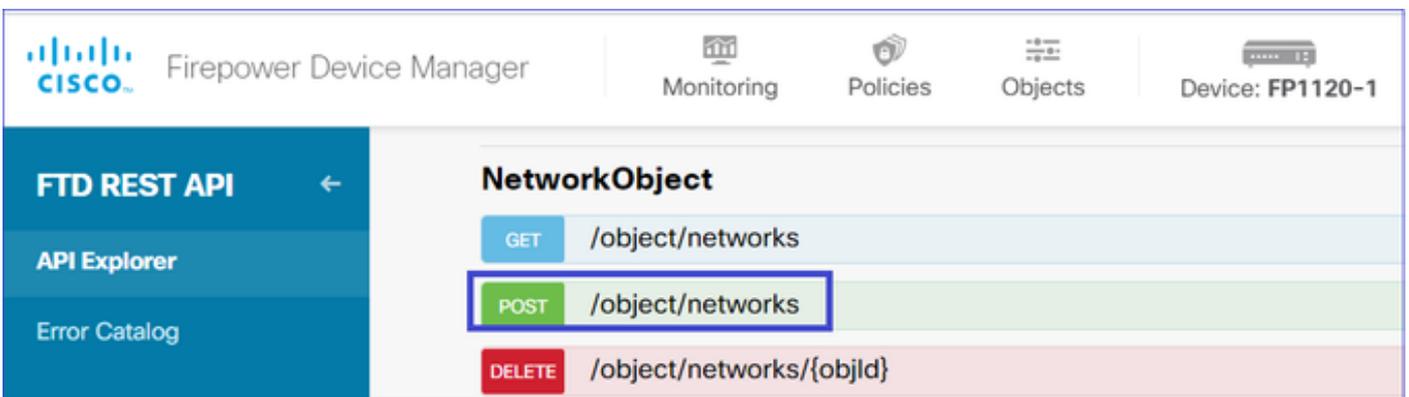
1. FDM API 탐색기 액세스

FDM GUI에서 FDM REST API 탐색기에 액세스하려면 3개의 점을 선택한 다음 API 탐색기를 선택합니다. 또는 URL https://FDM_IP/#/api-explorer으로 [이동합니다](#).



2. 네트워크 개체 컨피그레이션

SNMP 호스트에 대한 새 네트워크 객체를 생성합니다. FDM API 탐색기에서 NetworkObject를 선택한 다음 POST/object/networks를 선택합니다.

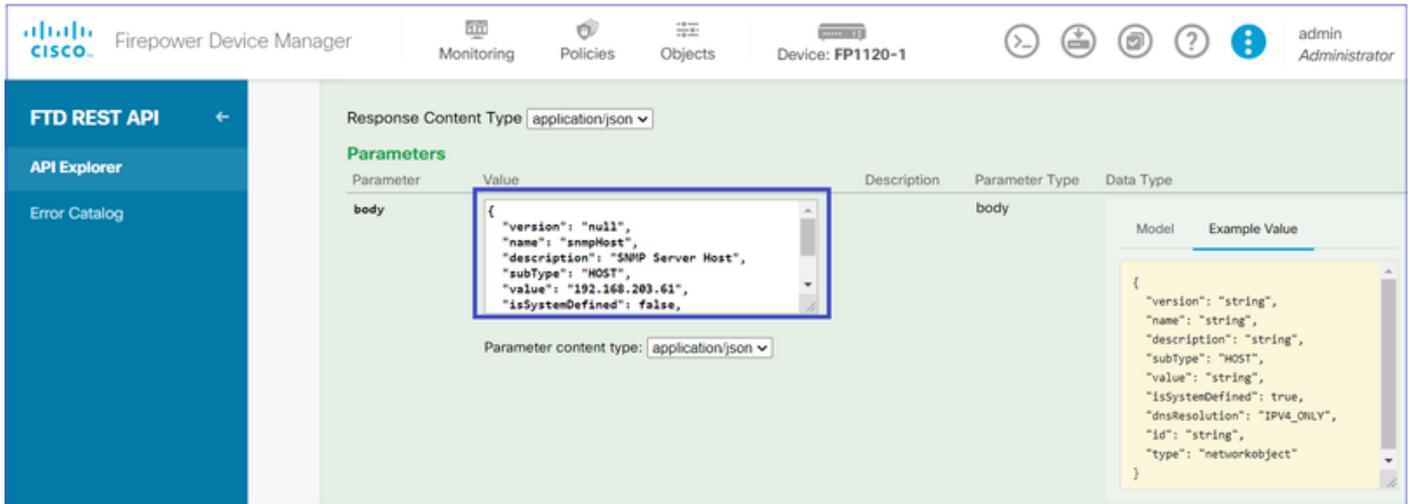


SNMP 호스트 JSON 형식은 다음과 같습니다. 이 JSON을 본문 섹션에 붙여넣고 "value"의 IP 주소를 SNMP 호스트 IP 주소와 일치하도록 변경합니다.

```

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

```



아래로 스크롤하여 TRY IT OUT! 버튼을 선택하여 API 호출을 실행합니다. 호출이 성공하면 응답 코드 200이 반환됩니다.

