Расширение возможностей протоколирования и мониторинга шлюза безопасности «С-Терра Шлюз»

Настоящий документ содержит описание способа совместного использования Продуктов компании ООО «С-Терра СиЭсПи» и Продуктов третьих производителей.
ООО «С-Терра СиЭсПи» осуществляет сопровождение настоящего сценария в части настроек Продуктов Компании. Упоминание наименований, продуктов, торговых марок третьих организаций исключительно неформально и не является поддержкой, рекомендацией либо рекламой. ООО «С-Терра СиЭсПи» не несет какой-либо ответственности в отношении работоспособности и использования этих Продуктов.
Документ имеет статус вспомогательного материала, который может быть использован технологическими партнерами, компаниями-интеграторами, при разработке собственных решений.
Решения, разработанные на базе данного сценария, могут применяться в действующих сетях/системах только после тестовой и/или опытной эксплуатации.

Назначение документа

В данном документе рассматриваются:

- настройка SNMP агентов для сбора статистики со шлюза безопасности «С-Терра Шлюз»;
- примеры скриптов для получения расширенной статистики по SNMP;
- использование шаблона для системы мониторинга Zabbix (6.0 LTS, 6.2, 7.0 LTS);
- настройки Syslog;
- настройка отправки почты;
- прочее.

Описание стенда

С устройства Server будет запрашиваться статистика шлюза GW1 по протоколу SNMP. Построение защищенного IPsec туннеля в данном документа рассматриваться не будет.

Канал между системой мониторинга Server и шлюзом GW1 должен быть доверенным. В данном документе мониторинг будет осуществляться через подсеть 192.168.1.0/24, которая является доверенной. Если требуется осуществлять мониторинг через Интернет, то трафик нужно защитить при помощи IPsec.

Схема стенда (Рисунок 1):



Рисунок 1

Общая информация

В состав Продукта «С-Терра Шлюз» входит два SNMP агента. Первый (встроенный агент VPNGATE) отвечает за предоставление информации о подсистемах IKE/IPsec, второй (пакет `snmpd`) – за выдачу

общей информации о системе (счетчики сетевых интерфейсов, загрузка ЦПУ/ОЗУ и так далее) и о информации о третьих сервисах (Keepalived и так далее).

Два SNMP агента работают следующим образом. Агент SNMPD принимает все SNMP запросы на порту 161 и проксирует определенные объектные идентификаторы (OIDs, список можно посмотреть в `/etc/snmp/snmpd.conf`) агенту VPNGATE.

SNMP агент VPNGATE не поддерживает SNMP v3, но благодаря проксированию запросов от SNMPD к VPNGATE становится возможным получение статистики о подсистемах IKE/IPsec по SNMP v3.

Дополнительную информацию о настройке SNMP можно прочитать в документации на «С-Терра Шлюз 5.0» (https://doc.s-terra.ru/ -> С-Терра Шлюз -> С-Терра Шлюз 5.0 -> Мониторинг).

В данном документе для настройки SNMP агента VPNGATE будет использоваться cisco-like консоль.

Настройка SNMP v2

Базовая настройка SNMP v2 состоит из двух этапов. Первый – правка конфигурационного файла `/etc/snmp/snmpd.conf` агента SNMPD. Второй – настройка SNMP агента VPNGATE через cisco-like консоль.

Для базовой настройки SNMP v2 следует выполнить следующие действия:

1. В конфигурационном файле `/etc/snmp/snmpd.conf` нужно отредактировать следующую строку, указав желаемые параметры:

   ```
   agentaddress udp:<адрес>:<порт>
   ```

   Где:

   - `agentaddress udp:<IP адрес шлюза>:<порт>` – локальный адрес шлюза и UDP порт (161 по умолчанию), на которых SNMPD будет слушать запросы по SNMP.

   При необходимости стоит также поменять community в строке:

   ```
   rocommunity public default -V customview
   ```

   Где:

   - `rocommunity` – задает значение community-строки (в данном случае – `public`) с правами только на чтение, а следующее за ним ключевое слово `default -V <имя>` указывает какие OID будут выдаваться.

   Пример конфигурационного файла `/etc/snmp/snmpd.conf` приведен в Приложении.

2. После внесения изменений в конфигурационный файл `/etc/snmp/snmpd.conf` перезапустите демон SNMPD и добавьте его в автозапуск:

   ```
   root@GW1:~# systemctl restart snmpd.service
   root@GW1:~# systemctl enable snmpd.service
   ```

3. В cisco-like консоли перейдите в режим конфигурирования и настройте встроенный SNMP агент VPNGATE:

   ```
   GW1#configure terminal
   Enter configuration commands, one per line.  End with CNTL/Z.
   GW1(config)#snmp-server community public
   GW1(config)#snmp-server polling address 127.0.0.1
   GW1(config)#snmp-server polling udp-port 1161
   GW1(config)#end
   GW1#exit
   ```

   Community строка в команде `snmp-server community` (в данном случае `public`) и UDP порт в `snmp-server polling udp-port` должны соответствовать строкам из настроек proxy в файле `/etc/snmp/snmpd.conf` (строки `proxy -v 2c -c "public" …`). Данную настройку менять не рекомендуется.

Проверка работоспособности SNMP v2

Для проверки можно воспользоваться предустановленными утилитами `snmpget` или `snmpwalk`.

На настроенном устройстве GW1 выполните команду:

```
root@GW1:~# snmpwalk -v 2c -c public <IP адрес шлюза>:161
```

Где:

- `<IP адрес шлюза>` – это адрес, на который должны поступать SNMP запросы.

На экран будут выведены все SNMP значения, предоставляемые SNMPD.

Выполните команду:

```
root@GW1:~# snmpwalk -v 2c -c public <IP адрес шлюза>:161 .1.3.6.1.4.1.9
```

Где:

- `<IP адрес шлюза>` — это адрес, на который должны поступать SNMP запросы.

На экран будут выведены SNMP значения, предоставляемые встроенным агентом VPNGATE.

Также следует проверить работу утилиты `snmptranslate`. Если передать ей в качестве параметра объектный идентификатор в численном виде, то в стандартный вывод она должна вернуть его в текстовом. Выводы должны соответствовать представленным ниже.

```
root@GW1:~# snmptranslate .1.3.6.1.4.1.9
CISCO-SMI::cisco
root@GW1:~# snmptranslate .1.3.6.1.2.1.31.1.1.1.1
IF-MIB::ifName
root@GW1:~# snmptranslate
.1.3.6.1.4.1.8072.1.3.2.3.1.2.29.103.101.116.95.100.97.121.115.95.117.110.116.105.108
.95.108.111.99.97.108.95.99.101.114.116.95.101.120.112
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_days_until_local_cert_exp"
```

После локальной проверки следует таким же образом проверить работу `snmpwalk` с устройства Server, выполнив те же команды. В случае, если команда работает нормально на устройстве GW1, но не работает с устройства Server, можно прийти к выводу, что проблема кроется не в настройках SNMP, а в сетевых настройках.

В приложении так же можно найти примеры использования SNMP запросов для различных случаев.

Настройка SNMP v3

1. В конфигурационном файле `/etc/snmp/snmpd.conf` нужно отредактировать следующую строку, указав желаемые параметры (если это уже не было сделано ранее):

   ```
   agentaddress udp:<адрес>:<порт>
   ```

   Где:

   - `agentaddress udp:<IP адрес шлюза>:<порт>` – локальный адрес шлюза и UDP порт (161 по умолчанию), на которых SNMPD будет слушать запросы по SNMP.

2. Создайте учетную запись пользователя SNMP v3. Добавьте в файл конфигурации `/etc/snmp/snmpd.conf` следующие строки:

   ```
   rouser <username> <security level> -V customview
   ```

   Где:

   - `rouser` – указывает права создаваемого пользователя. В данном примере пользователь будет создан с правами только на чтение (для прав на чтение и запись укажите `rwuser`);
   - `<username>` – имя учетной записи;
   - `<security level>` - способ аутентификации пользователя, должен быть одним из следующих:
     - `noauth` – только по имени пользователя (без шифрования соединения);
     - `auth` – по имени пользователя и паролю `<authpass>` (без шифрования соединения);
     - `priv` – по имени пользователя и паролям `<authpass>` и `<privpass>` (с шифрованием соединения);
   - `-V customview` – указывает, что по умолчанию будут выдаваться OID, входящие в `customview`.

   ```
   createUser <username> [(MD5|SHA) <authpass> [DES|AES [<privpass>]]]
   ```

   Где:

   - `<username>` – имя учетной записи;

- `MD5|SHA` – варианты аутентификации;
- `<authpass>` – пароль аутентификации (должен быть не короче 8 символов);
- `DES|AES` – протоколы шифрования (используется OpenSSL);
- `<privpass>` – пароль шифрования (если не указан, используется тот же пароль, что указан в `<authpass>`).

Пример 1:

```
rouser user auth -V customview
createUser user MD5 password
```

Пример 2:

```
rouser test_user priv -V customview
createuser test_user SHA supersecret123 AES superpassword456
```

После завершения всех настроек и проверки работоспособности, строку `createUser` можно удалить из конфигурационного файла, чтобы пароли доступа не оставались лежать в открытом виде.

3. Опционально: чтобы отключить SNMP v2 закомментируйте следующую строку:

```
rocommunity public default -V customview
```

4. После внесения изменений в конфигурационный файл `/etc/snmp/snmpd.conf` перезапустите демон SNMPD и добавьте его в автозапуск:

```
root@GW1:~# systemctl restart snmpd.service
root@GW1:~# systemctl enable snmpd.service
```

5. В cisco-like консоли перейдите в режим конфигурирования и настройте встроенный SNMP агент VPNGATE (если это уже не было сделано ранее):

```
GW1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
GW1(config)#snmp-server community public
GW1(config)#snmp-server polling address 127.0.0.1
GW1(config)#snmp-server polling udp-port 1161
GW1(config)#end
GW1#exit
```

Community строка в команде `snmp-server community` (в данном случае `public`) и UDP порт в `snmp-server polling udp-port` должны соответствовать строкам из настроек proxy в файле `/etc/snmp/snmpd.conf` (строки `proxy -v 2c -c "public" …`). Данную настройку менять не рекомендуется.

Проверка работоспособности SNMP v3

Для проверки можно воспользоваться предустановленными утилитами `snmpget` или `snmpwalk`.

Если используется метод аутентификации `auth`, выполните на настроенном устройстве GW1 команду (на основе Примера 1 выше):

```
root@GW1:~# snmpwalk -v 3 -u user -l authNoPriv -a MD5 -A password <IP адрес шлюза>:161
```

Где:

- `<IP адрес шлюза>` – это адрес, на который должны поступать SNMP запросы.

Если используется метод аутентификации `priv`, выполните команду (на основе Примера 2 выше):

```
root@GW1:~# snmpwalk -v 3 -u test_user -l authPriv -a SHA -A supersecret123 -x AES -X
superpassword456 <IP адрес шлюза>:161
```

Где:

- `<IP адрес шлюза>` – это адрес, на который должны поступать SNMP запросы.

Альтернативный способ управления настройками SNMP v3

Описанный в этом разделе способ управления настройками SNMP v3 является опциональным. Утилиты `snmpusm` и `snmpvacm` позволяют удаленно и без внесения изменений в конфигурационный файл `snmpd.conf` управлять пользователями, группами и view, в том числе создавать, удалять, активировать/деактивировать пользователей, определять доступные для выдачи OID и прочее. Все настройки, заданные таким образом, применяются сразу и не требуют перезапуска службы `snmpd`. Все действия выполняются из-под предварительно созданного пользователя SNMP v3.

Предварительные настройки

Создайте пользователя, из-под которого будут выполняться все дальнейшие настройки. Для этого добавьте в конфигурационный файл `/etc/snmp/snmpd.conf` следующие строки (в данном примере будет создан пользователь с именем `snmp_admin` с правами на чтение и запись, методом аутентификации `priv` и двумя паролями):

```
rwuser snmp_admin priv
createUser snmp_admin SHA adminpass123 AES adminsecret123
```

Затем перезапустите сервис `snmpd`:

```
root@GW1:~# systemctl restart snmpd.service
```

Создание и удаление пользователей

С использованием утилиты `snmpusm`, подключившись по SNMP v3 под пользователем `snmp_admin`, создайте нового пользователя (в данном примере будет создан пользовать с именем `user1`):

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> create user1 snmp_admin
```

```
User successfully created.
```

Обратите внимание, что создание нового пользователя осуществляется посредством клонирования уже существующего пользователя. Новый пользователь унаследует метод аутентификации и пароли того пользователя, от которого он был клонирован.

В общем виде команда создания нового пользователя выглядит следующим образом:

```
snmpusm <опции подключения> <адрес шлюза> create <имя нового пользователя> <имя существующего пользователя>
```

Для созданного пользователя `user1` задайте новые пароли:

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> passwd -Ca adminpass123 userpass123 user1
```

```
SNMPv3 Key(s) successfully changed.
```

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> passwd -Cx adminsecret123 usersecret123 user1
```

```
SNMPv3 Key(s) successfully changed.
```

В общем виде команда обновления паролей пользователя выглядит следующим образом:

```
snmpusm <опции подключения> <адрес шлюза> passwd (-Cx|-Ca) <старый пароль> <новый пароль> <имя пользователя>
```

Под пользователем `user1` на данный момент нельзя подключиться по SNMP v3, до тех пор, пока для него не будет задана группа, и не будут определены права доступа к тем или иным OID.

Посмотреть существующих пользователей можно следующей командой:

```
root@GW1:~# snmpwalk -v 3 -u snmp_admin -l authpriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> usmUserSecurityName
```

```
SNMP-USER-BASED-SM-MIB::usmUserSecurityName."....]..21W.d...."."user1"  =  STRING: user1
```

```
SNMP-USER-BASED-SM-MIB::usmUserSecurityName."....]..21W.d...."."snmp_admin"          =
STRING: snmp_admin
```

Для удаления пользователя `user1` используйте следующую команду:

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> delete user1
```

```
User successfully deleted.
```

Подробное описание параметров утилиты `snmpusm` содержится в выводе команд `man snmpusm` и `snmpusm --help`.

