

Управляйте множественными случаями OSPF с snmp context

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Знающий SNMP Context](#)

[!--- конфигурацию](#)

[Проверка](#)

[Проверка SNMPv2](#)

[Проверка SNMPv3](#)

[Дополнительные сведения](#)

Введение

Этот документ предоставляет примеры конфигурации для SNMPv2 и SNMPv3, которые описывают, как использовать Snmp context для управления множественными случаями Протокола OSPF.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.](#)

Общие сведения

MIB OSPF, определенный IETF ([RFC 1850](#)), был разработан для работы только с одним процессом OSPF / экземпляра на данном маршрутизаторе.

Например, существует только одиночный объект *ospfRouterId*, не таблица их. Для обработки множественных случаев [RFC 4750](#) предлагает, чтобы вы использовали контексты SNMPv3 для обеспечения представлений на экземпляр.

Знающий SNMP Context

До создания кода SNMP OSPF IOS, с учетом контекста, система выбрала бы более или менее случайный экземпляр "по умолчанию", когда это возвратило скалярные объекты и некоторые таблицы. В этих случаях информация от других экземпляров не была доступна через SNMP. Для некоторых других таблиц SNMP делал бы пюре вместе из записей от всех экземпляров без любого способа различить, который был который. Во многих случаях это могло привести неоднозначный или дублированные записи. Это был особенно не полезный прием в конфигурациях PE-CE, где IP-адреса и идентификаторы соседнего маршрутизатора не могли бы быть уникальными. Этот сделанный мониторинг и устранение проблем отдельных экземпляров CE, трудных или невозможных.

С текущим Кодом IOS с учетом контекста (когда никакой контекст не задан), все еще существует старое поведение для скалярных объектов. Единственное изменение - то, что это теперь ограничивает все, а не просто некоторые таблицы к тому же экземпляру OSPF "по умолчанию" как скаляры. Когда контексты будут предоставлены, запросы SNMP могут быть предназначены к определенному экземпляру OSPF, и вся информация для того экземпляра может быть получена последовательным и однозначным способом.

Если SNMPv3 используется, строка контекста может быть предоставлена непосредственно с опросом. SNMPv2c не предоставляет контекст. Однако можно сопоставить Строки имени и пароля SNMP с контекстами в конфигурации IOS, и эти контексты могут использоваться к прямым опросам SNMPv2 к определенному экземпляру OSPF.

!--- конфигурацию

Этот пример конфигурации основывается на SNMPv2:

```
Маршрутизатор 1
Router1#
router ospf 1
  router-id 1.1.1.111
  log-adjacency-changes
  snmp context context1
!
router ospf 2
  router-id 4.4.4.111
  log-adjacency-changes
```

```
snmp context context2
!--- Associates the SNMP context with the instance. !
snmp-server user u2 g2 v2c !--- Configures the user u2
to the SNMP group g2 and !--- specifies the group is
using the SNMPv2c security model. snmp-server group g2
v2c !--- Configures the SNMP group g2 and specifies !---
the group is using the SNMPv2c security model. snmp-
server group g2 v2c context context1 snmp-server group
g2 v2c context context2 snmp-server community public RO
!--- Community access string to permit access !--- to
the SNMP. snmp-server community cx1 RO snmp-server
community cx2 RO snmp-server context context1 snmp-
server context context2 snmp mib community-map cx1
context context1 security-name u2 !--- Associates the
SNMP community cx1 with !--- the context context 1. snmp
mib community-map cx2 context context2 security-name u2
```

Этот пример конфигурации основывается на SNMPv3:

Маршрутизатор 1

```
Router1#
router ospf 1
  router-id 1.1.1.111
  log-adjacency-changes
  snmp context context1
!
router ospf 2
  router-id 4.4.4.111
  log-adjacency-changes
  snmp context context2
!
snmp-server user u1 g1 v3
snmp-server group g1 v3 noauth
snmp-server group g1 v3 noauth context context1
snmp-server group g1 v3 noauth context context2
snmp-server context context1
snmp-server context context2
```

Примечание: [Чтобы получить подробные сведения о командах в данном документе, используйте Средство поиска команд \(только для зарегистрированных клиентов\).](#)

Проверка

Можно использовать команду **snmpwalk** на любом клиентском компьютере для проверки выходных данных.

Примечание: [Средство Output Interpreter \(OIT\) \(только для зарегистрированных клиентов\) поддерживает определенные команды show.](#) Посредством OIT можно анализировать выходные данные команд **show**.

Проверка SNMPv2

SNMPv2

```
linux>snmpwalk -c public -v2c irp-view14:7890 OSPF-
MIB::ospfRouterId.0 OSPF-MIB::ospfRouterId.0 =
IpAddress: 4.4.4.111 linux>snmpwalk -c cx1 -v2c irp-
```

```
view14:7890 OSPF-MIB::ospfRouterId.0 OSPF-
MIB::ospfRouterId.0 = IPAddress: 1.1.1.111
linux>snmpwalk -c cx2 -v2c irp-view14:7890 OSPF-
MIB::ospfRouterId.0 OSPF-MIB::ospfRouterId.0 =
IPAddress: 4.4.4.111
```

Проверка SNMPv3

SNMPv3

```
linux>snmpwalk -u u1 -v3 irp-view14:7890 OSPF-
MIB::ospfRouterId.0 OSPF-MIB::ospfRouterId.0 =
IPAddress: 4.4.4.111 linux>snmpwalk -u u1 -v3 -n
context1 irp-view14:7890 OSPF-MIB::ospfRouterId.0 OSPF-
MIB::ospfRouterId.0 = IPAddress: 1.1.1.111
linux>snmpwalk -u u1 -v3 -n context2 irp-view14:7890
OSPF-MIB::ospfRouterId.0 OSPF-MIB::ospfRouterId.0 =
IPAddress: 4.4.4.111
```

Дополнительные сведения

- [Настройка OSPF с помощью SNMP](#)
- [Cisco Systems – техническая поддержка и документация](#)