

Использование CISCO-BULK-FILE-MIB

Содержание

[Введение](#)

[Перед началом работы](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Использование CISCO-BULK-FILE-MIB](#)

[Создание операции BULK-FILE](#)

[Пошаговые инструкции](#)

[Передача файла с помощью CISCO-FTP-CLIENT-MIB](#)

[Пошаговые инструкции](#)

[Проверка результата](#)

[Устранение ошибок результата](#)

[Предупреждения](#)

[Дополнительные сведения](#)

Введение

В этом документе рассказывается о том, как можно использовать CISCO-BULK-FILE-MIB и передавать файлы, созданные этой базой MIB, с помощью CISCO-FTP-CLIENT-MIB.

Запускаясь с выпуска 12.0 программного обеспечения Cisco IOS, Cisco внедрила способ хранить объект Протокола SNMP или таблицу как файл на устройстве. Этот файл может тогда быть получен с помощью CISCO-FTP-CLIENT-MIB. Эта технология позволяет передавать большие объемы данных методом надежной передачи.

Перед началом работы

Требования

Прежде чем использовать эту конфигурацию, убедитесь, что выполняются эти требования:

- У вас есть устройство Cisco рабочий выпуск 12.0 программного обеспечения Cisco IOS или позже. Используйте инструмент MIB Locator, чтобы убедиться, что CISCO-BULK-FILE-MIB поддерживается устройством. Ссылка на программное средство может быть найдена на странице [Cisco IOS MIB Tools](#). **Примечание:** Этот MIB не поддерживается устройствами с операционной системой Catalyst OS.
- В устройстве следует настроить SNMP с помощью строк имени и пароля для чтения и

записи. Данная процедура не описывается в этом документе. Для получения информации о настройке SNMP на устройствах IOS® читайте, [Как Настроить Строки имени и пароля SNMP на маршрутизаторах, Коммутаторах XL на основе ПО Cisco IOS, RSM, MSFC и Коммутаторах Catalyst.](#)

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- CISCO-BULK-FILE-MIB для сохранения ifTable маршрутизатора 7507, работающего с 12.1(12), в файл, затем используйте CISCO-FTP-CLIENT-MIB для передачи этого файла с маршрутизатора на сервер FTP.
- Набор команд SNMP [сетевого snmp](#), установленный на UNIX или Windows.
- Используются следующие MIB: TC SNMPv2S I SNMPv2CONF SNMPv2MIB SNMPv2IANAifType-MIB IF-MIB CISCO-SMI CISCO-TC CISCO-BULK-FILE-MIB CISCO-FTP-CLIENT-MIB

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 Технические рекомендации Cisco. Условные обозначения.](#)

Общие сведения

Удостоверьтесь, что у вас есть MIB в этой таблице, загруженной в вашу платформу управления. Это позволяет вам использовать имена объектов и значения, перечисленные выше, вместо числовых идентификаторов объектов (OIDs). По большей части в данном документе используются имена объектов, а не их идентификаторы (OID).

Версия 1 формата SMI	Формат S I версии 2
SNMPv2-SMI-V1SMI.my	SNMPv2-SMI.my
SNMPv2-TC-V1SMI.my	SNMPv2-TC.my
	SNMPv2-CONF.my
SNMPv2-MIB-V1SMI.my	SNMPv2-MIB.my
IANAifType-MIB-V1SMI.my	IANAifType-MIB.my
IF-MIB-V1SMI.my	IF-MIB.my
CISCO-SMI-V1SMI.my	CISCO-SMI.my
CISCO-TC-V1SMI.my	CISCO-TC.my
CISCO-BULK-FILE-MIB-V1SMI.my	CISCO-BULK-FILE-MIB.my
CISCO-FTP-CLIENT-MIB-V1SMI.my	CISCO-FTP-CLIENT-MIB.my

Использование CISCO-BULK-FILE-MIB

Создание операции BULK-FILE

В данном примере мы перехватываем `ifTable` от маршрутизатора и храним его в объемном файле. Однако можно использовать любой объект MIB или таблицу.

Используйте версию сетевого snmp `snmpset`. IP-адрес маршрутизатора `14.32.8.2`. Его строка имени и пароля для чтения и записи является частной. Идентификационная строка имени и пароля только для чтения является общей.

Каждое создание новой массовой файловой операции требует выбора двух произвольных чисел для экземпляра ряда. Это может быть любое число от 1 до 4294967295 включительно. В целях данного примера используйте 333 и 444.