Создание и удаление view

View используется для указания перечня выдаваемых OID. Для примера, создадим view с именем `disk_monitoring`, в который будут входить всего две ветви OID (`dskPath` и `dskUsed`, содержащие информацию о файловой системе диска и занятом объеме):

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> createView disk_monitoring dskPath
```

```
View successfully created.
```

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> createView disk_monitoring dskUsed
```

```
View successfully created.
```

Этот view будет использован далее в настройках. В общем виде команда добавления OID в view выглядит следующим образом:

```
snmpvacm <опции подключения> <адрес шлюза> createView <имя view> <OID>
```

Удаление view `disk_monitoring` выполняется аналогично:

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> deleteView disk_monitoring dskPath
```

```
View successfully deleted.
```

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> deleteView disk_monitoring dskUsed
```

```
View successfully deleted.
```

Подробное описание параметров утилиты `snmpvacm` содержится в выводе команд `man snmpvacm` и `snmpvacm --help`.

Создание и удаление групп

Группы используются для того, чтобы задавать права доступа пользователей SNMP к конкретным view. Для примера, создадим группу с именем `DiskWatchers` и поместим туда созданного ранее пользователя `user1`:

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> createSec2Group 3 user1 DiskWatchers
```

```
Sec2group successfully created.
```

В общем виде команда добавления пользователя в группу выглядит следующим образом:

```
snmpvacm <опции подключения> <адрес шлюза> createSec2Group <модель безопасности> <имя
пользователя> <имя группы>
```

`<модель безопасности>` в примере выше принимает значение 3, что соответствует `User-based Security Model (USM)`.

Для группы `DiskWatchers` выдадим права на просмотр view `disk_monitoring`, созданного ранее:

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> createAccess DiskWatchers 3 3 1 disk_monitoring none none
```

```
Access successfully created.
```

В общем виде команда назначения прав для группы выглядит следующим образом:

```
snmpvacm <опции подключения> <адрес шлюза> createAccess <имя группы> <модель безопасности> <минимальный уровень аутентификации> <тип совпадения контекста> <имя view для просмотра> <имя view для записи> <имя view для нотификаций>
```

<модель безопасности> в примере выше принимает значение 3, что соответствует User-based Security Model (USM).

<минимальный уровень аутентификации> в примере выше принимает значение 3, что соответствует уровню authPriv.

<тип совпадения контекста> в примере выше принимает значение 1, что соответствует точному совпадению контекста.

<имя view для просмотра> содержит имя view, который будет доступен на просмотр для данной группы.

<имя view для записи> содержит имя view, который будет доступен на изменение для данной группы. В данном примере – none (пустой view).

<имя view для нотификаций> содержит имя view, который будет доступен для использования в SNMP TRAP и INFORM сообщениях для данной группы. В данном примере – none (пустой view).

Обратите внимание, что пользователь может входить только в одну группу. Группа, в свою очередь, может включать в себя несколько пользователей. Для группы можно указать только по одному view для просмотра, записи и нотификаций.

Для удаления ранее назначенных прав группе DiskWatchers необходимо выполнить следующую команду:

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> deleteAccess DiskWatchers 3 3
```

```
Access successfully deleted.
```

Просмотреть назначенные права можно следующей командой:

```
root@GW1:~# snmpwalk -v3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X adminsecret123 <адрес шлюза> vacmAccessTable
```

```
SNMP-VIEW-BASED-ACM-MIB::vacmAccessContextMatch."DiskWatchers"."".3.authPriv = INTEGER: exact(1)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessContextMatch."grpsnmp_admin"."".3.authPriv = INTEGER: prefix(2)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessReadViewName."DiskWatchers"."".3.authPriv = STRING: disk_monitoring
SNMP-VIEW-BASED-ACM-MIB::vacmAccessReadViewName."grpsnmp_admin"."".3.authPriv = STRING: _all_
SNMP-VIEW-BASED-ACM-MIB::vacmAccessWriteViewName."DiskWatchers"."".3.authPriv = STRING:
SNMP-VIEW-BASED-ACM-MIB::vacmAccessWriteViewName."grpsnmp_admin"."".3.authPriv = STRING: _all_
SNMP-VIEW-BASED-ACM-MIB::vacmAccessNotifyViewName."DiskWatchers"."".3.authPriv = STRING:
SNMP-VIEW-BASED-ACM-MIB::vacmAccessNotifyViewName."grpsnmp_admin"."".3.authPriv = STRING: _all_
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStorageType."DiskWatchers"."".3.authPriv = INTEGER: nonVolatile(3)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStorageType."grpsnmp_admin"."".3.authPriv = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStatus."DiskWatchers"."".3.authPriv = INTEGER: 0
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStatus."grpsnmp_admin"."".3.authPriv = INTEGER: active(1)
```

Для удаления группы `DiskWatchers` используйте следующую команду:

```
root@GW1:~# snmpvacm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> deleteSec2Group 3 user1
```

```
Sec2group successfully deleted.
```

Если в группу было добавлено несколько пользователей, то эту команду нужно выполнить для каждого из входящих в эту группу пользователей.

Посмотреть существующие группы можно следующей командой:

```
root@GW1:~# snmpwalk -v 3 -u snmp_admin -l authpriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> vacmGroupName
```

```
SNMP-VIEW-BASED-ACM-MIB::vacmGroupName.3."user1" = STRING: DiskWatchers
SNMP-VIEW-BASED-ACM-MIB::vacmGroupName.3."snmp_admin" = STRING: grpsnmp_admin
```

Подробное описание параметров утилиты `snmpvacm` содержится в выводе команд `man snmpvacm` и `snmpvacm --help`.

Проверка работоспособности

Для проверки заданных параметров, с помощью утилиты `snmpwalk`, запросите все доступные OID, подключившись под пользователем `user1`:

```
root@GW1:~# snmpwalk -v 3 -u user1 -l authPriv -a SHA -A userpass123 -x AES -X
usersecret123 <адрес шлюза> .1
```

```
UCD-SNMP-MIB::dskPath.1 = STRING: /run
UCD-SNMP-MIB::dskPath.2 = STRING: /
UCD-SNMP-MIB::dskPath.3 = STRING: /dev/shm
UCD-SNMP-MIB::dskPath.4 = STRING: /run/lock
UCD-SNMP-MIB::dskPath.5 = STRING: /sys/fs/cgroup
UCD-SNMP-MIB::dskPath.6 = STRING: /var/log
UCD-SNMP-MIB::dskPath.7 = STRING: /tmp
UCD-SNMP-MIB::dskPath.8 = STRING: /boot/efi
UCD-SNMP-MIB::dskUsed.1 = INTEGER: 10000
UCD-SNMP-MIB::dskUsed.2 = INTEGER: 894952
UCD-SNMP-MIB::dskUsed.3 = INTEGER: 0
UCD-SNMP-MIB::dskUsed.4 = INTEGER: 0
UCD-SNMP-MIB::dskUsed.5 = INTEGER: 0
UCD-SNMP-MIB::dskUsed.6 = INTEGER: 532
UCD-SNMP-MIB::dskUsed.7 = INTEGER: 0
UCD-SNMP-MIB::dskUsed.8 = INTEGER: 3975
UCD-SNMP-MIB::dskUsed.8 = No more variables left in this MIB View (It is past the end
of the MIB tree)
```

Видно, что вывод включает в себя только те OID, которые входят в view `disk_monitoring`.

Активация и деактивация пользователей

Существующих пользователей можно деактивировать, чтобы отключить им доступ, при этом не удаляя самих пользователей. Для примера, деактивируем пользователя `user1`:

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> deactivate user1
```

```
User successfully deactivated.
```

Теперь при попытке подключения под этим пользователем будет отображаться соответствующее сообщение:

```
root@GW1:~# snmpwalk -v 3 -u user1 -l authPriv -a SHA -A userpass123 -x AES -X
usersecret123 <адрес шлюза> .1
```

```
snmpwalk: Unknown user name
```

Для активации пользователя `user1` выполните команду:

```
root@GW1:~# snmpusm -v 3 -u snmp_admin -l authPriv -a SHA -A adminpass123 -x AES -X
adminsecret123 <адрес шлюза> activate user1
```

```
User successfully activated.
```

Расширенные настройки Syslog

Настройка с помощью cisco-like консоли

В «С-Терра Шлюз» версии 5.0 в cisco-like консоли появилась возможность добавлять сервера (по адресу, порту и протоколу) для отправки логов от VPN подсистемы шлюза. Для этого используются следующие команды в конфигурационном режиме:

```
GW1(config)#logging host A.B.C.D transport udp port <1-65535>
GW1(config)#logging host A.B.C.D transport tcp port <1-65535>
GW1(config)#logging host A.B.C.D transport relp port <1-65535>
```

Где:

- `A.B.C.D` – адрес сервера для отправки логов. Поддерживаются IPv4 и IPv6 адреса.

Параметры применяются сразу после введения команд, перезапуск сервиса `rsyslog` выполняется автоматически.

Настройки сохраняются в конфигурационный файл `/etc/rsyslog.d/05-sterragate.conf`, редактировать его вручную не рекомендуется.

Можно задавать несколько серверов, отправка логов будет происходить на каждый из них. Например:

```
GW1(config)#logging host 10.0.1.10 transport udp port 514
GW1(config)#logging host 10.0.1.20 transport tcp port 514
GW1(config)#logging host 10.0.2.50 transport relp port 10514
```

Для удаления сервера используйте команду с префиксом `no`. Например:

```
GW1(config)#no logging host 10.0.1.20 transport udp port 514
```

Добавление серверов для отправки логов не отключает логирование в локальный файл `/var/log/cspvpngate.log`, а работает в дополнение к нему.

Настройка с помощью пользовательских файлов rsyslog

Для настройки Syslog необходимо создать пользовательский файл в директории `/etc/rsyslog.d/`. Файл должен иметь такое название, чтобы он выполнялся раньше файла `05-sterragate.conf`. Для этого его можно назвать любым именем, начинающимся на число менее «05» и заканчивающимся на «.conf» (`0[0-4]-[[:word:]]*.conf`).

В файле (назовем его для примера `04-custom.conf`) можно настроить отправку логов на несколько серверов по заданным IP адресам и номерам портов, добавив для этого в файл конфигурационные строки.

Настройки для отправки логов имеют следующий вид:

```
*.* <@|@@|:omrelp:><IP адрес>:<порт>
```

Где:

- `*.*` – в первом поле указывается канал протоколирования событий (factility), на который отправляются записи, во втором – уровень детализации;
- `<@|@@|:omrelp:>` – здесь указывается по какому протоколу будет осуществляться передача логов. `@` – передача осуществляется по UDP (на данный момент рекомендуется), `@@` – передача осуществляется по TCP, `:omrelp:` – передача осуществляется по протоколу RELP (в этом случае следует в начало файла добавить строку `module(load="omrelp")`);
- `<IP адрес>:<порт>` – логи должны передаваться на указанные здесь порт и IP адрес.

В качестве примера ниже разобраны несколько строк:

Пример:

```
*.* @192.168.0.10:516
```

Строка указывает на то, что все логи всех уровней важности нужно отправлять на адрес, заданный после символа `@` (для передачи используется протокол UDP).

При таких настройках `rsyslog` на принимающей стороне следует позаботиться о ротации логов (к примеру, воспользоваться `logrotate`).

Для того, чтобы добавить имя хоста в `syslog` сообщения, задайте соответствующее имя в файле `/etc/hosts` в следующей строке:

```
127.0.0.1 <имя хоста> localhost
```

Так же следует обратить внимание на то, что некоторые `syslog` сообщения берут имя хоста из файла `/etc/hostname`.

