



Программное обеспечение Kornfeld OS

Руководство по эксплуатации



Содержание

Глава 1. Аннотация.....	5
Глава 2. Глоссарий.....	6
Глава 3. Базовая настройка.....	10
Подключение к устройству.....	10
Подключение оборудования.....	10
Получение сетевых настроек	13
Работа с Kornfeld CLI.....	14
Режимы командной строки.....	14
Синтаксис.....	14
Функции управления.....	15
Контекстная подсказка.....	15
Автодополнение команд.....	16
Горячие клавиши командной строки.....	16
Сообщения об ошибках.....	17
Файловая система и работа с файлами.....	18
Файлы конфигурации.....	18
Виды конфигураций.....	19
Сохранение конфигурации.....	20
Применение конфигурации.....	21
Просмотр конфигурации.....	22
Глава 4. Системные настройки.....	23
Настройка баннера.....	23
Настройка системного времени и даты.....	23
Настройка протокола сетевого времени NTP.....	24
Настройка протокола сетевого времени NTP по умолчанию.....	24
Настройка протокола сетевого времени NTP.....	25
Настройка интерфейса источника протокола сетевого времени NTP.....	25
Настройка аутентификации протокола сетевого времени NTP.....	26
Просмотр NTP-ассоциаций.....	27
Отображение настроек протокола сетевого времени NTP.....	28
Глава 5. Управление устройством.....	30
SNMP.....	30
Обзор SNMP.....	30
Настройка SNMP.....	32



Глава 6. Настройка коммутации L2.....	36
VLAN.....	36
Глава 7. Настройка коммутации L3.....	40
Статическая маршрутизация.....	40
Глава 8. Настройка доступа и аутентификации пользователей.....	42
Служба аутентификации, авторизации и учета.....	42
Локальные пользователи.....	42
Управление пользователями.....	43
Настройка AAA.....	45

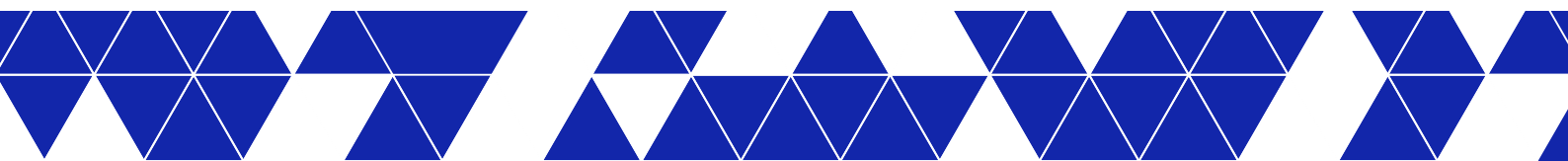


1 Аннотация

В руководстве приведено описание настройки и управления параметрами операционной системы Kornfeld OS.

В данном руководстве представлена информация о настройке и оптимизации сетевой инфраструктуры в соответствии с конкретными потребностями организации, а также подробное описание различных функций и возможностей коммутатора, включая устранение распространенных сетевых проблем и настройку дополнительных параметров.

Данное руководство предназначено для специалистов в области сетевых технологий, которые работают с сетевыми устройствами Kornfeld.



2 Глоссарий

Табл. 1. Глоссарий

Аббревиатура	Термин	Описание
AAA	Authentication, Authorization and Accounting	Аутентификация, авторизация и учет - службы управления доступом пользователей к сетевым ресурсам.
ARP	Address Resolution Protocol	Протокол распознавания адреса - интернет-протокол, используемый для сопоставления пар IP- и MAC-адресов. Описан в документе RFC 826.
BFD	Bidirectional Forwarding Detection	Обнаружение двунаправленной передачи - сетевой протокол, используемый для обнаружения неисправностей между узлами сети.
BGP	Border Gateway Protocol	Пограничный межсетевой протокол - протокол, лежащий в основе глобальной системы маршрутизации Интернета.
Broadcast	Broadcast	Широковещательная передача - передача данных всем участникам сети.
CLI	Command Line Interface	Интерфейс командной строки - текстовый интерфейс для взаимодействия с компьютерной системой.
DHCP	Dynamic Host Configuration Protocol	Протокол динамической конфигурации сетевого узла - сетевой протокол, который обеспечивает автоматическое назначение устройствам в сети IP-адресов.
DNS	Domain name server	Сервер имен доменов - иерархическая и распределенная система, которая позволяет получать информацию о символьных именах.
ECMP	Equal-cost multi-path routing	Мультитракт с каналами равной стоимости - технология маршрутизации, которая позволяет распределять трафик до получателя несколькими путями, определяемыми по метрике.
IP	Internet Protocol	Интернет-протокол - стандартный сетевой протокол в рамках TCP/IP, который используется для передачи данных в компьютерных сетях, включая сеть Интернет. Он представляет собой набор правил и стандартов, определяющих формат данных и их адресацию для обмена информацией между устройствами в сети.

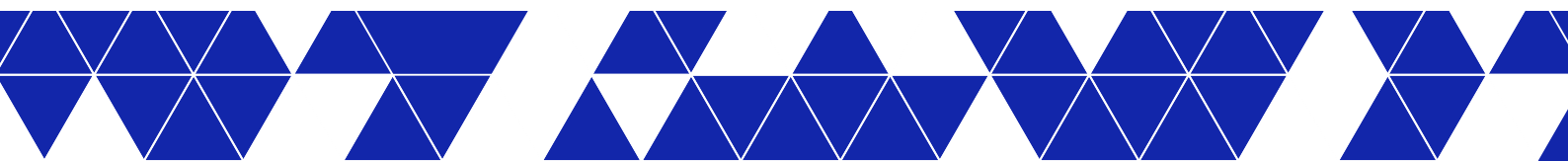


Табл. 1. Глоссарий

Аббревиатура	Термин	Описание
IP address	IP address	IP-адрес - уникальный числовой идентификатор, который присваивается каждому устройству в компьютерных сетях, использующих протокол IP. Состоит из трех основных компонентов: номер сети, дополнительный номер подсети и номер хоста. Номера сети и подсети необходимы для правильной маршрутизации данных, в то время как номер хоста позволяет идентифицировать каждый конкретный хост внутри сети или подсети.
IPv4	Internet Protocol version 4	IPv4 - четвертая версия Интернет-протокола, используемая для идентификации и адресации устройств в компьютерных сетях. IPv4-адрес представляет собой 32-битное число и состоит из четырех десятичных чисел, разделенных точками.
IPv6	Internet Protocol version 6	IPv6 - шестая версия Интернет-протокола, используемая для идентификации и адресации устройств в компьютерных сетях. В отличие от предыдущей версии IPv4, которая использует 32-битные адреса, IPv6 использует 128-битные адреса. IPv6-адрес представляется в виде шестнадцатиречных чисел, разделенных двоеточиями.
ICMP	Internet Control Message Protocol	Протокол управляющих сообщений в Интернете - вспомогательный протокол в составе протоколов Интернета. Используется сетевыми устройствами для передачи сообщений об ошибках и оперативной информации, указывающей на успех или неудачу при взаимодействии с другим IP-адресом.
LACP	Link Aggregation Control Protocol	Протокол управления агрегированием каналов - сетевой протокол для формирования единого логического канала из нескольких физических.
LLDP	Link Layer Discovery Protocol	Протокол распределения меток - сетевой протокол обнаружения и передачи информации о соседних сетевых устройствах на канальном уровне.
MC-LAG	Multi-Chassis Link Aggregation Group	Агрегирование каналов различных шасси - сетевая технология, объединяющая несколько физических каналов между коммутаторами в один логический.

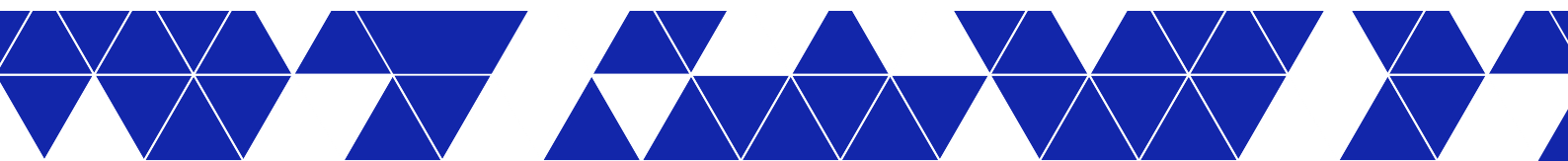


Табл. 1. Глоссарий

Аббревиатура	Термин	Описание
MIB	Management Information Base	База данных управляющей информации - база данных для управления объектами в сети связи.
Multicast	Multicast	Многоадресная передача - передача данных выбранной группе сети.
NTP	Network Time Protocol	Протокол сетевого времени - сетевой протокол для синхронизации часов.
OSPF	Open Shortest Path First	Протокол маршрутизации с определением кратчайшего пути - протокол маршрутизации с состоянием связей для сети Интернет, который связывает между собой различные маршрутизаторы и сетевые устройства.
RADIUS	Remote Authentication in Dial-In User Service	Сетевой протокол для аутентификации и авторизации пользователей в сети.
REST	Representational State Transfer	Передача состояния представления - набор правил для взаимодействия компонентов распределенного приложения в сети. REST - набор правил для организации серверного приложения, упрощающий обмен данными между системами и обеспечивающий масштабируемость.
API	Application Programming Interface	Прикладной программный интерфейс - набор протоколов и средств, который позволяет приложениям взаимодействовать с другими программными системами или сервисами.
sFLOW	Sampled Flow	Выборочный поток - протокол, используемый для сбора, отправки и анализа информации о сетевом трафике в целях мониторинга.
SNMP	Simple Network Management Protocol	Простой протокол управления сетью - протокол для сбора и организации информации об управляемых устройствах в сети Интернет.
SONiC	Software for Open Networking in the Cloud	Программное обеспечение для открытых сетей в облаке - сетевая операционная система с открытым исходным кодом на базе Linux.
SSH	Secure Shell	Протокол безопасной оболочки - защищенный протокол для работы сетевых служб через незащищенную сеть.
STP	Spanning Tree Protocol	Протокол связующего дерева - протокол, строящий нециклическую логическую топологию для Ethernet-сетей. STP определяет наименьший путь от корневого моста до каждого коммутатора в сети, блокируя некоторые порты, чтобы избежать циклических петель.

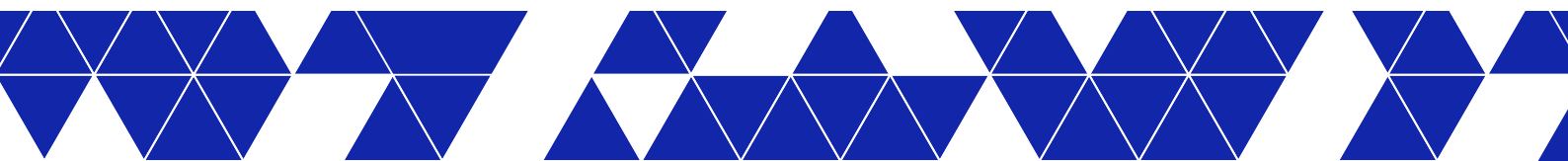


Табл. 1. Глоссарий

Аббревиатура	Термин	Описание
TACACS+	Terminal Access Controller Access-Control System+	Сетевой протокол для централизованной аутентификации, авторизации и учета (AAA) пользователей в компьютерных сетях.
TFTP	Trivial File Transfer Protocol	Простейший протокол передачи файлов - протокол, который позволяет клиенту получать файлы с удаленного узла или отправлять файлы на него.
UDLD	Unidirectional Link Detection	Обнаружение однонаправленного канала - протокол для обнаружения однонаправленных связей.
Unicast	Unicast	Одноадресная передача - передача данных к одному получателю.
VLAN	Virtual Local Area Network	Виртуальная локальная сеть - метод логического разделения физической сети на отдельные сегменты с помощью программного обеспечения, что позволяет управлять трафиком и обеспечивать безопасность сети.
VRF	Virtual Routing and Forwarding	Виртуальная маршрутизация и переадресация - технология, позволяющая создавать и управлять несколькими экземплярами таблицы маршрутизации на одном маршрутизаторе.
VRRP	Virtual Router Redundancy Protocol	Протокол резервирования виртуального маршрутизатора - сетевой протокол, обеспечивающий возможность резервирования и восстановления работоспособности сети.
VxLAN	Virtual Extensible LAN	Виртуальная расширяемая локальная сеть - технология виртуализации сети.
ZTP	Zero-touch provisioning	Автоматическая подготовка или автоматическая регистрация - функция автоматической настройки и инициализации коммутатора.



3 Базовая настройка

К базовой настройке можно приступить сразу после включения устройства.

Управление коммутатором осуществляется посредством интерфейса командной строки Kornfeld CLI. Другие способы управления устройством приведены в разделе «[Управление устройством](#)».

3.1 Подключение к устройству

1. Включение устройства.

Чтобы включить устройство, подсоедините блоки электропитания коммутатора к одному или двум источникам электропитания. Количество используемых источников зависит от требуемого типа резервирования электропитания.

2. Подключение к оборудованию.

Подключение рабочей станции к устройству возможно через:

- консольный порт (раздел [Подключение оборудования через консольный порт](#));
- порт MGMT (раздел [Подключение через порт MGMT](#)).

В случае подключения устройства к сети через порт MGMT, он должен быть достижим для рабочей станции, с которой будет производиться базовая настройка коммутатора. Для этого оба устройства должны находиться в одной сети Management, и должна быть настроена маршрутизация между ними. Сетевые настройки для коммутатора и рабочей станции необходимо либо ввести вручную, либо получить автоматически от DHCP-сервера.

3. Получение сетевых настроек.

Чтобы подключиться к устройству, необходимо назначить IP-адрес его интерфейсу MGMT. Есть два способа настройки IP-адреса на MGMT интерфейсе на коммутаторе:

- по DHCP по умолчанию (раздел [Получение IP-адреса по DHCP](#));
- вручную (раздел [Настройка IP-адреса вручную](#)).

3.1.1 Подключение оборудования

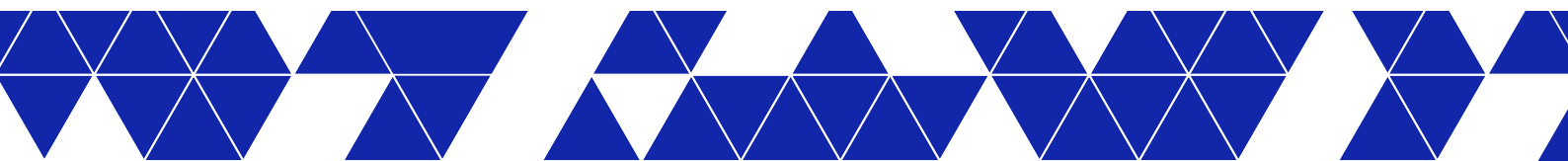
Подключение оборудования через консольный порт

Данный способ подключения позволяет получить доступ к интерфейсу управления Kornfeld CLI. Доступ осуществляется с рабочей станции на базе ОС Windows или Linux. Рабочая станция подключается кабелем Console RJ-45 - DB-9 к порту CONS на лицевой стороне коммутатора.

Вход в интерфейс Kornfeld CLI с рабочей станции с ОС Windows

❗ На рабочей станции должно быть установлено приложение PuTTY или аналог.

1. Подключите рабочую станцию к порту CONS на лицевой стороне коммутатора с помощью консольного кабеля.
2. Запустите диспетчер устройств и определите имя консольного порта коммутатора в списке портов.
3. Запустите приложение PuTTY или аналог.
4. Выберите категорию **Connection -Serial** и введите параметры подключения:



```
Speed (baud) - 115200
Data bits - 8
Stop bits - 1
Parity - None
Flow control - XON/XOFF
```

5. Выберите в блоке **Connection type** тип подключения **Serial**.
6. Введите в поле **Serial line** имя порта, который вы определили на шаге 2.
7. Нажмите **Open** для запуска сессии подключения.
8. Введите учетные данные пользователя по умолчанию:
 - имя - `admin`;
 - пароль - `admin`.

