

了解简单网络管理协议(SNMP)陷阱

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[使用 SNMP 陷阱](#)

[Cisco IOS 发送的陷阱示例](#)

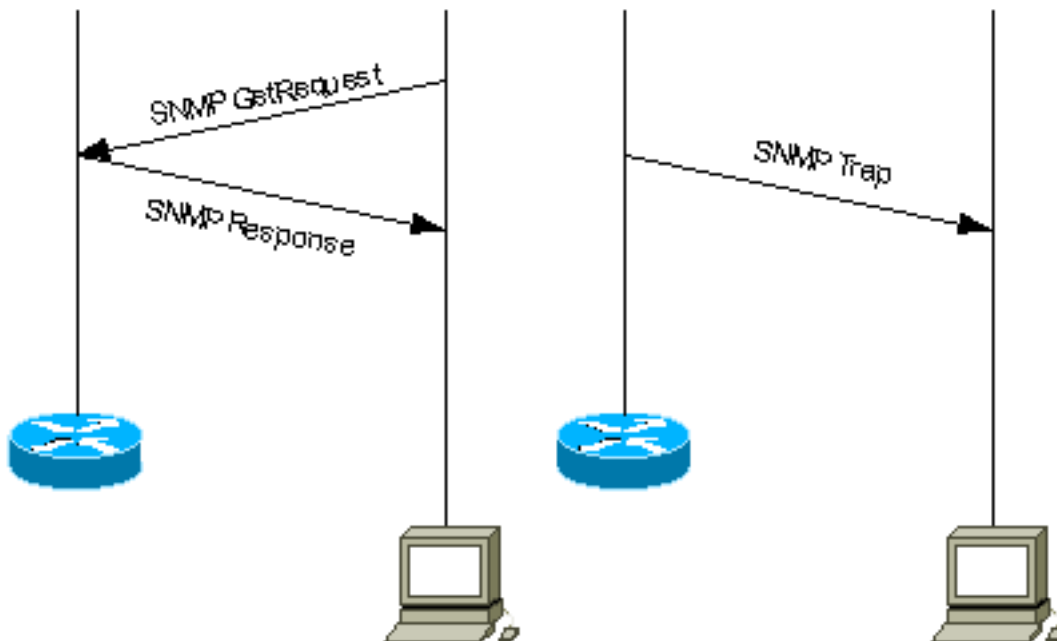
[Related Information](#)

Introduction

本文档提供对 SNMP 陷阱的介绍。文中将展示在数据网络的管理中，SNMP 陷阱如何使用及扮演何种角色。

SNMP 陷阱支持代理以未经请求的 SNMP 消息的形式，向管理站发送关于重大事件的通知。

在此图表中，左边的设定显示的是轮询信息并获得回应的网络管理系统。右边的设定显示的是向网络管理系统 (NMS) 发送未经请求的陷阱或异步陷阱的代理。



Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to [Cisco Technical Tips Conventions](#) for more information on document conventions.

使用 SNMP 陷阱

SNMPv1 (简单网络管理协议) 和 SNMPv2c, 与关联的管理信息库 (MIB) 相结合, 支持陷阱定向的通知。

如果一个管理器要负责大量的设备, 而每台设备都包含大量的对象, 要让管理器向每台设备上的每个对象都进行轮询或请求数据是不切实际的。陷阱定向的通知这一想法由此产生。其解决办法是受管设备上的每个代理不经请求就向管理器发送通知。这一操作通过发送一条称为事件陷阱的消息来完成。

收到事件后, 管理器会显示事件, 并可以选择根据事件采取措施。例如: 管理器可以直接轮询代理, 也可以轮询其他关联的设备代理, 以便更好地了解事件。

通过消除对无意义 SNMP 请求的需求, 陷阱定向的通知能节省大量的网络和代理资源。不过, 要完全消除 SNMP 轮询是不可能的。网络发现和拓扑更改必须使用 SNMP 请求。另外, 如果设备遭遇灾难性断电, 受管设备代理将不能发送陷阱。

RFC 1157 中对 SNMPv1 陷阱进行了定义, 包括以下字段:

- *Enterprise* - 识别生成陷阱的受管对象的种类。
- *Agent address* - 提供生成陷阱的受管对象的地址。
- *Generic trap type* - 表示这是若干种常规陷阱类型之一。
- *Specific trap code* - 表示这是若干种特定陷阱类型之一。
- *Time stamp* - 提供从上次网络重新初始化到生成陷阱之间经过的时间。
- *Variable bindings* - 陷阱中包含 PDU 的数据字段。每个可变绑定都将一个特定 MIB 对象实例与其当前值相关联。

标准的常规陷阱有: coldStart、warmStart、linkDown、linkUp、authenticationFailure、egpNeighborLoss。在常规 SNMPv1 陷阱中, *Enterprise* 字段包含发送陷阱的设备的 [sysObjectID](#) 值。[在供应商特定陷阱中](#), *Generic trap type* 字段会设置为 enterpriseSpecific(6)。思科以非传统的方式实施了自己的特定陷阱。不同于让陷阱的 *Enterprise* 字段保留 [sysObjectID](#), 并用 *Specific trap code* 字段识别所有思科设备支持的所有特定陷阱的办法, 思科使用多个陷阱 *Enterprise* 和 *Specific trap code* 字段来进行陷阱识别。您可以从 [SNMP Object Navigator](#) 中看到实际的值。[此外](#), 思科还在 [CISCO-GENERAL-TRAPS MIB](#) 中重新定义了部分常规陷阱, 并添加了更多约束变量。[在这些陷阱中](#), *Generic trap type* 字段保留不变, 不会设置为 enterpriseSpecific(6)。

在 SNMPv2c 中, 陷阱被定义为“通知”, 并且格式设置与 SNMPv1 不同。它具有以下参数:

- *sysUpTime* - 与 SNMPv1 陷阱中的 Time stamp 相同。
- [snmpTrapOID](#) - 陷阱识别字段。常规陷阱中的值在 RFC 1907 中已定义; 供应商特定陷阱中, *snmpTrapOID* 实际上是 SNMPv1 *Enterprise* 参数和另外两个子标识符 (“0”和 SNMPv1

Specific trap code 参数) 的联结。

- *VarBindList* - 可变绑定的列表。

为了理解代理给自己发送的陷阱，管理系统必须知道对象标识符 (OID) 定义了什么。所以，它必须拥有加载的陷阱的 MIB。这提供了正确的 OID 信息，以便网络管理系统能够了解发送给它的陷阱。

要了解特定 MIB 中思科设备支持的陷阱，请访问 [Cisco SNMP Object Navigator](#)。[这里列出了特定 MIB 可用的陷阱。要接收这些陷阱，您的 Cisco IOS® 软件版本必须支持列出的 MIB。要找出您的思科设备上支持哪些 MIB，请访问 \[www.cisco.com/go/mibs\]\(http://www.cisco.com/go/mibs\)。MIB 必须加载到您的网络管理系统中。此操作通常称为编译。请参阅您网络管理系统 \(例如 HP OpenView 或 NetView\) 的用户指南，了解在您的 NMS 平台上进行 MIB 编译的信息。同时请参阅 \[SNMP：MIB 常见问题\]\(#\) 及 \[MIB 编译器与 MIB 加载\]\(#\)。](#)

此外，除非经过配置，否则设备不会向网络管理系统发送陷阱。设备必须知道它应该发送陷阱。陷阱目标通常用 IP 地址定义，但如果为设备设置了域名系统 (DNS) 服务器查询，陷阱目标也可以是主机名。在较新的 Cisco IOS 软件版本中，设备管理员可以选择他们希望发送哪些陷阱。有关如何为思科设备配置 SNMP 及如何发送陷阱，请参阅相应的设备配置指南和[基本拨号网络管理系统实施指南](#)，还有[受支持的 Cisco IOS SNMP 陷阱及其配置方法](#)和[如何支持与配置 Cisco CatalystOS SNMP 陷阱](#)。

Note: 管理器通常会在 UDP 端口 162 上接收 SNMP 通知 (陷阱或通知)。

[Cisco IOS 发送的陷阱示例](#)