После проведения настройки перезапустите сервис:

```
root@GW1:~# systemctl restart rsyslog.service
```

Дополнительные настройки SNMP

Также опционально в файл `/etc/snmp/snmpd.conf` могут быть добавлены следующие настройки:

Описание платформы

```
sysDescr <строка с описанием платформы>
```

Для корректной работы данной настройки следует закомментировать в `/etc/snmp/snmpd.conf` строку `proxy -v 2c -c "public" 127.0.0.1:1161 .1.3.6.1.2.1.1.1.0`

```
sysContact <строка с контактными данными администратора платформы>
sysLocation <строка с описанием местоположения платформы>
```

Например:

```
sysDescr S-Terra Gate 2000
sysContact A.I.Ivanov, admin@example.com
sysLocation Moscow, DC-2, unit 123
```

Настройка SNMP для мониторинга BGP Peer Status

1. Добавьте в файл конфигурации `/etc/snmp/snmpd.conf` строки, отвечающие за настройку сокетов AgentX:

   ```
   agentXSocket /var/agentx/master,tcp:localhost:705
   ```

2. Добавьте в файл конфигурации `/etc/snmp/snmpd.conf` строку с добавлением нужных OID и комментарий:

   ```
   # BGP4 MIB
   view customview included .1.3.6.1.2.1.15
   ```

3. Добавьте в файл конфигурации `/etc/snmp/snmpd.conf` строки, исключающие потенциально опасные с точки зрения производительности OID (https://docs.frrouting.org/en/latest/snmp.html#net-snmp-configuration) и комментарии к ним:

   ```
   # Remove ipRouteTable from view
   view all    excluded  .1.3.6.1.2.1.4.21
   # Remove ipNetToMediaTable from view
   view all    excluded  .1.3.6.1.2.1.4.22
   # Remove ipNetToPhysicalPhysAddress from view
   view all    excluded  .1.3.6.1.2.1.4.35
   # Remove ipCidrRouteTable  from view
   view all    excluded  .1.3.6.1.2.1.4.24
   # Optionally protect SNMP private/secret values
   view all    excluded  .1.3.6.1.6.3.15
   view all    excluded  .1.3.6.1.6.3.16
   view all    excluded  .1.3.6.1.6.3.18
   ```

4. Создайте файл `/etc/snmp/zebra.conf` и добавьте туда следующую строку:

   ```
   master agentx
   ```

5. В файле `/etc/frr/daemons` замените значения `zebra_options`, `bgpd_options`, `ospfd_options` и `ripd_options` на следующие (нужно добавить опцию `-M snmp`):

   ```
   zebra_options="  -A 127.0.0.1 -M  snmp -s 90000000"
   bgpd_options="   -A 127.0.0.1 -M  snmp"
   ospfd_options="  -A 127.0.0.1 -M  snmp"
   ripd_options="   -A 127.0.0.1 -M snmp"
   ```

6. Перезапустите сервис `frr`:

   ```
   root@GW1:~# systemctl restart frr.service
   ```

7. Добавьте AgentX в конфигурацию `vtysh`:

```
root@GW1:~# vtysh

Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

GW1# configure terminal
GW1(config)# agentx
GW1(config)# end
GW1# write
GW1# exit
```

8. Перезапустите сервис `snmpd`:

```
root@GW1:~# systemctl restart snmpd.service
```

Посмотреть статусы всех пиров можно следующей командой:

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 BGP4-MIB::bgpPeerState
BGP4-MIB::bgpPeerState.10.10.20.2 = INTEGER: established(6)
BGP4-MIB::bgpPeerState.10.10.20.3 = INTEGER: established(6)
BGP4-MIB::bgpPeerState.192.168.255.1 = INTEGER: established(6)
```

Для того чтобы выяснить статус конкретного пира можно воспользоваться утилитой `snmpget`, в качестве OID подав `BGP4-MIB::bgpPeerState.<IP_адрес_пира>`:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 BGP4-MIB::bgpPeerState.10.10.20.2
BGP4-MIB::bgpPeerState.10.10.20.2 = INTEGER: established(6)
```

SNMP трапы

Для приема и разбора SNMP трапов на сервере мониторинга должны быть произведены соответствующие настройки, которые в данной инструкции рассматриваться не будут. Шаблон Zabbix, о котором идет речь в приложении, содержит настройку, позволяющую отслеживать все трапы, получаемые системой, но для того, чтобы ей воспользоваться требуется настройка соответствующих сервисов (пример: `snmptrapd` и `snmptt`). Со списком SNMP трапов, отправляющихся с устройства С-Терра Шлюз по умолчанию, можно ознакомиться на портале документации https://doc.s-terra.ru.

Отправка SNMP трапов, доступных встроенному агенту VPNGATE

Для включения автоматической отправки SNMP трапов, доступных встроенному VPNGATE агенту, выполните следующие настройки в cisco-like консоли:

Встроенный VPNGATE агент может отправлять только трапы версий SNMP v1 и SNMP v2

1. Войдите в режим конфигурации:

```
GW1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

2. Задайте адрес хоста, на который будут отправляться трапы (в данном примере – 192.168.1.5), версию SNMP трапов (в данном примере – SNMP v2), community строку (в данном примере – `public`) и порт (в данном примере – 162):

```
GW1(config)#snmp-server host 192.168.1.5 traps version 2c public udp-port 162
```

3. Опционально: укажите интерфейс, с которого будут отправляться SNMP трапы:

```
GW1(config)#snmp-server trap-source gigabitEthernet0/0
```

4. Включите отправку SNMP трапов:

```
GW1(config)#snmp-server enable traps
```

Отправка SNMP трапов вручную

Утилиты `snmptrap` и `snmpinform` на шлюзе отвечают за отправку SNMP трапов. Шаблон команды для отправки трап сообщения из linux консоли может быть следующим:

```
snmptrap -v <версия SNMP> -c <community-строка> <IP адрес сервера>:<порт> <uptime>
<OID> <object/OID> <тип переменной> <значение>
```

Значение параметра `uptime` задавать необязательно и можно заменить пустыми кавычками – “” или ‘’. Примеры отправки тестового SNMP v2-трапа с использованием кастомного OID:

```
root@GW1:~# snmptrap -v 2c -c public 192.168.1.5:162 '' .1.3.6.1.4.1.31342.1.2
.1.3.6.1.4.1.31342.1.2 s "Test message"
root@GW1:~# snmpinform -v 2c -c public 192.168.1.5:162 '' .1.3.6.1.4.1.31342.1.2
.1.3.6.1.4.1.31342.1.2 i 1234
```

Отправка SNMP v3 трапов требует указания дополнительных параметров аутентификации. Пример отправки SNMP v3 трапа:

```
root@GW1:~# snmpinform -v 3 -l authpriv -u User -a MD5 -x DES -A "UserPassword" -X
"UserSecret" 192.168.1.5:162 '' .1.3.6.1.4.1.31342.1.2 .1.3.6.1.4.1.31342.1.2 s "Test
message v3"
```

Расширенное использование SNMP при помощи NET-SNMP-EXTEND-MIB

Сервис SNMPD позволяет расширить использование SNMP для тех случаев, когда какие-то значения нельзя напрямую получить по SNMP от какой-то подсистемы. Расширенное использование SNMP заключается в применении `NET-SNMP-EXTEND-MIB` для запуска произвольных программ или скриптов (директива `extend` в `/etc/snmp/snmpd.conf`).

Программы и скрипты запускаются демоном SNMPD от пользователя Debian-snmp.

На данный момент, посредством `NET-SNMP-EXTEND-MIB`, осуществляется сбор информации о количестве дней до истечения локальных сертификатов, сбор параметров `rtt, jitter, loss` при мониторинге состояния канала, получение статистики от утилиты `kstat_show`, получение статистики от подсистем IPSM/DPDK (для шлюзов DP), получение информации о статусе блоков питания (для некоторых аппаратных платформ). Примеры описаны ниже. При необходимости добавления произвольного функционала просьба руководствоваться данными примерами.

Получение информации о количестве дней до истечения локальных сертификатов

Метод, описанный в данном разделе, позволяет отслеживать сроки истечения локальных сертификатов. Этот метод лучше всего подходит для использования с Zabbix шаблоном.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

   ```
   includeFile /opt/snmp_monitoring/snmp_extend/certs_expiry/certs_expiry.snmpd_conf
   ```

   В соответствии с этими строками конфигурационного файла в системе станут доступны объектные идентификаторы, заданные в файле `/opt/snmp_monitoring/snmp_extend/certs_expiry/certs_expiry.snmpd_conf`:
   `NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_index"`
   `NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_days_til_exp"`

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
0 3 * * * root if [ -e $EXTEND_DIR/certs_expiry/certs_expiry.py ]; then
$PYTHON $EXTEND_DIR/certs_expiry/certs_expiry.py write_params_to_file; fi
```

Данная запись говорит `cron` о том, что скрипт `/opt/snmp_monitoring/snmp_extend/certs_expiry/certs_expiry.py` должен запускаться один раз в сутки (в 03:00 часов) с правами root. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Крайне не рекомендуется задавать вызов данного скрипта чаще чем один раз в сутки, так как, при наличии большого количества сертификатов в базе шлюза, вызов утилиты `cert_mgr` может быть очень ресурсоемкой и длительной операцией.
После настройки `cron` следует выполнить один раз скрипт вручную, чтобы сразу получить результаты: `root@GW1:~# /opt/snmp_monitoring/snmp_extend/certs_expiry/certs_expiry.py write_params_to_file`

Проверка

Выполните запрос на получение номеров локальных сертификатов (обратите внимание на экранирование кавычек в теле запроса):

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_certs_index\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_index".1 = STRING: 1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_index".2 = STRING: 2
```

Из ответа видно, что на шлюзе присутствуют локальные сертификаты с порядковыми номерами 1 и 2.

Номера, о которых идёт речь выше не являются идентификаторами из вывода `cert_mgr`. См. `/opt/snmp_monitoring/snmp_extend/certs_expiry/README.md`

Чтобы получить количество дней до их истечения выполните следующий запрос:

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_certs_days_til_exp\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_days_til_exp".1 = STRING: 351
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_days_til_exp".2 = STRING: 364
```

Первая строка в ответе соответствует локальному сертификату с порядковым номером 1, вторая – локальному сертификату с номером 2.

Чтобы получить количество дней до истечения конкретного сертификата, например, с номером 2, выполните следующий запрос:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_certs_days_til_exp\".2
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_certs_days_til_exp".2 = STRING: 364
```

Описание работы скрипта

Скрипт `certs_expiry.py` с опцией `write_params_to_file` запускается по расписанию в `cron` и пишет информацию из вывода команд `cert_mgr show -i <id>`, где `<id>` это идентификаторы локальных сертификатов, в следующий файл:

`/opt/snmp_monitoring/snmp_extend/certs_expiry/local_certs.info`

Скрипт `certs_expiry.py` с опцией `print_parameter_values --parameter_name {indexes, days}` запускается по соответствующему snmp запросу и выводит последовательно либо значения индексов, либо соответствующее им количество дней до истечения локального сертификата.

Если информация в файле `local_certs.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт

ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Логирование выполняется при помощи syslog в facility local7 (в файл `/var/log/cspvpngate.log` по умолчанию). При логировании указывается название скрипта и суть ошибки.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/certs_expiry/README.md` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите правило обнаружения (Discovery rule) `Local Certificate Discovery` в состояние `Enabled`. По умолчанию, обнаружение локальных сертификатов будет выполняться каждый день в 00 часов 00 минут. Также переведите следующий элемент данных (Items) из раздела `Application: Extended` в состояние `Enabled`:

- Local Certificates (Script Exit Code).

Получение информации о количестве дней до истечения предварительно распределенных ключей

Метод, описанный в данном разделе, позволяет отслеживать сроки истечения предварительно распределенных ключей. Этот метод лучше всего подходит для использования с Zabbix шаблоном. Мониторинг доступен для платформ STK и Aquarius.

Мониторинг для платформ STK и Aquarius доступен в обновленной версии пакета monitoring, начиная с версии `0.12+sterra1-debian12`. Процесс проверки версии пакета и его обновления описан в Приложении.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

```
includeFile /opt/snmp_monitoring/snmp_extend/keys_expiry/keys_expiry.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станут доступны объектные идентификаторы, заданные в файле `/opt/snmp_monitoring/snmp_extend/keys_expiry/keys_expiry.snmpd_conf`:
`NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_index"`
`NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_name"`
`NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_days_til_exp"`

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
0 3 * * * root if [ -e $EXTEND_DIR/keys_expiry/keys_expiry.py ]; then $PYTHON $EXTEND_DIR/keys_expiry/keys_expiry.py write_params_to_file; fi
```

Данная запись говорит `cron` о том, что скрипт `/opt/snmp_monitoring/snmp_extend/keys_expiry/keys_expiry.py` должен запускаться один раз в сутки (в 03:00 часов) с правами root. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Крайне не рекомендуется задавать вызов данного скрипта чаще чем один раз в сутки, так как, при наличии большого количества предварительно распределенных ключей в базе шлюза, вызов утилиты `key_mgr` может быть очень ресурсоемкой и длительной операцией.
После настройки `cron` следует выполнить один раз скрипт вручную, чтобы сразу получить результаты: `/opt/snmp_monitoring/snmp_extend/keys_expiry/keys_expiry.py write_params_to_file`

Проверка

Выполните запрос на получение номеров предварительно распределенных ключей (обратите внимание на экранирование кавычек в теле запроса):

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_keys_index\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_index".1 = STRING: 1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_index".2 = STRING: 2
```

Из ответа видно, что на шлюзе присутствуют предварительно распределенные ключи с порядковыми номерами 1 и 2.