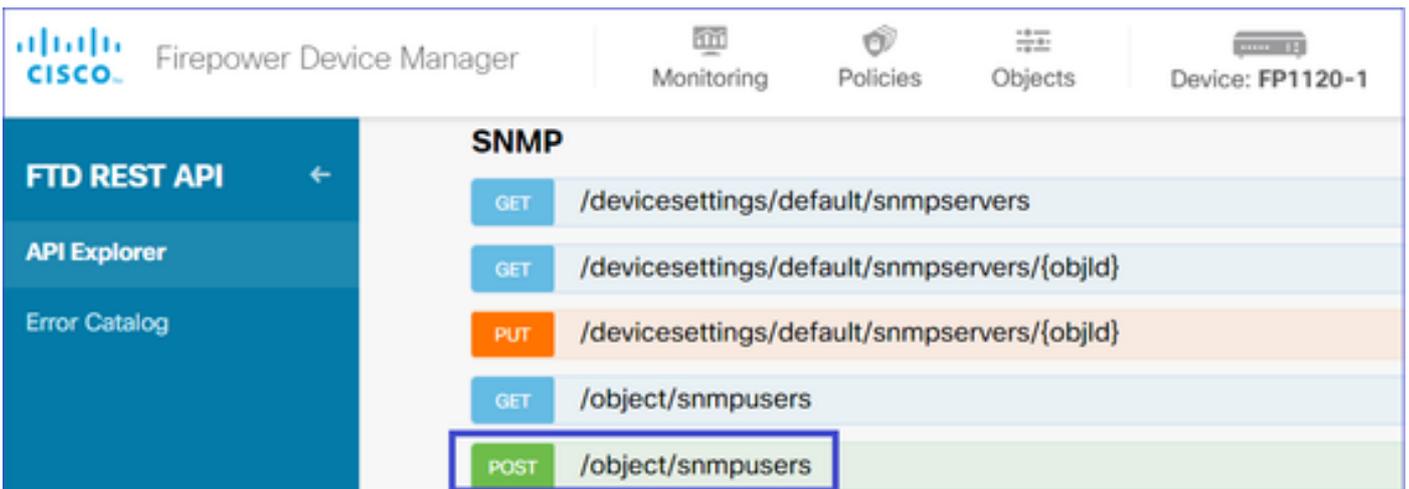


응답 본문의 JSON 데이터를 메모장에 복사합니다. 나중에 SNMP 호스트에 대한 정보를 입력해야 합니다.



