2 SNMP агент

Обзор

Вы возможно захотите использовать SNMP мониторинг устройств таких как принтеры, сетевые коммутаторы, маршрутизаторы или ИБП, как правило, которые как правило поддерживают SNMP и для которых было бы непрактично пытаться настраивать комплексные системы управления или Zabbix агенты.

Чтобы была возможность получать данные переданные SNMP агентами с этих устройств, Zabbix сервер должен быть изначально сконфигурирован с поддержкой SNMP.

SNMP проверки выполняются только через UDP протокол.

Начиная с версии 2.2.3 демоны Zabbix сервера и прокси опрашивают устройства SNMP множественными значениями за один запрос. Это поведение повлияет на все виды SNMP элементов данных (простые SNMP элементы данных, элементы данных с динамическими индексами и также низкоуровневые SNMP обнаружения) и обработка SNMP элементов данных сейчас должна быть более эффективной. Пожалуйста обратите внимание на раздел с техническими подробностями ниже, описывающий как работает изнутри этот функционал. Начиная с Zabbix 2.4 у каждого интерфейса также имеется настройка “Использовать массовые запросы”, которая позволяет отключать массовые запросы у устройств, которые не способны обработать их должным образом.

Начиная с Zabbix 2.2.7 и Zabbix 2.4.2 процессы сервера и прокси будут журналировать строки похожие на следующие в случае получения неправильного/искаженного SNMP ответа:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Пока они не покрывают все возможные проблемные случаи, но они являются удобным удобным идентификатором отдельных SNMP устройств на которых необходимо отключить массовые запросы.

Начиная с версии Zabbix 2.2 демоны сервера и прокси корректно обрабатывают параметр конфигурации Timeout при выполнении SNMP проверок. Дополнительно демоны не выполняют повторных запросов после одного неуспешного (по превышении времени ожидания/неверные настройки учетных данных) SNMP запроса. Ранее на самом деле использовались стандартные для библиотеки SNMP значения времени ожидания и количества повторов (1 секунда и 5 повторов соответственно).

Начиная с версии Zabbix 2.2.8 и Zabbix 2.4.2 демоны сервера и прокси всегда выполняют один повторный запрос: либо через механизм библиотеки SNMP, либо через внутренний механизм сбора множества значений за один запрос (bulk).

Если выполняется мониторинг устройств по SNMPv3, убедитесь что msgAuthoritativeEngineID (также известное как snmpEngineID или “Engine ID”) никогда не будет общим для двух и более устройств. Согласно RFC 2571 (раздел 3.1.1.1) оно должно быть уникальным для каждого устройства.

Настройка мониторинга по SNMP

Для начала мониторинга устройства по SNMP, должны быть выполнены следующие шаги:

Шаг 1

Создайте узел сети для устройства с SNMP интерфейсом.

Введите IP адрес. Вы можете использовать один из поставляемых шаблонов SNMP (*Template SNMP Device* и другие), которые автоматически добавят некоторый набор элементов данных. Тем не менее, шаблон может быть не совместим с узлом сети. Нажмите на *Добавить* для сохранения узла сети.

SNMP проверки не используют *Порт агента*, он игнорируется.

Шаг 2

Узнайте строку SNMP (или OID) элемента данных, которую вы хотите мониторить.

Для получения списка строк SNMP, используйте команду **snmpwalk** (часть программного обеспечения net-snmp, которое вы должны были установить как часть инсталляции Zabbix) или эквивалентную утилиту:

```
shell> snmpwalk -v 2c -c public <IP хоста> .
```

'2c' здесь означает версию SNMP, вы также можете заменить его на '1', чтобы использовать 1 версию SNMP на устройстве.

Эта команда должна показать вам список SNMP строк и их последние значения. Если это не произойдет, то возможно что SNMP 'community' отличается от стандартного 'public', в этом случае вам необходим узнать это имя.

Вы можете пройтись по списку пока не найдете строку которую вы хотите мониторить, например, если вы хотите мониторить входящее количество байт на вашем коммутаторе на 3 порту вы могли бы использовать `IF-MIB::ifInOctets.3` из этой строки:

```
IF-MIB::ifInOctets.3 = Counter32: 3409739121
```

Сейчас вы можете воспользоваться командой **snmpget** для того чтобы определить цифровой OID для 'IF-MIB::ifInOctets.3':

```
shell> snmpget -v 2c -c public -On 10.62.1.22 IF-MIB::ifInOctets.3
```

Обратите внимание, что последнее число в строке это номер порта, который вы ищите для мониторинга. Смотрите также: Динамические индексы.

