

Поиск

E-mail

Подписаться

☒ [www.samag.ru](#) ☐ Web

0 товаров
сумма 0 руб.

[О ЖУРНАЛЕ](#)

[ЖУРНАЛ «БИТ»](#)

[ПОДПИСКА](#)

[ГДЕ КУПИТЬ](#)

[АВТОРАМ](#)

[РЕКЛАМОДАТЕЛЯМ](#)

[МАГАЗИН](#)

[АРХИВ НОМЕРОВ](#)

[ВАКАНСИИ](#)

[КОНТАКТЫ](#)

УПРАВЛЯЕМ СЕТЕВЫМ ОБОРУДОВАНИЕМ С ПОМОЩЬЮ ПРОТОКОЛА SNMP

[Архив номеров](#) / [2006](#) / [Выпуск №3 \(40\)](#) / [Управляем сетевым оборудованием с помощью протокола SNMP](#)

Рубрика: Сети / Сети | [Дополнительные материалы](#)

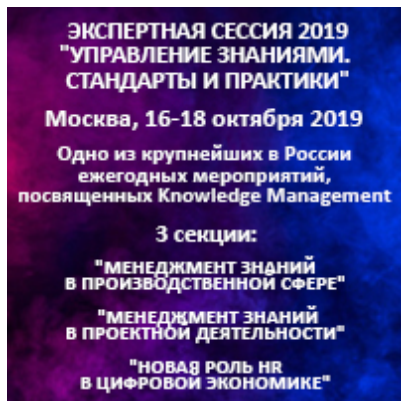


Андрей Бирюков

Управляем сетевым оборудованием с помощью протокола SNMP

Управлять активным сетевым оборудованием можно различными средствами, например, с помощью Telnet или SSH. Но одним из наиболее быстрых и удобных средств взаимодействия является протокол SNMP.

Задача автоматизации управления различным сетевым оборудованием существует со времени появления первых сетевых устройств. На сегодняшний день практически в любой сети можно найти активное сетевое оборудование, управление которым можно, а как правило, нужно автоматизировать. Для решения подобных задач был разработан протокол SNMP (Simple Network Management Protocol). Существует масса готовых коммерческих решений по управлению различными устройствами с помощью SNMP, например HP Open View, однако не каждой организации по карману приобретение подобного ПО, к тому же эти программные продукты предназначены для управления большим количеством устройств, и их использование в небольших сетях будет нецелесообразным.



► Опросы

Какие курсы вы бы выбрали для себя? ◀

- ☐ Очные
- ☐ Онлайновые
- ☐ Платные
- ☐ Бесплатные
- ☐ Я и так все знаю

Голосовать

Читайте далее...

1001 и 1 книга

28.05.2019г.

Просмотров: 552

Комментарии: 1

Анализ вредоносных программ

Читайте далее...

28.05.2019г.

Просмотров: 688

Комментарии: 1

Рассмотрим теоретические основы работы протокола SNMP и практическую реализацию решения некоторых задач с помощью сценариев на языке Perl.


Описание протокола и его компонентов

SNMP использует UDP в качестве транспортного протокола порт 161 и предназначен для использования сетевыми управляющими станциями, как правило серверами, в качестве управляющих и активного сетевого оборудования в качестве управляемых систем. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение управляющих станций.

SNMP-сообщения не имеют фиксированного формата и фиксированных полей. При работе протокол SNMP использует управляющую базу данных MIB – (Management Information Base), которая определяется стандартами RFC1213, 1212.

Общая архитектура взаимодействия по протоколу SNMP имеет следующий вид: на устройстве, которым необходимо управлять, запущен агент SNMP (см. рис. 1).


Микросервисы и контейнеры Docker

 Читать далее...

28.05.2019г.

Просмотров: 540
Комментарии: 0


Django 2 в примерах

 Читать далее...

28.05.2019г.

Просмотров: 435
Комментарии: 0


Введение в анализ алгоритмов

 Читать далее...


27.03.2019г.