本节包含一些 Cisco IOS 发送的陷阱的示例，使用的命令是 `debug snmp packet`。

SNMPv1 常规陷阱，由思科重新定义过：

```
Nov 21 07:44:17: %LINK-3-UPDOWN: Interface Loopback1, changed state to up
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V1 Trap, ent products.45, addr 172.17.246.9, gentrap 3, spectrap 0
  ifEntry.1.23 = 23
  ifEntry.2.23 = Loopback1
  ifEntry.3.23 = 24
  lifEntry.20.23 = up
```

此输出结果显示来自 [CISCO-GENERAL-TRAPS](#) MIB、经过思科重新定义的 linkUp 陷阱，带有四个约束变量。它包含以下字段：

- *Enterprise* = products.45 (发送陷阱的设备的 [sysObjectID](#) 值，在本例中为 c7507 路由器)
- *Generic trap type* = 3 (linkUp)
- *Specific trap code* = 0

SNMPv1 思科专有陷阱：

```
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 172.17.246.9, gentrap 6, spectrap 1
  clogHistoryEntry.2.954 = LINK
  clogHistoryEntry.3.954 = 4
  clogHistoryEntry.4.954 = UPDOWN
  clogHistoryEntry.5.954 = Interface Loopback1, changed state to up
  clogHistoryEntry.6.954 = 43021184
```

此输出结果显示来自 [CISCO-SYSLOG-MIB](#) 的思科专有 clogMessageGenerated 陷阱，带有五个约束变量。[它包含以下字段：](#)

- *Enterprise* = clogMessageGenerated 陷阱的 Enterprise 值
- *Generic trap type* = 6 (enterpriseSpecific)
- *Specific trap code* = 1 (clogMessageGenerated 的 specific trap code 值)

SNMPv2c 思科专有陷阱：

```
4d23h: SNMP: Queuing packet to 172.17.246.162
4d23h: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 43053404
snmpTrapOID.0 =
clogHistoryEntry.2.958 = SYS
clogHistoryEntry.3.958 = 6
clogHistoryEntry.4.958 = CONFIG_I
clogHistoryEntry.5.958 = Configured from console by vty0 (10.10.10.10)
clogHistoryEntry.6.958 = 43053403
```

此输出结果显示来自 [CISCO-CONFIG-MAN-MIB](#) 的思科专有 [ciscoConfigManEvent](#) SNMPv2c 通知，带有三个约束变量：

- [ccmHistoryEventCommandSource](#)
- [ccmHistoryEventConfigSource](#)
- [ccmHistoryEventConfigDestination](#)

如果对设备配置进行过任何更改，可以使用此陷阱。最后两个组件的值用于确定是否发出过 **show** 命令，或配置是否被改动过。

```
6506E#term mon
6506E#debug snmp packet
SNMP packet debugging is on
```

```
6506E#sh run
Building configuration...
...
6506E#
19:24:18: SNMP: Queuing packet to 10.198.28.80
19:24:18: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 6981747
snmpTrapOID.0 = ciscoConfigManMIB.2.0.1
ccmHistoryEventEntry.3.100 = 1
!--- 1 -> commandLine. Executed via CLI. ccmHistoryEventEntry.4.100 = 3 !--- 3 -> running
ccmHistoryEventEntry.5.100 = 2 !--- 2 -> commandSource. Show command was executed.
```

```
6506E#term mon
6506E#debug snmp packet
SNMP packet debugging is on
```

```
6506E#conf t
Enter configuration commands, one per line. End with CNTL/Z.
6506E(config)#exit
```

```
22:57:37: SNMP: Queuing packet to 10.198.28.80
22:57:37: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 8261709
snmpTrapOID.0 = ciscoConfigManMIB.2.0.1
ccmHistoryEventEntry.3.108 = 1
!--- 1 -> commandLine. Executed via CLI. ccmHistoryEventEntry.4.108 = 2 !--- 2 -> commandSource
ccmHistoryEventEntry.5.108 = 3 !--- 3 -> running. Change was destined to the running
configuration.
```

[Related Information](#)

- [Technical Support & Documentation - Cisco Systems](#)