3. 새 SNMPv3 사용자 생성

FDM API 탐색기에서 SNMP, POST/object/snmpusers 순으로 선택합니다

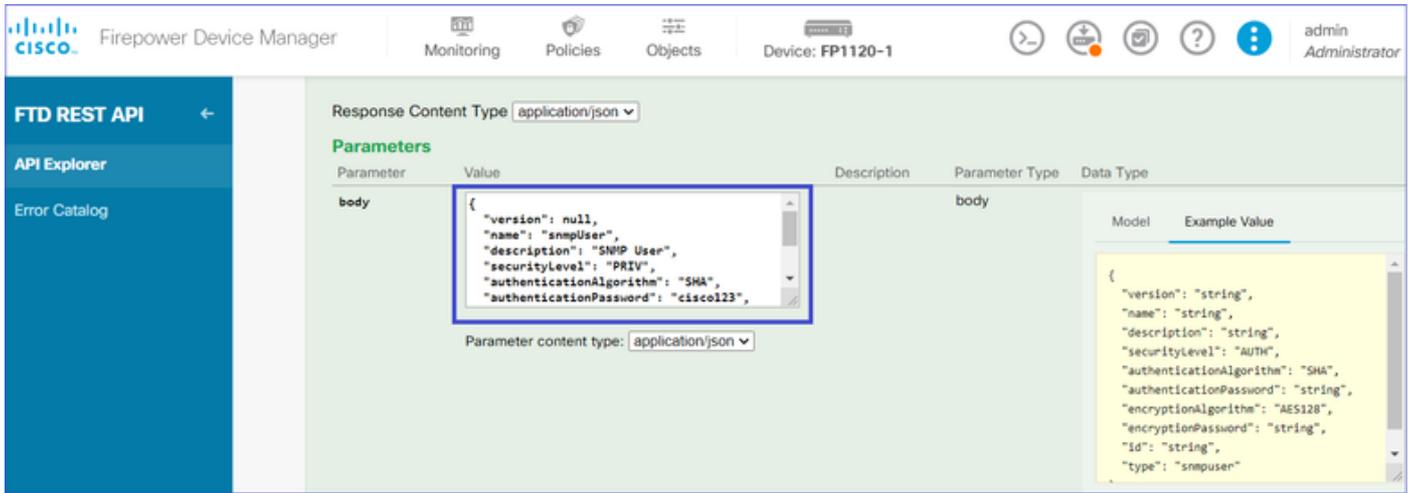


이 JSON 데이터를 메모장에 복사하고 관심 있는 섹션(예: "authenticationPassword", "encryptionPassword" 또는 알고리즘)을 수정합니다.

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

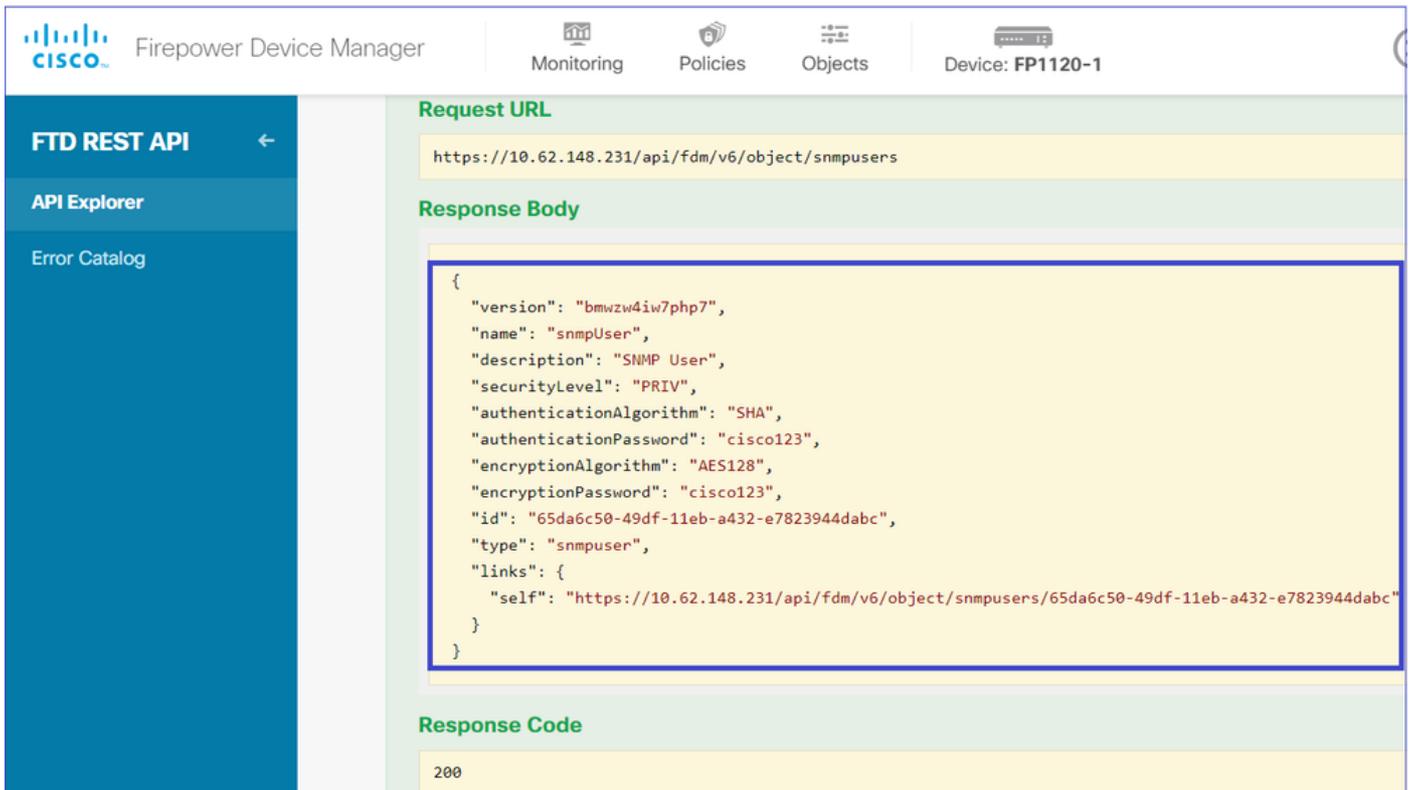
⚠ 주의: 예에서 사용되는 비밀번호는 데모용으로만 사용됩니다. 프로덕션 환경에서는 강력한 비밀번호를 사용해야 합니다