! После первого входа в систему пароль по умолчанию для локального пользователя `admin` необходимо сменить. Процедура изменения пароля содержится в разделе «Последовательность базовой настройки».

После успешной авторизации на экране появится приглашение Kornfeld CLI" />:

```
kornfeld login: admin
Password:

Kornfeld OS Distribution by YADRO
Copyright (c) 2021-2023 YADRO, https://www.yadro.com
kornfeld#
```

Вход в интерфейс Kornfeld CLI с рабочей станции с ОС Linux

! На рабочей станции должно быть установлено приложение Minicom.

1. Подключите рабочую станцию к порту CONS на лицевой стороне коммутатора с помощью консольного кабеля.
2. Выполните команду `dmesg | grep /dev/tty` и определите имя консольного порта коммутатора в выводе команды.
3. Выполните команду `sudo minicom -s`.
4. Выберите пункт **Serial port setup** и настройте параметры подключения:

```
A - Serial port: /dev/<port-name>
E - Bps/par/Bits: 115200 8N1
F - Hardware Flow Control: No
G - Software Flow Control: Yes
```

Здесь `<port-name>` - имя консольного порта, который вы определили на шаге 2.

5. Выберите пункт **Save setup as dfl**.
6. Выберите пункт **Exit**.
7. Выполните команду `sudo minicom`.
8. Введите учетные данные пользователя:
 - имя - `admin`;
 - пароль - `admin`.

! После первого входа в систему пароль по умолчанию для локального пользователя `admin` необходимо сменить. Процедура изменения пароля содержится в разделе «Последовательность базовой настройки».

После успешной авторизации на экране появится приглашение Kornfeld CLI:

```
kornfeld login: admin
Password:

Kornfeld OS Distribution by YADRO
Copyright (c) 2021-2023 YADRO, https://www.yadro.com
kornfeld#
```

Подключение через порт MGMT

Данный способ подключения позволяет получить доступ к интерфейсу управления Kornfeld CLI. Доступ осуществляется с рабочей станции на базе ОС Windows или Linux, подключенной к порту MGMT коммутатора.

При таком способе подключения IP-адрес интерфейса MGMT для рабочей станции и коммутатора можно либо настроить вручную, либо получить автоматически по DHCP. Более подробно о получении сетевых настроек по DHCP - в разделе «[Получение сетевых настроек](#)».

Вход в интерфейс Kornfeld CLI с рабочей станции с ОС Windows

❗ На рабочей станции должно быть установлено приложение PuTTY или аналог.

1. Запустите приложение PuTTY или аналог.
2. Убедитесь, что в блоке **Connection type** выбран тип подключения **SSH**.
3. Введите IP-адрес в поле **Host Name** или **IP Address**.
4. Убедитесь, что в поле **Port** установлено значение **22**.
5. Нажмите **Open** для запуска сессии подключения.
6. Нажмите **Yes** в открывшемся окне предупреждения о безопасности.
7. Введите учетные данные пользователя по умолчанию:
 - имя - `admin`;
 - пароль - `admin`.

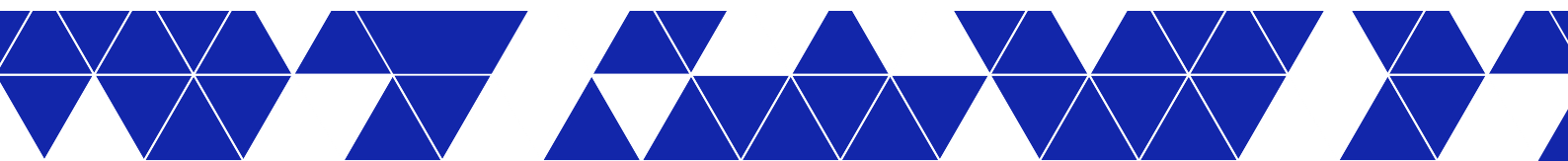
❗ После первого входа в систему пароль по умолчанию для локального пользователя `admin` необходимо сменить. Процедура изменения пароля содержится в разделе «Последовательность базовой настройки».

Вход в интерфейс Kornfeld CLI с рабочей станции с ОС Linux

1. Выполните команду `ssh <user>@<IP>`, где `<IP>` - IP-адрес, `<user>` - имя учетной записи.
2. Введите учетные данные пользователя по умолчанию:
 - имя - `admin`;
 - пароль - `admin`.

❗ После первого входа в систему пароль по умолчанию для локального пользователя `admin` необходимо сменить. Процедура изменения пароля содержится в разделе «Последовательность базовой настройки».

После успешной авторизации на экране появится приглашение Kornfeld CLI:



```
kornfeld login: admin
Password:

Kornfeld OS Distribution by YADRO
Copyright (c) 2021-2023 YADRO, https://www.yadro.com
kornfeld#
```

3.1.2 Получение сетевых настроек

Получение IP-адреса по DHCP

Обращение к интерфейсу Kornfeld CLI через порт MGMT происходит по IP-адресу. Назначить коммутатору IP-адрес можно как вручную, так и с помощью DHCP-сервера.

1. Подключите коммутатор и рабочую станцию к локальной сети с помощью кабелей Ethernet.
2. Убедитесь, что DHCP-сервер настроен и доступен в этой сети. Чтобы сетевые устройства могли автоматически получать сетевые настройки от DHCP-сервера, он должен быть настроен и доступен.
3. Чтобы узнать IP-адрес, который назначен DHCP-сервером порту MGMT коммутатора, введите команду:

```
show interface Management0
```

Настройка IP-адреса вручную

В ситуации, когда получение параметров для подключения к сети от DHCP-сервера невозможно, IP-адрес, маску подсети и адрес шлюза можно настроить вручную.

1. Войдите в систему как пользователь admin. Пароль - admin.

```
kornfeld login: admin
```

! После первого входа в систему пароль по умолчанию для локального пользователя admin необходимо сменить. Процедура изменения пароля содержится в разделе «Последовательность базовой настройки».

2. Войдите в режим конфигурирования.

```
configure terminal
```

3. Войдите в режим конфигурирования интерфейса MGMT.

```
show interface Management0
```

4. Введите параметры подключения.

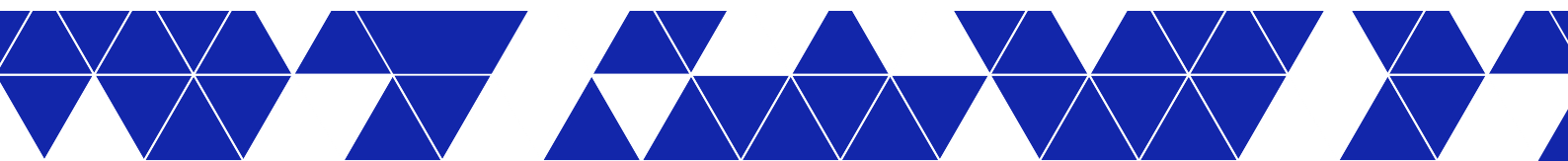
```
ip address <IP-адрес>/<маска подсети> addr <шлюз>
```

5. Сохраните изменения в конфигурации startup.

```
do write memory
```

6. Выйдите из режима конфигурирования.

```
end
```



3.2 Работа с Kornfeld CLI

3.2.1 Режимы командной строки

Интерфейс командной строки Kornfeld CLI имеет два режима: режим просмотра и режим конфигурирования.

Режим просмотра (execution mode)

Используется для поиска неполадок, проверки состояния устройства, просмотра конфигурации, выключения и перезагрузки устройства. Данный режим устанавливается при входе в Kornfeld CLI по умолчанию.

Командная строка режима просмотра:

```
kornfeld#
```

Чтобы вернуться в режим просмотра из любого другого режима, используйте команду `end`.

Режим конфигурирования (configuration mode)

Используется для изменения параметров устройства. Вводимые в режиме конфигурирования команды изменяют текущую конфигурацию. Для перехода в режим конфигурирования используйте команду `configure terminal`.

Из режима конфигурирования можно переходить в подрежимы для конфигурирования отдельных сущностей и функций устройства, например, интерфейсов, протоколов маршрутизации и т. д.

Командная строка режима конфигурирования:

```
kornfeld(config)#
```

Чтобы выйти из режима конфигурирования и вернуться в режим просмотра, используйте команду `exit`. Также используйте эту команду, чтобы вернуться из подрежима конфигурирования функции в режим конфигурирования коммутатора.

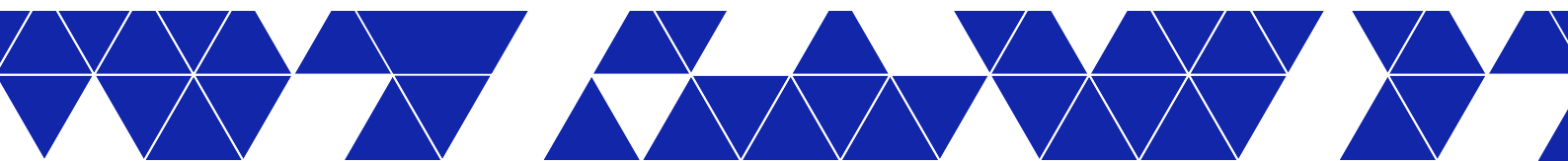
В каждом режиме предусмотрен свой набор разрешенных команд Kornfeld CLI. Кроме того, существует общий для двух режимов работы набор команд, субкоманд и клавиш управления.

Оба режима поддерживают вызов контекстной подсказки (помощи), автодополнение команд, прокрутку многостраничного режима, поиск, фильтрацию, фильтр исключения, сохранение результатов вывода команды в файл.

3.2.2 Синтаксис

Ниже представлены следующие соглашения по синтаксису команд, которые используются в справочнике по командам:

- **Полужирным** шрифтом выделяются команды и ключевые слова, которые вводятся буквально, как показано в примерах реальной конфигурации и сообщений системы. Полужирным шрифтом выделяются команды, которые вводятся пользователем вручную.
- *Курсивом* выделяются аргументы команды, для которых пользователь указывает необходимое значение.
- Вертикальная черта | разделяет взаимоисключающие элементы.



- Фигурные скобки { } включают в себя обязательные элементы.
- Квадратные скобки [] включают в себя необязательные элементы.
- Фигурные скобки, помещенные в квадратные скобки [x {y | z}], указывают на выбор обязательных элементов внутри необязательного элемента.

В рассматриваемых примерах используются следующие условные обозначения:

- Сеансы терминала и данные, выводимые Kornfeld OS, выделяются **экранным** шрифтом.
- Непечатаемые символы, такие как пароли или символы табуляции, заключаются в угловые скобки <>.

3.2.3 Функции управления

Функция **отображения информации**. Например, команда `show` отображает информацию об аппаратном или программном обеспечении.

Функция **очистки данных**. Команда `clear` стирает различные типы системной информации.

Функция **автодополнения** или **завершения частично введенной команды**. Чтобы завершить частично введенную команду или параметр, нажмите клавишу **Tab** или клавишу **Space** (пробела). Если частично введенная строка однозначно идентифицирует команду - отобразится полное имя команды. Более подробная информация о способах автодополнения ввода команд приведена в разделе «Автодополнение команд» ([Автодополнение команд](#)).

Функция **фильтрации выходных данных команды**. Для фильтрации вывода введите после команды `show` **вертикальную черту** (|), за которой следует любая из этих команд:

- `except` - отображает только текст, который не соответствует шаблону.
- `find` - ищет первое появление шаблона и отображает все последующие совпадения.
- `grep` - отображает только текст, соответствующий указанному шаблону.
- `no-more` - выводит данные сплошным текстом.
- `save` - сохраняет выходные данные в файл.

3.2.4 Контекстная подсказка

Для просмотра списка доступных в любом режиме (просмотра или конфигурирования) команд необходимо ввести клавишу вопросительного знака `?`.

```
kornfeld# ?
ack      Acknowledges the running configuration
bash     Runs the Bash shell
clear    Clear commands
configure Enters to configuration mode
copy     Performs configuration copy operations
exit     Exits from the CLI in the execution mode
image    Image related commands
logger   Enters messages into the system log
no       No commands under Exec mode
ping     Sends ICMP ECHO_REQUEST to network hosts
ping6    Sends ICMPv6 ECHO_REQUEST to network hosts
poweroff Power-off the switch
reboot   Reboots the switch
renew    Renew commands
show     Displays running system information
terminal Set terminal settings
traceroute Traces a packet route to the host
traceroute6 Traces a packet route to the IPv6 host
write    Saves the running configuration to the startup configuration
ztp      Restarts the ZTP function
```

Если начать вводить команду и на середине слова нажать клавишу вопросительного знака `?`, будет выведен список доступных команд, которые начинаются с введенных символов.



```
kornfeld# co?
configure  Enters to configuration mode
copy       Performs configuration copy operations
```

Если сначала ввести команду полностью и потом нажать клавишу вопросительного знака `?`, будет выведен список возможных вариантов данной команды или список параметров этой команды.

```
kornfeld# copy ?
running-configuration  Copies the running configuration
saved-configuration    Copies saved configuration
startup-configuration  Copies the startup configuration
```

Если в режиме просмотра нажать клавишу вопросительного знака `?` после ввода параметра, будет отображен список параметров данной команды, которые следуют после введенного параметра.

```
kornfeld# show ssh-server ?
ciphers      Displays enabled SSH ciphers
configuration Displays SSH server configuration
hostkey      Displays enabled SSH host key types
```

Если в режиме конфигурирования нажать клавишу вопросительного знака `?` после ввода параметра, будут выведены возможные диапазоны значений этих параметров и их описание.

```
kornfeld(config)# ssh-server timeout ?
<0..4294967295> SSH timeout
```

3.2.5 Автодополнение команд

В Kornfeld CLI реализована возможность автоматически дописывать окончание вводимой команды, если она однозначно определена в системе.

Ввод любой команды, любого параметра или значения можно автоматически дополнить нажатием клавиши табуляции **Tab**. Если пользователь начал вводить команду и не дописал ее, он может нажать клавишу **Tab**, и команда будет полностью дописана.

Если команду можно продолжить разными способами, то есть существуют разные команды, которые начинаются с введенных символов, то Kornfeld CLI предложит список вариантов продолжения написания команды. В таком случае можно ввести первый отличающийся символ и снова нажать клавишу **Tab**, чтобы затем ввести команду полностью. Аналогичным образом можно вводить любые параметры и их значения.

```
kornfeld# pi<TAB>
ping ping6
-I -M -V -W -c -h -i -s -t -v vrf
```

Автоматически дополнять команды можно также по нажатию клавиш **Enter** и **Space**. Такой способ работает только в том случае, если в системе нет других команд, начинающихся с введенных символов. В противном случае команда выполнится с ошибкой, указывающей на то, что введенных символов недостаточно, чтобы однозначно трактовать команду.

```
kornfeld# re<ENTER>
reboot renew
Error: Ambiguous command.
```

При автодополнении клавишей **Enter** команда дополняется и сразу выполняется, а при дополнении клавишей **Space** команда только дополняется. Чтобы выполнить команду, нужно нажать клавишу **Enter**.

3.2.6 Горячие клавиши командной строки

В Kornfeld CLI поддерживаются следующие горячие клавиши командной строки:



Табл. 2. Горячие клавиши командной строки

Горячие клавиши	Описание
up arrow	Выводит предыдущую команду из списка введенных команд.
down arrow	Выводит следующую команду из списка введенных команд.
left arrow	Сдвигает курсор на один символ влево.
right arrow	Сдвигает курсор на один символ вправо.
q	Останавливает вывод команды, если он занимает более одного экрана, и возвращает пользователя обратно в командный режим.
Ctrl + C	Останавливает запущенную команду.
?	Выводит список всех допустимых для текущего режима работы команд с описанием. Выводит список допустимых параметров или ключевых слов с их описанием через пробел после ключевого слова. Выводит все допустимые команды, начинающиеся с введенной строки.
Tab	Выводит все допустимые для текущего режима работы команды без описания или все допустимые варианты продолжения написания команды по введенным символам. Завершает ввод команды, аргумента, ключевого слова, если в системе такое написание однозначно определено.
Space	Завершает ввод команды, аргумента, ключевого слова, если в системе такое написание однозначно определено.
Enter	Завершает ввод команды, аргумента, ключевого слова, если в системе такое написание однозначно определено, и запускает выполнение команды.