Просмотров: 1024
Комментарии: 0

Arduino Uno и Raspberry Pi 3: от схемотехники к интернету вещей

 Читать далее...

Друзья сайта 

Форум системных администраторов 

sysadmins.ru

Рисунок 1. Структура взаимодействия SNMP

Рисунок 1. Структура взаимодействия SNMP

Агенты имеют доступ к информации об управляемом устройстве, на котором они запущены и делают ее доступной для систем сетевого управления NMS (Network Management Systems).

Управляемое устройство может быть любого типа, лишь бы оно было подключено к сети, это могут быть как компьютеры, так и сервера, принтеры, маршрутизаторы, коммутаторы и даже DSL-модемы.

Для примера: устройство может отслеживать следующие параметры:

- Количество и состояние своих виртуальных соединений (Virtual circuits).
- Количество принятых сообщений об ошибках определенного рода (Number of certain kinds of error messages received).
- Количество байт и пакетов, принятых и посланных этим устройством.
- Максимальное значение длины очереди (для маршрутизаторов и других межсетевых устройств).
- Количество принятых и отправленных ширококестельных сообщений.
- Состояние каждого из своих интерфейсов.

Забегая вперед, скажу, что задача получения значений некоторых из этих параметров будет реализована далее в этой статье.

Отправляем объекты

Существует несколько стандартов на базы данных управляющей информации для протокола SNMP. Основные – стандарты MIB-I и MIB-II и версия базы данных для удаленного управления RMON MIB. Спецификация MIB-I определяла только операции чтения значений переменных. Операции изменения или установки значений объекта являются частью спецификаций MIBII.

Версия MIB-I определяет порядка 114 объектов, которые подразделяются на 8 групп.

- **System** – общие данные об устройстве (например, идентификатор поставщика, время последней инициализации системы).
- **Interfaces** – параметры сетевых интерфейсов устройства (например, их количество, типы, скорости обмена, максимальный размер пакета).
- **Address Translation Table** – описание соответствия между сетевыми и физическими адресами (например, по протоколу ARP).
- **Internet Protocol** – данные протокола IP (адреса IP-шлюзов, хостов, статистика о IP-пакетах).
- **ICMP** – данные протокола обмена управляющими сообщениями ICMP.
- **TCP** – данные протокола TCP (например, о TCP-соединениях).
- **UDP** – данные протокола UDP (число переданных, принятых и ошибочных UDP-дейтаграмм).
- **EGP** – данные протокола обмена маршрутной информацией Exterior Gateway Protocol (число принятых с ошибками и без ошибок сообщений).

В версии MIB-II был расширен набор стандартных объектов, а число групп увеличилось до 10. В число объектов, описывающих каждый конкретный интерфейс устройства, включены следующие:

- **ifType** – тип протокола, который поддерживает интерфейс. Этот объект принимает значения всех стандартных протоколов канального уровня, например rfc877-x25, ethernet-csmacd, iso88023-csmacd, iso88024-tokenBus, iso88025-tokenRmg и т. д.
- **ifMtu** – максимальный размер пакета сетевого уровня, который можно послать через этот интерфейс.
- **ifSpeed** – пропускная способность интерфейса в битах в секунду (100 для Fast Ethernet).
- **ifPhysAddress** – физический адрес порта, для Fast Ethernet им будет MAC-адрес. ifAdminStatus – желаемый статус порта:
- **up** – готов передавать пакеты;
- **down** – не готов передавать пакеты;
- **testing** – находится в тестовом режиме.
- **ifOperStatus** – фактический текущий статус порта, имеет те же значения, что и ifAdmin-Status.
- **ifIn Octets** – общее количество байт, принятое данным портом, включая служебные, с момента последней инициализации SNMP-агента.
- **ifInUcastPkts** – количество пакетов с индивидуальным адресом интерфейса, доставленных протоколу верхнего уровня.
- **ifInNUcastPkts** – количество пакетов с ширококестельным или мультивещательным адресом интерфейса, доставленных протоколу верхнего уровня.
- **ifInDiscards** – количество пакетов, которые были приняты интерфейсом, оказались корректными, но не были доставлены протоколу верхнего уровня, скорее всего из-за переполнения буфера пакетов или же по иной причине.