수정된 JSON 데이터를 본문 섹션으로 복사합니다.



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 parameter named 'body' is highlighted with a blue box, showing a JSON object with fields like 'version', 'name', 'description', 'securityLevel', 'authenticationAlgorithm', and 'authenticationPassword'. Below the table, there is a 'Parameter content type' dropdown set to 'application/json'. On the right, there is a 'Model' section with an 'Example Value' field containing a JSON object with similar fields.

아래로 스크롤하여 API 호출을 실행하려면 TRY IT OUT! 버튼을 선택합니다. 호출이 성공하면 응답 코드 200이 반환됩니다. 응답 본문의 JSON 데이터를 메모장에 복사합니다. 나중에 SNMP 사용자에 대한 정보를 입력해야 합니다.



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 that, there is a 'Response Body' section with a JSON object containing fields like 'version', 'name', 'description', 'securityLevel', 'authenticationAlgorithm', 'authenticationPassword', 'encryptionAlgorithm', 'encryptionPassword', 'id', 'type', and 'links'. Below the response body, there is a 'Response Code' section showing '200'.

4. 인터페이스 정보 가져오기

FDM API 탐색기에서 Interface를 선택한 다음 GET/devices/default/interfaces를 선택합니다. SNMP 서버에 연결하는 인터페이스에서 정보를 수집해야 합니다.

아래로 스크롤하여 API 호출을 실행하려면 TRY IT OUT! 버튼을 선택합니다. 호출이 성공하면 응답 코드 200이 반환됩니다. 응답 본문의 JSON 데이터를 메모장에 복사합니다. 나중에 인터페이스에 대한 정보를 입력해야 합니다.

```

"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,

```

JSON 데이터에서 인터페이스 "version", "name", "id" 및 "type"을 기록해 둡니다. 인터페이스 내부의 JSON 데이터 예:

```

<#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,

"id": "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"
}
```

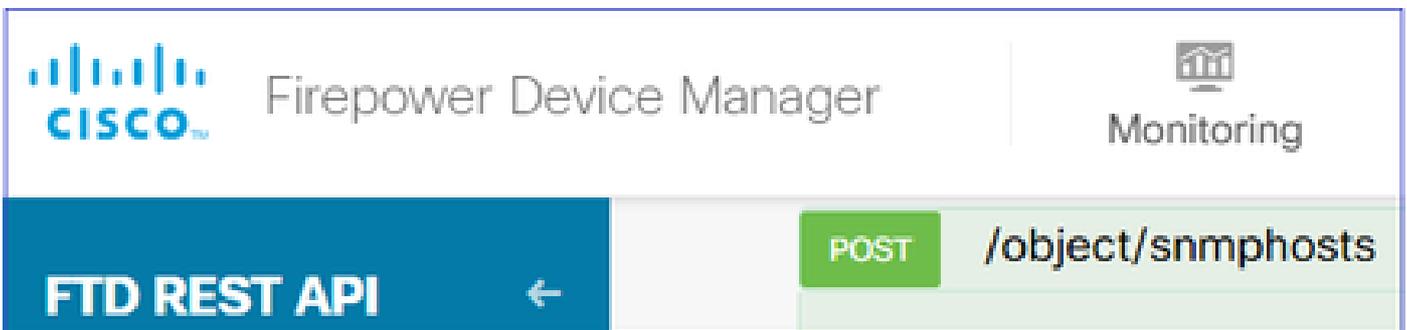
},

JSON 데이터에서 'inside' 인터페이스에 SNMP 서버와 연결해야 하는 이 데이터가 있음을 확인할 수 있습니다.