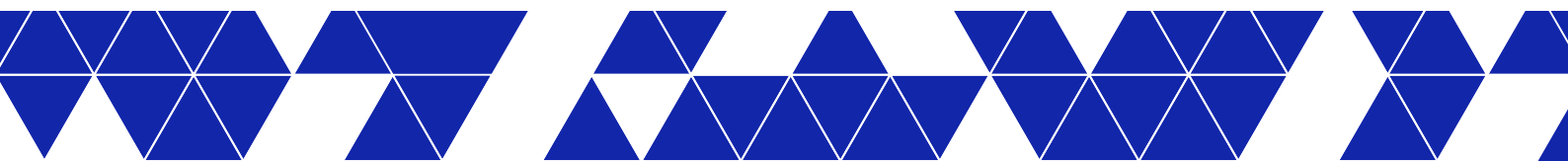
3.2.7 Сообщения об ошибках

В Kornfeld CLI проверяется корректность всех вводимых команд. Если система коммутатора не распознает команду, на экране отображается сообщение об ошибке.

Ниже в таблице приведен перечень ошибок, которые могут отображаться при вводе команд, с описаниями.

Табл. 3. Перечень ошибок

Сообщение об ошибке	Описание
<code>%Error: Invalid input detected at "^" marker.</code>	Команда введена некорректно. Символ «^» отмечает начало ошибки в строке.
<code>%Error: The command is not completed.</code>	В введенной строке отсутствуют обязательные параметры или ключевые для выполнения команды слова.
<code>%Error: Ambiguous command.</code>	Ошибка возникает при автодополнении ввода и указывает на то, что введенных символов недостаточно для однозначного определения слова (команды, параметра, аргумента, значения или ключевого слова).



3.3 Файловая система и работа с файлами

Прямого доступа к файловой системе в Kornfeld CLI нет. Однако реализована функция сохранения вывода команд `show` в текстовый файл.

Пользователь может создавать файлы только в своей директории в разделе файловой системы `/home`. Доступ в любые другие разделы файловой системы запрещен системными параметрами Kornfeld OS.

Сохранение вывода в файл

Команда `save` позволяет сохранять вывод в текстовый файл. Для этого в командной строке Kornfeld CLI необходимо ввести команду `show`, перенаправить ее вывод команде `save` и указать имя файла. Файл будет сохранен в директории текущего пользователя `/home/<user>`.

```
save filename.txt
```

Чтобы сохранить файл в поддиректории внутри директории текущего пользователя, необходимо указать относительный путь до файла.

```
save subdir1/subdir2/subdir3/filename.txt
```

Продуманная система поддиректорий позволяет организовать хранение файлов конфигураций, избежать загрузки коммутатора с некорректной конфигурацией, потери данных и незапланированного простоя оборудования.

Загрузка файлов с коммутатора

Файлы можно скачивать с коммутатора и загружать на коммутатор, если подключаться к нему по SSH с файлового клиента. Для этого можно использовать утилиты SCP, WinSCP и подобные. Более подробная информация приведена в разделе «[Управление устройством](#)».

Пример сохранения вывода в файл

1. Подключитесь к коммутатору через консольный порт и сохраните вывод команды в текстовый файл.

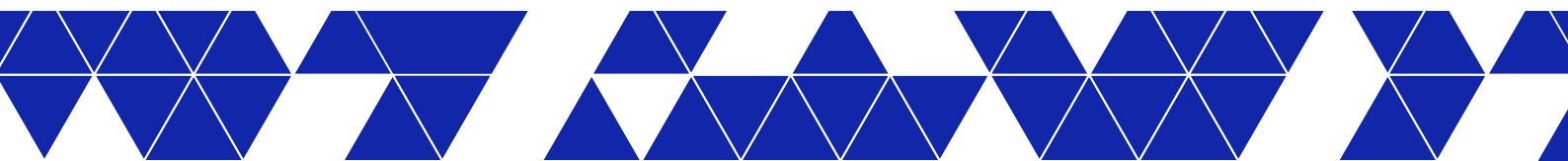
```
kornfeld# show running-configuration | save sw1-config.conf
```

2. Созданный файл загрузите с коммутатора на удаленную машину по SFTP.

```
<user>@PC1:~/Downloads$ sftp -oUser=admin A.B.C.D:/home/admin/sw1-config.conf .
admin@A.B.C.D's password:
Connected to A.B.C.D.
Fetching /home/admin/sw1-config.conf to ./sw1-config.conf
sw1-config.conf                                100% 13KB 39.1KB/s 00:00
<user>@PC1:~/Downloads$ ls -la | grep sw1-config.conf
-rw-r--r--  1 <user> <group>    13682 Jul 28 12:44 sw1-config.conf
```

3.4 Файлы конфигурации

Файл конфигурации - это текстовый файл, который содержит строки с командами, выполняемыми на коммутаторе. Перечисленные в файле конфигурации команды выполняются на коммутаторе в автоматическом режиме и определяют работу всех функций и подсистем. Пользователь может сохранять, просматривать, переименовывать, архивировать, восстанавливать и удалять файлы конфигурации.



! На коммутаторе можно запускать только ту конфигурацию, которая ранее была сохранена именно на этом физическом устройстве. Версия микропрограммы при этом должна быть той же, при которой конфигурация сохранялась. В противном случае устройство может стать неработоспособным.

Правила именования файлов конфигурации

1. Имя файла должно соответствовать правилам именования файлов в операционной системе Linux.
2. Длина имени файла конфигурации не должна превышать 50 символов.
3. Рекомендуется поддерживать единый формат имен файлов конфигураций: `<имя хоста>-[краткое описание]-<гггг-мм-дд>_<ччмм>`.

Единый формат имен файлов облегчает работу с ними и позволяет избежать случайной перезаписи файлов конфигурации, а также загрузки коммутатора с некорректной конфигурацией.

4. Рекомендуется использовать в имени файла конфигурации только разрешенные символы:
 - буквы латиницы в любом регистре,
 - цифры,
 - символ «.» (точка),
 - символ «_» (подчеркивание),
 - символ «-» (дефис).

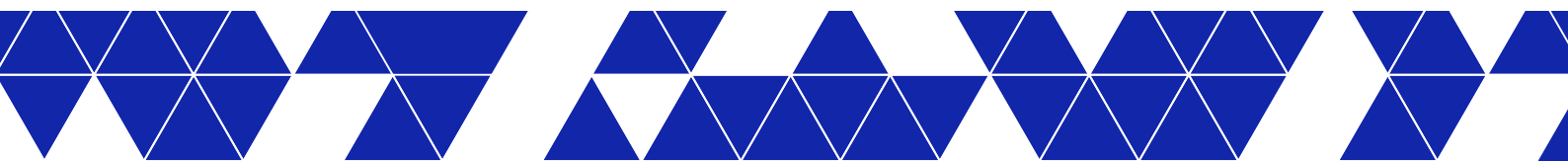
Использовать другие символы не рекомендуется, поскольку файлы конфигурации с некорректными именами могут неправильно сохраниться и не загрузиться.

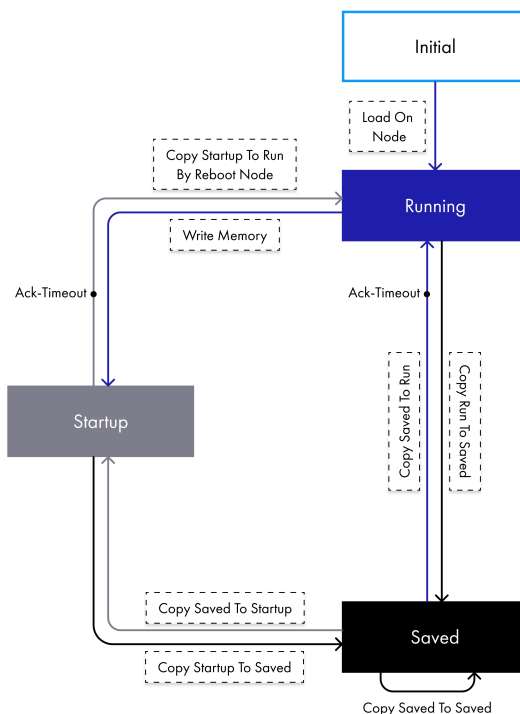
3.4.1 Виды конфигураций

В Kornfeld OS предусмотрены следующие виды конфигураций:

- исходная конфигурация, которая применяется при первом запуске коммутатора (initial configuration),
- текущая конфигурация (running configuration),
- конфигурация, которая применяется после перезагрузки коммутатора (startup configuration),
- конфигурация, которая хранится в памяти коммутатора (saved configuration).

Действия с конфигурациями





В первый раз коммутатор загружается с исходной конфигурацией (initial configuration), которая содержит заводские значения параметров по умолчанию. Эта конфигурация становится текущей конфигурацией (running configuration). Все изменения настроек коммутатора применяются в текущей конфигурации.

После перезагрузки коммутатор всегда ищет конфигурацию startup и загружает ее. Чтобы коммутатор после перезагрузки применял нужную конфигурацию, ее необходимо сохранить как startup. Для этого по завершении конфигурирования параметров коммутатора необходимо вводить команду `do write memory`.

Любую текущую конфигурацию можно сохранить в памяти коммутатора, тогда она становится сохраненной конфигурацией (saved configuration). Количество сохраненных файлов конфигураций ограничивается только размерами дискового пространства коммутатора.

3.4.2 Сохранение конфигурации

Существует два способа сохранить любую конфигурацию:

- в текстовый файл;
- перевести конфигурацию в вид saved.

Сохранение конфигурации в файл

Сохранение конфигурации в файл осуществляется с помощью команды `save`. Более подробная информация об этом, а также о правилах именования файлов конфигурации представлена в разделе [Файловая система и работа с файлами](#).

! Если имя сохраняемого файла конфигурации совпадает с именем файла другой конфигурации, то новая конфигурация будет перезаписана поверх старой.

Сохранение текущей конфигурации (running)

Чтобы сохранить текущую конфигурацию (running) в виде saved, введите команду `copy` и укажите имя файла сохраняемой конфигурации.

```
copy running-configuration saved-configuration <имя файла конфигурации>
```

Сохранение стартовой конфигурации (startup)

Чтобы сохранить стартовую конфигурацию (startup), применяемую при запуске, в виде конфигурации saved, введите команду `copy` и укажите имя файла сохраняемой конфигурации.

```
copy startup-configuration saved-configuration <имя файла конфигурации>
```

Сохранение изменений в стартовую конфигурацию (startup)

Чтобы введенные настройки сохранились в памяти устройства и применились после перезагрузки, они должны быть сохранены в стартовой конфигурации (startup). Для этого по завершении конфигурирования необходимо ввести команду `do write memory` в режиме конфигурирования.

```
do write memory
```

3.4.3 Применение конфигурации

Под применением конфигурации понимается перевод любой конфигурации в вид текущей конфигурации (running configuration). Иначе данную операцию можно называть восстановлением конфигурации.

Применение сохраненной конфигурации (saved) в качестве текущей (running)

Любую сохраненную ранее конфигурацию можно восстановить в качестве текущей. Для этого введите следующую команду и укажите имя файла той конфигурации, которую требуется восстановить:

```
copy saved-configuration <имя файла конфигурации> running-configuration
```

Применение конфигурации (startup) в качестве текущей (running)

Любые совершенные в текущей конфигурации изменения можно откатить до тех, которые применялись при загрузке коммутатора. Для этого необходимо перевести конфигурацию из startup в running.

```
copy startup-configuration running-configuration
```

Применение стартовой конфигурации (startup) в качестве текущей (running) с подтверждением

В качестве дополнительной меры безопасности конфигурацию можно применить на заданный промежуток времени, в течение которого система будет ожидать от пользователя подтверждения применения новой конфигурации в качестве текущей. Если подтверждение получено, новая конфигурация применяется и становится текущей. В противном случае внесенные изменения удаляются, и текущая конфигурация восстанавливается до прежней версии.

```
copy startup-configuration running-configuration ack-timeout <время в секундах>
```

Параметр `ack-timeout` указывает время, в течение которого система ожидает подтверждения пользователя о применении конфигурации. Параметр принимает значение в диапазоне от 60 до 600 секунд. При применении конфигурации временно может быть потеряно соединение. Оно восстанавливается автоматически.

Если в течение интервала `ack-timeout` подтверждение о смене конфигурации от пользователя не было получено, система восстановит прежнюю текущую конфигурацию и предупредит об этом.

```
ATTENTION! The running configuration is not acknowledged.  
The previous running configuration will be restored.
```

Чтобы подтвердить применение конфигурации до истечения интервала времени `ack-timeout`, необходимо ввести команду:

```
ack running-configuration
```



Данная функция защищает устройство от загрузки из непроверенной конфигурации и позволяет в случае ошибки автоматически вернуться к последней рабочей версии конфигурации.

3.4.4 Просмотр конфигурации

В системе предусмотрены четыре вида конфигураций, которые подробно описаны в разделе «[Виды конфигураций](#)». Команда `show` позволяет выводить на экран и проверять содержимое нужной конфигурации.

Просмотр конфигурации, которая применяется после перезагрузки (startup)

Чтобы вывести на экран содержимое конфигурации (startup), которая применяется после перезагрузки, введите команду:

```
show startup-configuration
```

Просмотр текущей конфигурации (running)

Чтобы вывести на экран содержимое текущей конфигурации (running), введите команду:

```
show running-configuration
```

Просмотр сохраненных конфигураций (saved)

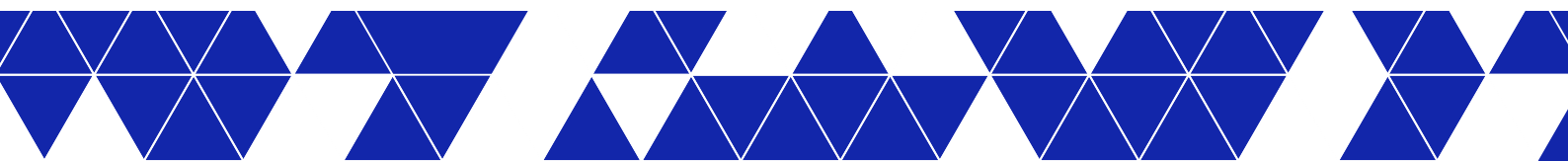
1. Вывести на экран список всех сохраненных конфигураций.

```
show saved-configuration list
```

Вывод команды будет содержать все сохраненные в памяти коммутатора конфигурации. В выводе команды указывается имя конфигурации (Name), имя пользователя (Author), который сохранил конфигурацию, а также дата и время сохранения (Saved) конфигурации.

2. Вывести на экран одну из сохраненных конфигураций. В параметре `имя файла конфигурации` указывается имя сохраненной конфигурации, которую необходимо отобразить на экране.

```
show saved-configuration name <имя файла конфигурации>
```



4 Системные настройки

Настоящий раздел описывает основные аспекты базовой настройки коммутатора, включая параметры системного баннера, системные часы и протокол сетевого времени, способствующие эффективной работе и управлению устройством.

4.1 Настройка баннера

Баннер MOTD (Message of the Day) представляет собой текстовое сообщение, которое отображается пользователям при попытке входа в коммутатор. Может содержать информацию о правилах использования системы, предупреждениях о доступе и другие полезные сообщения для пользователей перед тем, как они получают доступ к командному интерфейсу устройства.

Баннер MOTD (Message of the Day) появляется в командной строке перед входом в систему коммутатора. Включите баннер MOTD с помощью команды `banner motd motd text` в режиме настройки.