- **ifInErrors** – количество пришедших пакетов, которые не были переданы протоколу верхнего уровня из-за обнаружения в них ошибок.

Каждый объект в дереве значений SNMP определяется с помощью уникального идентификатора OID (Object ID). OID можно представить либо в числовом виде (то есть в том виде, который использует SNMP), либо в текстовом, с использованием MIB-файлов. Следует также отметить, что существуют базы MIB от различных производителей как аппаратного, так и программного обеспечения, которые позволяют представлять числовые значения параметров своих SNMP-агентов в символьном виде. Формат MIB-файлов описан в RFC-1212 [3].

Составное числовое имя объекта SNMP MIB соответствует полному имени этого объекта в дереве регистрации объектов стандартизации ISO.

Пространство имен объектов ISO имеет древовидную иерархическую структуру. От корня этого дерева отходят три ветви, соответствующие стандартам, контролируемым ISO, ITU и совместно ISO-ITU. В свою очередь, организация ISO создала ветвь для стандартов, создаваемых национальными и международными организациями (ветвь org). Объекты любых стандартов, создаваемых под эгидой ISO, однозначно идентифицируются составными символьными именами, начинающимися от корня этого дерева. В сообщениях протоколов символьные имена не используются, а применяются однозначно соответствующие им составные числовые имена. Каждая ветвь дерева имен объектов нумеруется в дереве целыми числами слева направо начиная с единицы, и эти числа и заменяют символьные имена. Поэтому полное символьное имя объекта MIB имеет вид: iso.org.dod.internet.mgmt.mib, а полное числовое имя: 1.3.6.1.2.1 (см. рис. 2).


 Рисунок 2. Древоподобная структура MIB



Рисунок 2. Древоподобная структура MIB

Таким образом, искомые значения можно получить, обратившись к соответствующим ветвям MIB с помощью числовых имен. На практике это выглядит следующим образом, для того чтобы получить информацию о наименовании устройства необходимо обратиться к следующей ветке MIB `iso.org.dod.internet.mgmt.mib.system.sysName`, а в числовом представлении это будет выглядеть так `1.3.6.1.2.1.1.5`.

Ловушки

Еще одним важным понятием являются Traps, или ловушки, реагирующие на определенные события отправкой сообщений управляющей системе. Какая информация отправляется управляющей системе в сообщении Trap?

Посылаются следующие данные:

- Uptime устройства в виде пары: соответствующий OID uptime, значение. То есть время, которое прошло с момента включения устройства.
- OID, содержащий информацию о произошедшем событии.

Любые пары OID и его значение, которые могут дать дополнительную информацию. Например, когда коммутатор отправляет сообщение о подключении по протоколу Telnet, он также передает OID, содержащий информацию о том, с какого IP-адреса, с какого и на какой порт было осуществлено подключение и OID, идентифицирующий сессию. Информация, переданная в данном уведомлении Trap, будет выглядеть следующим образом (см. рис. 3).