Номера, о которых идёт речь выше не являются идентификаторами из вывода `key_mgr`. См. /opt/snmp_monitoring/snmp_extend/keys_expiry/README.md

Чтобы получить количество дней до их истечения выполните следующий запрос:

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_keys_days_til_exp\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_days_til_exp".1 = STRING: 29
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_days_til_exp".2 = STRING: 123
```

Первая строка в ответе соответствует предварительно распределенному ключу с порядковым номером 1, вторая – предварительно распределенному ключу с номером 2.

Чтобы получить количество дней до истечения конкретного предварительно распределенного ключа, например, с номером 2, выполните следующий запрос:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_keys_days_til_exp\".2
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_days_til_exp".2 = STRING: 123
```

Чтобы получить имена ключей выполните следующий запрос:

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutLine.\"local_keys_name\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_name".1 = STRING: cs_preshared_cont_file1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."local_keys_name".2 = STRING: cs_preshared_cont_file2
```

Первая строка в ответе соответствует имени предварительно распределенного ключа с порядковым номером 1, вторая – предварительно распределенному ключу с номером 2.

Описание работы скрипта

Скрипт `keys_expiry.py` с опцией `write_params_to_file` запускается по расписанию в `cron` и записывает информацию из вывода команды `key_mgr show` в следующий файл:

`/opt/snmp_monitoring/snmp_extend/keys_expiry/local_keys.info`

Скрипт `keys_expiry.py` с опцией `print_parameter_values --parameter_name {indexes, days, names}` запускается по соответствующему snmp запросу и выводит последовательно либо значения индексов, либо соответствующее им количество дней до истечения предварительно распределенного ключа, либо имя предварительно распределенного ключа.

Если информация в файле `local_keys.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Логирование выполняется при помощи syslog в facility local7 (в файл `/var/log/cspvpngate.log` по умолчанию). При логировании указывается название скрипта и суть ошибки.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/keys_expiry/README.md` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите правило обнаружения (Discovery rule) `Local PSK Discovery` в состояние `Enabled`. По умолчанию, обнаружение предварительно распределенных ключей будет выполняться каждый день в 00 часов 00 минут.

Получение параметров при мониторинге состояния канала с помощью ICMP

Данный метод позволяет отслеживать состояние канала и получать такие параметры как `rtt`, `jitter` и `loss` до заданного сетевого адреса при помощи ICMP скрипта.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

```
includeFile
/opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/rtt_jitter_loss.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станут доступны объектные идентификаторы, заданные в файле `/opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/rtt_jitter_loss.snmpd_conf`:

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_rtt"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_jitter"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_loss"
```

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
* * * * * root if [ -e $EXTEND_DIR/get_rtt_jitter_loss/get_rtt_jitter_loss.py ]; then
$PYTHON $EXTEND_DIR/get_rtt_jitter_loss/get_rtt_jitter_loss.py write_params_to_file --
address 8.8.8.8; fi
```

Данная запись говорит `cron` о том, что скрипт `opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/get_rtt_jitter_loss.py` должен запускаться один раз в минуту с правами root. В параметре `--address` задайте желаемый сетевой адрес. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Проверка

Выполните запрос для получения значения `rtt`:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"get_rtt\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_rtt" = STRING: 0.222
```

Выполните запрос для получения значения `jitter`:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"get_jitter\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_jitter" = STRING: 0.260
```

Выполните запрос для получения значения `loss`:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"get_loss\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_loss" = STRING: 0
```

Описание работы скрипта

Скрипт `get_rtt_jitter_loss.py` с опцией `write_params_to_file` и заданным сетевым адресом в параметре `--address`, запускается по расписанию в `cron` и пишет значения `rtt, jitter` и `loss` до указанного сетевого узла на основе команды `ping`, в следующий файл:

`/opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/rtt_jitter_loss.info`

Скрипт `certs_expiry.py` с опцией `print_parameter_values` и параметром `--parameter_name {rtt, jitter, loss}` запускается по соответствующему SNMP запросу и выводит указанные в параметре значения из файла.

Если информация в файле `rtt_jitter_loss.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/README.txt` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите следующие элементы данных (Items) из раздела `Application: Extended` в состояние `Enabled`:

- SLA. RTT;
- SLA. RTT (Script exit code);
- SLA. Jitter;
- SLA. Jitter (Script exit code);
- SLA. Loss;
- SLA. Loss (Script exit code).

Получение статистики от системы шифрования

Для мониторинга по SNMP статистики системы шифрования по количеству входящих, исходящих, потерянных IPsec пакетов, потерям низкоприоритетных и высокоприоритетных пакетов и процентам потерь на шлюзе есть специальный скрипт, работающий с выводом утилиты `kstat_show`.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

```
includeFile /opt/snmp_monitoring/snmp_extend/check_kstat_show/kstat_show.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станут доступны объектные идентификаторы, заданные в файле `/opt/snmp_monitoring/snmp_extend/check_kstat_show/kstat_show.snmpd_conf`:

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."high_prio_dropped_pkts"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."low_prio_dropped_pkts"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."prio_total_dropped_pkts"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."ipsec_prio_dropped_percentage"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."ipsec_dropped_percentage"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."ipsec_total_passed_pkts"
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."ipsec_total_dropped_pkts"
```

На шлюзе DP из вышеуказанных параметров будут доступны только `ipsec_total_passed_pkts`, `ipsec_total_dropped_pkts` и `ipsec_dropped_percentage`.

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле /etc/cron.d/monitoring раскомментируйте следующую строку:

```
* * * * * root if [ -e $EXTEND_DIR/check_kstat_show/check_kstat_show.py ]; then
$PYTHON $EXTEND_DIR/check_kstat_show/check_kstat_show.py write_params_to_file; fi
```

Данная запись говорит cron о том, что скрипт /opt/snmp_monitoring/snmp_extend/check_kstat_show/check_kstat_show.cron.py должен запускаться один раз в минуту с правами root. После редактирования файла сервис cron перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Проверка

Выполните запрос на получение одного из параметров, например, числа потерянных пакетов с меткой high priority:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"high_prio_dropped_pkts\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."high_prio_dropped_pkts" = STRING: 0.0
```

Описание работы скрипта

Скрипт check_kstat_show.py с опцией write_params_to_file запускается по расписанию в cron и пишет значения из вывода команды kstat_show в следующий файл:

/opt/snmp_monitoring/snmp_extend/check_kstat_show/kstat_show.info

Скрипт check_kstat_show.py с опцией print_parameter_values и параметром --parameter_name {HIGH_PRIO_DROPPED_PACKETS, LOW_PRIO_DROPPED_PACKETS, PRIO_TOTAL_DROPPED_PACKETS, PRIO_DROP_PERCENTAGE, IPSEC_DROP_PERCENTAGE, IPSEC_TOTAL_PASSED_PACKETS, IPSEC_TOTAL_DROPPED_PACKETS} запускается по соответствующему SNMP запросу и выводит указанное в параметре значение из файла.

Если информация в файле kstat_show.info устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной THRESHOLD в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Логирование выполняется при помощи syslog в facility local7 (в файл /var/log/cspvpngate.log по умолчанию). При логировании указывается название скрипта и суть ошибки.

Подробное описание работы скрипта присутствует в файле /opt/snmp_monitoring/snmp_extend/check_kstat_show/README.md на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста (кроме шлюзов DP) переведите следующие элементы данных (Items) из раздела Application: Extended в состояние Enabled:

- High priority packets dropped;
- High priority packets dropped (Script exit code);
- Low priority packets dropped;
- Low priority packets dropped (Script exit code);
- IPsec dropped packets percentage;
- IPsec dropped packets percentage (Script exit code).

Получение статистики от подсистем IPSM/DPDK на шлюзах DP

Метод мониторинга подсистем IPSM/DPDK, описанный в данном разделе, предназначен для использования с Zabbix шаблоном. Данный метод позволяет получать статистику по интерфейсам vEth<N> со шлюзов DP, включая IP/MAC/PCI адреса, статус, счетчики полученных/отправленных пакетов и байтов, счетчики различных видов ошибок и потерь трафика, и др.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации /etc/snmp/snmpd.conf раскомментируйте следующую строку:

```
includeFile /opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/ipsm_dpdk.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станут доступны объектные идентификаторы, заданные в файле `/opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/ipsm_dpdk.snmpd_conf`.

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В конфигурационном файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
* * * * * root if [ -e $EXTEND_DIR/ipsm_iface_stats/ipsm_iface_stats.py ]; then $PYTHON $EXTEND_DIR/ipsm_iface_stats/ipsm_iface_stats.py write_params_to_file; fi
```

Данная запись говорит `cron` о том, что скрипт `/opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/ipsm_iface_stats.py` должен запускаться раз в минуту с правами root. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Проверка

Выполните запрос на получение индексов `vEth` интерфейсов:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutputFull.\"ipsm_interface_id\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."ipsm_interface_id" = STRING: 1
0
```

Описание работы скрипта

Скрипт `ipsm_iface_stats.py` с опцией `write_params_to_file` запускается по расписанию в `cron` и пишет значения параметров, указанных в файле `parameters.conf` в следующий файл:

`/opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/ipsm_iface.info`

Скрипт `ipsm_iface_stats.py` с опцией `print_ipsm_iface_ids` запускается по соответствующему SNMP запросу и выводит последовательно значения индексов IPSM интерфейсов (на каждой новой строке), читая их из файла.

Скрипт `ipsm_iface_stats.py` с опциями `print_parameter_values --parameter_group <группа> --parameter_name <имя_параметра>` запускается по соответствующему SNMP запросу и выводит значения указанных параметров из соответствующих групп для каждого из IPSM интерфейсов.

Если информация в файле `ipsm_iface.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Логирование выполняется при помощи syslog в facility local7 (в файл `/var/log/cspvpngate.log` по умолчанию). При логировании указывается название скрипта и суть ошибки.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/README.md` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите правило обнаружения (Discovery rule) `IPSM Interface Discovery` в состояние `Enabled`.

Получение информации о состоянии блоков питания

Мониторинг состояния блоков питания доступен для платформ STK, Aquarius (T50), Lanner NCA-5210C и Lanner FW-8894. На платформах STK и Aquarius мониторинг БП возможен только при наличии установленного IPMI.

Мониторинг состояния блоков питания для платформ STK и Aquarius доступен в обновленной версии пакета monitoring, начиная с версии `0.12+sterra1-debian12`. Процесс проверки версии пакета и его обновления описан в Приложении.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

```
includeFile /opt/snmp_monitoring/snmp_extend/get_power_state/power_state.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станет доступен объектный идентификатор, заданный в файле `/opt/snmp_monitoring/snmp_extend/get_power_state/power_state.snmpd_conf`:
`NET-SNMP-EXTEND-MIB::nsExtendOutLine."get_power_state".1`
`NET-SNMP-EXTEND-MIB::nsExtendOutLine."get_power_state".2`

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
*/15 * * * * root if [ -e $EXTEND_DIR/get_power_state/get_power_state.py ]; then
$PYTHON $EXTEND_DIR/get_power_state/get_power_state.py --write; fi
```

Данная запись говорит `cron` о том, что скрипт `/opt/snmp_monitoring/snmp_extend/get_power_state/get_power_state.py` должен запускаться один раз в 15 минут с правами root. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Проверка

Выполните запрос на получение состояния блоков питания:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-MIB::nsExtendOutputFull.\"get_power_state\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_power_state" = STRING: "up"
"up"
```

Описание работы скрипта

Скрипт `get_power_state.py` с опцией `--write` запускается по расписанию в `cron` и пишет состояние блоков питания в следующий файл:

`/opt/snmp_monitoring/snmp_extend/get_power_state/power_state.info`

Скрипт `get_power_state.py` с опцией `--print` запускается по соответствующему SNMP запросу и выводит информацию о состоянии блоков питания из файла.

Если информация в файле `power_state.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/get_power_state/README.md` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите следующие элементы данных (Items) из раздела `Application: Power Supply` в состояние `Enabled`:

- Power supply 1 state
- Power Supply 2 state
- Power Supply state (Script exit code).

Получение информации о состоянии RAID

Мониторинг состояния RAID доступен для платформ STK и Aquarius, где диски объединены в RAID массив.

Мониторинг состояния RAID для платформ STK и Aquarius доступен в обновленной версии пакета monitoring, начиная с версии `0.12+sterra1-debian12`. Процесс проверки версии пакета и его обновления описан в Приложении.

Настройка расширенного объектного идентификатора в SNMPD

1. В файле конфигурации `/etc/snmp/snmpd.conf` раскомментируйте следующую строку для получения объектных идентификаторов:

```
includeFile
/opt/snmp_monitoring/snmp_extend/get_raid_status/get_raid_status.snmpd_conf
```

В соответствии с этими строками конфигурационного файла в системе станет доступен объектный идентификатор, заданный в файле `/opt/snmp_monitoring/snmp_extend/get_raid_status/get_raid_status.snmpd_conf`: `NET-SNMP-EXTEND-MIB:: nsExtendOutputFull."get_raid_status"`

2. Перезапустите сервис `snmpd` командой `systemctl restart snmpd.service`.

Настройка cron

В файле `/etc/cron.d/monitoring` раскомментируйте следующую строку:

```
* * * * * root if [ -e $EXTEND_DIR/get_raid_status/get_raid_status.py ]; then $PYTHON
$EXTEND_DIR/get_raid_status/get_raid_status.py --write; fi
```

Данная запись говорит `cron` о том, что скрипт `/opt/snmp_monitoring/snmp_extend/get_raid_status/get_raid_status.py` должен запускаться один раз в минуту с правами root. После редактирования файла сервис `cron` перезапускать не обязательно, запись должна подхватиться автоматически в течение минуты.