Можно ввести не более 255 символов. Используйте управляющие последовательности для отображения символа новой строки, одиночной кавычки или некоторых других символов в тексте баннера. В таблице ниже приведены управляющие последовательности и символы, которые они обозначают.

Табл. 4. Управляющие последовательности и символы, которые они обозначают

Управляющие последовательности	Символы на баннере
<code>\t</code>	Табуляция
<code>\n</code>	Перевод строки
<code>\r</code>	Возврат каретки
<code>\"</code>	Двойная кавычка
<code>\\</code>	Обратный косая черта

Настройте баннер входа в систему

```
sonic# configure terminal
      sonic(config)# banner motd "Welcome!\nThis device is owned by YADRO.\nUnauthorised access is prohibited.\nAll rights reserved."
      sonic(config)# exit
```

По умолчанию MOTD баннер пуст. Чтобы отключить настроенный баннер MOTD, используйте команду `no banner motd`.

Проверьте содержимое баннера входа в систему

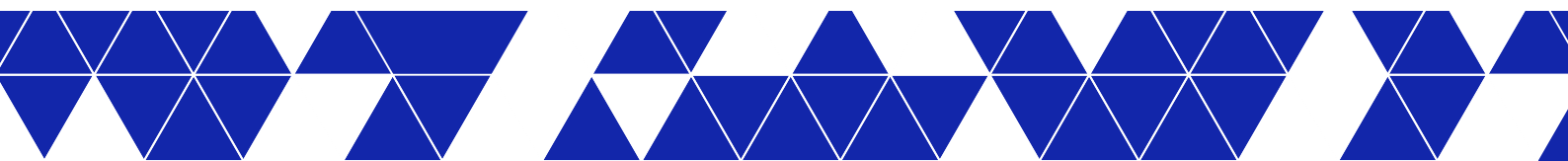
```
sonic# show banner motd
```

Проверьте баннер входа в систему в текущей конфигурации

```
sonic# show running-configuration banner motd
```

4.2 Настройка системного времени и даты

Системное время - фундаментальный элемент в настройках коммутатора. Они функционируют с момента включения системы и служат для отслеживания даты и времени в соответствии с координированным всемирным временем (UTC). Системные часы осуществляют контроль за статусом времени, определяя, является ли оно авторитетным (т.е. было установлено источником, считающимся надежным). В случае, если время не считается авторитетным, оно доступно только для просмотра и не передается дальше.



Настройка системного времени на сетевых устройствах очень важна. Синхронизация сетевого времени обязательна, поскольку все аспекты управления сетью, обеспечения безопасности, планирования и устранения неисправностей тесно связаны с правильным определением времени возникновения событий. Без синхронизации времени невозможно точно сопоставить журналы событий на устройствах в случае анализа безопасности или мониторинга сети.

Синхронизация времени также уменьшает беспорядок в общих файловых системах. Важно, чтобы метки времени модификации было одинаковым независимо от того, на какой машине находятся файловые системы.

Управлять системным временем и датой на коммутаторе можно с помощью автоматических настроек, например, протокола сетевого времени (NTP). При включении NTP коммутатор динамически синхронизирует время устройства с временем, полученным от сервера NTP.

Для корректного отображения системного времени вы можете вручную настроить информацию о локальном часовом поясе. Коммутатор поддерживает названия часовых поясов, действующих в Linux. По умолчанию используется часовой пояс UTC.

Задаёт часовой пояс

```
sonic# configure terminal
sonic(config)# clock timezone Europe/Moscow
sonic(config)# end
```

Чтобы вернуться к часовому поясу по умолчанию, используйте команду `no clock timezone` в режиме настройки.

Просматривает системную дату, время и настроенный часовой пояс.

```
sonic# show clock
Tue Mar 21 10:13:33 MSK 2023
```

4.3 Настройка протокола сетевого времени NTP

Протокол сетевого времени (NTP) синхронизирует время между серверами и клиентами. Протокол координирует распределение времени в сети. NTP-клиенты синхронизируют свои часы с NTP-серверами, которые обеспечивают точное измерение времени. NTP-клиенты выбирают один из нескольких доступных NTP-серверов, определяя самый надежный и доступный источник времени для синхронизации.

Для настройки коммутатора на опрос определенных узлов, предоставляющих NTP-время, производится отправка сообщений на NTP-серверы. Затем коммутатор обрабатывает полученные ответы и анализирует информацию в NTP-сообщении, чтобы оценить характеристики времени своих коллег. Эта информация позволяет коммутатору выбирать наилучший источник времени, синхронизировать свои локальные часы и распространять информацию о времени по сети.

4.3.1 Настройка протокола сетевого времени NTP по умолчанию

В таблице 5 приведена настройка NTP по умолчанию.

Табл. 5. Настройка протокола сетевого времени по умолчанию

Характеристика	Настройки по умолчанию
NTP-сервер	Не настроено.
NTP в VRF-интерфейсе	VRF по умолчанию.

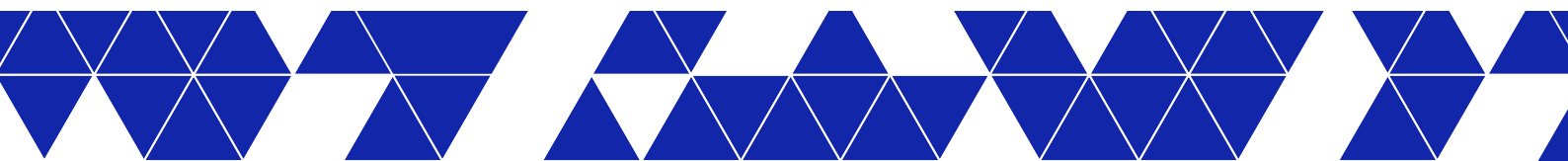


Табл. 5. Настройка протокола сетевого времени по умолчанию

Характеристика	Настройки по умолчанию
NTP интерфейс источника	Не настроено.
NTP аутентификация	Выключено. Ключ аутентификации не указан.
NTP-ассоциации	Не настроено.

4.3.2 Настройка протокола сетевого времени NTP

! Перед настройкой службы NTP на коммутаторе необходимо сначала вручную установить часовой пояс. Смотреть раздел: [Настройка системного времени и даты](#).

Для включения NTP на коммутаторе настройте NTP-сервер с помощью команды `ntp server`. Укажите IP-адрес или доменное имя сервера, с которым будет синхронизироваться время системы.

```
sonic# configure terminal
sonic(config)# ntp server 198.51.100.1
sonic(config)# end
```

В сети можно настроить несколько NTP-серверов. Оптимальной практикой считается наличие не менее 3 NTP-серверов. Текущая версия Kornfeld OS позволяет настроить до 10 NTP-серверов. Система выбирает один из этих NTP-серверов для синхронизации, основываясь на надежности связи.

Чтобы сбросить настройку NTP-сервера, введите версию `no` команды `complete`.

Настройка NTP в VRF

Можно изолировать трафик NTP от сетевого трафика, настроив NTP на работу в VRF. Для этого воспользуйтесь командой режима настройки `ntp vrf {vrf-name}`. Параметр `vrf-name` может быть либо `mgmt`, либо `default` либо строкой до 15 символов, начинающейся с символов `Vrf`.

Когда VRF не настроен, служба NTP включается в VRF по умолчанию.

```
sonic# configure terminal
sonic(config)# ntp vrf mgmt
sonic(config)# end
```

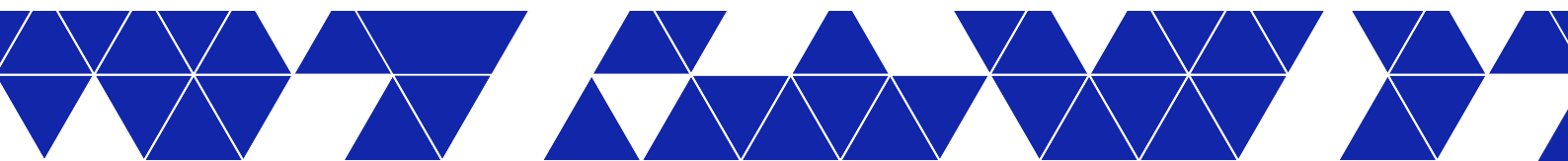
4.3.3 Настройка интерфейса источника протокола сетевого времени NTP

Можно настроить NTP на использование определенного IP-адреса для отправки и получения всех пакетов NTP. По умолчанию в качестве IP-адреса для NTP-пакетов используется IP-адрес интерфейса, через который система выходит в сеть.

По умолчанию IP-адресом источника пакетов NTP является `Management0`. Можно настроить только один интерфейс источника NTP.

Для предотвращения использования IP-адреса определенного интерфейса в качестве адреса для ответных сообщений, можно использовать данную команду, чтобы указать интерфейс, который будет служить источником для NTP-сообщений. Таким образом, в NTP-сообщениях будет использоваться основной IP-адрес интерфейса в качестве адреса источника.

Настройте интерфейс на коммутаторе с помощью команды `ntp source-interface {Ethernet | Loopback | Management | PortChannel | Vlan}` в режиме настройки.



- `Ethernet` для имени интерфейса или субинтерфейса Ethernet.
- `Loopback` для имени Loopback-интерфейса.
- `Management` для имени Management-интерфейса.
- `PortChannel` для имени интерфейса или подинтерфейса port-channel.
- `Vlan` для имени VLAN-интерфейса.

Чтобы удалить интерфейс источника NTP, введите версию `no` команды `complete`.

Настройка источника IP-адреса

```
sonic# configure terminal
sonic(config)# ntp source-interface Ethernet 1
sonic(config)# end
```

Просмотр настройки IP-адреса источника

```
sonic# configure terminal
sonic(config)# do show running-configuration | grep source
ntp source ethernet1/1/1
sonic(config)# end
```

4.3.4 Настройка аутентификации протокола сетевого времени NTP

NTP-аутентификация гарантирует, что NTP-клиенты синхронизируют свое время с надежными источниками времени. При включении NTP-аутентификации коммутатор синхронизируется с сервером времени только в том случае, если он содержит один из ключей аутентификации, известных коммутатору. Коммутатор отбрасывает пакеты, не прошедшие проверку подлинности, и не позволяет им обновлять локальные часы.

По умолчанию аутентификация NTP отключена. Чтобы включить аутентификацию NTP на коммутаторе и предотвратить синхронизацию с неаутентифицированными источниками, используйте команду `ntp authenticate` в режиме конфигурирования.

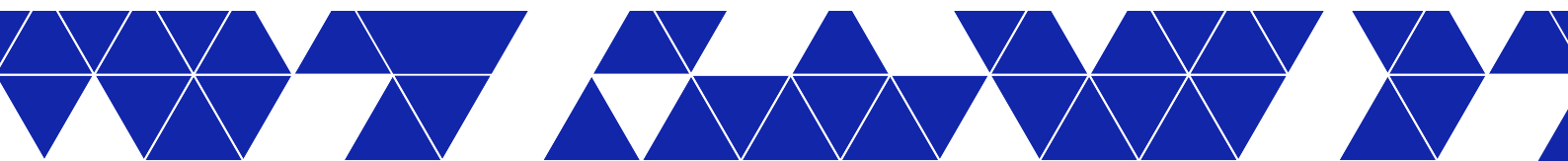
Процесс аутентификации начинается с настройки ключа и генерации первого NTP-пакета. Ключ встраивается в пакет синхронизации, который отправляется источнику времени NTP. В первый раз пароль аутентификации можно ввести открытым текстом. Он будет зашифрован в работающей настройке. В дальнейшем зашифрованный пароль (с ключевым словом `encrypted`) можно скопировать и вставить из вывода команды `show running configuration`.

1. Задайте номер и ключ аутентификации с помощью команды `ntp authentication-key key-number auth-type key` в режиме конфигурирования.

- `key-number` - число от 1 до 65535, значение по умолчанию не задано.
- `auth-type` - означает алгоритм аутентификации и поддерживает значения `md5`, `sha1`, и `sha2-256`.
- `key` - это строка длиной до 64 символов. Она может содержать любые печатаемые символы ASCII, кроме `#`, `|`, ПРОБЕЛА и ЗАПЯТОЙ. Ключи длиной более 20 символов должны быть представлены в шестнадцатеричном формате.

```
sonic# configure terminal
sonic(config)# ntp authentication-key 1 sha1 8ecf2f51b25b2cbdae69a1030aee728d66859817
sonic(config)# ntp authentication-key 2 md5 104D000A0618
sonic(config)# ntp authentication-key 3 sha1 534e8d1d211dfbf1793be16d2dbf53dfd5879ad1
sonic(config)# end
```

2. Определите доверенный ключ с помощью команды `ntp trusted-key key-number` в режиме настройки. `key-number` (от 1 до 65535) должен совпадать с настроенным ключом аутентификации NTP. Ключи определяют доверенные источники - NTP-серверы, от которых коммутатор принимает синхронизацию времени.



```
sonic# configure terminal
sonic(config)# ntp trusted-key 1
sonic(config)# ntp trusted-key 3
sonic(config)# end
```

3. Настройте NTP-сервер, указав, какой ключ использовать для этого сервера. Используйте команду `ntp server host [key key-number]` в режиме настройки.

Настройка NTP-аутентификации

```
sonic# configure terminal
sonic(config)# ntp authenticate
sonic(config)# ntp authentication-key 3 sha1 534e8d1d211dfbf1793be16d2dbf53dfd5879ad1
sonic(config)# ntp trusted-key 3
sonic(config)# ntp server 198.51.100.1 key 3
sonic(config)# end
```

В данном примере показано использование аутентификации на основе ключа SHA-1. При использовании аутентификацию на основе другого алгоритма, используйте соответствующий ключ. Настройте те же параметры аутентификации NTP на удаленном NTP-сервере, который служит источником времени NTP, или на соответствующем NTP-клиенте.

Чтобы удалить ключ аутентификации для NTP, используйте команду `no ntp authentication-key` в режиме настройки. Чтобы отключить аутентификацию NTP, введите `no ntp authenticate key-number`.

4.3.5 Просмотр NTP-ассоциаций

Связь между NTP-сервером и коммутатором, на котором работает NTP-клиент, называется ассоциацией. Для отображения настроенных серверов времени и надежности их связи используйте команду `show ntp associations` в режиме выполнения.

```
sonic# show ntp associations
-----
remote      refid      st  t  when  poll  reach  delay  offset  jitter
-----
*192.0.2.13 .GPS.      1  u  0     64    1     11.333 69.327  0.074
-----
* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured
```

Табл. 6. Отображение значений и их описание

Значение в строке	Описание
*	Коммутатор синхронизируется с этим сервером или аналогом.
number	Коммутатор еще не синхронизирован с этим сервером или аналогом.
+	Коммутатор выбрал этот сервер или аналог для синхронизации времени.
-	Коммутатор рассматривает этот сервер или аналог как кандидата на синхронизацию времени.
~	Сервер или пир сконфигурирован статически.
remote	IP-адрес NTP-сервера или пирингового сервера.
refid	IP-адрес удаленного устройства, с которым синхронизируется сервер или пиринг.
st	Stratum, т.е. количество переходов NTP-сервера от источника времени, от 0 до 16. В качестве NTP-клиента коммутатор автоматически использует сервер с наименьшим stratum. 0 означает, что устройство является источником времени. 1 означает, что устройство напрямую подключено к источнику времени. 2 означает, что устройство подключено к устройству stratum 1 , и так далее. 16 означает, что устройство не синхронизировано с источником времени.