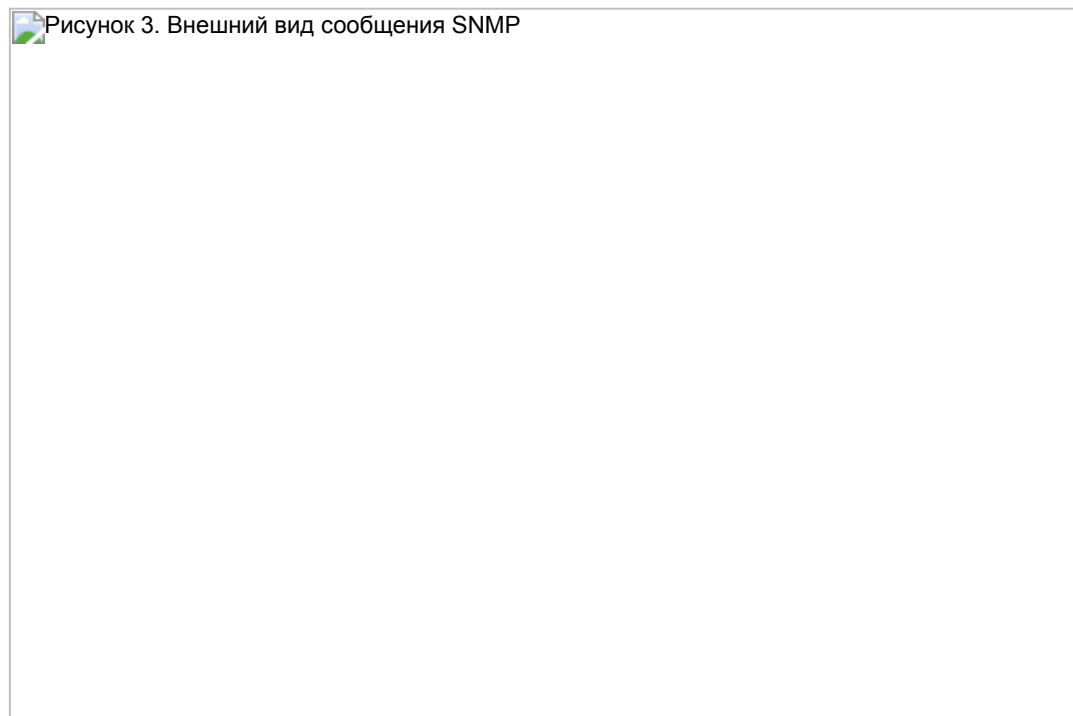


Рисунок 3. Внешний вид сообщения SNMP

Расшифровав это уведомление Trap (сообщение SNMP) с помощью базы MIB для коммутаторов Cisco, можно получить OID и их значения в символьном виде. Например, идентификатор `.1.3.6.1.4.1.9.2.9.3.1.1` в символьном виде будет представлен так: `.iso.org.dod.internet.private.enterprises.cisco.local.its.itsLineSessionTable.itsLineSessionEntry.tslineSesType`, и содержать значение 5. Из описания данного OID узнаем, что это telnet(5), тип сессии Телнет.

Также уведомления Trap весьма полезны при различных внештатных ситуациях, например, отключения порта на коммутаторе или маршрутизаторе.

Симулятор SNMP

Для лучшего понимания теоретических основ функционирования протокола SNMP можно воспользоваться специальным симулятором Advent Net, испытательную версию которого можно загрузить по адресу [6].

Данный симулятор позволяет воспроизвести точную копию различных устройств, как управляющих (MIB-браузер, Trap Recorder), так и управляемых (Agent Simulator, Trap Stormer) с поддержкой протокола SNMP, причем компоненты AdventNet можно использовать и при тестировании взаимодействия с реальными устройствами. Например, можно, включив на маршрутизаторе поддержку SNMP, получать уведомления Trap с помощью Trap Recorder и конфигурировать элементы Management Information Base через MIB-браузер. И наоборот, отправлять уведомления Traps из Trap Stormer и получать их с помощью Perl-сценариев, о которых речь пойдет далее. Следует также сказать несколько слов о поддержке MIB в Advent Net. Симулятор содержит целый ряд баз для различных агентов, в частности имеется MIB для оборудования Cisco, протоколов маршрутизации OSPF и BGP, операционной системы Windows.

Настраиваем оборудование

Прежде чем приступить к настройке SNMP на конкретном управляемом устройстве, необходимо определиться с топологией сети, в которой находятся управляемое устройство и сервер управления SNMP. Как уже упоминалось, данный протокол использует UDP. В связи с этим следует отметить, что при использовании технологии NAT (Network Address Translation) или соединения через VPN-канал между управляющим устройством и сервером управления могут возникнуть трудности с прохождением SNMP-сообщений. Также необходимо позаботиться о том, чтобы в соответствующих списках доступа для портов 161 и 162 (SNMP Traps) был разрешен UDP-трафик.

Приступим к настройке работы SNMP на управляемом устройстве.