Вывод команды покажет вам что-то наподобие этого:

```
.1.3.6.1.2.1.2.2.1.10.3 = Counter32: 3472126941
```

Опять же, последнее число в OID является номером порта.

3COM кажется использует номера портов сотнями, например 1 порт = 101 порт, 3 порт = 103 порт, но в Cisco используются обычные номера, например, 3 порт = 3.
Некоторые из наиболее часто используемых SNMP OID'ов автоматически конвертируются Zabbix'ом в числовое представление.

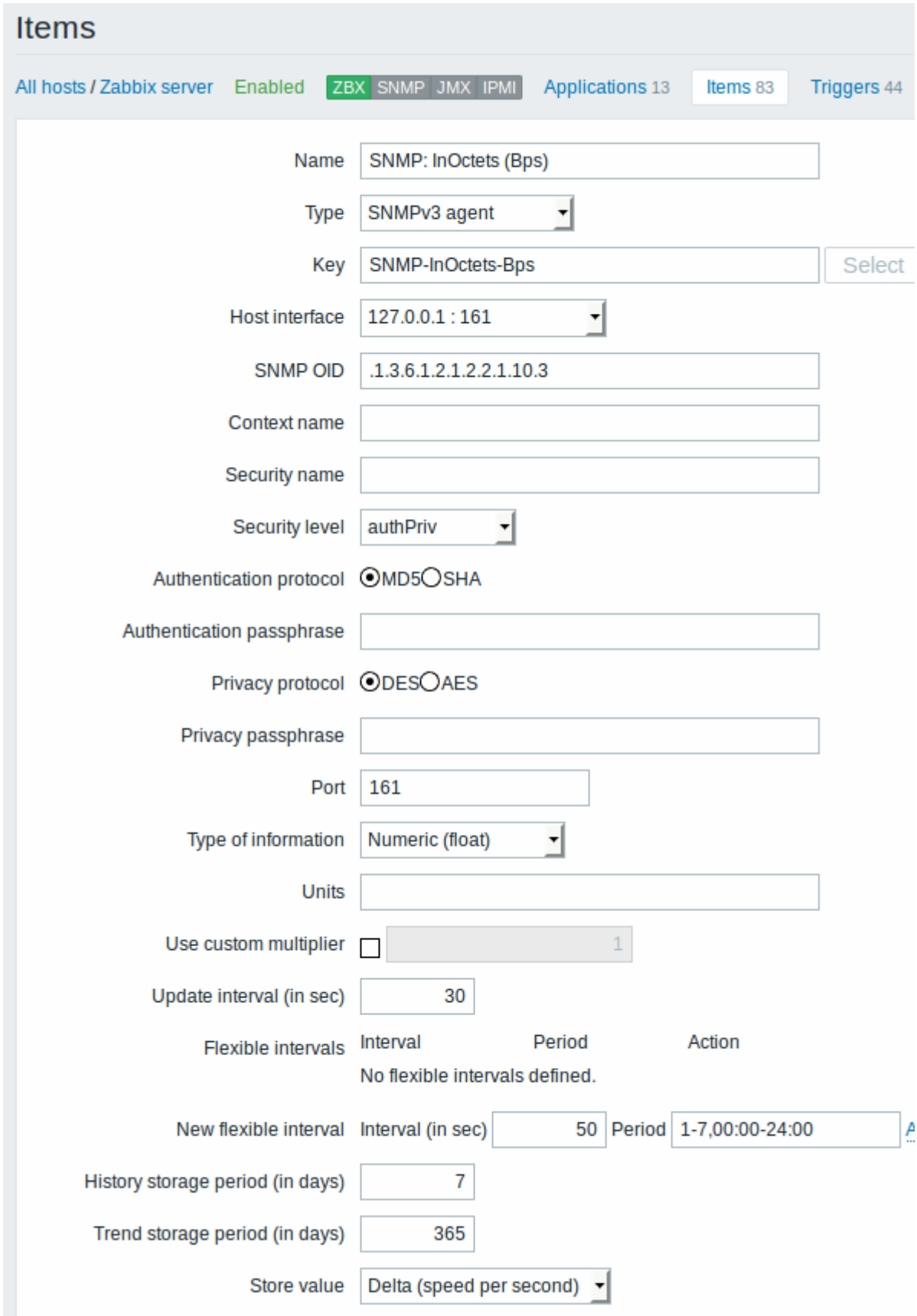
В последнем примере выше тип значение “Counter32” (32-битный счетчик), что внутренне соответствует типу ASN_COUNTER. Полный список поддерживаемых типов ASN_COUNTER, ASN_COUNTER64, ASN_UINTEGER, ASN_UNSIGNED64, ASN_INTEGER, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS, ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR и ASN_OBJECT_ID (с 2.2.8, 2.4.3). Приведенные типы грубо соответствуют “Counter32”, “Counter64”, “UInteger32”, “INTEGER”, “Float”, “Double”, “Timeticks”, “Gauge32”, “IpAddress”, “OCTET STRING”, “OBJECT IDENTIFIER” в выводе **snmpget** утилиты, но могут также отображаться как “STRING”, “Hex-STRING”, “OID” и другие, в зависимости от наличия полученной подсказки.