Пошаговые инструкции

Для создания операции BULK-FILE выполните эти шаги:

1. Установите файл, который будет создан.
`$snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 5 $ snmpset -c private 14.32.8.2 cbfDefineFileName.333 s ifTable.txt $ snmpset -c private 14.32.8.2 cbfDefineFileFormat.333 i bulkASCII`
2. Укажите объект MIB для сбора данных. Этот объект требует двух индексов для нормальной работы. Эти 333 являются 333 от таблицы создания файла выше. 444 -
`, cbfDefineObjectTable. cbfDefineObjectID (ifTable)`. Здесь также можно использовать полную информацию об идентификаторе объекта (OID).
`$snmpset -c private 14.32.8.2 cbfDefineObjectID.333.444 o ifTable`
3. Активация вновь созданных строк.
`cbfDefineObjectTable. $ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 1 $ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 1`
4. Создайте файл.
`$ snmpset -c private 14.32.8.2 cbfDefineFileNow.333 i 3` Создается файл-болванка.
5. Убедитесь, что файл был создан успешно с помощью `snmpget` объекта `cbfStatusFileState`. Для данного объекта требуется два индекса. Первый индекс – это случайное число, выбранное для таблицы файлов (в этом примере – 333). Значение второго индекса зависит от числа созданных на маршрутизаторе файлов. Так как это - ваш первый файл, индекс равняется 1. Поэтому используйте команду:
`$ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1 running(1) , .` Значение (2) средства, что файл был создан успешно и ждет, чтобы быть считанным.

Однако этот файл нельзя открыть прямо на маршрутизаторе. Используйте CISCO-FTP-CLIENT-MIB, чтобы прочитать этот файл.

Передача файла с помощью CISCO-FTP-CLIENT-MIB

Для каждой операции FTP Client следует указать случайный номер экземпляра строки. Можно использовать один из одинаково случайных номеров, использованных выше. В этом примере используется 555.

Пошаговые инструкции

Для передачи файла с помощью CISCO-FTP-CLIENT-MIB выполните эти шаги:

1. Создайте экземпляр строки клиента FTP.
`$ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 5`
2. Укажите требуемые параметры. LocalFile должен иметь такое же имя, как и файл, созданный ранее! Используйте параметр putASCII для передачи файлов bulkASCII.
`cbfDefineFileFormat bulkBinary, cfcRequestOperation putBinary.$ snmpset -c private 14.32.8.2 cfcRequestOperation.555 i putASCII $ snmpset -c private 14.32.8.2 cfcRequestLocalFile.555 s ifTable.txt $ snmpset -c private 14.32.8.2 cfcRequestRemoteFile.555 s /home/Marcus/ifTable.txt $ snmpset -c private 14.32.8.2 cfcRequestServer.555 s 172.18.123.33 $ snmpset -c private 14.32.8.2 cfcRequestUser.555 s Marcus $ snmpset -c private 14.32.8.2 cfcRequestPassword.555 s marcus123`
3. Начните передачу, установив ряд в активный режим.
`$ snmpset -c private 14.32.8.2 cfcRequestEntryStatus.555 i 1` Передача FTP начинается. Когда завершённый, файл сохранен к/home/Marcus/ifTable.txt.
4. Для получения статуса Передачи FTP используйте snmpget снова на объекте cfcRequestResult. Этот объект использует тот же самый индекс, который был использован с другими объектами FTP.
`$ snmpget -c public 14.32.8.2 cfcRequestResult.555`
`pending(1) , . A value of success(2) means the file transferred successfully. Любое другое значение является ошибкой.`
5. После передачи файла еще раз попробуйте snmpget объекта cbfStatusFileState. Теперь имеет другое значение.
`$ snmpget -c public 14.32.8.2 cbfStatusFileState.333.1`
`enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfStatus.cbfStatusFileTable.cbfStatusFileEntry.cbfStatusFileState.333.1 = emptied(3) emptied(3) , . Файл нельзя передать повторно.`
6. Теперь можно безопасно удалить этот файл, уничтожив строку состояния файла. Этот объект берет те же индексы в качестве cbfStatusFileState ВЫШЕ.
`$ snmpset -c private 14.32.8.2 cbfStatusFileEntryStatus.333.1 i 6`
7. Как только файл удален, удалите соответствующие строки объекта и файла.
`$ snmpset -c private 14.32.8.2 cbfDefineObjectEntryStatus.333.444 i 6 $ snmpset -c private 14.32.8.2 cbfDefineFileEntryStatus.333 i 6`

Этим способом можно использовать CISCO-FTP-CLIENT-MIB для передачи любого файла прочь маршрутизатора с помощью FTP.

Проверка результата

Этот раздел ведет вас посредством чтения части синтаксиса для этого файла.