На некоторых видах оборудования, таких как аппаратные принт-сервера, ADSL-модемы, а также в некоторых операционных системах, агент SNMP включен по умолчанию. Поэтому мы рассмотрим процедуру настройки взаимодействия по протоколу SNMP на маршрутизаторах Cisco. Данные настройки осуществляются довольно просто. В режиме конфигууратора необходимо ввести:

```
snmp-server community <community_name> RO <access-list>

snmp-server enable traps

snmp-server host <host_address>
```

Первая команда определяет community или «сообщество», представляющее строку, используемую в процессе доступа к устройству. На многих устройствах по умолчанию это public. Однако лучше придумать название подлиннее и посложнее, дабы те, кому не положено, не смогли даже просмотреть значения MIB для данного устройства.

Второй параметр RO – это уровень доступа Read Only, существует также уровень доступа, позволяющий запись (RW), однако настоятельно рекомендую без особой необходимости не давать такого доступа, так как это может привести к самым неприятным последствиям, вплоть до полной потери рабочей конфигурации.

Наконец, третий параметр – это номер Access-list, списка доступа, который определяет, кому разрешен доступ к данному управляемому устройству. Этот параметр не является обязательным, если его не указывать, то сообщения об ошибке не будет, но для ограничения безопасности access-list необходим [1].

Вторая команда включает Trap, то есть SNMP-сообщение, являющееся реакцией на событие или изменение состояния. В данном примере мы будем получать отклики обо всех событиях на управляемом устройстве.

Третья команда определяет SNMP-сервер, на который будут отправляться уведомления Trap.

Итак, мы сконфигурировали SNMP на маршрутизаторе Cisco, теперь посмотрим, как можно с помощью сценариев Perl обратиться к управляемому устройству и получить необходимую информацию.

Программная реализация

Рассмотрим пример, позволяющий выполнить основные операции взаимодействия с управляемым устройством. Исходный код, а также библиотеки, которые были взяты за основу при написании сценариев, можно загрузить по адресу [5]. Однако эти исходные коды были немного доработаны и теперь позволяют получать информацию через веб-интерфейс, так что не забудьте подправить путь к интерпретатору Perl в первой строке сценария. Итак, попробуем прочитывать значения следующих объектов MIB (для простоты воспользуемся MIB RFC 1213), находящихся в ветках iso.org.dod.internet.mgmt.mib.system и iso.org.dod.internet.mgmt.mib.ip:

- **sysDescr** – описание системы;
- **sysContact** – контактная информация;
- **sysName** – наименование системы;
- **sysLocation** – расположение устройства;
- **ipInDelivers** – сколько IP-пакетов отправлено;
- **ipInReceives** – сколько IP-пакетов получено;
- **ipInAddrErrors** – сколько пакетов с неверными адресами.

Значения первых четырех параметров, как правило указываются при первом конфигурировании устройства и затем остаются неизменными, а последние три являются счетчиками и постоянно изменяются.

Как уже упоминалось, приведенный здесь сценарий позволяет получать информацию через веб-интерфейс, создавать странички «на лету», то есть является CGI-сценарием. Не увлекаясь особо веб-программированием, будем выводить эти значения в виде таблицы. Общий алгоритм работы сценария следующий, устанавливаем SNMP-сессию, в которой запрашиваем нужные ветки MIB. Затем полученные значения преобразовываются и выводятся на экран.