Проверка

Выполните запрос на получение состояния блоков питания:

```
root@GW1:~# snmpget -v 2c -c public <адрес шлюза>:161 NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"get_raid_status\"
```

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."get_raid_status" = STRING: 0
```

Описание работы скрипта

Скрипт `get_raid_status.py` с опцией `--write` запускается по расписанию в `cron` и пишет состояние RAID массива в следующий файл:

`/opt/snmp_monitoring/snmp_extend/get_raid_status/raid_status.info`

Скрипт `get_raid_status.py` с опцией `--print` запускается по соответствующему SNMP запросу и выводит информацию о состоянии RAID массива из файла.

Если информация в файле `raid_status.info` устарела (в случае, когда информация не обновлялась дольше порогового значения, заданного в переменной `THRESHOLD` в скрипте), то скрипт ничего не выводит, и выходит с кодом возврата 2. Если файл пустой, то скрипт так же ничего не будет выводить, выйдет с кодом возврата 2.

Подробное описание работы скрипта присутствует в файле `/opt/snmp_monitoring/snmp_extend/get_raid_status/README.md` на шлюзе.

Настройка Zabbix

На Zabbix сервере для соответствующего хоста переведите следующие элементы данных (Items) из раздела `Application: Extended` в состояние `Enabled`:

- RAID status;
- RAID status (Script exit code).

Использование SNMP шаблона на Zabbix сервере

Для Продукта «С-Терра Шлюз» был создан шаблон для системы мониторинга Zabbix. Загрузить его можно на портале документации https://doc.s-terra.ru -> в разделе «Типовые сценарии применения» -> Версия 5.0 -> сценарий «Расширение возможностей протоколирования и мониторинга шлюза безопасности «С-Терра Шлюз» ->файл `zbx_sterra_gates_snmpv2_template.zip`. Шаблон протестирован на версиях Zabbix Appliance 6.0 LTS, 6.2, 7.0 LTS.

Данный шаблон позволяет получать следующую информацию:

- Информация об IKE/IPsec SA;
- Информация о количестве свободного места на различных разделах;
- Информация о количестве дней до истечения срока действия локальных сертификатов;
- Информация о состоянии канала (jitter, rtt, loss);
- Информация о сетевых интерфейсах;
- Информация о загрузке ЦПУ, ОЗУ;
- Информация о температуре ЦПУ;
- Информация о состоянии ноды VRRP кластера;
- Информация о состоянии пиров BGP;
- Информация о состоянии блоков питания;
- Информация о состоянии RAID;
- Информация от подсистем IPSM/DPDK (для шлюзов DP).

Импорт шаблона

1. Для импорта шаблона требуется в веб интерфейсе выбрать вкладку `Configuration -> Templates -> Import` и импортировать файл шаблона с параметрами по умолчанию.
2. Для корректной работы шаблона на Zabbix сервере должны присутствовать MIB файлы. Взять их можно со Шлюза 5.0 из директории `/opt/snmp_monitoring/mibs` (включая все поддиректории), либо на портале документации https://doc.s-terra.ru -> в разделе «Типовые сценарии применения» -> Версия 5.0 -> сценарий «Расширение возможностей протоколирования и мониторинга шлюза безопасности «С-Терра Шлюз» -> файл `mibs.zip`. Файлы следует поместить на сервер Zabbix также в директорию `/usr/share/snmp/mibs/` (актуально для Zabbix Appliance 6.0 LTS, 6.2, 7.0 LTS, для других версий директория может отличаться).
3. Перезагрузите Zabbix сервер.

Замечания и ограничения по использованию шаблона

1. Мониторинг срока действия локальных сертификатов использует правило обнаружения (Discovery rule) «Local Certificate Discovery», которое по SNMP запросу получает индексы локальных сертификатов со шлюза, а затем на основе полученных значений создает элементы данных (Items) вида `Local Certificate Index <i> Days Until Expiry`, где i – индекс (id) локального сертификата в базе шлюза. Эти объекты показывают оставшееся число дней до истечения локальных сертификатов с соответствующим id. По умолчанию, данное правило обнаружения выполняется каждый день в 00:00 часов. Для использования данного способа требуется выполнение действий, описанных в разделе «Получение информации о количестве дней до истечения локальных сертификатов». Информация на сервере Zabbix появится не сразу, а только после того, как успешно отработает скрипт `certs_expiry.py` на шлюзе по расписанию в `cron` и успешно выполнится правило обнаружения в Zabbix. Дополнительно, следует включить `Local Certificates (Script Exit Code)` для отслеживания кода возврата скрипта.
2. Значения «SLA. Jitter», «SLA. Loss» и «SLA. RTT» будут доступны только после выполнения инструкций из раздела «Получение параметров при мониторинге состояния канала с помощью ICMP».
3. Значения «IPsec dropped packets percentage», «High priority packets dropped» и «Low priority packets dropped» будут доступны только после выполнения инструкций из раздела «Получение статистики от системы шифрования».

4. Статистика от подсистем IPSM/DPDK для шлюзов DP будет доступна только после выполнения инструкций из раздела «Получение статистики от подсистем IPSM/DPDK на шлюзах DP».
5. Информация о состоянии блоков питания будет доступна только после выполнения инструкций из раздела «Получение информации о состоянии блоков питания».
6. При импорте шаблона в Zabbix 7.0 LTS группы хостов (Host groups) не создаются автоматически. При необходимости можно создать группу `S-Terra Gates` вручную в меню `Data collection > Host groups > Create host group`.

Добавление устройства

1. Перейти на страницу `Configuration -> Hosts -> Create host` для того, чтобы создать узел сети.
2. Во вкладке `Host` в поле `Host name` следует ввести название для добавляемого устройства, в поле `Templates` выбрать `Template S-Terra Gates SNMPv2`, в поле `Groups` выбрать группу `S-Terra Gates`. Добавить SNMP интерфейс, при этом следует задать актуальные адрес, порт и SNMP community. Завершить добавление устройства по кнопке `Add`.
3. Затем, нужно перейти на страницу `Monitoring -> Latest data` для того, чтобы проверить, начал ли осуществляться мониторинг добавленного устройства. Если всё сделано верно, то в подменю `Last value` будут изменяться отслеживаемые параметры.

Для того, чтобы пользоваться шаблоном для SNMP v3, нужно изменить SNMP version с SNMP v2 на SNMP v3. После чего нужно ввести параметры пользователя (нового или ранее созданного), которые были внесены в главе «Настройка SNMP v3» данной инструкции.

Использование обнаружения

В шаблоне присутствуют правила автоматического обнаружения объектных идентификаторов. Каждое обнаружение выполняется с некоторым заданным периодом и по результатам выполнения создаются объекты (Items), по которым можно отслеживать состояние оборудования.

Делается это для того, чтобы отслеживать состояние объектов, названия которых заранее неизвестны. В качестве примера можно привести VRRP. Изначально неизвестно какой именно идентификатор виртуального маршрутизатора (VRID) будет задан пользователем, поэтому неизвестно за чем именно предстоит следить. После выполнения обнаружения (в данном шаблоне `VRRP Instance Discovery`) будут автоматически созданы объекты для отслеживания состояния каждого VRID.

В случае если объекты не создаются сразу после импорта шаблона, можно запустить обнаружение вручную (в веб интерфейсе Zabbix): `Configuration -> Hosts`, найти нужное устройство, перейти в его раздел `Discovery`, выбрать требуемое правило и нажать кнопку `Execute now`.

Использование Zabbix агента

Для мониторинга параметров, помимо SNMP, можно использовать Zabbix агент.

Для того, чтобы использовать Zabbix агент, необходимо выполнить следующие настройки на криптошлюзе:

1. В файле `/etc/zabbix/zabbix_agentd.conf` найти строку `Server=127.0.0.1` и задать адрес Zabbix сервера, с которого будет осуществляться мониторинг криптошлюза:

```
### Option: Server
...
Server=192.168.1.5
...
```

2. В файле `/etc/zabbix/zabbix_agentd.conf` раскомментировать строку "`#ListenPort=10050`" для того, чтобы порт 10050 использовался для мониторинга криптошлюза:

```
### Option: ListenPort
...
ListenPort=10050
...
```

3. Запустить Zabbix агент и добавить его в автозагрузку:

```
root@GW1:~# systemctl start zabbix-agent.service
root@GW1:~# systemctl enable zabbix-agent.service
```

После чего на Zabbix сервере для криптошлюза нужно указать интерфейс мониторинга – `Agent`.

Шаблон, указанный в главе «Использование SNMP шаблона на Zabbix сервере», может быть использован только для SNMP агента. Ключи шаблонов для Zabbix агента и для SNMP агента отличаются. Для работы с Zabbix агентом можно воспользоваться предустановленными на Zabbix сервере шаблонами или же создать свой.

Определение загруженности подсистемы шифрования

Для просмотра счетчиков IPsec драйвера можно воспользоваться утилитой `kstat_show`. Выполните команду:

```
root@GW1:~# kstat_show -e
```

Если значение хоть одного из перечисленных счетчиков `skb allocation errors`, `send queue overflows`, `high priority packets dropped`, `low priority packets dropped` в выводе утилиты `kstat_show` увеличивается, то это свидетельствует о перегруженности подсистемы шифрования «С-Терра Шлюз».

Для определения количества пропускаемого трафика на интерфейсе «С-Терра Шлюз» воспользуйтесь утилитой `vnstat` (например командой `vnstat -l -i eth0`).

Отправка почтовых уведомлений

Для отправки электронных писем и уведомлений можно воспользоваться утилитой `msmtp`. Для отправки почтовых сообщений следует указать SMTP сервер, который их будет перенаправлять. Настройки для подключения к такому серверу следует указать в файле конфигураций `$HOME/.msmtprc` (где `$HOME` – домашняя директория пользователя которым будет отправляться почта). Если такой файл отсутствует, создайте его и введите требуемые настройки. Подробнее о настройках `msmtp` можно узнать здесь: https://marlam.de/msmtp/documentation/.

Пример

Создадим файл `/root/.msmtprc` и добавим в него нужные настройки.

Ниже указаны настройки для SMTP сервера `yandex.ru`:

Оптимальная работа при данных настройках не гарантируется. Отсылаемые письма могут быть заблокированы `yandex.ru` как спам. Для отправления писем рекомендуется использовать собственный SMTP сервер.

```
# Default values
defaults
auth on
port 587
tls on
tls_certcheck off

# Yandex account
account yandex
host smtp.yandex.ru
from <имя пользователя>@yandex.ru
user <имя_пользователя>@yandex.ru
password <пароль_пользователя>

# What account will be used if 'msmtp' is used without '-a' option
account default : yandex
```

Где:

- `defaults` – указывает на то, что дальше будут заданы значения по умолчанию;
- `auth on` – указывает на то, что метод аутентификации будет выбран автоматически;
- `port 587` – говорит о том, что для соединения с SMTP сервером будет использован указанный порт (587);
- `tls on` – соединения будут защищаться при помощи TLS;
- `tls_certcheck off` – отключает проверку сертификата сервера;
- `account yandex` – указывает на то, что дальше будут заданы значения для аккаунта с названием `yandex` (можно указать любое название);
- `host smtp.yandex.ru` – указывает на то, что письма связанные с аккаунтом будут отсылаться на SMTP сервер с указанным названием (smtp.yandex.ru);
- `from <имя_пользователя>@yandex.ru` – устанавливает обратный адрес (на этот адрес будут отправляться уведомления в случае если письмо не удалось доставить);
- `user <имя_пользователя>@yandex.ru` – задает имя аутентифицируемого пользователя;
- `password <пароль_пользователя>` – здесь требуется указать пароль пользователя;
- `account default : yandex` – указывает на то, что по умолчанию утилитой `msmtp` будут использоваться настройки, указанные в account Yandex.

После указания настроек можно отправлять письма. Для этого можно использовать файлы следующего содержания:

```
To: <адрес_получателя>
From: <имя пользователя>@yandex.ru
```

```
Subject: <тема_письма>

[текст_письма]
```

Дополнительный перенос строки после `Subject` не является опечаткой. При отсутствии переноса текст письма будет утерян.

Отправить письмо можно следующей командой:

```
root@GW1:~# cat <имя_файла> | msmtp <адрес_получателя>
```

Получение данных о температуре процессора

Получить информацию о температуре процессора можно с использованием пакета `lm-sensors`, либо из подкаталога `/sys`:

- С использованием `lm-sensors`:

```
root@GW1:~# sensors
```

```
coretemp-isa-0000
Adapter: ISA adapter
Core 0:       +29.0 C  (high = +98.0 C, crit = +98.0 C)
Core 1:       +29.0 C  (high = +98.0 C, crit = +98.0 C)

nct6776-isa-0a10
Adapter: ISA adapter
Vcore:        +0.83 V  (min =  +0.00 V, max =  +1.74 V)
in1:          +1.50 V  (min =  +0.00 V, max =  +0.00 V)  ALARM
AVCC:         +3.33 V  (min =  +2.98 V, max =  +3.63 V)
+3.3V:        +3.31 V  (min =  +2.98 V, max =  +3.63 V)
in4:          +1.00 V  (min =  +0.00 V, max =  +0.00 V)  ALARM
in5:          +1.02 V  (min =  +0.00 V, max =  +0.00 V)  ALARM
3VSB:         +3.31 V  (min =  +2.98 V, max =  +3.63 V)
Vbat:         +3.23 V  (min =  +2.70 V, max =  +3.63 V)
fan1:           0 RPM  (min =    0 RPM)  ALARM
fan2:        7627 RPM  (min =    0 RPM)  ALARM
fan3:           0 RPM  (min =    0 RPM)  ALARM
SYSTIN:       +28.0 C  (high =  +0.0 C, hyst =  +0.0 C)  ALARM  sensor = CPU diode
CPUTIN:       +51.0 C  (high = +80.0 C, hyst = +75.0 C)  sensor = CPU diode
AUXTIN:       +26.0 C  (high = +80.0 C, hyst = +75.0 C)  sensor = thermistor
cpu0_vid:    +1.708 V
intrusion0:  ALARM
intrusion1:  ALARM
```

Примечание – В случае, если команда `sensors` не показывает требуемой информации, следует единожды выполнить команду `sensors-detect`. Она запустит поиск сенсоров. На все вопросы следует ответить положительно.

- Из подкаталога `/sys` можно получить информацию в следующем виде:

```
root@GW1:~# cat /sys/bus/platform/devices/coretemp.0/temp*_input
```

```
29000
29000
```

В зависимости от аппаратной платформы путь может немного отличаться, например:

```
/sys/bus/platform/devices/coretemp.0/hwmon/hwmon1/temp*_input
/sys/bus/platform/devices/coretemp.0/hwmon/hwmon2/temp*_input
```

Здесь указана температура процессоров в миллиградусах. Для форматирования вывода можно воспользоваться следующей командой:

```
root@GW1:~# paste <(ls /sys/devices/platform/coretemp.0/temp*_label | sort -V | xargs
cat) <(ls /sys/devices/platform/coretemp.0/temp*_input | sort -V | xargs cat) |column
-s $'\t' -t | sed 's/\(.\)..$/.\1 C/'
```

```
Core 0  29.0 C
Core 1  31.0 C
```

Получение информации о системе при помощи утилиты ipmi-sensors

На поддерживаемых платформах можно получить различную информацию о состоянии системы, включая информацию о температуре процессора и памяти, статус блоков питания и прочее:

```
root@GW1:~# ipmi-sensors
```

ID	Name	Type	Reading	Units	Event
1	3VSB	Voltage	3.34	V	'OK'
2	5VSB	Voltage	4.86	V	'OK'
3	CPU1_VCORE	Voltage	1.78	V	'OK'
4	CPU2_VCORE	Voltage	1.78	V	'OK'
5	VCCM ABC	Voltage	1.20	V	'OK'
6	VCCM DEF	Voltage	1.20	V	'OK'
7	VCCM GHI	Voltage	1.20	V	'OK'
8	VCCM JKL	Voltage	1.20	V	'OK'
9	1.05_PCH	Voltage	1.03	V	'OK'
10	1.80_PCH	Voltage	1.79	V	'OK'
11	BAT	Voltage	3.06	V	'OK'
12	3V	Voltage	3.30	V	'OK'
13	5V	Voltage	4.89	V	'OK'
14	12V	Voltage	12.00	V	'OK'
16	ChassisIntr	Physical Security	N/A	N/A	'OK'
17	CPU2_PROCHOT	Processor	N/A	N/A	'OK'
18	CPU1_PROCHOT	Processor	N/A	N/A	'OK'
20	PCH Temp	Temperature	35.00	C	'OK'
21	CPU1_THERMTRIP	Processor	N/A	N/A	'OK'
22	CPU2_THERMTRIP	Processor	N/A	N/A	'OK'
39	CPU2_FAN1	Fan	2700.00	RPM	'OK'
40	CPU1_FAN1	Fan	2700.00	RPM	'OK'
41	FRNT_FAN4	Fan	3400.00	RPM	'OK'
42	FRNT_FAN1	Fan	3400.00	RPM	'OK'
43	FRNT_FAN2	Fan	3400.00	RPM	'OK'
44	FRNT_FAN3	Fan	3400.00	RPM	'OK'
45	CPU1 Temp	Temperature	33.00	C	'OK'
46	CPU2 Temp	Temperature	32.00	C	'OK'
47	DDR4_A Temp	Temperature	28.00	C	'OK'
48	DDR4_B Temp	Temperature	29.00	C	'OK'
49	DDR4_C Temp	Temperature	N/A	C	N/A
50	DDR4_D Temp	Temperature	N/A	C	N/A
51	DDR4_E Temp	Temperature	N/A	C	N/A
52	DDR4_F Temp	Temperature	N/A	C	N/A
53	DDR4_G Temp	Temperature	24.00	C	'OK'
54	DDR4_H Temp	Temperature	24.00	C	'OK'
55	DDR4_I Temp	Temperature	N/A	C	N/A
56	DDR4_J Temp	Temperature	N/A	C	N/A
57	DDR4_K Temp	Temperature	N/A	C	N/A
58	DDR4_L Temp	Temperature	N/A	C	N/A
59	PSU1 Status	Power Supply	N/A	N/A	'Presence detected'
60	PSU2 Status	Power Supply	N/A	N/A	'Presence detected'
61	FRNT_FAN1_2	Fan	N/A	RPM	N/A
62	FRNT_FAN2_2	Fan	N/A	RPM	N/A
63	FRNT_FAN3_2	Fan	N/A	RPM	N/A
64	FRNT_FAN4_2	Fan	N/A	RPM	N/A
65	Card Side Temp	Temperature	25.00	C	'OK'
66	MB Temp	Temperature	20.00	C	'OK'
67	PSU2 AC lost	Power Supply	N/A	N/A	'OK'
68	PSU1 AC lost	Power Supply	N/A	N/A	'OK'
70	CPU_CATERR	Processor	N/A	N/A	'OK'
71	CORE_PCH	Voltage	1.01	V	'OK'
72	PSU1 IOUT	Current	5.70	A	'OK'
73	PSU2 IOUT	Current	6.00	A	'OK'
74	PSU1 VIN	Voltage	238.00	V	'OK'
75	PSU2 VIN	Voltage	238.00	V	'OK'
76	PSU1 PIN	Power Supply	76.00	W	'OK'
77	PSU2 PIN	Power Supply	79.00	W	'OK'
78	PSU1 POUT	Power Supply	56.00	W	'OK'

```
79 | PSU2 POUT      | Power Supply      | 64.00      | W     | 'OK'
81 | PSU1 Temp      | Temperature       | 20.00      | C     | 'OK'
82 | PSU2 Temp      | Temperature       | 20.00      | C     | 'OK'
```

Получение информации о состоянии RAID на платформах Aquarius

На аппаратных платформах Aquarius, в которых дисковые накопители находятся в отказоустойчивом RAID массиве есть возможность посмотреть информацию о состоянии накопителей и RAID контроллера с использованием утилиты `MegaCli64`. Данная утилита находится в директории `/opt/MegaRAID/MegaCli` на шлюзе.

Для просмотра информации о физических дисковых накопителях, входящих в RAID, можно воспользоваться следующей командой:

```
root@GW1:~# /opt/MegaRAID/MegaCli/MegaCli64 -PDList -aALL
```

```
Adapter #0

Enclosure Device ID: 252
Slot Number: 0
Drive's position: DiskGroup: 0, Span: 0, Arm: 0
Enclosure position: N/A
Device Id: 14
WWN: 5002538f0271ed5d
Sequence Number: 2
Media Error Count: 0
Other Error Count: 0
Predictive Failure Count: 0
Last Predictive Failure Event Seq Number: 0
PD Type: SATA

Raw Size: 223.570 GB [0x1bf244b0 Sectors]
Non Coerced Size: 223.070 GB [0x1be244b0 Sectors]
Coerced Size: 223.062 GB [0x1be20000 Sectors]
Sector Size:  512
Logical Sector Size:  512
Physical Sector Size:  4096
Firmware state: Online, Spun Up
Commissioned Spare : No
Emergency Spare : No
Device Firmware Level: 304Q
Shield Counter: 0
Successful diagnostics completion on :  N/A
SAS Address(0): 0x4433221100000000
Connected Port Number: 0(path0)
Inquiry Data: S663NE0T705199      SAMSUNG MZ7L3240HCHQ-00A07              JXTC304Q
FDE Capable: Not Capable
FDE Enable: Disable
Secured: Unsecured
Locked: Unlocked
Needs EKM Attention: No
Foreign State: None
Device Speed: 6.0Gb/s
Link Speed: 6.0Gb/s
Media Type: Solid State Device
Drive:  Not Certified
Drive Temperature :28C (82.40 F)
PI Eligibility:  No
Drive is formatted for PI information:  No
PI: No PI
Drive's NCQ setting : Enabled
Port-0 :
Port status: Active
```

```
Port's Linkspeed: 6.0Gb/s
Drive has flagged a S.M.A.R.T alert : No



Enclosure Device ID: 252
Slot Number: 1
Drive's position: DiskGroup: 0, Span: 0, Arm: 1
Enclosure position: N/A
Device Id: 15
WWN: 5002538f0271ed5b
Sequence Number: 2
Media Error Count: 0
Other Error Count: 0
Predictive Failure Count: 0
Last Predictive Failure Event Seq Number: 0
PD Type: SATA

Raw Size: 223.570 GB [0x1bf244b0 Sectors]
Non Coerced Size: 223.070 GB [0x1be244b0 Sectors]
Coerced Size: 223.062 GB [0x1be20000 Sectors]
Sector Size:  512
Logical Sector Size:  512
Physical Sector Size:  4096
Firmware state: Online, Spun Up
Commissioned Spare : No
Emergency Spare : No
Device Firmware Level: 304Q
Shield Counter: 0
Successful diagnostics completion on :  N/A
SAS Address(0): 0x4433221101000000
Connected Port Number: 1(path0)
Inquiry Data: S663NE0T705197      SAMSUNG MZ7L3240HCHQ-00A07              JXTC304Q
FDE Capable: Not Capable
FDE Enable: Disable
Secured: Unsecured
Locked: Unlocked
Needs EKM Attention: No
Foreign State: None
Device Speed: 6.0Gb/s
Link Speed: 6.0Gb/s
Media Type: Solid State Device
Drive:  Not Certified
Drive Temperature :28C (82.40 F)
PI Eligibility:  No
Drive is formatted for PI information:  No
PI: No PI
Drive's NCQ setting : Enabled
Port-0 :
Port status: Active
Port's Linkspeed: 6.0Gb/s
Drive has flagged a S.M.A.R.T alert : No

Exit Code: 0x00
```

Для просмотра информации о виртуальном дисковом RAID накопителе можно воспользоваться следующей командой:

```
root@GW1:~# /opt/MegaRAID/MegaCli/MegaCli64 -LDInfo -LALL -aALL
```

```
Adapter 0 -- Virtual Drive Information:
Virtual Drive: 0 (Target Id: 0)
Name                :vd1
```

```
RAID Level          : Primary-1, Secondary-0, RAID Level Qualifier-0
Size                : 223.062 GB
Sector Size         : 512
Is VD emulated      : Yes
Mirror Data         : 223.062 GB
State               : Optimal
Strip Size          : 256 KB
Number Of Drives    : 2
Span Depth          : 1
Default Cache Policy: WriteBack, ReadAhead, Direct, No Write Cache if Bad BBU
Current Cache Policy: WriteThrough, ReadAhead, Direct, No Write Cache if Bad BBU
Default Access Policy: Read/Write
Current Access Policy: Read/Write
Disk Cache Policy   : Disk's Default
Encryption Type     : None
Bad Blocks Exist: No
PI type: No PI

Is VD Cached: No

Exit Code: 0x00
```

Для просмотра полной информации о RAID контроллере, его параметрах и других свойствах, можно воспользоваться следующей командой:

```
root@GW1:~# /opt/MegaRAID/MegaCli/MegaCli64 -AdpAllinfo -aAll
```

Для выгрузки лога (журнала) RAID контроллера в файл, который в случае необходимости можно использовать для диагностики работоспособности, воспользуйтесь следующей командой:

```
root@GW1:~# /opt/MegaRAID/MegaCli/MegaCli64 -AdpAlILog -aAll > log.txt
```

Получение информации о состоянии RAID на платформах STK

На аппаратных платформах STK, в которых дисковые накопители находятся в отказоустойчивом RAID массиве есть возможность посмотреть информацию о состоянии накопителей и RAID контроллера с использованием утилиты `ExSS_64`. Данная утилита находится в директории `/opt/` на шлюзе. Утилита является интерактивной и при вызове запускает свою собственную консоль для взаимодействия с RAID контроллером.

Для просмотра состояния RAID массива запустите утилиту `ExSS_64` и в ее консоли выполните команду `DC C0`:

```
root@sterragate:~# /opt/ExSS_64
```

```
 ------------------------------------------------------------------------------
-
 | ExSS RAID Card Utility v1.0.5                                               |
 | (C) 2005-2018 Innodisk Corp. Command Line Interface Tool For ExSS Series Product.
|
 ------------------------------------------------------------------------------
-
```