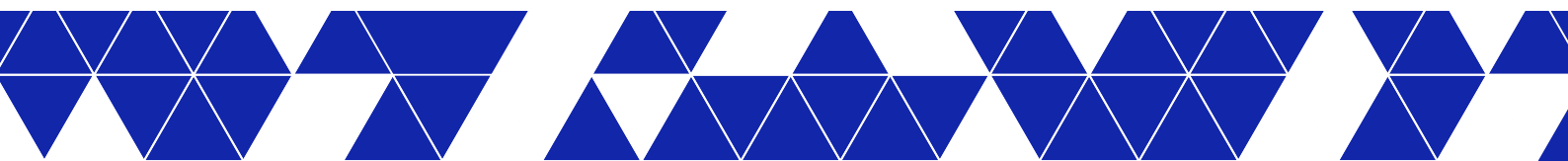


Табл. 6. Отображение значений и их описание

Значение в строке	Описание
t	Тип NTP-устройства: u - одноадресная или многоадресная передача NTP-клиента; b - широковещательная или многоадресная передача NTP-клиента; l - локальные опорные часы на коммутаторе; s - симметричный пиринг; A - многоадресная передача NTP-сервера; B - широковещательная передача NTP-сервера; M - многоадресная передача NTP-сервера.
when	Время (в секундах), прошедшее с момента получения пакета обновления NTP или с момента последней синхронизации времени.
poll	Интервал опроса (в секундах), используемый коммутатором для отправки запросов времени NTP, от 8 до 5160 (36 часов).
reach	Доступность NTP-сервера. Если значение reach не равно нулю, сервер доступен; если значение равно нулю, сервер недоступен. Значение reach - это одноранговая переменная, которая фиксирует момент получения действительного NTP-пакета и момент отправки NTP-пакета.
delay	Задержка в пути (в миллисекундах) до NTP-сервера. Разница во времени смещения (в миллисекундах) между коммутатором и сервером NTP или другим NTP пирингом.
offset	Рассчитанное смещение (в миллисекундах) между временем клиента и сервера.
jitter	Среднее отклонение во времени между коммутатором и сервером NTP на основе множества выборок времени.

4.3.6 Отображение настроек протокола сетевого времени NTP

Вы можете отобразить настроек NTP коммутатора с помощью команд `show ntp associations`, `show ntp server`, `show ntp global` в режиме выполнения.

Просмотр текущих настроек NTP

```
sonic# show running-configuration | grep ntp
ntp authenticate
ntp authentication-key 3 md5 U2FsdGVkX1+e0qbeFoKsXP7G009ZSMViCERTD43NqpD6v0sciMsCXPTaZgt7Js2F encrypted
ntp server 192.0.2.4 key 3 minpoll 6 maxpoll 10 version 4
ntp source-interface Management 0
ntp source-interface Ethernet1
ntp trusted-key 3
ntp vrf Vrf1050
```

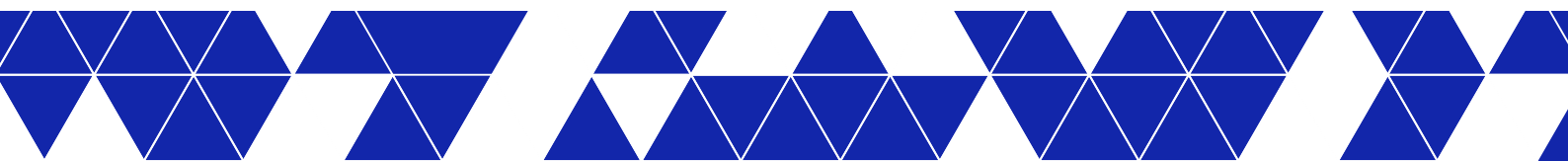
Просмотр настроенных NTP-серверов

```
sonic# show ntp server
-----
NTP Servers          minpoll maxpoll version Authentication key ID
-----
192.0.2.4            6        10        4
```

Опции `minpoll` и `maxpoll` указывают интервалы опроса запросов, отправляемых на NTP-сервер, в секундах, равных степени двойки. Например, `minpoll = 6` ($2^6 = 64$ с), `maxpoll = 10` ($2^{10} = 1024$ с).

Просмотр глобальных настроек NTP

```
sonic# show ntp global
-----
NTP Global Configuration
-----
NTP source-interface: eth0
Loopback100
NTP vrf: mgmt
```



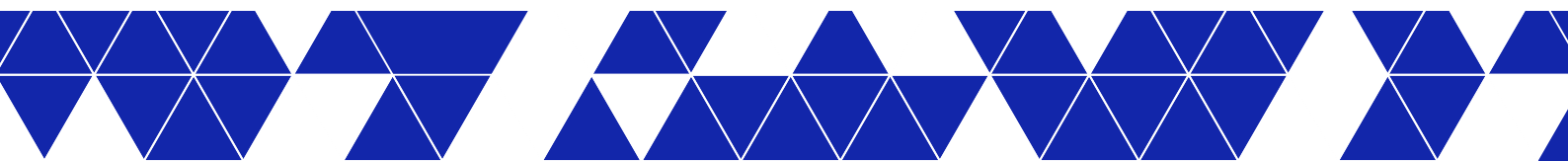
Просмотр NTP-ассоциаций

Убедитесь, что перед IP-адресом NTP-сервера стоит знак звезды (*) (мастер определен) и этот знак не меняется при повторном выводе команды.

```
sonic# show ntp associations
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*192.0.2.13	.GPS.	1	u	0	64	1	11.333	69.327	0.074

* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured



5 Управление устройством

Настоящий раздел описывает способ управления устройством с помощью взаимодействия между менеджерами и агентами в клиент-серверной архитектуре

5.1 SNMP

Данная глава содержит описание основных функций и настроек SNMP на сетевых устройствах Kornfeld.


5.1.1 Обзор SNMP

SNMP - это протокол прикладного уровня, обеспечивающий взаимодействие между менеджерами и агентами в клиент-серверной архитектуре.

SNMP-агент работает на управляемой системе и поддерживает на ней MIB. SNMP-менеджер генерирует запросы для получения информации из базы данных управляющей информации и обрабатывает ответы.

Менеджер может отправлять запросы агенту либо для получения информации из MIB (SNMP GET запрос), либо для внесения изменений в MIB (SNMP SET запрос). Агент также может отправлять незапрошенные сообщения менеджеру (SNMP-уведомления).

В качестве транспортного протокола SNMP обычно использует протокол UDP.

 Kornfeld OS не поддерживает операции SNMP SET.

5.1.1.1 MIB

MIB-файл содержит полный список и описание всех объектов конкретного устройства, которые можно запросить или контролировать с помощью SNMP. MIB имеют иерархическую структуру и доступны через OID. Данный файл служит интерфейсом между системой управления сетью (NMS) и SNMP-агентом, что позволяет NMS взаимодействовать и управлять переменными устройства через запросы и настройки.

Kornfeld OS поддерживает следующие MIB.

Табл. 7. Перечень поддерживаемых MIB

Модуль	Стандарт
RFC1213-MIB	RFC 1213
SNMP-FRAMEWORK-MIB	RFC 3411
SNMP-MPD-MIB	RFC 3412
SNMP-TARGET-MIB	RFC 3413
SNMP-VIEW-BASED-ACM-MIB	RFC 3415
SNMPv2-MIB	RFC 3418
IF-MIB	RFC 2863
IP-MIB	RFC 4293
IP-FORWARD-MIB	RFC 4292
TCP-MIB	RFC 4022
UDP-MIB	RFC 4113
HOST-RESOURCES-MIB	RFC 2790

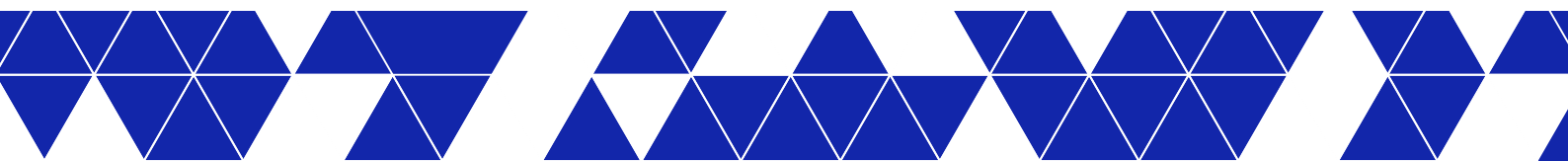


Табл. 7. Перечень поддерживаемых MIB

Модуль	Стандарт
BGP4-MIB	RFC 4273
ENTITY-MIB	RFC 6933
ENTITY-SENSOR-MIB	RFC 3433
LLDP-MIB	IEEE 802.1AB
NET-SNMP-AGENT-MIB	RFC 2579
NOTIFICATION-LOG-MIB	RFC 3014

5.1.1.2 SNMP-версии

Разработаны три версии SNMP: SNMPv1, SNMPv2c и SNMPv3.

SNMPv1 предоставляет основные функции управления сетью. Данная версия имеет ограниченный уровень безопасности с использованием аутентификации на основе паролей community и поддерживает небольшой набор ошибок.

SNMPv2c - это улучшенная версия SNMPv1, которая поддерживает расширенный набор кодов ошибок, дополнительные типы данных и новые операции, такие как GETBULK и INFORM.

SNMPv3 является наиболее безопасной версией, которая обеспечивает три уровня пользовательской защиты:

- noAuthNoPriv - без аутентификации и шифрования;
- authNoPriv - с аутентификацией, но без шифрования;
- privUser - с аутентификацией и шифрованием.

❗ Kornfeld OS поддерживает SNMPv2c и SNMPv3.

5.1.1.3 Идентификатор сущности SNMP

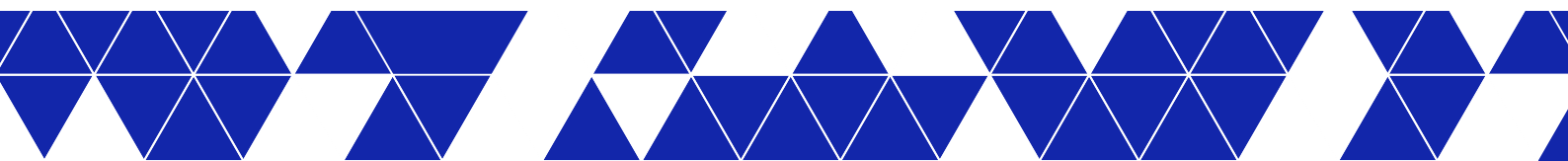
Идентификатор сущности (Engine ID) - это уникальный идентификатор, присваиваемый локальному SNMP-агенту на коммутаторе. Он задается октетами, разделенными двоеточиями, например, 80:00:02:b8:04:61:62:63.

В SNMPv3 при настройке SNMP-пользователя применяется локальный идентификатор сущности для генерации локализованных ключей аутентификации и/или конфиденциальности, что повышает безопасность сообщений SNMPv3.

5.1.1.4 SNMP-группы и пользователи

SNMP-пользователь - это член SNMP-группы, который взаимодействует с локальным SNMP-агентом. На удаленном устройстве пользователь SNMP идентифицируется по IP-адресу и UDP-порту, используемому для доступа к локальному агенту.

В Kornfeld OS SNMP-пользователям присваиваются определенные привилегии доступа в зависимости от назначенной группы. Группа задает набор профилей view SNMP MIB, к которым пользователи могут получить доступ.



5.1.1.5 Профиль SNMP view

В Kornfeld OS профили SNMP view настраиваются на уровне моделей безопасности и уровней внутри группы SNMP-пользователей. Каждый профиль определяет начальный OID в иерархии дерева MIB и указывает, включаются ли в данный профиль остальные структуры дерева или исключаются из него.

- Профиль **read view** предоставляет доступ к указанному дереву OID только для чтения, позволяя пользователям получать информацию из MIB.
- Профиль **write view** предоставляет доступ к указанному дереву OID в режиме «чтение-запись», позволяя пользователям как получать информацию из MIB, так и изменять ее.
- Профиль **notify view** позволяет отправлять SNMP-уведомления из указанного дерева OID другим членам группы.

Путем настройки данных профилей можно контролировать уровень доступа и разрешений, которые предоставляются SNMP-пользователям, гарантируя взаимодействие только с определенными частями MIB в соответствии с назначенной моделью и уровнем безопасности.

5.1.1.6 SNMP-уведомления

SNMP-уведомления (SNMP notifications) разделяются на два типа: **TRAP** и **INFORM**.

TRAP - это спонтанные события, генерируемые управляемым устройством, чтобы уведомить NMS об особых ситуациях. К ним относятся такие события, как перезапуск устройства или определенные события-триггеры. Когда условия выполняются, SNMP-агент отправляет уведомление Trap NMS, что позволяет администраторам оперативно реагировать на особые ситуации.

Trap считается ненадежным, потому что не требуют подтверждения от получателя, и отправитель не может подтвердить получение уведомления. Использование Trap позволяет сократить обмен управляющей информацией между NMS и управляемыми устройствами.

В свою очередь Inform также является спонтанным событием управляемого устройства, но требует подтверждения. Когда управляемое устройство посылает в NMS сообщение InformRequest, NMS отвечает сообщением InformResponse. Если управляемое устройство не получает подтверждения, оно сохраняет событие в буфере Inform, повторяет отправку InformRequest и записывает его в журнал событий. Inform потребляет больше системных ресурсов по сравнению с Trap.

❗ Текущая версия Kornfeld OS поддерживает только SNMP Inform.

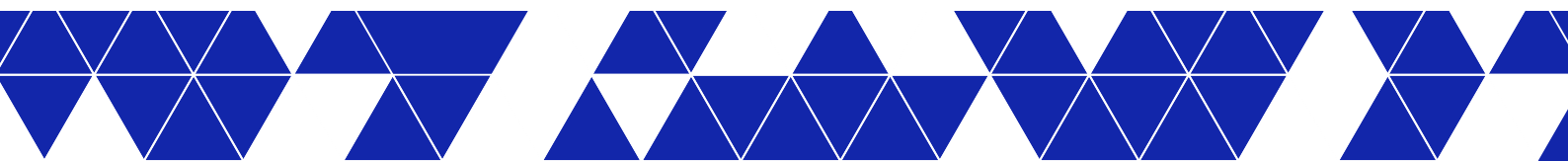
5.1.2 Настройка SNMP

Чтобы использовать SNMPv2c, настройте следующие параметры:

- локальный адрес агента;
- SNMP-группы;
- пароли SNMP community для взаимодействия с управляющими станциями;
- доступ пользователей к SNMP-агенту на коммутаторе;
- профили views структуры дерева MIB;
- уведомления SNMP Traps и Informs.

5.1.2.1 Настройка SNMP-агента

При включении VRF управления необходимо настроить локальный SNMP-агент на прослушивание запросов от управляющих станций на указанных IPv4/IPv6-адресах. Кроме того, можно указать UDP-порты или интерфейс, связанный с VRF управления. По умолчанию используется UDP-порт 161 и VRF default.



Чтобы настроить дополнительные адреса локального агента, введите соответствующую команду с нужными IPv4/IPv6-адресами.

Настройка SNMP-агента с номером UDP-порта, не заданным по умолчанию

1. Настройте интерфейс коммутатора.

```
kornfeld# configure terminal
kornfeld(config)# interface Ethernet 39
kornfeld(config-if-Ethernet39)# ip address 192.0.2.2/30
kornfeld(config-if-Ethernet39)# exit
```

2. Настройте IP-адрес и номер порта SNMP-агента.

```
kornfeld(config)# snmp-server agentaddress 192.0.2.2 port 1024
```

3. Настройте идентификатор сущности SNMP.

```
kornfeld(config)# snmp-server engine 8100013703525400abcd
```

4. Настройте местоположение SNMP-сервера.

```
kornfeld(config)# snmp-server location "Lab1, RACK-10"
```

5. Настройте контактные данные ответственного за SNMP-сервер.

```
kornfeld(config)# snmp-server contact "Yadro Support"
```

6. Проверьте настройку SNMP-сервера.