- "버전": "kpkibjlu6qro"
- "이름": "내부",
- "id": "fc3d07d4-49d2-11eb-85a8-65aec636a0fc",
- "type": "physicalinterface",

5. 새 SNMPv3 호스트 생성

FDM API 탐색기에서 SNMP를 선택한 다음 SNMP 아래에서 POST/object/snmphosts/를 선택합니다



이 JSON을 템플릿으로 사용합니다. 이전 단계의 데이터를 템플릿에 적절하게 복사하여 붙여넣습니다.

```
{
  "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"
}
```

참고:

- managerAddress id, type, version 및 name의 값을 1단계에서 받은 정보로 바꿉니다
- 인증의 값을 2단계에서 받은 정보로 대체합니다.
- 인터페이스의 값을 3단계에서 수신한 데이터로 바꿉니다.
- SNMP2의 경우 인증이 없으며 유형은 snmpv3securityconfiguration 대신 snmpv2csecurityconfiguration입니다

수정된 JSON 데이터를 본문 섹션으로 복사합니다.

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 displays the REST API configuration for a device. The 'Response Content Type' is set to 'application/json'. A table titled 'Parameters' has columns for 'Parameter', 'Value', and 'Description'. The 'body' parameter is highlighted with a blue box, and its value is a JSON object: {"version": null, "name": "snmpv3-host", "description": null, "managerAddress": {"version": "bsha3bhghu3vmk", "name": "snmpHost"}, ...}. Below the table, 'Parameter content type' is set to 'application/json'.

아래로 스크롤하여 API 호출을 실행하려면 TRY IT OUT! 버튼을 선택합니다. 호출이 성공하면 응답 코드 200이 반환됩니다.

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

FDM GUI로 이동하여 변경 사항을 구축합니다. 대부분의 SNMP 컨피그레이션을 볼 수 있습니다.

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 LEGEND																				
<p>Network Object Added: snmpHost</p> <table border="1"> <tr><td>-</td><td>subType: Host</td></tr> <tr><td>-</td><td>value: 192.168.203.61</td></tr> <tr><td>-</td><td>isSystemDefined: false</td></tr> <tr><td>-</td><td>dnsResolution: IPV4_ONLY</td></tr> <tr><td>-</td><td>description: SNMP Server Host</td></tr> <tr><td>-</td><td>name: snmpHost</td></tr> </table>		-	subType: Host	-	value: 192.168.203.61	-	isSystemDefined: false	-	dnsResolution: IPV4_ONLY	-	description: SNMP Server Host	-	name: snmpHost								
-	subType: Host																				
-	value: 192.168.203.61																				
-	isSystemDefined: false																				
-	dnsResolution: IPV4_ONLY																				
-	description: SNMP Server Host																				
-	name: snmpHost																				
<p>snmpHost Added: snmpv3-host</p> <table border="1"> <tr><td>-</td><td>udpPort: 162</td></tr> <tr><td>-</td><td>pollEnabled: true</td></tr> <tr><td>-</td><td>trapEnabled: true</td></tr> <tr><td>-</td><td>name: snmpv3-host</td></tr> <tr><td colspan="2">snmpInterface:</td></tr> <tr><td>-</td><td>inside</td></tr> <tr><td colspan="2">managerAddress:</td></tr> <tr><td>-</td><td>snmpHost</td></tr> <tr><td colspan="2">securityConfiguration.authentication:</td></tr> <tr><td>-</td><td>snmpUser</td></tr> </table>		-	udpPort: 162	-	pollEnabled: true	-	trapEnabled: true	-	name: snmpv3-host	snmpInterface:		-	inside	managerAddress:		-	snmpHost	securityConfiguration.authentication:		-	snmpUser
-	udpPort: 162																				
-	pollEnabled: true																				
-	trapEnabled: true																				
-	name: snmpv3-host																				
snmpInterface:																					
-	inside																				
managerAddress:																					
-	snmpHost																				
securityConfiguration.authentication:																					
-	snmpUser																				

MORE ACTIONS ▼ CANCEL DEPLOY NOW ▼

SNMP v2c

v2c의 경우 사용자를 생성할 필요는 없지만 다음을 수행해야 합니다.

1. 네트워크 개체 컨피그레이션 생성(SNMPv3 섹션에 설명된 대로)
2. 인터페이스 정보 가져오기(SNMPv3 섹션에 설명된 것과 동일)
3. 새 SNMPv2c 호스트 개체 만들기

다음은 SNMPv2c 객체를 생성하는 JSON 페이로드의 샘플입니다.

```
{
  "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": "snmpghost"
}

```

POST 메서드를 사용하여 JSON 페이로드를 배포합니다.