```
ExSS>DC C0
```

```
Controller[0]
-- ChipID       = 10
-- SerialNumber = 427491329
-- SuperUserPwd = ÿÿÿÿÿÿÿÿ
-- Sata[0]
------ ModelName    = FOXLINE FLSSD256M42T6
------ SerialNumber = 202411010128
------ FirmwareVer  = SBFM61.5
------ Capacity     = 238 GB
------ PortType     = RAID Disk
```

```
------ PortSpeed    = (null)
------ Page0State   = Hooked to PM
------ Page0RaidIdx = 0
------ Page0MbrIdx  = 1
-- Sata[1]
------ ModelName    = FOXLINE FLSSD256M42T6
------ SerialNumber = 202411010123
------ FirmwareVer  = SBFM61.5
------ Capacity     = 238 GB
------ PortType     = RAID Disk
------ PortSpeed    = (null)
------ Page0State   = Hooked to PM
------ Page0RaidIdx = 0
------ Page0MbrIdx  = 0
-- Raid[0]
------ ModelName       = H/W RAID1
------ SerialNumber    = PFAH0FHX26N3QN93EJM7
------ Capacity        = 238 GB
------ RaidStatus      = Normal
------ RaidLevel       = RAID 1
------ RaidMbrCount    = 2
------ RebuildPriority = 3
------ RaidMember[0]
--------- Ready        = 1
--------- LBA48Support = 1
--------- SataPage     = 0
--------- SataPort     = 1
--------- SataBase     = 0
--------- SataSize     = 238 GB
------ RaidMember[1]
--------- Ready        = 1
--------- LBA48Support = 1
--------- SataPage     = 0
--------- SataPort     = 0
--------- SataBase     = 0
--------- SataSize     = 238 GB
```

Получение информации о накопителях с помощью smartctl

Кроме информации, доступной по SNMP, можно получить информацию о состоянии жестких дисков при помощи утилиты `smartctl`. Данная утилита использует технологию SMART (Self Monitoring Analysing and Reporting Technology) для получения информации о различного рода проблемах и повреждениях (проблемы блока магнитных головок, проблемы механики, температура и др.)

Для определения дисков в системе можно использовать утилиту с опцией `--scan`:

```
root@GW1:~# smartctl --scan
```

```
/dev/sda -d scsi # /dev/sda, SCSI device
```

Для вывода информации по диску можно использовать утилиту с опцией `-a`:

```
root@GW1:~# smartctl -a /dev/sda
```

```
smartctl 6.6 2016-05-31 r4324 [x86_64-linux-4.9.0-11-amd64] (local build)
Copyright (C) 2002-16, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:     Innodisk 3IE3/3ME3 SSDs
Device Model:     mSATA mini 3ME3
Serial Number:    B2A11807170060805
LU WWN Device Id: 5 24693f 22a221959
Firmware Version: S16425
User Capacity:    16,013,942,784 bytes [16.0 GB]
Sector Size:      512 bytes logical/physical
Rotation Rate:    Solid State Device
```

```
Form Factor:      2.5 inches
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   ATA8-ACS (minor revision not indicated)
SATA Version is:  SATA 3.0, 6.0 Gb/s (current: 3.0 Gb/s)
Local Time is:    Thu Nov  3 23:02:14 2023 MSK
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x00)     Offline data collection activity
                                    was never started.
                                    Auto Offline Data Collection: Disabled.
Total time to complete Offline
data collection:              (   32) seconds.
Offline data collection
capabilities:                  (0x00)       Offline data collection not supported.
SMART capabilities:            (0x0003)     Saves SMART data before entering
                                    power-saving mode.
                                    Supports SMART auto save timer.
Error logging capability:        (0x00)     Error logging NOT supported.
                                    No General Purpose Logging support.
SCT capabilities:             (0x0039)       SCT Status supported.
                                    SCT Error Recovery Control supported.
                                    SCT Feature Control supported.
                                    SCT Data Table supported.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x0000   000   000   000    Old_age   Offline      -       0
  2 Throughput_Performance  0x0000   000   000   000    Old_age   Offline      -       0
  3 Spin_Up_Time            0x0000   000   000   000    Old_age   Offline      -       0
  5 Later_Bad_Block         0x0013   100   100   001    Pre-fail  Always       -       0
  7 Seek_Error_Rate         0x0000   000   000   000    Old_age   Offline      -       0
  8 Seek_Time_Performance   0x0000   000   000   000    Old_age   Offline      -       0
  9 Power_On_Hours          0x0002   119   000   000    Old_age   Always       -       1143
 10 Spin_Retry_Count        0x0000   000   000   000    Old_age   Offline      -       0
 12 Power_Cycle_Count       0x0002   115   000   000    Old_age   Always       -       371
163 Total_Bad_Block_Count   0x0000   000   000   000    Old_age   Offline      -       11
168 SATA_PHY_Error_Count    0x0000   000   000   000    Old_age   Offline      -       0
169 Remaining_Lifetime_Perc 0x0000   099   000   000    Old_age   Offline      -       99
175 Bad_Cluster_Table_Count 0x0000   000   000   000    Old_age   Offline      -       0
192 Power-Off_Retract_Count 0x0000   000   000   000    Old_age   Offline      -       0
194 Temperature_Celsius     0x0000   030   100   000    Old_age   Offline      -       30 (2 100
0 0 0)
197 Current_Pending_Sector  0x0012   000   100   000    Old_age   Always       -       0
225 Data_Log_Write_Count    0x0000   000   032   000    Old_age   Offline      -       569317
240 Write_Head              0x0000   000   000   000    Old_age   Offline      -       0
165 Max_Erase_Count         0x0002   025   001   000    Old_age   Always       -       25
167 Average_Erase_Count     0x0002   010   001   000    Old_age   Always       -       10
170 Spare_Block_Count       0x0003   100   001   000    Pre-fail  Always       -       143
171 Program_Fail_Count      0x0002   000   001   000    Old_age   Always       -       0
172 Erase_Fail_Count        0x0002   000   001   000    Old_age   Always       -       0
176 RANGE_RECORD_Count      0x0000   100   001   000    Old_age   Offline      -       0
187 Reported_Uncorrect      0x0002   000   001   000    Old_age   Always       -       0
229 Flash_ID              0x0002   100   001   000    Old_age   Always       -       0x517693943a98
232 Spares_Remaining_Perc   0x0003   100   001   000    Pre-fail  Always       -       0
235 Later_Bad_Blk_Inf_R/W/E 0x0002   000   000   000    Old_age   Always       -       0 0 0
241 Host_Writes_32MiB       0x0002   100   001   000    Old_age   Always       -       2223
242 Host_Reads_32MiB        0x0002   100   001   000    Old_age   Always       -       24116

SMART Error Log not supported

SMART Self-test Log not supported

Selective Self-tests/Logging not supported
```

Если по какой-то причине технология SMART на диске отключена, ее можно включить при помощи опции `-s`:

```
root@GW1:~# smartctl -s on /dev/sda
```

```
smartctl 6.6 2016-05-31 r4324 [x86_64-linux-4.9.0-11-amd64] (local build)
Copyright (C) 2002-16, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF ENABLE/DISABLE COMMANDS SECTION ===
SMART Enabled.
```

Получение информации о накопителях при использовании RAID

Если в системе дисковые накопители находятся в RAID, то для получения информации по каждому отдельному диску сначала следует определить номера дисков в выводе команды:

```
root@GW1:~# smartctl --scan
```

```
/dev/sda -d scsi # /dev/sda, SCSI device
/dev/bus/0 -d megaraid,5 # /dev/bus/0 [megaraid_disk_05], SCSI device
/dev/bus/0 -d megaraid,6 # /dev/bus/0 [megaraid_disk_06], SCSI device
```

А затем выполнить следующую команду для получения статистики конкретного диска, например:

```
root@GW1:~# smartctl -a -d megaraid,6 /dev/sda
```

```
smartctl 7.3 2022-02-28 r5338 [x86_64-linux-6.1.0-13st1-amd64] (local build)
Copyright (C) 2002-22, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:     Seagate Nytro SATA SSD
Device Model:     XA480LE10063
Serial Number:    HKQ0346A
LU WWN Device Id: 5 000c50 03ea3dc5c
Firmware Version: SF42147L
User Capacity:    480,103,981,056 bytes [480 GB]
Sector Sizes:     512 bytes logical, 4096 bytes physical
Rotation Rate:    Solid State Device
TRIM Command:     Available, deterministic, zeroed
Device is:        In smartctl database 7.3/5319
ATA Version is:   ACS-4, ACS-2 T13/2015-D revision 3
SATA Version is:  SATA 3.3, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Tue Apr  2 16:16:50 2024 MSK
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART Status not supported: ATA return descriptor not supported by controller firmware
SMART overall-health self-assessment test result: PASSED
Warning: This result is based on an Attribute check.

General SMART Values:
Offline data collection status:  (0x00)     Offline data collection activity
                                    was never started.
                                    Auto Offline Data Collection: Disabled.
Self-test execution status:      (   0)     The previous self-test routine completed
                                    without error or no self-test has ever
                                    been run.
Total time to complete Offline
data collection:             (    0) seconds.
Offline data collection
capabilities:                 (0x59) SMART execute Offline immediate.
                                    No Auto Offline data collection support.
                                    Suspend Offline collection upon new
                                    command.
                                    Offline surface scan supported.
                                    Self-test supported.
                                    No Conveyance Self-test supported.
                                    Selective Self-test supported.
SMART capabilities:           (0x0002)     Does not save SMART data before
```

```
                                    entering power-saving mode.
                                    Supports SMART auto save timer.
Error logging capability:        (0x01)     Error logging supported.
                                    General Purpose Logging supported.
Short self-test routine
recommended polling time:      (   1) minutes.
Extended self-test routine
recommended polling time:      (  30) minutes.
SCT capabilities:              (0x103d)       SCT Status supported.
                                    SCT Error Recovery Control supported.
                                    SCT Feature Control supported.
                                    SCT Data Table supported.

SMART Attributes Data Structure revision number: 11
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x002f   100   100   090    Pre-fail  Always       -       5
  5 Reallocated_Sector_Ct   0x0032   100   100   000    Old_age   Always       -       0
  9 Power_On_Hours          0x0032   100   100   000    Old_age   Always       -       6670
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -       202
100 Flash_GB_Erased         0x0032   100   100   000    Old_age   Always       -       26
102 Lifetime_PS4_Entry_Ct   0x0032   100   100   000    Old_age   Always       -       821
103 Lifetime_PS3_Exit_Ct    0x0032   100   100   000    Old_age   Always       -       0
170 Grown_Bad_Block_Ct      0x0032   100   100   000    Old_age   Always       -       0
171 Program_Fail_Count      0x0032   100   100   000    Old_age   Always       -       0
172 Erase_Fail_Count        0x0032   100   100   000    Old_age   Always       -       0
173 Avg_Program/Erase_Ct    0x0032   100   100   000    Old_age   Always       -       14
174 Unexpected_Pwr_Loss_Ct  0x0032   100   100   000    Old_age   Always       -       154
177 Wear_Range_Delta        0x0023   100   100   089    Pre-fail  Always       -       0 0 0
183 SATA_Downshift_Count    0x0032   100   100   000    Old_age   Always       -       0x000000000000000
187 Uncorrectable_ECC_Ct    0x0032   100   100   000    Old_age   Always       -       0
194 Temperature_Celsius     0x0022   022   037   000    Old_age   Always       -       22 (Min/Max
17/37)
195 RAISE_ECC_Cor_Ct        0x0032   100   100   000    Old_age   Always       -       0
198 Uncor_Read_Error_Ct     0x0032   100   100   000    Old_age   Always       -       0
199 UDMA_CRC_Error_Count    0x0032   100   100   000    Old_age   Always       -       0
230 Drv_Life_Protect_Status 0x0023   100   100   091    Pre-fail  Always       -       100
231 SSD_Life_Left           0x0033   100   100   010    Pre-fail  Always       -       0x001b4a00000000
232 Available_Reservd_Space 0x0027   100   100   003    Pre-fail  Always       -       0
233 Lifetime_Wts_To_Flsh_GB 0x0032   100   100   000    Old_age   Always       -       96
241 Lifetime_Wts_Frm_Hst_GB 0x0032   100   100   000    Old_age   Always       -       298
242 Lifetime_Rds_Frm_Hst_GB 0x0032   100   100   000    Old_age   Always       -       18124
243 Free_Space              0x0026   100   098   003    Old_age   Always       -       0x0726f80007e0b7
```

Получение информации о состоянии блоков питания

Получение информации о состоянии блоков питания на Lanner FW-8894

На платформе Lanner FW-8894 для получения информации о состоянии блоков питания следует использовать утилиту `pwr_tst`.

Перед вызовом утилиты следует выполнить загрузку модуля `pwr_drv` следующей командой:

```
root@GW1:~# modprobe pwr_drv
```

Примеры использования:

- Оба БП подключены к питанию:

```
root@GW1:~# pwr_tst
```

```
=== Lanner platform miscellaneous utility ===
MB-8896 Redundant Power Supply V1.1 2019-02-19

Power supply 1/2 ....Normal state
```

- 1 БП подключен, 2 не подключен:

```
root@GW1:~# pwr_tst
```

```
=== Lanner platform miscellaneous utility ===
MB-8896 Redundant Power Supply V1.1 2019-02-19