Листинг 1. Взаимодействие по протоколу SNMP

```
#!/usr/bin/perl

# Первая строка - путь к интерпретатору Perl

use SNMP_Session;

use BER;

# Эти две библиотеки можно найти в исходных кодах к статье

use strict;

# Объявляем основную процедуру, выводящую значения параметров на экран

sub snmp_get($@);

$SNMP_Session::suppress_warnings = 1;
```

```

my $ipv4_only_p = 0;
my $snmp_version = 1;

# По умолчанию используется версия 1
my $hostname = '172.17.39.1'; # управляемое устройство
my $community = 'admin12345'; # SNMP community

my %ugly_oids = qw(      sysDescr 1.3.6.1.2.1.1.1.0
                        sysContact 1.3.6.1.2.1.1.4.0
                        sysName 1.3.6.1.2.1.1.5.0
                        sysLocation 1.3.6.1.2.1.1.6.0
                        ipInDelivers 1.3.6.1.2.1.4.9.0
                        ipInReceives 1.3.6.1.2.1.4.3.0
                        ipInAddrErrors 1.3.6.1.2.1.4.5.0
                        );

# объекты MIB (OID) в числовом виде
my %pretty_oids;

foreach (keys %ugly_oids) {
    $ugly_oids{$_} = encode_oid (split (/\.\/, $ugly_oids{$_}));
    $pretty_oids{$ugly_oids{$_}} = $_;
}

# в цикле преобразовываем OID
srand();

# устанавливаем SNMP-сессию
my $session = ($snmp_version == 1)
    ? SNMPv1_Session->open ($hostname, $community, 161, undef, undef, undef, undef,undef, $ipv4_only_p)
    : SNMPv2c_Session->open ($hostname, $community, 161, undef, undef, undef, undef, undef, $ipv4_only_p)
    or die "Couldn't open SNMP session to $hostname: $SNMP_Session::errmsg";

# ошибка, если не удалось установить SNMP-сессию

snmp_get ($session, qw(sysDescr sysContact sysName sysLocation ipInDelivers ipInReceives ipInAddrErrors));

$session->close ();

```

```

sub snmp_get ($@) {

    my($session, @oids) = @_ ;

    my($response, $bindings, $binding, $value, $oid);

    grep ($_ = $ugly_oids{$_}, @oids);

    my $interf='';

    my $stroka='';

    if ($session->get_request_response (@oids)) {

        $response = $session->pdu_buffer;

        ($bindings) = $session->decode_get_response ($response);

        print "Content-type: text/html\n\n";

        print "<html> <title> SNMP Management monitor </title> <body> <center><b> SNMP Management Monitor </b><hr>
<table cellpadding=1

            cellspacing=1 border=2>";

        while ($bindings ne '') {

            ($binding, $bindings) = decode_sequence ($bindings);

            ($oid, $value) = decode_by_template ($binding, "%O%@",);

            print "<tr><td bgcolor=yellow><b>", $pretty_oids{$oid}, " </b></td> ";

            $stroka= pretty_print($value);

            print '<td bgcolor=magenta><b>', $stroka, '</b></td></tr>';

        }

        print "</table></body></html>";

    } else {

        warn "SNMP problem: $SNMP_Session::errmsg\n";

    }

}

```

В результате работы данного сценария получаем таблицу, содержащую значения параметров MIB (см. таблицу 1).

Таблица 1. SNMP Management Monitor

sysDescr	Cisco Router
sysContact	SysAdmin
sysName	Core router

sysLocation	Wiring closet
ipInDelivers	12991
ipInReceives	23632
ipInAddrErrors	246

Данный сценарий вполне можно использовать на практике.

Например, для получения информации о состоянии того или иного устройства на текущий момент времени. Однако при добавлении OID своего устройства необходимо убедиться, что управляемое устройство их поддерживает (например, с помощью MIB-браузера), так как в противном случае сценарий вернет ошибку выполнения.

Приведем еще один фрагмент сценария, который осуществляет прием сообщений Trap. Общий алгоритм работы аналогичен предыдущему, поэтому приведу лишь ту часть сценария, которая касается собственно получения сообщений.