The screenshot shows the Cisco Firepower Device Manager interface. The left sidebar contains 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'FTD REST API' and shows a configuration for a REST API endpoint. The 'Response Content Type' is set to 'application/json'. Under 'Parameters', there is a table with one row: 'body'. The value for 'body' is a JSON object:


```

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

```

 The 'Parameter content type' is also set to 'application/json'.

아래로 스크롤하여 TRY IT OUT! 버튼을 선택하여 API 호출을 실행합니다. 호출이 성공하면 응답 코드 200이 반환됩니다.

The screenshot shows the results of the API call. The 'Request URL' is 'https://10.62.148.231/api/fdm/v6/object/snmpghosts'. 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": "snmpghost",
  "links": {
    "self": "https://10.62.148.231/api/fdm/v6/object/snmpghosts/1bfd1f0-4ac6-11eb-a432-e76cd376bca7"
  }
}

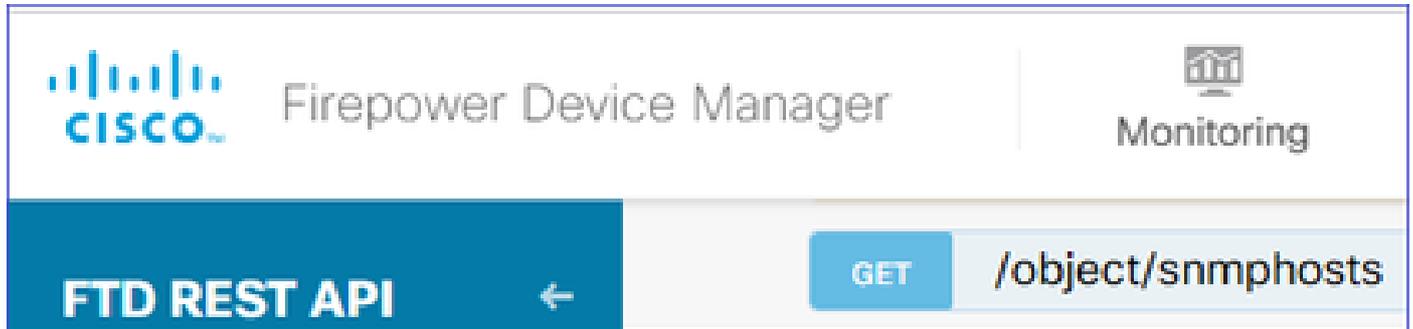
```

 The 'Response Code' is 200.

SNMP 컨피그레이션 제거

1단계.

SNMP 호스트 정보(SNMP > /object/snmphosts)를 가져옵니다.



아래로 스크롤하여 TRY IT OUT! 버튼을 선택하여 API 호출을 실행합니다. 호출이 성공하면 응답 코드 200이 반환됩니다.

개체 목록이 표시됩니다. 제거할 snmpHost 객체의 ID를 기록해 둡니다.

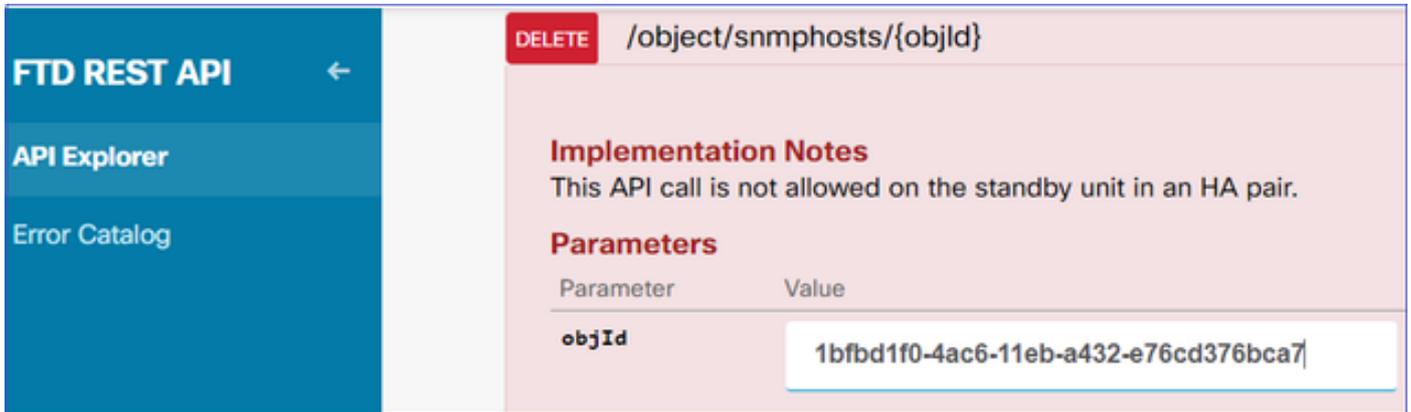
```
<#root>
```