Шаг 3

Создайте элемент данных для мониторинга.

Итак, вернитесь назад в Zabbix и нажмите на Элементы данных, выберите созданный ранее узел сети SNMP. В зависимости от того использовали ли вы шаблон при создании узла сети или нет, вы должны будете увидеть список элементов данных SNMP, связанных с вашим узлом сети или попросту окно нового элемента данных. Мы будем исходить из предположения, что вы собираетесь создать элемент данных самостоятельно, с помощью информации, которую вы только что собрали используя snmpwalk или snmpget, так что введите простое описание на русском языке (или английском) в поле 'Описание' в диалоге нового элемента данных. Убедитесь, что в поле 'Узел сети' находится ваш коммутатор/роутер и измените поле 'Тип' в значение “SNMPv* агент”. Введите community (обычно public) и укажите текстовый или числовой OID, который вы получили ранее, в поле 'SNMP OID', например: .1.3.6.1.2.1.2.2.1.10.3

Введите 'Порт SNMP' - 161 и 'Ключ' - что-то осмысленное, например, SNMP-InOctets-Bps. Выберите множитель, если желаете, и укажите 'Интервал обновления', и 'Хранение истории', если вы хотите чтобы значения параметров отличались от умолчаний. Установите 'Тип информации' в значение равное *Числовой (с плавающей точкой)* и 'Хранение значения' как *Дельта (скорость в секунду)* (важно, в противном случае вы будете получать накопленные значения с SNMP устройства вместо последнего изменения).



Теперь сохраните элемент данных и перейдите в *Мониторинг → Последние данные*, чтобы увидеть ваши данные SNMP!

Обратите внимание на специфичные опции доступные только для SNMPv3 элементов данных:

Параметр	Описание
Имя контекста	Введите контекстное имя для определения элемента данных в SNMP подсети. *Имя контекста* поддерживается для SNMPv3 элементов данных с Zabbix 2.2. В данном поле раскрываются пользовательские макросы.
Имя безопасности	Введите имя безопасности. В данном поле раскрываются пользовательские макросы.
Уровень безопасности	Выберете уровень безопасности: **noAuthNoPriv** - ни аутентификация, ни протокол безопасности не используются **AuthNoPriv** - используется протокол аутентификации, протокол безопасности нет **AuthPriv** - используются и протокол аутентификации, и протокол безопасности
Протокол аутентификации	Выберете протокол аутентификации - *MD5* или *SHA*.
Фраза-пароль аутентификации	Введите фразу-пароль для аутентификации В данном поле раскрываются пользовательские макросы.
Протокол безопасности	Введите протокол безопасности - *DES* или *AES*.
Фраза-пароль безопасности	Введите фразу-пароль безопасности. В данном поле раскрываются пользовательские макросы.

При изменениях в *Протокол аутентификации*, *Фраза-пароль аутентификации*, *Протокол безопасности* или *Фраза-пароль безопасности*, чтобы эти изменения применились, необходимо перезапустить сервер/прокси.

Пример 1

Общий пример:

Параметр	Описание
Community	public
OID	1.2.3.45.6.7.8.0 (или .1.2.3.45.6.7.8.0)
Ключ	<Уникальная строка, которая используется как ссылка в триггерах> Например, “my_param”.

Обратите внимание, что OID можно задать в числовом или строковом представлении. Тем не менее, в некоторых случаях, строковый OID должен быть сконвертирован в числовое представление. Для этого можно использовать утилиту snmpget:

```
shell> snmpget -On localhost public
enterprises.ucdavis.memory.memTotalSwap.0
```

Мониторинг SNMP параметров возможен, если указан флаг --with-net-snmp при конфигурировании исходных кодов Zabbix.