Power supply 1 ....Normal state
Power supply 2 ....Wrong state
```

Получение информации о состоянии блоков питания на Lanner NCA-5210C

На платформе Lanner NCA-5210C для получения информации о состоянии блоков питания следует использовать утилиту `pms`.

Примеры использования:

- Оба БП подключены к питанию:

```
root@GW1:~# pms
```

```
************************************************************
** Lanner Electronics Inc.  2019-07-05
** PMBUS utility  Ver, 1.0 for NCB-5210
************************************************************
<<<<< Power Information: >>>>>
Power Module 1 Power Mode:AC Power Mode
Power Module 1 SN:SA180J811724003455
Power Module 2 Power Mode:AC Power Mode
Power Module 2 SN:SA180J811724003444

================= Power Module 1  =================
Power Module Status:
12V: 12.047             5Vsb: 5.234
Fan Speed: 12400        Temp1: 38       Temp2: 38

================= Power Module 2  =================
Power Module Status:
12V: 12.078             5Vsb: 5.234
Fan Speed: 5800         Temp1: 37       Temp2: 40
```

- 1 БП подключен, 2 не подключен:

```
root@GW1:~# pms
```

```
************************************************************
** Lanner Electronics Inc.  2019-07-05
** PMBUS utility  Ver, 1.0 for NCB-5210
************************************************************
<<<<< Power Information: >>>>>
Power Module 1 Power Mode:AC Power Mode
Power Module 1 SN:SA180J811724003455
Power Module 2 Power Mode:AC Power Mode
Power Module 2 SN:SA180J811724003444

================= Power Module 1  =================
Power Module Status:
12V: 12.047             5Vsb: 5.234
Fan Speed: 5096         Temp1: 37       Temp2: 38

================= Power Module 2  =================
Power Module Status:
```

```
Unit not providing power to the output.
POWER_GOOD# present.
12V: 2.719              5Vsb: 2.797
Fan Speed: 0    Temp1: 40       Temp2: 43

================================================
PMBUS Value fail
```

Настройка LLDP

Также, можно получить информацию о соседних устройствах при помощи протокола LLDP. Информация об устройстве, которая может передаваться с помощью LLDP:

- Имя устройства (System Name),
- Описание устройства (System Description),
- Идентификатор порта (Port ID),
- Описание порта (Port Description),
- Возможности устройства (System Capabilities),
- Управляющий адрес (Management Address).

Запустите сервис `lldpd` (по умолчанию, он выключен):

```
root@GW1:~# systemctl start lldpd.service
```

После запуска проверьте список доступных устройств:

```
root@GW1:~# lldpctl
```

```
-------------------------------------------------------------------------------
LLDP neighbors:
-------------------------------------------------------------------------------
Interface:    eth0, via: LLDP, RID: 1, Time: 0 day, 00:00:07
  Chassis:
    ChassisID:    mac 08:00:27:17:ad:b4
    SysName:      sterragate
    SysDescr:     S-Terra_Gate_4.3
    TTL:          120
    MgmtIP:       0.0.0.0
    Capability:   Bridge, off
    Capability:   Router, on
    Capability:   Wlan, off
    Capability:   Station, on
  Port:
    PortID:       mac 08:00:27:17:ad:b4
    PortDescr:    eth0
    PMD autoneg:  supported: yes, enabled: yes
      Adv:          10Base-T, HD: yes, FD: yes
      Adv:          100Base-TX, HD: yes, FD: yes
      Adv:          1000Base-T, HD: no, FD: yes
      MAU oper type: 1000BaseTFD - Four-pair Category 5 UTP, full duplex mode
-------------------------------------------------------------------------------
```

Настройка параметров запуска сервиса `lldpd` находится в файле `/etc/default/lldpd`. По умолчанию, конфигурационный файл выглядит так:

```
# Uncomment to start SNMP subagent and enable CDP, SONMP and EDP protocol
#DAEMON_ARGS="-x -c -s -e"
```

Для изменений параметров запуска `lldpd` следует раскомментировать переменную `DAEMON_ARGS` и отредактировать ее значение. Некоторые из поддерживаемых параметров (полный список параметров можно посмотреть в выводе команды `lldpd --help`):

- `-I <список интерфейсов>` – позволяет явно указать интерфейсы, на которых будет работать LLDP. Например, `-I eth0,eth1` включает LLDP только на интерфейсах eth0 и eth1.

- `-x` – включает SNMP agentx, позволяя получать собранную LLDP информацию по SNMP (требуется дополнительная настройка snmpd).
- `-m <IP адрес>` – анонсирует заданный management IP адрес устройства.
- `-S` <описание> – анонсирует заданное описание устройства.
- `-c` – дополнительно включает протокол CDP (Cisco).
- `-e` – дополнительно включает протокол EDP (Extreme Networks).
- `-f` – дополнительно включает протокол FDP (Foundry).
- `-s` – дополнительно включает протокол SONMP (Nortel).

Для того, чтобы можно было по SNMP получить информацию об устройствах, обнаруженных с использованием LLDP, необходимо добавить в конфигурационный файл `/etc/snmp/snmpd.conf` соответствующую ветвь OID и указать сокет для подключения `agentx`:

```
# LLDP OID
view customview included .1.0.8802.1.1.2

agentXSocket /var/agentx/master
```

Затем следует перезапустить службы `snmpd` и `lldpd`:

```
root@GW1:~# systemctl restart snmpd.service
root@GW1:~# systemctl restart lldpd.service
```

Приложение

Конфигурационный файл /etc/snmp/snmpd.conf

По умолчанию файл `/etc/snmp/snmpd.conf` выглядит следующим образом:

```
##############################################################
# Listen for connections from the local system only.
agentaddress  udp:127.0.0.1:161
# Listen for connections on all interfaces.
#agentaddress udp:161

##############################################################################
# ATTENTION: change community and restrict access to SNMP requests #
#            from the specified sources for security reasons.      #
##############################################################################

# Read-only access from anywhere with 'public' community.
rocommunity public default -V customview
# Read-only access from 10.0.0.0/16 with 'public' community.
#rocommunity public 10.0.0.0/16 -V customview

# .iso.org.dod.internet.mgmt.mib-2
view customview included .1.3.6.1.2.1
# .iso.org.dod.internet.private.enterprises.cisco
view customview included .1.3.6.1.4.1.9
# S-TERRA-CSP-MIB.
view customview included .1.3.6.1.4.1.31342
# .iso.org.dod.internet.private.enterprises.Debian.project.keepalived
view customview included .1.3.6.1.4.1.9586.100.5
# .iso.identified-organization.dod.internet.private.enterprise.ucdavis
view customview included .1.3.6.1.4.1.2021
#                                                                .iso.identified-
organization.dod.internet.private.enterprise.netSnmpObjects.nsExtensions
# For using 'extend' directive (NET-SNMP-EXTEND-MIB).
view customview included 1.3.6.1.4.1.8072.1.3

# Uncomment to enable local certificates expiration monitoring
#includeFile /opt/snmp_monitoring/snmp_extend/certs_expiry/certs_expiry.snmpd_conf

# Uncomment to enable statistics from kstat_show
#includeFile /opt/snmp_monitoring/snmp_extend/check_kstat_show/kstat_show.snmpd_conf

# Uncomment to enable PSU monitoring
#includeFile /opt/snmp_monitoring/snmp_extend/get_power_state/power_state.snmpd_conf

# Uncomment to enable rtt/jitter/loss statistics from ICMP script
#includeFile
/opt/snmp_monitoring/snmp_extend/get_rtt_jitter_loss/rtt_jitter_loss.snmpd_conf

# Uncomment for getting IPSM/DPDK parameters (available only for S-Terra Gate DP)
#includeFile /opt/snmp_monitoring/snmp_extend/ipsm_iface_stats/ipsm_dpdk.snmpd_conf

# Configures  monitoring  of all disks found on the system, using the specified
(percentage) threshold.
includeAllDisks 10%

# Proxy directive to pass part of the OID tree to S-Terra CSP SNMP agent to retrieve
IKE and IPsec stats.
# Example of S-Terra Gate SNMP configuration in Cisco-like console:
#   snmp-server community public RO
```

```
#   snmp-server polling address 127.0.0.1
#   snmp-server polling udp-port 1161
proxy -v 2c -c "public" 127.0.0.1:1161 .1.3.6.1.4.1.9
proxy -v 2c -c "public" 127.0.0.1:1161 .1.3.6.1.2.1.1.1.0
proxy -v 2c -c "public" 127.0.0.1:1161 .1.3.6.1.4.1.31342

# Enable the AgentX functionality and cause the agent to start listening for incoming
AgentX registrations
# (for example used by a Keepalived).
master agentx
##############################################################
```

Где:

- `agentaddress <IP адрес шлюза>:<порт>` – локальный адрес шлюза, на котором SNMPD слушает запросы от систем мониторинга; 161 – локальный порт (стандартный).
- `rocommunity` – задает значение community-строки (в данном случае – `public`) с правами только на чтение, а следующее за ним ключевое слово `default -V <имя>` указывает какие объектные идентификаторы (OID) будут выдаваться. Если нужны права не только на чтение, но и на запись значений OID, замените `rocommunity` на `rwcommunity`.
- `view <имя> included <OID>` – задает значения, объектных идентификаторов, доступные для использования. В качестве `<OID>` в конфигурации указаны поддеревья идентификаторов, т.е. доступными для использования становятся все присутствующие в системе значения, являющиеся дочерними к указанным.
- `includeFile <путь до конфигурационного файла>` – директива, позволяющая включить содержимое заданного конфигурационного файла в конфигурацию `snmpd.conf`.
- `includeAllDisks <число>%` – включает мониторинг дискового пространства. Числом указывается процент оставшегося свободного места на разделе. В случае, если свободного места оказывается меньше указанного, переменная `UCD-SNMP-MIB::dskErrorFlag.<номер>` для соответствующего раздела примет значение «1».
- `proxy -v 2c -c "public" 127.0.0.1:1161 <OID>` – указывает, что запросы к `<OID>` нужно перенаправлять встроенному агенту VPNGATE.

Проверка версии пакета monitoring и обновление

1. Проверьте текущую версию пакета командой:

   ```
   root@GW1:~# dpkg -l monitoring
   ```

   ```
   INFO [dpkg-sxm]: Calling original dpkg: '/usr/share/linux-image-sterra-userspace-
   tools/dpkg -l monitoring'
   Desired=Unknown/Install/Remove/Purge/Hold
   | Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
   |/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
   ||/ Name           Version               Architecture Description
   +++-==============-=====================-============-
   ===============================================
   ii  monitoring     0.11+sterra1-debian12 all          Package contains auxillary
   monitoring stuff.
   ```

   Видно, что установлена версия `0.11+sterra1-debian12`.

2. Скачайте новую версию пакета с портала документации https://doc.s-terra.ru -> в разделе «Типовые сценарии применения» -> Версия 5.0 -> сценарий «Расширение возможностей протоколирования и мониторинга шлюза безопасности «С-Терра Шлюз» -> файл `monitoring.zip` и перенесите на криптошлюз.

3. Распакуйте архив и установите пакет(ы) в зависимости от класса защиты криптошлюза:

   3.1. Для шлюзов КС1 или КС2:

   ```
   root@GW1:~# dpkg -i monitoring_0.12+sterra1-debian12_all.deb
   ```

   3.2. Для шлюза КС3:

```
root@GW1:~#      dpkg      -i      monitoring_0.12+sterra1-debian12_all.deb      sxm-
monitoring_0.12+sterra1-debian12_all.deb
```

4. Проверьте, что пакет обновился командой:

```
root@GW1:~# dpl
```

```
INFO [dpkg-sxm]: Calling original dpkg: '/usr/share/linux-image-sterra-userspace-
tools/dpkg -l monitoring'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version               Architecture Description
+++-==============-=====================-============-
=============================================
ii  monitoring      0.12+sterra1-debian12 all            Package contains auxillary
monitoring stuff.
```

Видно, что версия пакета обновилась до `0.12+sterra1-debian12`.

Примеры SNMP запросов

Получение ARP таблицы по SNMP

Для получения ARP записей с использованием SNMP запроса можно использовать OID RFC1213-MIB::atPhysAddress (1.3.6.1.2.1.3.1.1.2):

```
root@GW1:~# snmpwalk -v 2c -c public <адрес шлюза>:161 RFC1213-MIB::atPhysAddress
```

```
RFC1213-MIB::atPhysAddress.2.1.172.16.100.1 = Hex-STRING: 00 0C 29 22 F4 35
RFC1213-MIB::atPhysAddress.3.1.192.168.100.11 = Hex-STRING: 00 0C 29 9D 26 88
RFC1213-MIB::atPhysAddress.3.1.192.168.100.100 = Hex-STRING: 00 0C 29 0A A3 5B
```

Пример содержимого ARP таблицы на шлюзе:

```
root@GW1:~# ip neigh
```

```
192.168.100.11 dev eth1 lladdr 00:0c:29:9d:26:88 REACHABLE
172.16.100.1 dev eth0 lladdr 00:0c:29:22:f4:35 REACHABLE
192.168.100.100 dev eth1 lladdr 00:0c:29:0a:a3:5b REACHABLE
```