```
{
  "items": [
    {
      "version": "ofaasthu26u1x",
      "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"
```

```
}  
},
```

2단계.

SNMP > /object/snmphosts{objId}에서 DELETE 옵션을 선택합니다. 1단계에서 수집한 ID를 붙여넣습니다.



The screenshot shows the FTD REST API interface. On the left is a blue sidebar with 'FTD REST API', 'API Explorer', and 'Error Catalog'. The main area is titled 'DELETE /object/snmphosts/{objId}'. It includes 'Implementation Notes' stating the call is not allowed on the standby unit in an HA pair, and 'Parameters' with a table:

Parameter	Value
objId	1bfbd1f0-4ac6-11eb-a432-e76cd376bca7

아래로 스크롤하여 TRY IT OUT! 버튼을 선택하여 API 호출을 실행합니다. 이 호출은 응답 코드 400을 반환합니다.

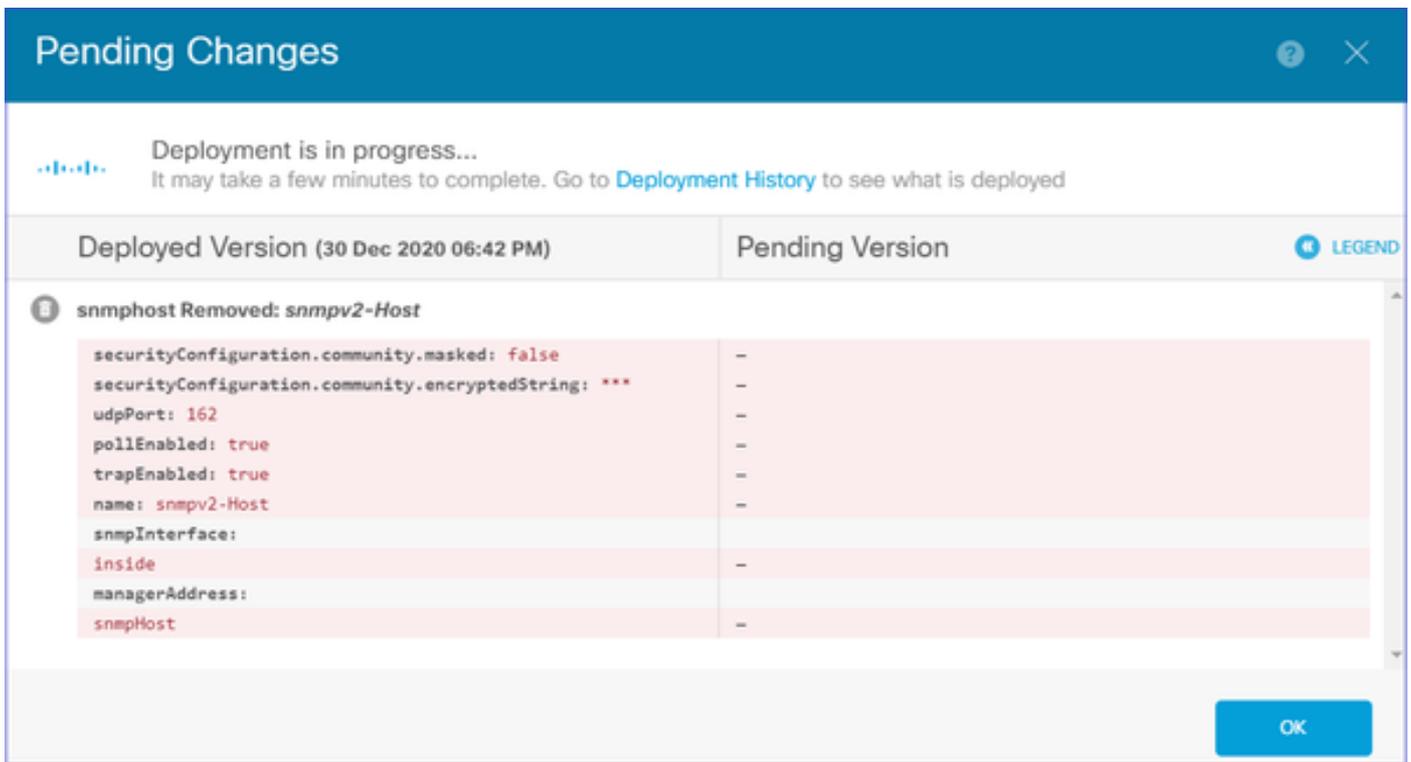


The screenshot shows the response details for the API call. It displays a 'Response Code' of 400 and a list of 'Response Headers' in JSON format:

```
{  
  "accept-ranges": "bytes",  
  "cache-control": "no-cache, no-store",  
  "connection": "close",  
  "content-type": "application/json;charset=UTF-8",  
  "date": "Wed, 30 Dec 2020 18:00:41 GMT",  
  "expires": "0",  
  "pragma": "no-cache",  
  "server": "Apache",  
  "strict-transport-security": "max-age=63072000; includeSubdomains; preload, max-age=31536000 ; includeSubDomains",  
  "transfer-encoding": "chunked",  
  "x-content-type-options": "nosniff",  
  "x-frame-options": "SAMEORIGIN, SAMEORIGIN",  
  "x-xss-protection": "1; mode=block"  
}
```

3단계.

변경 사항 구축:



구축에서 호스트 정보를 제거합니다.

```
<#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 *****
```

v2c용 snmpwalk 실패:

```
<#root>
```

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

```
snmpwalk -v2c -c cisco123 -Os 192.168.203.71
```

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

v3의 경우 이 순서대로 객체를 삭제해야 합니다.

1. SNMP 호스트(성공적인 반환 코드는 204임)

2. SNMP 사용자(성공적인 반환 코드는 204임)

개체를 잘못된 순서로 삭제하려고 하면 다음 오류가 발생합니다.

```
<#root>
{
  "error": {
    "severity": "ERROR",
    "key": "Validation",
    "messages": [
      {
        "description": "You cannot delete the object because it contains SNMPHost: snmpv3-host2, SNMPHost: snmpv3-host1.
        You must remove the object from all parts of the configuration before you can delete it.",
        "code": "deleteObjWithRel",
        "location": ""
      }
    ]
  }
}
```

다음을 확인합니다.

SNMP v3 확인

구축 후 FTD CLI로 이동하여 SNMP 컨피그레이션을 확인합니다. engineID 값은 자동으로 생성됩니다.

```
<#root>
FP1120-1#
connect ftd

>
system support diagnostic-cli

Attaching to Diagnostic CLI ... Press 'Ctrl+a then d' to detach.
Type help or '?' for a list of available commands.

FP1120-1>
enable

Password:
FP1120-1#
show run all snmp-server
```

```
snmp-server group AUTH v3 auth
snmp-server group PRIV v3 priv
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 테스트

```
<#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 9.8(2)K8"
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 확인

```
<#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 *****
```

v2c용 snmpwalk:

```
<#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
```

문제 해결

방화벽에서 추적을 통한 캡처 활성화:

```
<#root>
```

```
FP1120-1#
```

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

snmpwalk 툴을 사용하여 패킷이 표시되는지 확인합니다.

```
<#root>
```

FP1120-1#

show capture

capture CAPI type raw-data trace interface inside

[Capturing - 3137 bytes]

match udp any any eq snmp

캡처 내용:

<#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

SNMP 서버 통계 카운터에 SNMP Get 또는 Get-next 요청 및 응답이 표시되는지 확인합니다.

<#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

인그레스 패킷을 추적합니다. 패킷은 내부 NLP 인터페이스에 대한 UN-NAT입니다.

<#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

NAT 규칙은 SNMP 컨피그레이션의 일부로 자동으로 구축됩니다.

<#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 stat
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

백엔드 포트에서 UDP 4161은 SNMP 트래픽을 수신 대기합니다.

```
<#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 :::*
```

컨피그레이션이 잘못되거나 불완전한 경우 UN-NAT 단계가 없으므로 인그레스 SNMP 패킷이 삭제됩니다.

```
<#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 syslogs는 인그레스 패킷이 폐기되었음을 보여줍니다.

<#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.

질문과 대답

Q. FTD 관리 인터페이스를 사용하여 SNMP 메시지를 보낼 수 있습니까?

아니요. 현재 지원되지 않습니다.

관련 개선 결함: <https://bst.cloudapps.cisco.com/bugsearch/bug/CSCvu48012>

관련 정보

- [firepower Device Manager 버전 6.7용 Cisco Firepower Threat Defense 컨피그레이션 가이드](#)
- [Cisco Firepower Threat Defense REST API 가이드](#)
- [Cisco Firepower 릴리스 정보, 버전 6.7.0](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.