```
kornfeld# show snmp-server
Location      : Lab1, RACK-10
Contact       : Yadro Support
EngineID      : 8100013703525400abcd

Agent Addresses:

-----
IP Address      UDP Port      Interface
-----
192.0.2.2       1024
```

Для удаления определенной функции используйте соответствующую команду с префиксом `no` в режиме конфигурирования.

5.1.2.2 Настройка SNMP-группы

Настройте SNMP-группу

```
kornfeld# configure terminal
kornfeld(config)# snmp-server group group1 v2c notify no_view
```

Проверьте настройки SNMP-группы

```
kornfeld# show snmp-server group

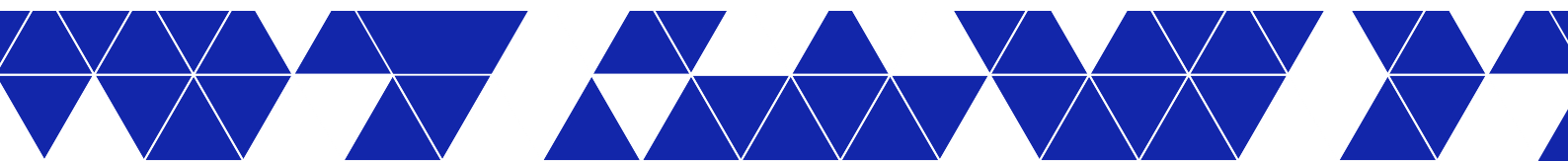
Group Name      Model: Security      Read View      Write View      Notify View
-----
group1          v2c: no-auth-no-priv None             None            no_view
```

Для удаления SNMP-группы используйте команду с префиксом `no` в режиме конфигурирования.

```
kornfeld(config)# no snmp-server group group1 v2c
```

5.1.2.3 Настройка пароля SNMP community

Настройте пароль SNMP community и назначьте его группе



```
kornfeld# configure terminal
kornfeld(config)# snmp-server community comm1 group group1
```

Проверьте настройку пароля SNMP community

```
kornfeld(config)# show snmp-server community
```

Community Name	Group Name
comm1	group1

Для удаления пароля SNMP community используйте команду с префиксом `no` в режиме конфигурирования.

```
kornfeld(config)# no snmp-server community comm1
```

5.1.2.4 Настройка SNMP-пользователя

⚠ Прежде чем настраивать удаленного пользователя с помощью команды `snmp-server user`, создайте удаленный идентификатор сущности с помощью команды `snmp-server engine`. При изменении настроенного идентификатора сущности для удаленного устройства необходимо заново настроить пароли аутентификации и конфиденциальности для всех удаленных пользователей, связанных с этим идентификатором сущности.

Настройте SNMP-пользователя и свяжите его с группой

```
kornfeld# configure terminal
kornfeld(config)# snmp-server user user1 group group1
```

Проверьте настройку SNMP-пользователя

```
kornfeld# show snmp-server user
```

User Name	Group Name	Auth	Privacy
user1	group1	None	None

Для удаления SNMP-пользователя используйте команду с префиксом `no` в режиме конфигурирования.

```
kornfeld(config)# no snmp-server user user1
```

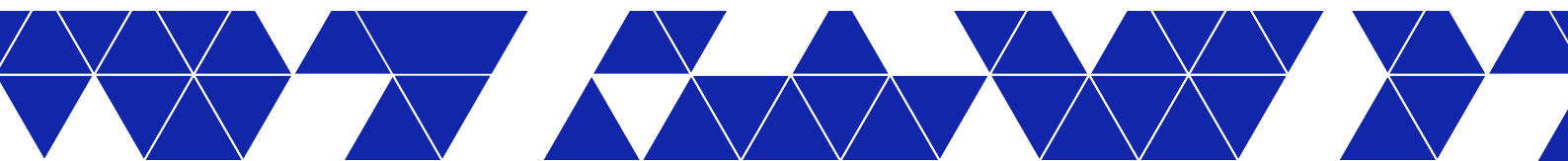
5.1.2.5 Настройка профиля SNMP view

Для настройки доступа к структуре дерева MIB в SNMP-агенте можно создать профиль view для чтения, чтения-записи или уведомления. Параметр `oid-tree` позволяет указать начальный OID в иерархии дерева MIB. Можно включить или исключить оставшееся содержимое поддерева MIB, указав соответственно `included` или `excluded`. При необходимости можно повторить команду, чтобы дополнительно исключить определенные элементы дерева из включенного содержимого. Гибкость в настройке профиля view позволяет определить область доступа SNMP в соответствии с требованиями, обеспечивая доступ SNMP-пользователей только к соответствующим частям структуры дерева MIB.

Настройте профиль SNMP view

```
kornfeld# configure terminal
kornfeld(config)# snmp-server view view2 1.2.3.4.5.6.7.8.9.2 excluded
```

Проверьте настройку профиля SNMP view



```
kornfeld(config)# show snmp-server view
```

View Name	OID Tree	Type
view2	1.2.3.4.5.6.7.8.9.2	excluded

Для удаления профиля SNMP view используйте команду с префиксом `no` в режиме конфигурирования.

```
kornfeld(config)# no snmp-server view view2 1.2.3.4.5.6.7.8.9.2
```

5.1.2.6 Настройка уведомлений SNMP Inform

Уведомления Inform отправляются на UDP-порт 162 и в VRF по умолчанию.

Настройте хост SNMP для получения уведомлений SNMP Inform

```
kornfeld# configure terminal
kornfeld(config)# snmp-server host 192.0.2.2 community comm1 informs timeout 150 retries 5
```

Проверьте настройки хоста SNMP

```
kornfeld# show snmp-server host
```

Target Address	Port	Type	Community	Ver	T-Out	Retries
192.0.2.2	162	inform	comm1	v2c	150.0	5

Target Address	Port	Type	User Name	Security	T-Out	Retries
----------------	------	------	-----------	----------	-------	---------

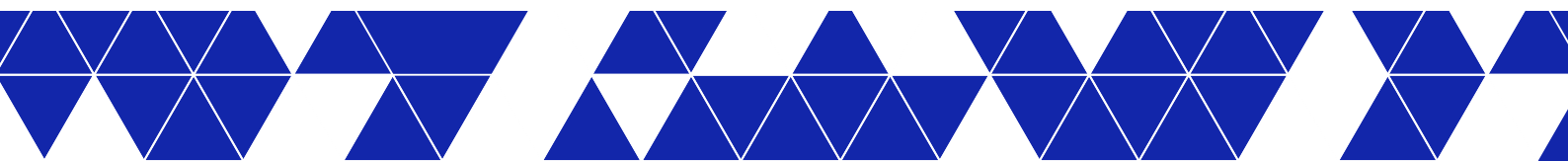
Для удаления хоста SNMP используйте команду с префиксом `no` в режиме конфигурирования.

```
kornfeld(config)# no snmp-server host 192.0.2.2
```

5.1.2.7 Пример настройки SNMP

В данном примере показана настройка SNMP на коммутаторе, включая адрес SNMP-агента, идентификатор сущности, контактные данные ответственного за SNMP-сервер, местоположение сервера, профили view, группы и пользователи.

```
# configure terminal
# interface Ethernet 39
# ip address 192.0.2.2/30
# exit
# snmp-server agentaddress 192.0.2.2 port 1024
# snmp-server engine 8100013703525400abcd
# snmp-server location "Lab1, RACK-10"
# snmp-server contact "Yadro Support"
# snmp-server group group1 v2c notify no_view
# snmp-server community comm1 group group1
# snmp-server user user1 group group1
# snmp-server view view2 1.2.3.4.5.6.7.8.9.2 excluded
# snmp-server host 192.0.2.2 community comm1 informs timeout 150 retries 5
# end
```



6 Настройка коммутации L2

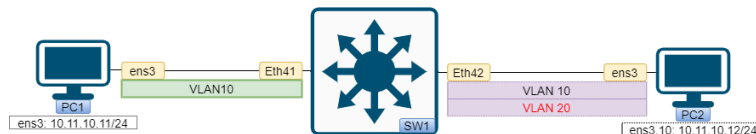
В настоящем разделе описаны способы коммутации сетевых устройств Kornfeld.

6.1 VLAN

Виртуальные локальные сети (VLAN) разделяют сеть на отдельные сегменты, что обеспечивает повышенную безопасность, улучшает производительность сети и упрощает управление. Когда узлам в одной VLAN нужно связаться с узлами в другой VLAN, используется механизм маршрутизации между VLAN.

Различают два режима работы портов на коммутаторах – режим доступа и транковый режим. Первый используется при подключении конечных хостов и указывает, в какой VLAN эти хосты будут работать. Второй предназначен для передачи нескольких VLAN между коммутаторами. Для разделения пакетов по VLAN используется механизм тегирования. Тег VLAN вставляется в Ethernet-кадр, добавляя необходимую информацию.

Пример конфигурации VLAN:



Описание:

Хосты PC1 и PC2 подключены к разным физическим портам коммутатора (SW1). Хост PC1 подключен к порту Ethernet 41, который пропускает только нетегированный трафик VLAN 10. Хост PC2 подключен к транковому порту Ethernet 42, который пропускает трафик двух VLAN - VLAN 10 и VLAN 20. При этом трафик для VLAN 10 является тегированным, а для VLAN 20 - нетегированным. Такая схема называется гибридной.

Создание VLAN и настройка портов на коммутаторе:

1. Войдите в режим конфигурирования:

```
kornfeld# configure terminal
```

2. Создайте отдельные VLAN:

```
kornfeld(config)# vlan 10
kornfeld(conf-Vlan10)# exit
kornfeld(config)# vlan 20
kornfeld(conf-Vlan20)# exit
```

3. Настройте Ethernet 41 в режиме доступа:

```
kornfeld(config)# interface Ethernet 41
kornfeld(conf-if-Ethernet41)# switchport access Vlan 10
kornfeld(conf-if-Ethernet41)# exit
```

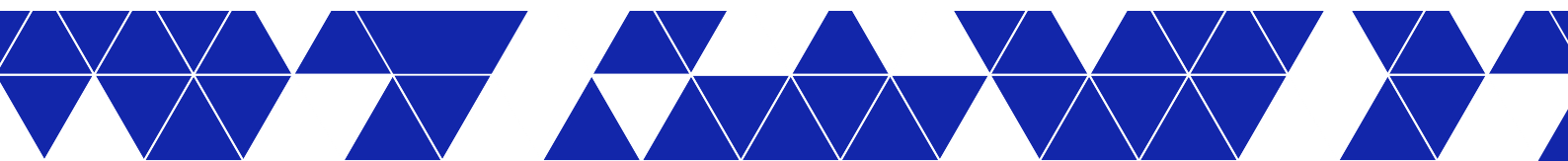
4. Настройте Ethernet 42 в транковом режиме:

```
kornfeld(config)# interface Ethernet 42
kornfeld(conf-if-Ethernet42)# switchport access Vlan 20
kornfeld(conf-if-Ethernet42)# switchport trunk allowed Vlan 10
```

5. Вернитесь в режим просмотра:

```
kornfeldj,zpfntkmyj ghzv nen (conf-if-Ethernet42)# end
```

6. Проверьте конфигурацию VLAN:



```
kornfeld# show vlan
```

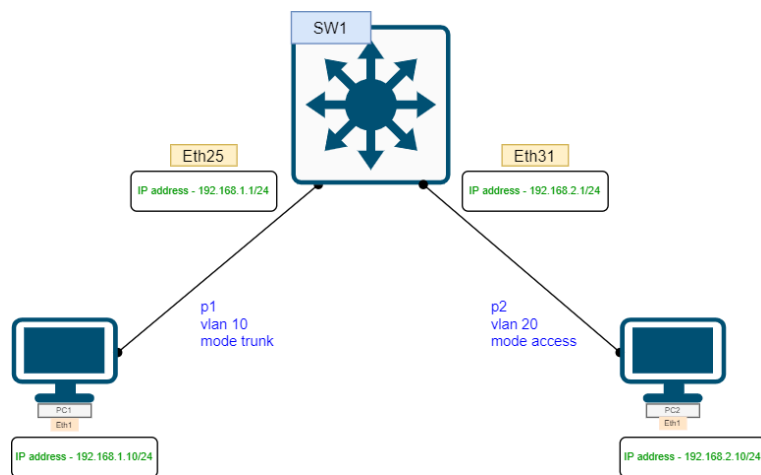
7. Проверьте связь между PC1 и PC2. Для этого выполните команду `ping` на PC1.

```
ping 10.11.10.12
```

Конфигурация:

```
# configure terminal
# vlan 10
# exit
# vlan 20
# exit
# interface Ethernet 41
# switchport access Vlan 10
# exit
# interface Ethernet 42
# switchport access Vlan 20
# switchport trunk allowed Vlan 10
# end
```

Пример настройки маршрутизации между VLAN:



Описание:

Хосты PC1 и PC2 подключены к разным физическим портам коммутатора (SW1). Хост PC1 подключен к транковому порту Ethernet 25, который пропускает тегированный трафик для VLAN 10. Хост PC2 подключен к порту доступа Ethernet 31, который пропускает трафик только для VLAN 20.

Настройка VLAN и виртуального интерфейса коммутатора (SVI) на SW1:

1. Войдите в режим конфигурирования:

```
kornfeld# configure terminal
```

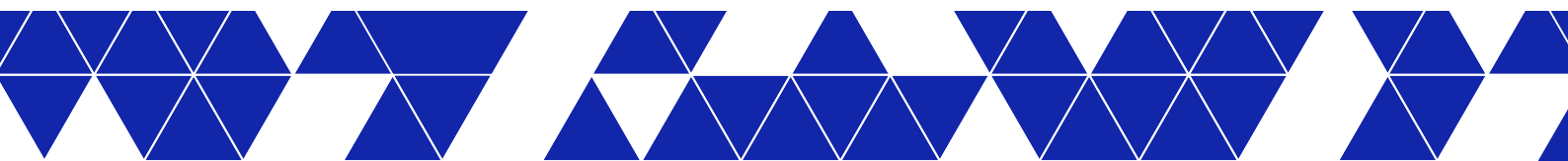
2. Создайте отдельные VLAN:

```
kornfeld(config)# vlan 10
kornfeld(conf-Vlan10)# exit
kornfeld(config)# vlan 20
kornfeld(conf-Vlan20)# exit
```

3. Настройте VLAN 10 интерфейс:

```
kornfeld(config)# interface Vlan 10
kornfeld(conf-if-Vlan10)# ip address 192.168.1.1/24
kornfeld(conf-if-Vlan10)# exit
```

4. Настройте VLAN 20 интерфейс:



```
kornfeld(config)# interface Vlan 20
kornfeld(conf-if-Vlan20)# ip address 192.168.2.1/24
kornfeld(conf-if-Vlan20)# exit
```

5. Настройте Ethernet 25 в транковом режиме:

```
kornfeld(config)# interface Ethernet 25
kornfeld(conf-if-Ethernet25)# switchport trunk allowed Vlan 10
kornfeld(conf-if-Ethernet25)# exit
```

6. Настройте Ethernet 31 в режиме доступа:

```
kornfeld(config)# interface Ethernet 31
kornfeld(conf-if-Ethernet31)# switchport access Vlan 20
kornfeld(conf-if-Ethernet31)# end
```

7. Убедитесь, что у SW1 есть IP-соединения с ПК1 и ПК2. Проверьте, что статусы соответствующих интерфейсов VLAN находятся в состоянии up/up:

```
kornfeld# show ip interfaces
Flags: U-Unnumbered interface, A-Anycast IP
-----
Interface IP address/mask VRF Admin/Oper Flags
-----
eth0 172.20.151.11/24 mgmt up/up
Loopback0 10.1.0.1/32 up/up
Vlan10 192.168.1.1/24 up/up
Vlan20 192.168.2.1/24 up/up
```

Конфигурация:

```
# configure terminal
# Vlan 10
# exit
# Vlan 20
# exit
# interface Vlan 10
# ip address 192.168.1.1/24
# exit
# interface Vlan 20
# ip address 192.168.2.1/24
# exit
# interface eth25
# switchport trunk allowed Vlan 10
# exit
# interface eth31
# switchport access Vlan 20
# end
```

Удаление VLAN:

1. Войдите в режим конфигурирования:

```
kornfeld# configure terminal
```

2. Войдите в режим конфигурирования VLAN:

```
kornfeld(config)# vlan vlan-id
```

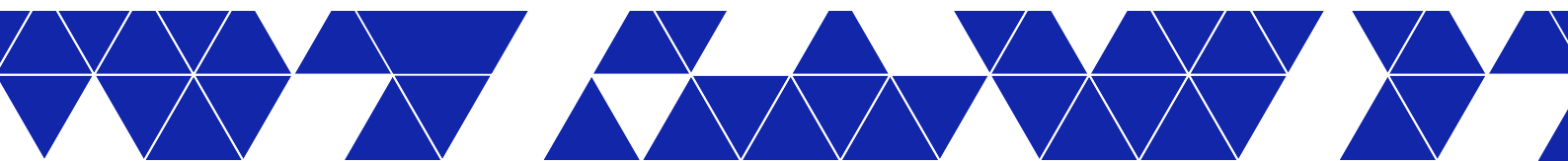
3. Удалите VLAN:

```
kornfeld(conf-Vlanvlan-id)# no vlan vlan-id
```

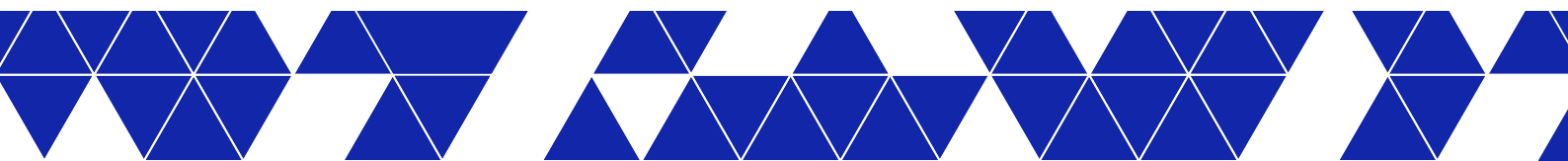
vlan-id

Идентификатор конкретного VLAN.

Пример:



```
# configure terminal
# vlan 10
# no vlan 10
```



7 Настройка коммутации L3

В настоящем разделе описаны способы статической коммутации сетевых устройств Kornfeld.

7.1 Статическая маршрутизация

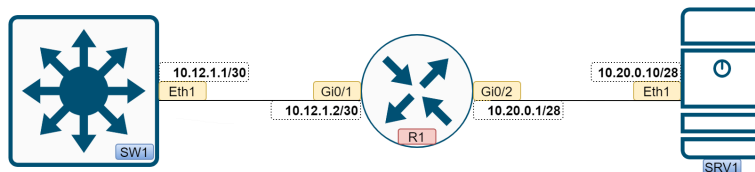
Статическая маршрутизация представляет собой метод управления потоком данных, основанный на заранее определенных путях передачи. Эти пути добавляются в таблицу маршрутизации устройства, при этом протоколы маршрутизации не используются.

В отличие от динамической статическая маршрутизация требует меньше ресурсов, так как не затрачивает ресурсы на определение маршрутов.

К недостаткам относятся плохая масштабируемость и отсутствие гибкости. В сложных и динамичных сетевых средах статическая маршрутизация требует частых ручных изменений конфигурации.

В разделе "Статическая маршрутизация" подробно рассматриваются этапы создания и управления статическими маршрутами.

Пример конфигурации статического маршрута:



Перед настройкой коммутатора (SW1) настройте интерфейсы на маршрутизаторе (R1) и сервере (SRV1).

Настройка Ethernet-порта на коммутаторе (SW1) и установка статического маршрута к SRV1:

1. Войдите в режим конфигурирования:

```
kornfeld# configure terminal
```

2. Настройте IP-адрес интерфейса Ethernet 1:

```
kornfeld(config)# interface Ethernet 1
kornfeld(config-if-Ethernet1)# ip address 10.12.1.1/30
kornfeld(config-if-Ethernet1)# exit
```

3. Установите статический маршрут:

```
kornfeld(config)# ip route 10.20.0.0/28 10.12.1.2
```

4. Вернитесь в режим просмотра:

```
kornfeld(config)# end
```

5. Проверьте настройку статической маршрутизации:

```
kornfeld# show ip route static
```

Конфигурация:

```
# configure terminal
# interface Ethernet 1
# ip address 10.12.1.1/30
# exit
# ip route 10.20.0.0/28 10.12.1.2
# end
```

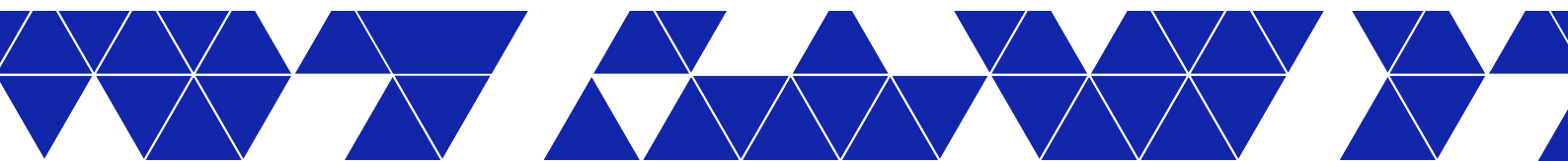

Удаление статического маршрута:

1. Войдите в режим конфигурирования:

```
kornfeld# configure terminal
```

2. Удалите настроенный статический маршрут:

```
kornfeld(config)# no ip route 10.20.0.0/28 10.12.1.2
```



8 Настройка доступа и аутентификации пользователей

В сетевых устройствах Kornfeld реализована защита конфигурации устройства и данных от несанкционированного доступа.

В настоящем разделе описаны следующие понятия:

- Общие принципы AAA
- Порядок настройки AAA
- Локальные пользователи
- RBAC
- RADIUS
- TACACS
- Отладка AAA

8.1 Служба аутентификации, авторизации и учета

Службы аутентификации, авторизации и учета (authentication, authorization and accounting - AAA) (далее - AAA) обеспечивают разграничение доступа к функционалу коммутатора и сетевым ресурсам, а также соответствие действий пользователей после входа в систему установленным для них правам.

Аутентификация - проверка имени и пароля пользователя.

Авторизация - предоставление пользователям, прошедшим аутентификацию, предусмотренных для них прав и установление ограничений на доступ к тем функциям, к которым эти пользователи доступа не имеют. В Kornfeld OS авторизация пользователей происходит на основании ролей.

Учет подразумевает запись в системный журнал сведений о действиях, выполняемых пользователями сетевых ресурсов.

В операционной системе коммутатора процессы AAA реализованы и могут осуществляться как локально, так и средствами централизованной системы, к которой коммутатор обращается по специальному протоколу RADIUS (Remote Authentication Dial-In User Service) или TACACS+ (Terminal Access Controller Access Control System plus).

8.1.1 Локальные пользователи

В сетевых устройствах Kornfeld реализовано разграничение доступа пользователей к командам и режимам коммутатора в соответствии с предоставленными им правами.

В операционной системе Kornfeld OS имеется один предустановленный пользователь `admin` с административными правами. Администратор может создавать учетные записи для других пользователей. Каждая учетная запись определяется именем пользователя, паролем и ролью. На основании роли пользователями выдаются права на доступ к режимам и командам коммутатора.

Пользователь `admin`

На коммутаторе предустановлен локальный пользователь `admin` с правами администратора и паролем по умолчанию. Ему можно менять пароль с использованием команды `username`.

```
kornfeld# configure terminal
kornfeld(config)# username admin password admin1 role admin
kornfeld(config)# exit
```



Если пароль пользователя `admin` был изменен, то хэш этого пароля выводится также, как и у любого другого пользователя.

В текущей конфигурации

```
kornfeld# show running-configuration | grep user
username admin password $6$3MwvJM303jsxE6WM$2SQOSFbHyQj8UggJ7TI3nPI/
dZ2zBn9MZZdNLPcoDTbNf9S3BugQQTKyQjHoInk7yPkTI7uypsoDFUB6zn1W3/ role admin
```

В startup конфигурации (заметьте что в startup конфигурации пароль `admin` установлен по умолчанию)

```
kornfeld# show startup-configuration | grep user
username admin role admin
username oper password $6$0n8oUHbBmJDONHE$Adn5Bi7a15fAOACDOTgxgIrr0Jsh4YTF3Xc6HXo3krNrJvBh89N3iXRRvxSikqCTqIrS3QzMZ/
AsAYrCs8.pj/ role operator
username robot password $6$3Y0bMk0y1kFM14KJ
$BymKmSi9mDx8diww2DYYKcx2tiakIJkJErgYrJwQ4fR3o.dwBTR21At2duz.bgfuVFkR2gVNgkh7AtEvK5y6c. role admin
```

В saved конфигурации

```
kornfeld# show saved-configuration name TEST1 | grep user
username admin password $6$3MwvJM303jsxE6WM$2SQOSFbHyQj8UggJ7TI3nPI/
dZ2zBn9MZZdNLPcoDTbNf9S3BugQQTKyQjHoInk7yPkTI7uypsoDFUB6zn1W3/ role admin
username robot password $6$3Y0bMk0y1kFM14KJ
$BymKmSi9mDx8diww2DYYKcx2tiakIJkJErgYrJwQ4fR3o.dwBTR21At2duz.bgfuVFkR2gVNgkh7AtEvK5y6c. role admin
```

Стандартными средствами восстановить пароль локального пользователя `admin` невозможно.

Если пользователь `admin` единственный в системе, удалить его невозможно.

```
host2(config)# no username admin
The user admin is logged in.
If you delete the user account all sessions of this user will be reset. [y/N]:y
%Error: Can't delete user 'admin', because it is the last user with admin role
host2(config)#
```

Роли

Каждая учетная запись пользователя в Kornfeld OS имеет уникальное имя, пароль и роль. Роль определяет набор прав доступа к функциям коммутатора, которые будут предоставлены пользователю после входа в систему. В операционной системе коммутатора предусмотрены две системные роли: администратора (`admin`) и оператора (`operator`). Переопределять системные роли каким-либо образом, иначе назначая права на вызов команд, изменять их или создавать новые роли нельзя. Каждому пользователю можно назначить только одну роль.

Пользователь с правами администратора имеет полный доступ ко всем режимам и командам коммутатора, тогда как пользователи с правами оператора могут работать только в пользовательском режиме (`execution mode`). Оператор также может просматривать текущую конфигурацию, перезагружать и выключать коммутатор.

8.1.2 Управление пользователями

Создание нового пользователя

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `username` и укажите имя пользователя, пароль и присваиваемую ему роль.

```
kornfeld(config)# username administrator password pass role admin
kornfeld(config)# username operator password pass role operator
```

3. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```



Просмотр пользователей

```
kornfeld# show running-configuration | grep username
username admin role admin
username administrator <hash_password> role admin
username operator password <hash_password> role operator
```

Изменение роли пользователя

1. Выведите на экран существующих пользователей и проверьте текущие роли.

```
kornfeld# show running-configuration | grep username
username administrator <hash_password> role admin
username operator password <hash_password> role operator
```

2. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

3. С помощью команды `username` укажите имя, пароль и новую роль пользователя.

```
kornfeld(config)# username administrator password pass role operator
kornfeld(config)# username operator password pass role admin
```

4. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

5. Убедитесь, что роль пользователя изменена.

```
kornfeld# show running-configuration | grep username
username administrator <hash_password> role operator
username operator password <hash_password> role admin
```

Просмотр активных сессий

```
kornfeld# show users
admin          pts/0  2022-09-07 13:10 (10.199.21.162)
administrator  pts/1  2023-01-18 09:13 (172.20.150.191)
operator       pts/1  2023-01-18 09:13 (172.20.150.191)
```

Удаление пользователя

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `no username` и укажите имя удаляемого пользователя.

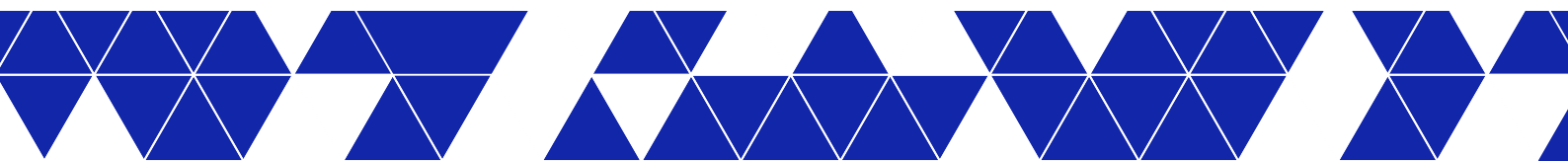
```
kornfeld(config)# no username operator
```

3. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Учет локальных пользователей

Когда пользователь входит в систему под своим уникальным именем и паролем, его можно однозначно идентифицировать и затем регистрировать его действия в системном журнале (`syslog`). Системный журнал находится в файле `auth.log` в директории `/var/log`. В него заносятся действия с учетными записями пользователей, такие как создание, удаление пользователя, смена пароля, а также вход пользователя



в систему, выход пользователя из системы, вход пользователя в режим конфигурации и выход пользователя из режима конфигурации в пользовательский режим.

8.1.3 Настройка AAA

Централизованные системы аутентификации имеют клиент-серверную архитектуру. Пользователь обращается к коммутатору за доступом к ресурсу, сообщая имя пользователя и пароль. Коммутатор передает данные серверу AAA. Сервер AAA проверяет учетную запись пользователя, определяет его роль и сообщает коммутатору, разрешен ли пользователю доступ к ресурсу. В соответствии с ответом сервера коммутатор либо обеспечивает пользователю доступ, либо нет.

Если сервер недоступен, коммутатор обращается к другому доверенному серверу из группы. Если недоступен ни один из серверов, то коммутатор действует согласно собственным настройкам и может либо отказать пользователю в доступе, либо проверять учетные данные пользователя в своей локальной базе данных.

Для централизованного управления группой серверов службы AAA настраиваются как глобально, так и локально. Глобальные параметры имеют значения по умолчанию и применяются ко всем серверам. Локальные параметры назначаются на каждом сервере индивидуально и имеют приоритет над глобальными. Таким образом, можно более гибко настраивать систему аутентификации в зависимости от применения.

Предварительные настройки

Прежде чем приступить к настройке коммутатора, убедитесь в том, что коммутатор и сервер AAA имеют достижимые маршруты друг к другу. Проведите предварительную настройку сервера, который будет использоваться для аутентификации и авторизации пользователей, согласно документации производителя сервера. Настройте на сервере AAA системные роли администратора и оператора, а также учетные записи пользователей.

Если системные роли пользователей на сервере (как RADIUS, так и TACACS+) не настроены, всем пользователям по умолчанию присваивается роль *operator*. Уровень привилегий роли *admin* равен 15, тогда как для роли *operator* от 1 до 6.

8.1.3.1 Настройка AAA с использованием сервера RADIUS

RADIUS (Remote Authentication Dial In User Service) - сетевой протокол, предназначенный для централизованной аутентификации, авторизации и учета пользователей, подключающихся к сетевым сервисам. В качестве транспортного протокола RADIUS использует User Datagram Protocol (UDP). По умолчанию порт 1812 используется для аутентификации и авторизации, а порт 1813 - для учета. Однако, администратор может назначить и другие порты. В ходе обмена данными по этому протоколу шифруется только поле пароля в пакете, остальные данные передаются открытым текстом.

Настройка методов аутентификации на коммутаторе

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите методы аутентификации в том порядке, в котором они будут применяться.

```
kornfeld(config)# aaa authentication login default group radius local
```

3. Разрешите коммутатору обращаться ко второму методу аутентификации, если аутентификация первым способом будет неудачной.

```
kornfeld(config)# aaa authentication failthrough enable
```



4. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Просмотр текущего порядка методов аутентификации

```
kornfeld(config)# show aaa
-----
AAA Authentication Information
-----
failthrough   : True
login-method  : radius, local
```

В данном примере значения параметров `failthrough : True` и `login-method : radius, local` указывают на то, что коммутатор будет обращаться сначала к серверам RADIUS, а если ни один из серверов не будет доступен, то к собственной локальной базе данных.

Изменение порядка методов аутентификации

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите методы аутентификации, которые будет применять коммутатор. Вводите системы в том порядке, в котором планируете осуществлять проверку.