Пример 2

Мониторинг времени работы:

Параметр	Описание
Community	public
Oid	MIB::sysUpTime.0
Ключ	router.uptime
Тип информации	Числовой (с плавающей точкой)
Единица измерения	uptime
Множитель	0.01

Обработка массовых SNMP запросов

Начиная с 2.2.3 Zabbix сервер и прокси одним опросом запрашивают множество SNMP элементов данных. Такое поведение затрагивает следующие типы SNMP элементов данных:

- обычные SNMP элементы данных;
- SNMP элементы данных с динамическими индексами;
- правила низкоуровневого SNMP обнаружения.

Все элементы данных SNMP с одного интерфейса запланированы на опрос в одно время. Первые два типа элементов данных собираются поллерами порциями не более чем по 128 элементов данных, в то время как правила низкоуровневого обнаружения обрабатываются индивидуально как и ранее.

На низком уровне, есть два вида операций выполняемых при опросе значений: получение нескольких заданных объектов и прохождение дерева OID-ов.

Для “получения” используется GetRequest-PDU с не более чем 128 привязанных переменных. Для “прохождения”, используется GetNextRequest-PDU для SNMPv1 и GetBulkRequest с полем “max-repetitions” с наибольшим количеством в 128 полученных значений используется для SNMPv2 и SNMPv3.

Таким образом преимущества массовой обработки для каждого типа SNMP элемента данных описаны ниже:

- простые SNMP элементы данных получают преимущество от улучшения “получения”;
- SNMP элементы данных с динамическими индексами получают преимущество и от улучшений “получения” и “прохождения”: “получение” используется для проверки индексов, а “прохождение” для построения кэша значений;
- правила низкоуровневого SNMP обнаружения получают преимущество от улучшения “прохождения”.

Тем не менее, есть техническая проблема что не все устройства способны вернуть 128 значений за один запрос. Некоторые всегда возвращают корректный ответ, но другие либо отвечают с ошибкой “tooBig(1)”, либо не отвечают вообще, когда потенциальный запрос превышает определенный лимит.

Для вычисления оптимального количества запрашиваемых объектов с устройства, Zabbix использует следующую стратегию. Начинается с осторожного запроса одного значения. Это запрос выполнен успешно, запрашивается 2 значения за один запрос. Если запрос снова выполнен успешно, запрашивается 3 значения за запрос и продолжается аналогично

умножением количества запрашиваемых значений на 1.5, в результате получается следующая последовательность размера запросов: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Однако если устройство отказывается от ответа на определенный запрос (к примеру, 42 переменных), Zabbix делает 2 вещи.

Первое, для текущей серии элементов данных Zabbix делит пополам количество элементов данных за один запрос и запрашивает 21 переменных. Если устройство доступно, далее запросы должны работать в большинстве случаев, потому что известно что 28 переменных забиралось, а 21 значительно меньше. Тем не менее если проблема с запросами продолжается, Zabbix уменьшает количество запросов последовательно согласно этому алгоритму. Если и далее проблемы с запросами все еще актуальны, значит устройство определенно не отвечает и количество запросов это не корень проблемы.

Второе дело, которое делает Zabbix для дальнейших порций элементов данных - это, начиная с последнего удачного количества переменных (28 в нашем случае), продолжает увеличивать количество переменных за запрос на 1 до достижения лимита. Например, предположим что максимально возможное количество запросов для данного устройства это 32, последующие запросы будут следующими 29,30,31,32 и 33. Последний запрос будет неудачным и Zabbix никогда более не запросит 33 значения за один запрос. С этого момента, Zabbix всегда будет запрашивать 32 значения для этого устройства.

Если большие запросы неудачно завершаются с определенным количеством переменных, это может означать одно из двух. Точный критерий по которому устройство может ограничивать запросы неизвестен, но мы можем приблизительно рассчитать количество переменных. Первая вероятность - что количество значений примерно равно действительному лимиту размера для данного устройства в общем случае: иногда запросов либо меньше чем лимит, иногда больше. Вторая вероятность, что UDP пакет был потерян. В этом случае, если Zabbix сталкивается с неудачным запросом, он уменьшает максимальное количество запрашиваемых значение за запрос для попытки получения с устройства корректного диапазона, но (начиная с 2.2.8) только до 2 раз.

В примере выше, если запрос с 32 переменными будет неудачен, Zabbix уменьшит количество до 31. Если неудача случиться снова, Zabbix уменьшит количество до 30. Тем не менее, Zabbix не будет уменьшать количество ниже 30, потому что он предположит, что следующие проблемы по причине потерянных UDP пакетов, чем скорее ограничение устройства.

Если, однако, устройство не может обрабатывать массовые запросы корректно и по другим причинам, начиная с Zabbix 2.4 имеется настройка “Использовать массовые запросы” у каждого интерфейса, которая позволяет отключить массовые запросы у этого устройства.