Листинг 2. Получение SNMP Traps

```
... # заголовок и объявление библиотек

my $trap_session = SNMPv1_Session->open_trap_session ()

    or die "cannot open trap session";

my ($trap, $sender_addr, $sender_port) = $trap_session->receive_trap ()

    or die "cannot receive trap";

my ($community, $enterprise, $agent,
    $generic, $specific, $sysUptime, $bindings)
    = $trap_session->decode_trap_request ($trap)

    # устанавливаем trap session

    or die "cannot decode trap received"

... # команды, связанные с получением сообщения,
... # аналогичные предыдущему листингу

my ($binding, $oid, $value);

while ($bindings ne '') {

    ($binding,$bindings) = &decode_sequence ($bindings);

    ($oid, $value) = decode_by_template ($binding, "%O%@" );

    # декодируем полученное сообщение

    print BER::pretty_oid ($oid), " => ", pretty_print ($value);

    # выводим ветку MIB и значение в строковом формате

    %mail = ( To          => 'admin@mynetwork.ru',
```

```

From      => 'snmptrap@mynetwork.ru',
Message => "SNMP trap received:". $bindings,

SMTP      => '10.0.1.2'

);

sendmail(%mail) or die $Mail::Sendmail::error;

# письмо администратору
}

# здесь собственно производится прием сообщений Trap, для того чтобы получать эти сообщения постоянно,
# необходимо использовать бесконечный цикл, который прерывается при нажатии любой клавиши
While (@_='') {
my ($trap, $sender_addr, $sender_port) = $trap_session->receive_trap ()
    or die "cannot receive trap";
}

```

В результате получим содержимое сообщений SNMP Trap, которые пришли от управляемых устройств.

Например, вот так будет выглядеть сообщение SNMP, которое прислал коммутатор после отключения его интерфейса.

```

.1.3.6.1.2.1.2.2.1.2.4 STRING FastEthernet0/4
.1.3.6.1.3.1.9.2.2.1.1.20.4 STRING LostCarrier

```

А так выглядит SNMP Trap, сообщающий о том, что интерфейс включен:

```

.1.3.6.1.2.1.2.2.1.2.3 STRING FastEthernet0/3
.1.3.6.1.4.1.9.2.2.1.1.20.3 STRING up

```

Аналогичным образом можно получить сообщения, касающиеся других событий, происходящих с управляемым устройством.

Заключение

В завершении хотелось бы отметить ряд моментов. Прежде всего, следует обратить внимание на то, что у многих крупных производителей имеются свои базы MIB и для более эффективного использования возможностей SNMP лучше использовать MIB для оборудования конкретного производителя. Для поиска баз MIB воспользуйтесь сайтом MIBSearch.com. Что же касается активного сетевого оборудования Cisco, то на CPAN.org можно найти множество Perl-сценариев для взаимодействия по протоколу SNMP, а также сценариев, осуществляющих разбор SNMP-сообщений для конкретных событий, например для событий, связанных с протоколами динамической маршрутизации, потерей пакетов, состоянием VLAN и так далее.

Также, возможно, у вас сложилось впечатление, что протокол SNMP на практике использует только активное сетевое оборудование, канального и сетевого уровней, с встроенным программным обеспечением, однако это не так. К примеру, некоторые виды

программного обеспечения, такие как корпоративные антивирусные системы, активно используют уведомления Trar для оповещения о вирусных инцидентах и эпидемиях. Также SNMP используют системы резервного копирования и системы бесперебойного питания.

Так что данный протокол полезен не только для организации систем управления сетевым оборудованием, но и для решения повседневных задач системного администрирования.

Литература, ссылки:

1. Основы организации сетей Cisco. Справочное руководство.
2. www.opennet.org – интернет-ресурс со множеством статей и примеров, посвященных реализации SNMP на различных устройствах.
3. www.ietf.org – ресурс содержит все стандарты RFC, в том числе и RFC 1212, 1213.
4. www.CPAN.org – ресурс, на котором можно найти большое количество исходных кодов Perl-сценариев для работы с SNMP.
5. <http://www.switch.ch/misc/leinen/snmp/perl/dist> – исходный код, а также библиотеки, использованные при написании сценария.
6. <http://www.adventnet.com/products/simulator> – дистрибутив симулятора SNMP.

Комментарии отсутствуют

[Добавить комментарий](#)

Комментарии могут оставлять только зарегистрированные пользователи



Copyright © Системный администратор

Рейтинг@Mail.ru 

Tel.: (499) 277-12-41
Fax: (499) 277-12-45
E-mail: sa@samag.ru