```
kornfeld(config)# aaa authentication login default local group radius
```

В данном примере коммутатор будет обращаться сначала к локальной базе данных, а затем к настроенным серверам RADIUS.

3. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Базовые настройки RADIUS

Для минимальной настройки аутентификации пользователей с использованием сервера RADIUS необходимо указать IP-адрес RADIUS сервера и индивидуальный ключ шифрования для аутентификации коммутатора на RADIUS сервере.

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Укажите индивидуальный ключ шифрования для аутентификации коммутатора на RADIUS сервере. Ключ может содержать любые символы ASCII, кроме пробела, запятой и #. Максимальная длина ключа составляет 65 символов.

```
kornfeld(config)# radius-server key testing123
```

3. Укажите IP-адрес RADIUS сервера.

```
kornfeld(config)# radius-server host 10.10.20.4
```

4. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Чтобы настроить несколько серверов RADIUS, повторите указанные выше шаги для каждого сервера.

Глобальная конфигурация RADIUS

Для более широких настроек коммутатора возможна глобальная настройка конфигурации.



Статистика RADIUS (statistics)

На коммутаторе можно включить сбор статистики аутентификации или учета для всех RADIUS серверов.

```
kornfeld(config)# radius-server statistics enable
```

Время ожидания ответа сервера (timeout)

Изменить время ожидания ответа от сервера RADIUS в секундах. По истечении этого времени соединение с сервером будет разрываться. Параметр может принимать значения от 0 до 1000, по умолчанию установлено время ожидания 5 секунд.

```
kornfeld(config)# radius-server timeout 10
```

Протокол аутентификации (auth-type)

Обмен данными между RADIUS сервером и коммутатором происходит по одному из трех протоколов аутентификации: PAP, CHAP или MS-CHAPv2. По умолчанию используется наименее безопасный PAP, однако, администратор может настроить и другие.

```
kornfeld(config)# radius-server auth-type chap
```

Просмотр глобальной конфигурации

```
kornfeld# show radius-server

-----
RADIUS Global Configuration
-----
statistics      : True
timeout         : 10
auth-type       : chap
key configured  : Yes
-----

HOST            AUTH-TYPE KEY-CONFIG AUTH-PORT PRIORITY TIMEOUT RTSMT VRF  SI
-----
10.10.20.4      -          No          1812     -         -         -   -   -
-----

RADIUS Statistics
-----
10.10.20.4:
access-accepts: 1
access-rejects: 0
access-requests: 1
timeout-access-requests: 0
access-challenges: 0
bad-authenticators: 0
invalid-packets: 0
```

Локальные параметры RADIUS

Расширенные настройки протокола RADIUS направлены на повышение безопасности и эффективности работы системы аутентификации и авторизации. Локальные параметры имеют приоритет над глобальными. Они могут иметь значения по умолчанию.

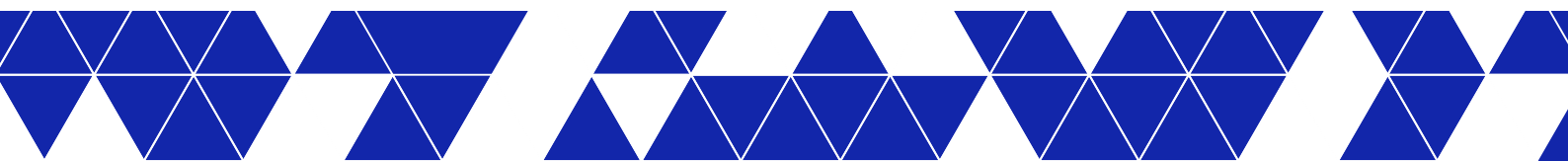
Количество повторных запросов (retransmit)

Коммутатор может обращаться к RADIUS серверу с запросом проверки пользователя несколько раз. Данный параметр может принимать значения от 0 до 100 и по умолчанию равен 3.

```
kornfeld(config)# radius-server retransmit 3
```

Приоритет сервера (priority)

Если для процессов AAA в системе предусмотрена группа RADIUS серверов, то коммутатор обращается к ним с запросами на аутентификацию пользователей по очереди. Настроить такую очередь можно с помощью параметра `priority`. Первым коммутатор обратится к серверу с наивысшим приоритетом.



Если тот не отвечает, то коммутатор перейдет к серверу с приоритетом ниже. Параметр может принимать значения от 1 до 64. Чем больше число, тем выше приоритет сервера.

```
kornfeld(config)# radius-server priority 55
```

Интерфейс источника (source-interface)

По умолчанию в качестве такого интерфейса источника для аутентификации пользователей на серверах RADIUS установлен интерфейс управления (management interface).

```
kornfeld(config)# radius-server source-interface Management 0
```

Адрес сервера NAS (nas-ip)

Относительно RADIUS серверов коммутатор выступает в роли сервера доступа (Network Access Server - NAS). По умолчанию в качестве IP-адреса сервера NAS используется IP-адрес интерфейса управления (management interface).

```
kornfeld(config)# radius-server nas-ip 2.2.2.2
```

VRF

Если обмен сообщениями между коммутатором и RADIUS сервером происходит в VRF, то необходимо указать имя такой VRF. По умолчанию для отправки пакетов AAA используется таблица маршрутизации по умолчанию (default VRF).

```
kornfeld(config)# radius-server vrf mgmt
kornfeld(config)# radius-server vrf vrf-name
```

Просмотр текущей конфигурации RADIUS

```
kornfeld# show running-configuration
...
radius-server host 1.2.4.5 key 9
3a95c26b2a5b96a6b80036839f296babe03560f4b0b7220d6454b3e71bdfc59b
radius-server retransmit 10
radius-server timeout 10
ip radius source-interface mgmt 1/1/1
...
```

Удаление конфигурации RADIUS

Прежде чем удалить настройки всех серверов RADIUS и отключить коммутатор от централизованной системы аутентификации, необходимо установить для ОС коммутатора только локальный метод аутентификации.

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите метод аутентификации `local`.

```
kornfeld(config)# aaa authentication login default local
```

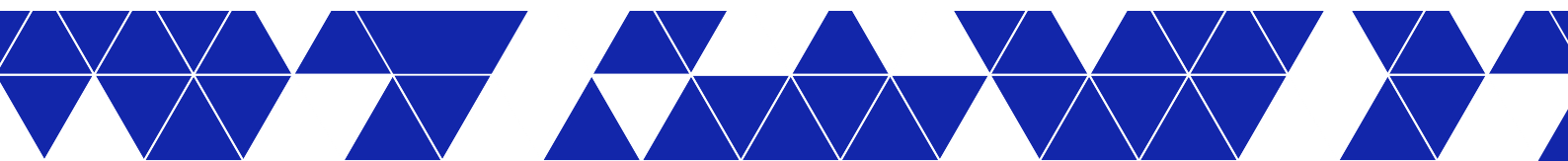
3. Удалите один за другим все настроенные RADIUS серверы.

```
kornfeld# no radius-server host 1.2.4.5
```

Учет средствами RADIUS

Отслеживать действия пользователя можно только через просмотр логов.

Например, ищем действия пользователя `sadmin`:




```
vSwitch-1# show logging lines | grep sadmin
Jun 14 09:25:42.169648 vSwitch-1 INFO mgmt-framework#supervisord: rest-server IJun 14 09:25:42.169053+00:00
2023      24 cliUserAuth.go:64] [REST-178] Authorization passed for user=sadmin, roles=[sadmin sudo docker admin]
Jun 14 09:25:42.197577 vSwitch-1 INFO mgmt-framework#clish[246]: User "sadmin" command "startup" status - success
Jun 14 09:38:41.288274 vSwitch-1 INFO mgmt-framework#supervisord: rest-server IJun 14 09:38:41.284560+00:00
2023      24 xfmr_system_infra.go:174] DbToYang_sys_infra_state_show_user_list_xfmr: %ssadmin pts/0      2023-06-14
09:25 (10.112.100.10)
Jun 14 09:38:41.345077 vSwitch-1 INFO mgmt-framework#clish[246]: User "sadmin" command "show users"
status - success
Jun 14 09:42:37.728735 vSwitch-1 INFO mgmt-framework#clish[246]: User "sadmin" command "show radius-
server" status - success
Jun 14 09:42:47.209715 vSwitch-1 INFO mgmt-framework#clish[246]: User "sadmin" command "show ip interfaces"
status - success
vSwitch-1 INFO mgmt-framework#clish[246]: User "sadmin" command "exit" status - success
Jun 14 09:50:12.001396 vSwitch-1 INFO mgmt-framework#supervisord: rest-server IJun 14 09:50:12.001014+00:00
2023      24 cliUserAuth.go:64] [REST-204] Authorization passed for user=sadmin, roles=[sadmin sudo docker admin]
```

Из записей в системном журнале видно когда пользователь вошел в систему ("Authorization passed for user=...." Jun 14 09:25:42.169648) и когда из нее вышел ("User "... command "exit" status - success" в Jun 14 09:44:32.032398), а также какие команды пользователь выполнял, когда и в какое время ("User "... command "show users" status - success).

8.1.3.2 Настройка AAA с использованием сервера TACACS+

Протокол TACACS+ (Terminal Access Controller Access Control System plus) позволяет реализовать аутентификацию и авторизацию на разных уровнях безопасности, разделяет AAA в соответствии с архитектурой AAA, обеспечивая модульность реализации сервера безопасности.

Протокол TACACS+ работает поверх протокола TCP и по умолчанию использует порт 49 для всех процессов.

TACACS+ обеспечивает большую безопасность данных за счет шифрования всей части протокола в пакете, отправляемом от коммутатора к серверу аутентификации, тогда как протокол RADIUS шифрует только пароли.

Настройка методов аутентификации на коммутаторе

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите методы аутентификации в том порядке, в котором они будут применяться.

```
kornfeld(config)# aaa authentication login default group tacacs+ local
```

3. Разрешите коммутатору обращаться ко второму методу аутентификации, если аутентификация первым способом будет неудачной.

```
kornfeld(config)# aaa authentication failthrough enable
```

4. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Просмотр текущего порядка методов аутентификации

```
kornfeld(config)# show aaa
-----
AAA Authentication Information
-----
failthrough   : True
login-method  : tacacs+, local
```

В данном примере значения параметров `failthrough : True` и `login-method : tacacs+, local` указывают на то, что коммутатор будет обращаться сначала к серверам TACACS+, а если ни один из серверов не будет доступен, то к собственной локальной базе данных.



Изменение порядка методов аутентификации

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите методы аутентификации, которые будет применять коммутатор. Вводите системы в том порядке, в котором планируете осуществлять проверку.

```
kornfeld(config)# aaa authentication login default local group tacacs+
```

В данном примере коммутатор будет обращаться сначала к локальной базе данных, а затем к настроенным серверам TACACS+.

3. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Базовые настройки TACACS+

Базовая настройка аутентификации с использованием TACACS+ включает указание IP-адреса сервера TACACS+ и индивидуального ключа шифрования для аутентификации коммутатора на TACACS+ сервере.

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Укажите индивидуальный ключ шифрования для аутентификации коммутатора на TACACS+ сервере. Ключ может содержать любые символы ASCII, кроме пробела, запятой, решетки (#) и других специальных символов. Максимальная длина ключа составляет 65 символов.

```
kornfeld(config)# tacacs-server key testing123
```

3. Укажите IP-адрес TACACS+ сервера.

```
kornfeld(config)# tacacs-server host 10.10.20.4
```

4. Вернитесь в пользовательский режим.

```
kornfeld(config)# exit
```

Чтобы настроить несколько серверов TACACS+, повторите указанные выше шаги для каждого сервера.

Глобальная конфигурация TACACS+

Глобальная конфигурация позволяет установить настройки по умолчанию.

Интерфейс источника (source-interface)

По умолчанию в качестве такого интерфейса источника для аутентификации пользователей на серверах RADIUS установлен интерфейс управления (management interface).

```
kornfeld(config)# tacacs-server source-interface Ethernet 0
```

Время ожидания ответа сервера (timeout)

Изменить время ожидания ответа от сервера TACACS+ в секундах. По истечении этого времени соединение с сервером будет разрываться. Параметр может принимать значения от 0 до 1000, по умолчанию установлено время ожидания 5 секунд.

```
kornfeld(config)# tacacs-server timeout 10
```

Протокол аутентификации (auth-type)



Обмен данными между TACACS+ сервером и коммутатором происходит по одному из трех протоколов аутентификации: PAP, CHAP или MS-CHAPv2. По умолчанию используется наименее безопасный PAP, однако, администратор может настроить и другие.

```
kornfeld(config)# tacacs-server auth-type chap
```

Просмотр глобальной конфигурации TACACS+

```
kornfeld(config)# show tacacs-server
```

```

-----
TACACS Global Configuration
-----
source-interface : eth0
timeout          : 10
auth-type        : chap
key configured   : Yes
-----

```

HOST	AUTH-TYPE	KEY-CONFIG	PORT	PRIORITY	TIMEOUT	VRF
10.112.101.2	chap	Yes	49	1	5	default

Локальные параметры TACACS+

Расширенные настройки протокола TACACS+ направлены на повышение безопасности и эффективности работы системы аутентификации и авторизации. Локальные параметры имеют приоритет над глобальными. Они могут иметь значения по умолчанию.

Количество повторных запросов (retransmit)

Коммутатор может обращаться к TACACS+ серверу с запросом проверки пользователя несколько раз. Данный параметр может принимать значения от 0 до 100 и по умолчанию равен 3.

```
kornfeld(config)# tacacs-server retransmit 3
```

Приоритет сервера (priority)

Если для процессов AAA в системе предусмотрена группа TACACS+ серверов, то коммутатор обращается к ним с запросами на аутентификацию пользователей по очереди. Настроить такую очередь можно с помощью параметра `priority`. Первым коммутатор обратится к серверу с наивысшим приоритетом. Если тот не отвечает, то коммутатор перейдет к серверу с приоритетом ниже. Параметр может принимать значения от 1 до 64. Чем больше число, тем выше приоритет сервера.

```
kornfeld(config)# radius-server priority 55
```

VRF

Если обмен сообщениями между коммутатором и TACACS+ сервером происходит в VRF, то необходимо указать имя такой VRF. По умолчанию для отправки пакетов AAA используется таблица маршрутизации по умолчанию (default VRF).

```
kornfeld(config)# tacacs-server vrf mgmt
kornfeld(config)# tacacs-server vrf vrf-name
```

Просмотр конфигурации TACACS+

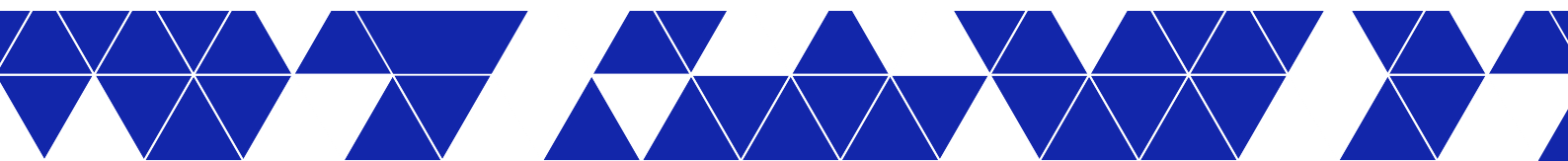
```
kornfeld# show running-configuration
```

```

...
tacacs-server host 1.2.4.5 key 9
3a95c26b2a5b96a6b80036839f296babe03560f4b0b7220d6454b3e71bdfc59b
ip tacacs source-interface loopback 2
...

```

Удаление конфигурации TACACS+



Прежде чем отключить коммутатор от централизованной системы аутентификации, необходимо отключить в настройках коммутатора метод аутентификации с помощью TACACS+ и убедиться, что метод аутентификации `local` включен.

1. Войдите в режим конфигурации.

```
kornfeld# configure terminal
```

2. Введите команду `aaa authentication login default` и укажите метод аутентификации `local`.

```
kornfeld(config)# aaa authentication login default local
```

3. Удалите по очереди все настроенные TACACS+ серверы

```
kornfeld# no tacacs-server host 1.2.4.5
```