1. Первая линия является линией prefix. Для нашего примера ifTable это:
`prefix 1.3.6.1.2.1.2.2.1 OID ifEntry. IfTable` составлен из одного или более ifEntries.
2. Следующая строка содержит число объектов в таблице. Строка включает таблицу ключевых слов, количество объектов в таблице и индекс каждого объекта. Пример:
`table 22 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22` Это состояния линии, что таблица содержит 22 объекта и каждый объект, имеет инкрементно увеличивающийся индекс. Эти объекты от примера ifTable:
`ifIndex
ifDescr
ifType
ifSpeed
...`
3. После этой строки находятся многостроковые записи. ifTable . Строки начинаются со строки ключевого слова, придерживавшейся их индексным

идентификатором и придерживавшейся объектами, перечисленными предыдущим элементом таблицы. Пример:

```
row 1 1 546F6B656E52696E67302F30 9 4464 16000000 0008B0851800 2 2 6551 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

4. Четвертой записью является `ifDescr` для интерфейса 1. `ifDescr ASCII`. Для перевода этой линии в больше удобочитаемого формата используйте эту команду Perl: `$ perl -e 'print pack("H*", "546F6B656E52696E67302F30")'` `TokenRing0/0` Эта запись соответствует интерфейсу `TokenRing0/0`. Все объекты, которые обычно являются строками, отображены как шестнадцатеричная кодировка ASCII в объемных файлах. Для преобразования строки ASCII в читаемый текст можно использовать эту команду языка Perl. Если у вас нет Perl, используйте эту [таблицу ASCII - символа](#) для перевода строки.

5. В некоторых записях есть значки `~` для значений. Это означает, что значение для того объекта является NULL. То есть экземпляр объекта на устройстве отсутствует. Пример: `row 9 9 41544D312F302F302D61746D206C61796572 37 ~ 0 1 1 5971 ~ ~ ~ ~ ~ ~ ~ ~ ~ ~` Это соответствует `ATM1/0/0-atm` интерфейс уровня. Заметьте, что `ifMtu` является NULL для этого интерфейса. Так как это - виртуальный интерфейс, он целесообразен, что он не имеет MTU. Если вы предпочитаете, можно заменить эти NULL 0 путем добавления этой команды к конфигурации устройства: `Router(config)#no snmp-server sparse-table`

Устранение ошибок результата

`cbfStatusFileState` , (1), (2) (3), . Это причины для ошибок:

<code>noSpace</code>	no data due to insufficient file space
<code>badName</code>	no data due to a name or path problem
<code>writeErr</code>	no data due to fatal file write error
<code>noMem</code>	no data due to insufficient dynamic memory
<code>buffErr</code>	implementation buffer too small
<code>aborted</code>	short terminated by operator command

Если количество объектов в файле является меньше, чем вы ожидаете, `cbfDefineMaxObjects` от CISCO-BULK-FILE-MIB может быть установлен слишком низко. Для определения текущего значения объекта используйте `snmpget`.

```
$ snmpget -c public 14.32.8.2 cbfDefineMaxObjects.0
```

Значение 0 средств, что не настроен никакой предел. Значение может быть установлено в любое целое число между 0 и 4294967295, включительно. Для установки максимальных объектов на файл к 10 используйте команду `snmpset`. Индекс этого объекта всегда равен 0.

```
$ snmpset -c private 14.32.8.2 cbfDefineMaxObjects.0 u 10
```

Возможно, объект невозможно настроить на всех платформах. Если сбой `snmpset` с этой ошибкой, объект не конфигурируем на вашей платформе:

```
Error in packet.
```

```
Reason: (noSuchName) There is no such variable name in this MIB.
```

```
Failed object:
```

```
enterprises.cisco.ciscoMgmt.ciscoBulkFileMIB.ciscoBulkFileMIBObjects.cbfDefine.cbfDefineMaxObjects.0
```

`cfcRequestResult` , "(1) "(2), FTP . Это причины для ошибок:

aborted	user aborted the transfer
fileOpenFailLocal	local bulk file was not found
fileOpenFailRemote	remote file could not be opened for writing
badDomainName	FTP server's hostname could not be resolved
unreachableIpAddress	route to the FTP server could not be found
linkFailed	connection could not be made to the remote server
fileReadFailed	local file could not be read
fileWriteFailed	remote file could not be written

Предупреждения

- В настоящее время нет никакого поддерживаемого способа обратиться к объемным файлам непосредственно. Для чтения файлов необходимо перейти через CISCO-FTP-CLIENT-MIB.
- Объект `cbfDefineFileStorage` определяет три типа: `ephemeral`, `volatile` и `permanent`. Currently, the only type supported in IOS is `ephemeral`. Ephemeral files exist in small amounts until read.
- Как только файлы считаны, они не могут быть перечитаны. Сначала необходимо создать их заново.
- Объект `cbfDefineFileFormat` определяет три типа: `standardBER`, `bulkBinary` и `bulkASCII`. Единственными поддерживаемыми форматами является `bulkBinary` и `bulkASCII`.
- Известно, что FTP-сервер Chameleon для Windows не работает с CISCO-FTP-CLIENT-MIB, так как он возвращает неправильные коды результатов.

Дополнительные сведения

- [Настройка строк имени и пароля SNMP на маршрутизаторах, XL-коммутаторах с ПО Cisco IOS, устройствах RSM, MSFC и коммутаторах Catalyst](#)
- [Техническая поддержка - Cisco Systems](#)